Федеральное государственное автономное образовательное
учреждение высшего образования
НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ
«ВЫСШАЯ ШКОЛА ЭКОНОМИКИ»

На правах рукописи

КАШНИЦКИЙ
ЮРИЙ САВЕЛЬЕВИЧ

МЕТОДЫ ЗАМКНУТЫХ ОПИСАНИЙ В ЗАДАЧЕ КЛАССИФИКАЦИИ ДАННЫХ СО СЛОЖНОЙ СТРУКТУРОЙ

Специальность 05.13.17 —
«Теоретические основы информатики»

Диссертация на соискание учёной степени
кандидата технических наук

Научный руководитель:
доктор физико-математических наук
С. О. Кузнецов

Москва – 2018

Оглавление

Введение

Актуальность темы. Чаще всего задачи анализа данных формулируются для данных, которые можно представить объектно-признаковыми таблицами. Если посмотреть на задачи машинного обучения в корпоративной среде или соревнования по анализу данных[1], то за редким исключением они сводятся к анализу объектно-признаковых таблиц. При этом данные со сложной структурой (тексты, изображения) тоже представляются в некотором признаковом пространстве (TF-IDF, word2vec, нейросетевые признаки изображений и т.д.). Однако в последнее время активно развиваются методы анализа сложно структурированных данных, для которых теоретически сложно либо практически неэффективно составлять признаковые описания, зато можно судить о свойствах объектов на основе сходства их описаний. Такие задачи встречаются в химической информатике [MMF11], анализе текстов [JM00] и изображений [Nav14]. Далее в этой работе под сложно структурированными данными мы будем понимать данные, для которых можно определить узорную структуру (Определение 28).

Важным аспектом в решении задач классификации является интерпретируемость полученных результатов. Во многих приложениях, особенно в медицине, необходима интерпретация результатов классификации в виде понятных человеку правил, к которым можно применить экспертный анализ и на его основе судить о релевантности используемых моделей, алгоритмов и мер сходства объектов в конкретной задаче. В разных задачах интерпретируемость определяется по-разному, но в данной работе под интерпретируемостью алгоритмов мы будем понимать их возможность объяснить классификацию тестовых примеров. Конкретней, под локальной интерпретируемостью классификации мы пониманием среднюю длину посылок правил, с помощью которых делается прогноз для тестового примера (Определение 21). В [Hol93b] показано, что методы классификации на основе коротких правил хорошо работают на большинстве наборах данных популярного репозитория UCI, при этом методы хорошо интерпретируемы, то есть полученные правила могут анализироваться экспертами.

Одним из успешных инструментов для анализа сложно структурированных данных является ДСМ-метод автоматического восстановления зависимостей из эмпирических данных [Фин83; Фин10a; Фин10b; Куз91; Дюк02] . Классификация на основе ДСМ-метода относится к интерпретируемым подходам, поскольку позволяет анализировать структурное сходство тестового примера и обучающих. Однако по качеству классификации, определяемому по метрике типа доли верных ответов на кросс-валидации или отложенной выборке, такой подход уступает ядерным методам

[1] www.kaggle.com/competitions

(kernel methods) [HSS08], в особенности, методу опорных векторов [CV95]. Было предложено множество ядерных функций для оценки сходства объектов со сложной структурой – строковые ядра [Lod+02], ядра для последовательностей [CMR08] и графовые ядра [Vis+10]. Недостатком метода опорных векторов является плохая интерпретируемость полученных результатов.

Необходимость анализа данных со сложными структурными описаниями и решения связанных с ними задач классификации делает актуальным применение методов, позволяющих работать со структурным сходством и использовать эффективные приближения описаний. Методы анализа формальных понятий и решеток замкнутых описаний (узорных структур) предоставляют удобный и эффективный математический аппарат для построения моделей в решении целого ряда важных научных и прикладных задач. В задачах обучения без учителя эти методы актуальны, поскольку позволяют находить и интерпретировать сходство произвольного множества объектов, а в задачах обучения с учителем – потому что с их помощью можно получить наборы классифицирующих правил, понятных человеку (интерпретируемых) и позволяющих далее применять к ним экспертный анализ. Аппарат проекций узорных структур позволяет эффективно работать с приближенными описаниями сложно структурированных объектов, учитывая основные свойства структуры и понижая вычислительную и временную сложность обработки таких описаний.

Таким образом, **объектом исследования** являются данные со сложной структурой. **Предметом исследования** являются методы, алгоритмы и программы для классификации данных со сложной структурой с помощью классифицирующих правил, а также для их экспертного анализа.

Целью диссертационного исследования является разработка единого подхода к классификации данных со сложной структурой. Результатами работы алгоритма должны быть как приемлемое для конкретной задачи качество классификации, так и интерпретируемый вывод алгоритма в виде классифицирующих правил, подходящий для дальнейшего экспертного анализа.

В соответствии с целью исследования были поставлены следующие **задачи**:

1. Предложить универсальный подход к классификации данных со сложной структурой на основе решеток замкнутых описаний;

2. В частном случае описаний в виде бинарных, категориальных и количественных признаков предложить подход к классификации на основе правил, решающий задачу классификации лучше (по точности), чем деревья решений, и порождающий более короткие правила, чем алгоритм случайного леса;

3. Разработать комплекс программ для классификации данных со сложной структурой и апробировать его в задачах классификации как с бинарными, категориальными и количественными признаками, так и с описаниями со сложной структурой в виде последовательностей и графов.

Следующие особенности работы определяют ее **научную новизну**:

1. Предложен новый подход к классификации данных со сложной структурой на основе узорных структур;

2. Предложен специальный вид проекций узорных структур для данных с количественными признаками, обобщающий подход к обучению на основе деревьев решений;

3. Создан комплекс программ для классификации данных со сложной структурой на основе решеток замкнутых описаний. Соответствующие алгоритмы были апробированы на многих наборах данных с категориальными и количественными признаками, а также на данных по токсичности химических веществ со сложной структурой в виде молекулярных графов.

Теоретическая ценность данной работы состоит

1. в представлении методов классификации числовых данных, в том числе деревьев решений, с помощью проекций интервальных узорных структур;

2. в представлении подхода к классификации на основе правил, гарантирующего нахождение правил с лучшим значением выбранного критерия информативности, чем правила, полученные с помощью деревьев решений;

3. во введении и исследовании дискретизирующей проекции для интервальных узорных структур.

Практическая ценность работы состоит

1. в получении качественных (по доле правильных ответов) и интерпретируемых решений задач классификации данных в виде последовательностей и графов;

2. в получении качества классификации в экспериментах с реальными данными, статистически значимо лучшего, чем у алгоритмов построения деревьев решений;

3. в представлении алгоритма классификации на основе правил, более коротких по длине (числу признаков в посылке), а потому легче интерпретируемых, чем правила, построенные алгоритмом случайного леса;

4. в разработке программного комплекса, позволяющего анализировать сложно структурированные данные и решать для них задачи классификации с помощью интерпретируемых наборов правил, подходящих для дальнейшего экспертного анализа.

Положения, выносимые на защиту:

1. Предложен универсальный подход к классификации данных со сложной структурой на основе решеток замкнутых описаний. При этом для каждого объекта порождаются наборы коротких и интерпретируемых классифицирующих правил;

2. Показано, что предложенный алгоритм классификации на основе правил демонстрирует более высокое качество классификации (в терминах средней доли правильных ответов и F1-метрики на кросс-валидации), чем деревья решений. Также он порождает в среднем более короткие и интерпретируемые правила, чем алгоритм случайного леса;

3. Показано, что для любого объекта можно найти подходящее классифицирующее правило, такое что его посылка будет замкнутым множеством признаков, а качество правила (измеряемое с помощью критерия типа прироста информации) – выше, чем у любого подходящего правила, построенного деревом решений.

4. Предложен вид приближений числовых описаний (в терминах проекций интервальных узорных структур), на основе которых представлены посылки правил, полученных с помощью деревьев решений. Эффективность использования таких проекций экспериментально подтверждена в задаче классификации для нескольких наборов данных с количественными признаками;

5. Разработан комплекс программ для анализа данных со сложной структурой на основе решеток замкнутых описаний. Поддерживаются 4 типа данных: числовые (бинарные, категориальные и количественные признаки), интервальные, последовательности и помеченные графы.

Достоверность полученных результатов опирается на строгость использованных математических моделей, их экспериментальное подтверждение и практическую эффективность программных реализаций.

Апробация работы. Основные результаты работы докладывались и обсуждались на следующих конференциях и семинарах:

1. Семинар Межфакультетской кафедры математического моделирования и компьютерных исследований МГУ имени М.В. Ломоносова 31 октября 2017 года, г. Москва, Россия;

2. Семинары отдела Интеллектуальных систем ВЦ РАН им. А.А. Дородницына 20 октября 2016 года, 7 июля 2017 года и 14 сентября 2017 года, г. Москва, Россия;

3. 23-ий Международный симпозиум по методологиям интеллектуальных систем (ISMIS 2017), июнь 2017 г., г. Варшава, Польша.

4. Семинары Департамента Анализа Данных и Искусственного Интеллекта НИУ ВШЭ (6 выступлений в мае и октябре 2015-2016 гг., а также в декабре 2016 г. и марте 2017 г.), г. Москва, Россия;

5. Пятнадцатая национальная конференция по искусственному интеллекту с международным участием (КИИ-2016), сентябрь 2016 г., г. Смоленск, Россия;

6. Семинар “What can FCA do for Artificial Intelligence?” при Европейской конференции по искусственному интеллекту ECAI, август 2016 г., г. Гаага, Нидерланды;

7. 13-ая международная конференция по решеткам понятий и их приложениям (The 13th International Conference on Concept Lattices and Their Applications), июль 2016 г., г. Москва, Россия;

8. Конференция “Технологии Больших Данных” (ТБД-2016), июнь 2016 г., г. Москва, Россия;

9. Пятая международная конференция по Анализу Изображений, Сетей и Текстов АИСТ 2016, апрель 2016 г., г. Екатеринбург, Россия (награда за лучший доклад в секции “Data Analysis, Graphs & Complex Data”);

10. Семинар “What can FCA do for Artificial Intelligence?” при международной объединенной конференции по искусственному интеллекту IJCAI, июль 2015 г., г. Буэнос-Айрес, Аргентина;

11. Ph.D.-семинар при Европейской конференции по машинному обучению и теоретическим основам и практике обнаружения знаний в базах данных ECML/PKDD, июль 2014 г., г. Нанси, Франция;

12. Семинар “What can FCA do for Artificial Intelligence?” при Европейской конференции по искусственному интеллекту ECAI, июль 2014 г., г. Прага, Чехия;

13. Третья международная конференция по Анализу Изображений, Сетей и Текстов АИСТ, апрель 2014 г., г. Екатеринбург, Россия;

Публикации. Основные результаты по теме диссертации изложены в 8 научных работах, 2 из которых изданы в изданиях, рекомендованных ВАК, 6 — в рецензируемых трудах международных конференций, индексируемых в базе данных научного цитирования Scopus.

Диссертация состоит из введения, 4 глав, заключения, списка литературы, а также списков рисунков, таблиц и приложений.

В Главе 1 рассматриваются некоторые базовые понятия теории решеток и Анализа Формальных Понятий, приводится обзор методов классификации в машинном обучении, основанных на ассоциативных правилах, а также рассматриваются критерии отбора классифицирующих правил.

В Главе 2 сначала дается введение в узорные структуры и их проекции для анализа сложно структурированных данных. Затем предлагается специальный вид проекций узорных структур, позволяющий интерпретировать алгоритмы построения деревьев решений в терминах АФП.

Глава 3 посвящена описанию предлагаемых алгоритмов классификации с помощью узорных структур и их проекций.

В Главе 4 приводятся результаты экспериментов, посвященных исследованию предлагаемых алгоритмов и их использованию в прикладных задачах анализа данных. Также описывается программный комплекс, реализующий предлагаемые алгоритмы. Рассматривается как абстрактный интерфейс для анализа данных с произвольной сложной структурой, так и частный случай интерфейса для данных, представимых в объектно-признаковом виде.

Глава 1

Анализ Формальных Понятий и классификация на основе ассоциативных правил

1.1. Введение

Термин “Анализ Формальных Понятий” (АФП, Formal Concept Analysis, FCA) был предложен Рудольфом Вилле (Rudolf Wille) [Wil09] в конце 1970-х годов в Техническом университете Дармштадта. АФП уходит корнями в предшествующие работы, посвященные соответствиям Галуа и решеткам замкнутых множеств [Bir40], и ранние работы о приложениях теории решеток к задачам информатики. Основной вклад в развитие этого направления математики сделан Рудольфом Вилле и Бернардом Гантером (Bernhard Ganter) [GW97].

Поначалу Анализ Формальных Понятий зачастую воспринимался как формализм для работы с таблицами из нулей и единиц. Однако сейчас актуальность АФП подтверждается его использованием в задачах обработки больших объёмов сложных динамических данных, связанных с дополнительными знаниями предметной области. В течение трех последних десятилетий основанные на АФП модели представления, выявления [Her02] и интенсивной обработки знаний разработаны и используются во многих научно-исследовательских и промышленных проектах во всем мире. Среди работ, освещающих методы и приложения АФП, можно выделить обзоры Йонаса Пульманса (J. Poelmans) с коллегами [Poe+13b; Poe+13a], обзор Клаудио Карпинето (C. Carpineto) и Джовани Романо (G. Romano) [CR04], обзор Уты Присс (U. Priss) по применению АФП к информационному поиску и выявлению знаний [Pri06], обзор программного обеспечения, использующего АФП [Til04], а также аналитический обзор с библиометрическим анализом публикаций и изучением научно-исследовательского сообщества АФП [DJS12]. В то же время были показаны связи с другими направлениями в области выявления и обработки знаний: дескриптивными логиками, понятийными структурами [Sow84], нахождением ассоциативных правил [Smi09], би- и

трикластеров [Ign+15], машинным обучением [Kuz04], теорией “неточных” (rough) и “нечетких” (fuzzy) множеств [Poe+14] и другими.

Отдельное важное направление АФП связано с уходом от представления данных в виде бинарного отношения для анализа сложно структурированных данных [GK01]. Этому посвящена Глава 2.

В данной работе основной акцент сделан на исследовании подходов к задаче классификации в машинном обучении, основанных на АФП, особенно в задачах со сложно структурированными данными. Поэтому далее в этой главе мы рассмотрим основные термины и идеи АФП [GW97] и теории решеток (Раздел 1.2), постановку задачи классификации в машинном обучении (Раздел 1.3), а также подходы к классификации данных на основе АФП (Раздел 1.4).

1.2. Теория решеток и Анализ Формальных Понятий

1.2.1. Частично-упорядоченные множества и решетки

Приведем стандартные определения из областей теории отношений, порядков и решеток [Бир84].

***Определение* 1.** *[Бир84] Бинарное отношение $\leq$ на множестве A называется **(нестрогого) частичного порядка**, если оно рефлексивное, антисимметричное и транзитивное, то есть для $a, b, c \in A$:*

1. $a \leq a$ *(рефлексивность);*
2. $a \leq b, b \leq a \Rightarrow a = b$ *(антисимметричность);*
3. $a \leq b, b \leq c \Rightarrow a \leq c$ *(транзитивность).*

*Множество A с определенным на нем отношением частичного порядка $\leq$ называется **частично-упорядоченным множеством** $(A, \leq)$. В случае $a \leq b$ $(a, b \in A)$ говорят, что элемент a **меньше** элемента b. Если при этом $a \neq b$, то элемент a **строго меньше** элемента b $(a < b)$. Если для $a \nexists b : a \leq b$, то a – максимальный элемент множества A относительно порядка $\leq$.*

***Определение* 2.** *[Бир84] Пусть имеется частичный порядок $\leq$. Тогда соответствующее ему **отношение покрытия** $\prec$ задаётся следующим образом:*

$$x \prec y := x \leq y, x \neq y, \nexists z \neq x,y : \quad x \leq z \leq y$$

или, эквивалентно,

$$x \prec y := x < y, \nexists z : \quad x < z < y.$$

***Определение* 3.** *[Бир84] Бинарная операция $\sqcap : A \times A \to A$ называется **полурешеточной**, если для некоторого $e \in A$ и любых $a, b, c \in A$:*

1. $a \sqcap a = a$ *(идемпотентность);*

2. $a \sqcap b = b \sqcap a$ *(коммутативность);*

3. $(a \sqcap b) \sqcap c = a \sqcap (b \sqcap c)$ *(ассоциативность);*

4. $a \sqcap e = e$.

Определение* 4.** *[Бир84] Множество A с определенной на нем полурешеточной операцией называется* ***полурешеткой $(A, \sqcap)$.

Полурешеточная операция $\sqcap$ задает два частичных порядка $\sqsubseteq$ и $\sqsupseteq$ на A $(a, b \in A)$:

$$a \sqsubseteq b \leftrightarrow a \sqcap b = a$$

При этом частичный порядок на полурешетке задаётся как $x \leq y \Leftrightarrow x \sqcap y = x$.

Определение* 5.** *[Бир84]* ***Решеткой *называется множество L, на котором определены две полурешёточные операции $\sqcap$ и $\sqcup$ такие что:*

- $x \sqcup (x \sqcap y) = x$
- $x \sqcap (x \sqcup y) = x$

Решётку можно также определить другим эквивалентными способом через два определения полурешётки.

Определение* 6.** *[Бир84]* ***Верхней гранью *подмножества X упорядоченного множества A называется элемент $l \in A$, такой что $l \geq x \; \forall x \in X$.* ***Точная (наименьшая) верхняя грань*** *множества X (также называется* ***супремумом X*** *$- \; supX$) - это верхняя грань l множества X, такая что $l \leq m$ для любой верхней грани m множества X. Аналогично определяется точная (наибольшая) нижняя грань множества X, или инфимум X.*

Определение* 7.** *[Бир84] Частично-упорядоченное множество $(SL, \leq)$ называется* ***верхней полурешеткой*, если для любой пары элементов множества $x, y \in SL$ существует супремум $sup\{x,y\}$.*

Двойственным образом вводится понятие **нижней полурешетки**, определяемой относительно инфимума.

Определение* 8.** *[Бир84] Частично-упорядоченное множество $(SL, \leq)$ называется* ***нижней полурешеткой*, если для любой пары элементов множества $x, y \in SL$ существует инфимум $inf\{x,y\}$.*

Определение* 9.** *[Бир84]* ***Решеткой *называется упорядоченное множество $(L, \leq)$, которое является верхней и нижней полурешёткой.*

Определение* 10.** *[Бир84]* ***Диаграмма (Хассе) *частично-упорядоченного множества* $(L, \leq)$ *– это плоский геометрический объект, состоящий из кругов, центры которых соответствуют элементам порядка, и связывающих центры кругов отрезков, соответствующих отношению покрытия* $(L, \prec)$*, со следующими свойствами:*

1. $a \prec b \Longrightarrow$ *точка, соответствующая вершине* a*, имеет строго меньшую вертикальную координату чем точка, соответствующая вершине* b*.*

2. Отрезки не проходят через круги, центры которых не являются их границами.

***Пример* 1.** *Граф и диаграмма частичного порядка [Kuz96].*

	a	*b*	*c*	*d*	*e*
a	*1*	*0*	*1*	*1*	*1*
b	*0*	*1*	*1*	*1*	*1*
c	*0*	*0*	*1*	*0*	*1*
d	*0*	*0*	*0*	*1*	*1*
e	*0*	*0*	*0*	*0*	*1*

ациклический граф

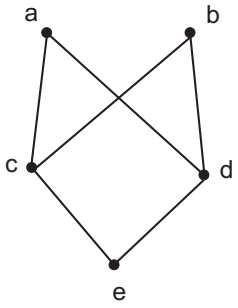


диаграмма порядка

Анализ Формальных Понятий и аппарат узорных структур основываются на следующем определении соответствия Галуа [GW97].

Определение* 11.** *[Бир84] Пусть* $(P, \leq_P)$ *и* $(Q, \leq_Q)$ *– частично упорядоченные множества.* ***Соответствием Галуа *между этими множествами называется пара отображений:* $\varphi : P \mapsto Q$ *и* $\psi : Q \mapsto P$ *такие, что для любых* $p_i, p_j \in P$ *и* $q_k, q_l \in Q$ *(*i,j,k,$l \in \mathbb{N}$*) верно:*

- $p_i \leq_P p_j \Rightarrow \varphi(p_i) \geq_Q \varphi(p_j)$*;*
- $q_k \leq_Q q_l \Rightarrow \psi(q_k) \geq_P \psi(q_l)$*;*
- $p_i \leq_P \psi(\varphi(p_i))$ *и* $q_l \leq_Q \varphi(\psi(q_l))$*.*

1.2.2. Анализ Формальных Понятий

Анализ Формальных Понятий (АФП) – это область прикладной теории решёток, методы которого используются для решения различных задач анализа и интеллектуального данных. Приведём основные определения АФП согласно [GW97].

Определение* 12.** *[GW97]* ***Формальный контекст *– это тройка* (G, M, I)*, в которой* G *– это множество объектов,* M *– множество признаков,* $I \subseteq G \times M$ *– бинарное отношение между* G *и* M*.*

В Таблице 1.1 дан пример формального контекста. Между множествами подмножеств объектов и признаков можно задать соответствие Галуа с помощью следующих отображений:

$$A' = \{m \in M \mid (g,m) \in I \text{ для всех } g \in A\}, \quad \text{где } A \subseteq G$$
$$B' = \{g \in G \mid (g,m) \in I \text{ для всех } m \in B\}, \quad \text{где } B \subseteq M$$

Для отдельных объектов и признаков $a \in A$ и $b \in B$ понимаем a' как $\{a\}'$ и b' как $\{b\}'$.

	3G	**LTE**	**GSM**	**jack**
Iphone 5	×		×	×
Galaxy S7	×	×	×	×
Iphone 7	×	×	×	
ThinkPad	×	×		×
Acer A200	×		×	

Таблица 1.1: Пример формального контекста

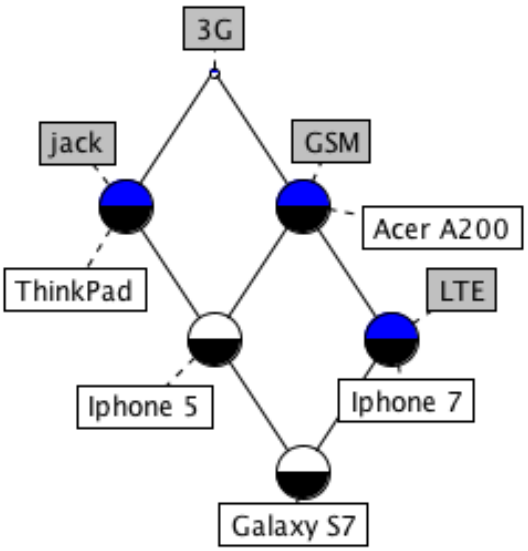


Рисунок 1.1: Решетка формальных понятий для формального контекста, изображенного Таблицей 1.1.

Соответствие Галуа сопоставляет множеству объектов максимальное множество признаков, каждый из которых находится в отношении с каждым объектом. Аналогично для множества признаков. Например, для формального контекста, изображенного Таблицей 1.1, $\{Iphone\ 5, Acer\ A200\}' = \{3G,\ GSM\}$, в то время как $\{LTE\}' = \{Galaxy\ S7, Iphone\ 7, ThinkPad\}$. Соответствие Галуа лежит в основе формальных понятий и соответствующей решетки формальных понятий.

Определение* 13.** *[GW97]* ***Формальное понятие *– это пара* (A,B)*, где* A *– это подмножество объектов,* $A \subseteq G$*;* B *– признаков,* $B \subseteq M$*, причём* $A' = B$*, а* $A = B'$*. Множество объектов* A *называют объёмом, а множество признаков* B *– содержанием формального понятия* (A,B)*.*

Примером формального понятия для контекста, изображенного в Таблице 1.1, является пара $(\{Galaxy\ S7, Iphone\ 7, ThinkPad\}, \{3G, LTE\})$, которой соответствует максимальное подмножество объектов, обладающих признаками $\{3G, LTE\}$, в то время как мы не можем расширить множество признаков, не изменив множество объектов, соответствующих ему. При этом $\{Galaxy\ S7, Iphone\ 7, ThinkPad\}'' = \{3G, LTE\}$ и $\{3G, LTE\}'' = \{Galaxy\ S7, Iphone\ 7, ThinkPad\}$. Множество понятий упорядочено согласно теоретико-множественному включению объемов или содержаний. Например, $(\{ThinkPad\}; \{3G, LTE, jack\}) \leq (\{Galaxy\ S7, Iphone\ 7, ThinkPad\}, \{3G, LTE\})$, так как $\{ThinkPad\} \subseteq \{Galaxy\ S7, Iphone\ 7, ThinkPad\}$, или двойственно $\{3G, LTE\} \subseteq \{3G, LTE, jack\}$. Данный частичный порядок является решеткой, то есть для любой пары понятий существуют верхняя и нижняя грани. Рисунок 1.1 показывает диаграмму решетки, соответствующей формальному контексту, изображенному в Таблице 1.1.

Для возможности работы с количественными признаками в АФП вводится понятие многозначного формального контекста.

Определение* 14.** *[GW97]* ***Многозначный формальный контекст *— это четверка* (G, M, W, I)*, где* G *– множество объектов,* M *– множество признаков,* W *– множество значений признаков,* $I \subseteq G \times M \times W$*, такое что* $((g,m,w) \in I)$ **&** $((g,m,v) \in I) \Rightarrow w = v$*. Признак* m ***полный****, если для всех* $g \in G$ *существует* $w \in W$ *, такое что* $(g, m, w) \in I$*. Многозначный контекст полон, если все его признаки полны. Для полных многозначных контекстов значение признака* m *на объекте* g *обозначается через* $m(g)$*, таким образом* $(g, m, m(g)) \in I$*.*

Далее нам также пригодится определение генератора замкнутого множества признаков.

Определение* 15.** *[GW97] Подмножество признаков* $D \subseteq M$ *есть* ***генератор *замкнутого подмножества признаков* $B \subseteq M$*,* $B'' = B$*, если* $D \subseteq B$*,* $D'' = B = B''$*.*

Подмножество $D \subseteq M$ есть **минимальный генератор**, если для любого $E \subset D$ имеет место $E'' \neq D'' = B''$.

Генератор $D \subseteq M$ называется **нетривиальным**, если $D \neq D'' = B''$. Множество всех нетривиальных минимальных генераторов B обозначим $mingen(B)$.

1.2.3. Алгоритмы построения решетки формальных понятий

Существует немалое число алгоритмов для нахождения множества формальных понятий, таких как “Замыкай по-Одному” (Close-by-One, CbO) [Куз93; Kuz96], его модификация In-Close [And09], NextClosure [Gan10] и др., а также для нахождения решётки формальных понятий, таких как AddIntent [Kou+09] (см. обзор алгоритмов [KO02]). Алгоритмическая сложность нахождения всех формальных понятий контекста для указанных алгоритмов составляет

$O(|G||M||L|\min(|G|,|M|))$, где $|G|$ – количество объектов, $|M|$ – количество признаков, $|L|$ – конечный размер решётки. Стоит отметить, что размер решётки может быть экспоненциальным от числа объектов или признаков, точнее $2^{\min(|G|,|M|)}$.

Далее в Главе 1.4 для представления деревьев решений в терминах АФП нам понадобится определение признакового CbO-дерева, перефразированное на основе [Куз93].

***Определение* 16.** *Пусть дан формальный контекст* $\mathbb{K} = (G,M,I)$ *и признаки из множества* M *пронумерованы, т.е. для множества признаков* M *задан порядок* $(\alpha(M), <), \forall m \in M\ \alpha(m) \in [1, |M|]$. *Пусть для* $B \subseteq M$ $min(B)$ *выдает признаки из* B *с минимальным номером:*
$min(B) = \{m \mid m \in B, \alpha(m) < \alpha(\tilde{m})\ \forall \tilde{m} \in B \backslash \{m\}\}$.

Обозначим $suc(B)$ *– множество всех наследников множества* B*: понятий с содержанием вида* $(B \cup \{i\})''$*, таких что* $min((B \cup \{i\})'' \setminus B) = \{i\}$. ***Признаковым CbO-деревом*** *для формального контекста* $\mathbb{K}$ *называется дерево, состоящее из всевозможных множеств* $suc(B)$*, дуги которого задаются отношением* $(B, suc(B))$.

При замене множеств признаков на множества объектов в определении выше получается определение **объектного CbO-дерева**.

1.3. Задача классификации в машинном обучении

1.3.1. Постановка задачи классификации

Машинное обучение – одно из ключевых направлений искусственного интеллекта и анализа данных. Определений у этого термина немало, в одном из них теория машинного обучения определяется через решаемые ей задачи предсказания будущего поведения сложных систем в том случае, когда отсутствуют точные гипотезы о механизмах, управляющих поведением таких систем [Вью13]. Классическим определением обучающейся программы считается данное Томом Митчеллом [Mit97] (хоть оно и не строго формальное): “Компьютерная программа обучается решению некоторого класса задач T согласно метрике качества P с накоплением опыта E, если качество решения задач класса T этой программой, измеренное с помощью метрики P, растет при накоплении опыта E”.

Существует несколько областей машинного обучения (остановимся на самых крупных): обучение с учителем (supervised learning), (в частности, задачи классификации и восстановления регрессии), обучение без учителя (unsupervised learning) (в частности, задачи кластеризации и снижения размерности), обучение с подкреплением (reinforcement learning) и др. Далее нас будет интересовать первый тип машинного обучения – обучение с учителем, а именно, задача классификации.

Классическая постановка задачи классификации в машинном обучении формулируется следующим образом [Вью13]. Имеется множество объектов (ситуаций), описанное с помощью некоторого множества признаков и разделенное некоторым образом на классы. Задано конечное множество объектов, для которых известно, к каким классам они относятся. Это множество называется

обучающей выборкой. Принадлежность остальных объектов к классам неизвестна. Требуется построить алгоритм, способный классифицировать произвольный объект из исходного множества, то есть, указать номер (или наименование класса), к которому относится данный объект.

Формализация постановки задачи [Вью13]:

Пусть X — множество объектов, Y — конечное множество ответов (меток, имён классов).

Существует неизвестная целевая зависимость $y^* : X \to Y$ — отображение, значения которого известны только на объектах конечной обучающей выборки $X^\ell = \{(x_1,y_1),...,(x_\ell,y_\ell)\}$. Требуется построить алгоритм $a : X \to Y$, способный приближать целевую функцию $y^*(x)$ для произвольного объекта $x \in X$.

Отметим, что мы выбрали именно это определение, поскольку оно не накладывает никаких ограничений на природу объектов множества X, хотя зачастую задачу классификации определяют сразу для элементов признакового пространства, то есть предполагают, что каждый объект $x \in X$ описан с помощью признаков $f_1,\dots,f_d$: $x = (f_1(x),\dots,f_d(x))$, где $f : X \to D_f$ называется признаком (D_f – множество допустимых значений признака, в зависимости от этого множества признаки делятся на бинарные, номинальные, порядковые и количественные). В главе 2 мы будем говорить про постановку задачи классификации и для объектов с произвольной сложной структурой, не только с признаковым описанием.

Примеры применения методов классификации для объектов, заданных признаковым описанием, можно найти в задачах медицинской диагностики [Вью13]. В роли объектов выступают пациенты. Признаки характеризуют результаты обследований, симптомы заболевания и применявшиеся методы лечения. Признаки могут быть бинарными (пол, наличие головной боли, слабости), порядковыми (тяжесть состояния – удовлетворительное, средней тяжести, тяжёлое, крайне тяжёлое), количественными (возраст, пульс, артериальное давление, содержание гемоглобина в крови, доза препарата). Признаковое описание пациента является, по сути дела, формализованной историей болезни. Накопив обучающую выборку, можно классифицировать вид заболевания (дифференциальная диагностика), определять наиболее целесообразный способ лечения, предсказывать длительность и исход заболевания, оценивать риск осложнений а также находить синдромы — наиболее характерные для данного заболевания совокупности симптомов [Вью13].

Помимо медицинских приложений, классификация используется в множестве других задач: распознавание образов и компьютерное зрение, распознавание речи, предсказание оттока клиентов, кредитный скоринг, обнаружение спама, классификация документов, семантический анализ текста и во многих других [PR08].

Методы классификации, основанные на сходстве объектов, образуют целое семейство алгоритмов машинного обучения (классификации и не только). Самый известный подход к работе с объектами на основе их сходства – ядерные методы [Mül+01]. Примером задачи классификации, решаемой на основе сходства объектов и в которой объекты задаются сложными описаниями, может служить задача прогнозирования мутагенности химических веществ [Hel+04]. Здесь описаниями объектов будут их молекулярные графы. Обзор методов классификации данных, представленных графами, можно найти в [Nav14].

Задача классификации известна также как задача корректного распознавания корректного распознавания образов, в случае целочисленных признаков задача эффективно решается методами логического подхода [Жур66; Жур71; Жур02; Руд87; Дья05; Вор00]. Вычислительные аспекты построения логических корректоров в таких задачах исследованы в работе [Про16].

Постановки задачи классификации в терминах АФП

Далее для единообразия изложения материала мы переформулируем задачу классификации в машинном обучении в терминах АФП. В частности, выделим задачу классификации для данных с бинарными признаками и для данных с вещественными признаками.

Определение* 17.** *Пусть даны* $\mathbb{K}_{train} = (G_{train}, M \cup \{t\}, I_{train})$ *и* $\mathbb{K}_{test} = (G_{test}, M \cup \{t\}, I_{test})$ *–* ***обучающий *и* ***тестовый*** *формальные контексты. Контекст* $\mathbb{K} = (G_{train} \cup G_{test}, M \cup \{t\}, I_{train} \cup I_{test})$ *–* ***классификационный****, признак* t *–* ***целевой****. Задачей* ***бинарной классификации*** *для классификационного контекста* $\mathbb{K}$ *называется построение функции* $y^* : G_{train} \cup G_{test} \to y$*, где* $y = t$ *или* $y = \neg t$*, которая каждому объекту* $g \in G_{train} \cup G_{test}$ *ставит в соответствие* t *или* $\neg t$*. При этом множество* M *называется* ***признаковым пространством****, а* g' *для* $\forall g \in \mathbb{K}$ *называется* ***признаковым описанием*** *объекта* g*. Контекст* $\mathbb{K}_{train}$ *также будем называть обучающей выборкой.*

Обучающий контекст из определения выше также будем называть *обучающей выборкой*. Чтобы сформулировать задачу классификации для данных с количественными признаками, используем Определение 14 многозначного формального контекста.

Определение* 18.** *Пусть даны* $\mathbb{K}_{train} = (G_{train}, M \cup t, W_{train}, I_{train})$ *и* $\mathbb{K}_{test} = (G_{test}, M \cup t, W_{test}, I_{test})$ *–* ***обучающий *и* ***тестовый*** *многозначные формальные контексты. Контекст* $\mathbb{K} = (G_{train} \cup G_{test}, M \cup \{t\}, W_{train} \cup W_{test}, I_{train} \cup I_{test})$ *–* ***классификационный****, признак* t *–* ***целевой****. Задачей* ***бинарной классификации*** *для многозначного классификационного контекста* $\mathbb{K}$ *называется построение функции* $y^* : G_{train} \cup G_{test} \to y$*, где* $y = t$ *или* $y = \neg t$*, которая каждому объекту* $g \in G_{train} \cup G_{test}$ *ставит в соответствие* t *или* $\neg t$*. Контекст* $\mathbb{K}_{train}$ *также будем называть обучающей выборкой.*

Если в условиях предыдущего определения целевой признак t является многозначным, то соответствующая задача классификации называется не бинарной, а **многоклассовой**.

***Пример* 2.** *В качестве примера возьмем задачу классификации для "классического" набора данных из [Mit97]. Классификационный контекст (обучающая и тестовая выборки) представлен Таблицей 1.2 :* $\mathbb{K} = (G_{train} \cup G_{test}, M \cup \{t\}, I)$*, где* $G_{train} = \{1, \ldots, 10\}$*,* $G_{test} = \{11, \ldots, 14\}$*,* $M = \{or, oo, os, tc, tm, th, hn, w\}$*,* $t = play$*, а* I *– бинарное отношение, определенное на* $G \times M \cup \{t\}$*, такое что элемент этого отношения представлен крестом (*$\times$*) в соответствующей клетке таблицы. Диаграмма решетки формальных понятий данного контекста представлена на Рис. 1.2.*

$G\backslash M$	*or*	*oo*	*os*	*tc*	*tm*	*th*	*hn*	*w*	*play*
1			×			×			
2			×			×		×	
3		×				×			×
4	×				×				×
5	×			×			×		×
6	×			×			×	×	
7		×		×		×	×	×	×
8			×		×				
9			×	×			×		×
10	×				×		×		×
11			×		×		×	×	*?*
12		×			×			×	*?*
13		×				×	×		*?*
14	×				×			×	*?*

Таблица 1.2: Формальный контекст, соответствующий задаче классификации из [Mit97]. Признаки: or – outlook = rainy, oo – outlook = overcast, os – outlook = sunny, tc – temperature = cool, tm – temperature = mild, th – temperature = high, hn – humidity = normal, w – windy, play – играть в теннис или нет (целевой признак).

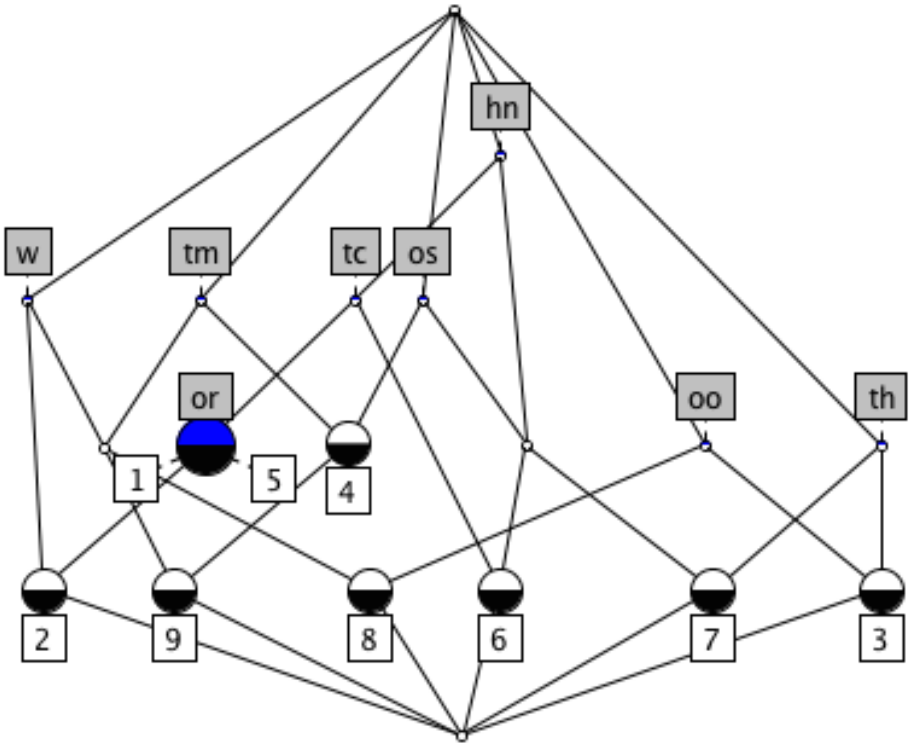


Рисунок 1.2: Решетка формальных понятий для формального контекста, представленного Таблицей 1.2.

1.3.2. Методы классификации на основе классифицирующих ассоциативных правил

В этой главе мы рассмотрим подход к классификации в машинном обучении, основанный на поиске классифицирующих ассоциативных правил, а также критерии отбора таких правил для улучшения качества и интерпретируемости классификации.

Ассоциативное правило – это утверждение об условной вероятности (называемой достоверностью) события по отношению к другому событию в совокупности с утверждением о совместной вероятности обоих событий (называемым поддержкой), где оба события описаны в терминах множеств признаков. Поиск ассоциативных правил – одна из основных задач интеллектуального анализа данных (data mining) [AS94]. Дадим определение ассоциативного правила в терминах Анализа Формальных Понятий.

Определение* 19.** ***Ассоциативное правило *формального контекста* (G, M, I) *– это выражение вида* $A \to_{c,s} B$*, где* A,$B \subseteq M$ *– подмножества признаков, а*

- $c, s \in [0, 1]$*;*
- $c = \frac{|(A \cup B)'|}{|A'|}$ *–* ***достоверность*** *(confidence, conf);*
- $s = \frac{|(A \cup B)'|}{|G|}$ *–* ***поддержка*** *(support, supp).*

Множество A *называется* ***посылкой*** *правила, а* B *–* ***заключением****.*

Если посылкой правила является набор признаков объектов обучающей выборки, а заключением – метка целевого класса объектов обучающей выборки, то такое ассоциативное правило называется **классифицирующим** [VMZ06]:

Определение* 20.** ***Классифицирующее ассоциативное правило *формального контекста* $(G, M \cup \{t\}, I)$ *– это выражение вида* $A \to_{c,s} y$*, где* $y = t$ *или* $y = \neg t$*,* $A \subseteq M$ *– подмножество признаков, а* c,s *– достоверность и поддержка правила соответственно, определяющиеся так же, как в Определении 19.*

Чаще всего мы не будем указывать поддержку правил и писать $A \to_c t$ для обозначения классифицирующего правила с достоверностью c, например $\{a,b,c\} \to_{0.8} t$. Также иногда мы будем опускать и обозначение достоверности правила. Еще в случае бинарной классификации будем обозначать целевой признак как “+” и писать $\{a,b,c\} \to$ “+”.

Определение* 21.** *Пусть в условиях Определения 17* $\{\{B_{ij} \to_{c_{ij}} t\}\}$ *– множество классифицирующих правил для классификации тестовых объектов из* G_{test}*. Здесь* $i \in 1, \ldots, |G_{test}|, j \in 1, \ldots, N_i$*, где* N_i *– число правил для классификации* i*-го примера.* ***Локальной интерпретируемостью *множества правил называется величина*

$$\frac{1}{|G_{test}|} \sum_{i=1}^{|G_{test}|} \frac{1}{N_i} \sum_{j=1}^{N_i} |B_{ij}|,$$

то есть длина посылки правила, усредненная по всем правилам для тестового объекта и по всем тестовым объектам.

Деревья решений

Дерево решений воспроизводит логические правила, позволяющие получить окончательное решение о классификации объекта с помощью ответов на иерархически организованную систему вопросов. Каждый узел в дереве решений представляет признак классифицируемого объекта, а каждая ветка – значение, которое может принимать признак. Проблема построения оптимального двоичного дерева решений NP-полна [Bre+84], поэтому существует множество эвристик для построения “почти оптимального” дерева [Mur97]. Признак, лучше всего разделяющий примеры из обучающей выборки согласно некоторому критерию типа неопределенности Джини или прироста информации [Qui93], помещается в корень дерева, далее в большинстве алгоритмов, например, в C4.5 [Qui93] и CART [Bre+84], построение дерева происходит рекурсивно, пока множество объектов, удовлетворяющих всем ограничениям значений признаков в узлах дерева (которые “читаются” по ветвям), не будет представлять один класс. Существует много методов нахождения оптимального признака для разделения обучающей выборки, но ни один из них не подтвердил наличие идеальной стратегии [Mur97]. Некоторые из этих методов будут рассмотрены далее.

Одно из главных полезных свойств деревьев решений — это их интерпретируемость. Человеку понятно, как именно, на основе каких доводов, дерево решений классифицировало какой-либо пример. Также бывает плюсом отсутствие параметров модели. Минус же деревьев решений — невысокое качество классификации по сравнению с другими, более продвинутыми алгоритмами классификации, такими как нейронные сети или композиции алгоритмов [CN06]. Например, случайный лес (random forest) [Bre01] может значительно повысить качество классификации по сравнению с одним отдельно взятым деревом, теряя, однако, при этом в интерпретируемости алгоритма. Далее в разделе 1.4.1 мы переформулируем деревья решений в терминах АФП.

Пример построенного дерева решений показан на Рис. 1.3. Здесь в каждой вершине дерева показано, какое условие в ней проверяется (кроме листовых вершин), число примеров, удовлетворяющих всем условиям от данной вершины до корня (samples), соответствующее значение неопределенности Джини (gini) и соотношение классов (value).

Классифицирующие (решающие) правила

Деревья решений можно представить набором правил, если следовать от корня дерева к листьям. Однако такие классифицирующие правила можно порождать и на основе самих данных без построения дерева. В статье [Für99] проведен обзор алгоритмов построения наименьшего возможного набора решающих правил, согласующихся с обучающей выборкой. Слишком большое количество порожденных решающих правил обычно свидетельствует о том, что алгоритм пытается “запомнить” данные, а не обнаружить закономерности в них, и часто ведет к проблеме переобучения. Поэтому большинство алгоритмов классификации, основанных на решающих

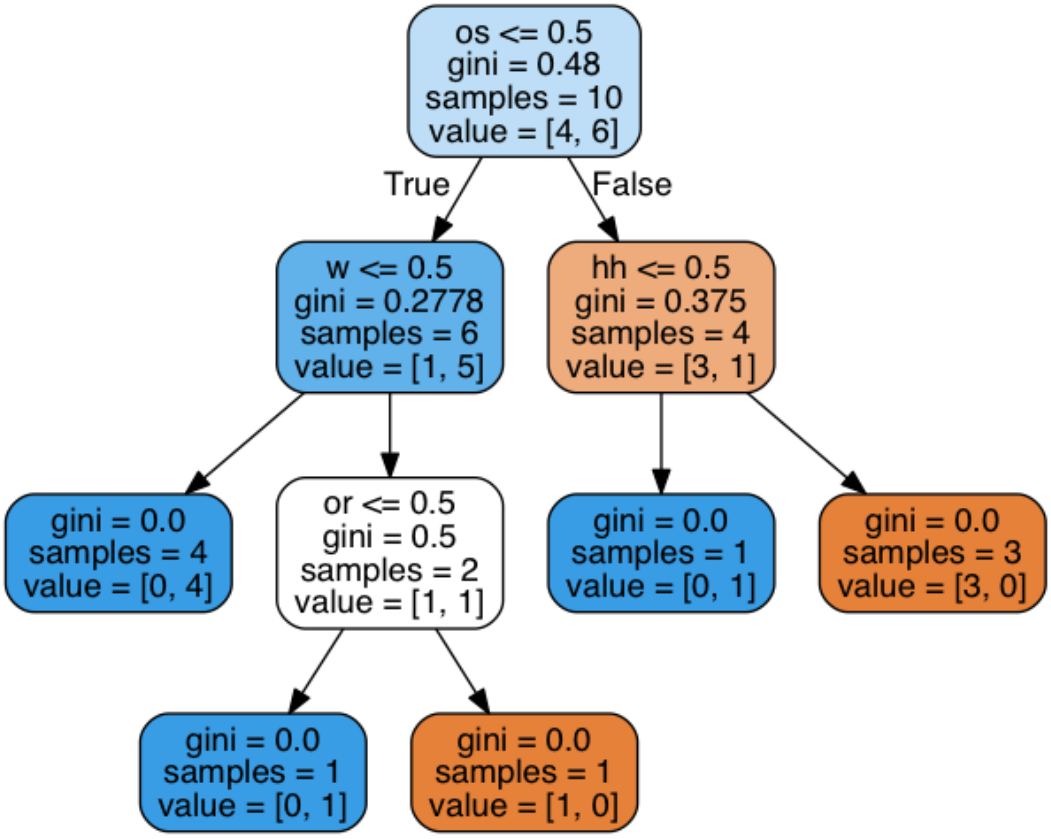


Рисунок 1.3: Дерево решений для задачи классификации с данными, представленными в Таблице 1.2 (В Scikit-learn деревья решений поддерживают только числовые признаки, так что запись $os \leq 0.5$ надо понимать как проверку на отсутствие признака os).

правилах, предлагают эвристики для отбора правил. Среди наиболее известных алгоритмов такого вида можно выделить RIPPER [Coh95] и PART [Für97]. Преимущества и недостатки решающих правил в целом те же, что и у деревьев решений – хорошая интерпретируемость, возможность учета экспертных знаний предметной области, представленных также в виде правил, но невысокое качество классификации в сложных задачах. Тем не менее, во многих задачах зачастую простые классифицирующие правила имеют хорошие результаты [Hol93a].

1.3.3. Критерии выбора классифицирующих правил

Как при построении дерева решений, так и при нахождении классифицирующих правил по выборке необходимо задать критерий отбора правил. При решении задачи классификации чаще всего такие критерии называют критериями информативности. При обучении без учителя в задаче интеллектуального анализа данных чаще такие критерии называют мерами “интересности” наборов признаков (pattern interestingness measures, см. обзор [GH06]).

Применительно к бинарным деревьям решений критерий информативности $Q(A, m)$ определяют для разбиения выборки $A \subseteq X$ по наличию или отсутствию признака m в задаче классификации с обучающей выборкой X и вектором ответов y.

***Определение* 22.** *Пусть в условиях Определения 17 A – некоторое множество объектов обучающей выборки в задаче бинарной классификации, $A \in G_{train}$. Пусть A_m – подмножество объектов из A, обладающих признаком m, а $A_{\neg m}$ – подмножество объектов из A, не обладающих признаком m, то есть $aI_{train}m \; \forall \; a \in A_m$ и $\neg(aI_{train}m) \; \forall \; a \in A_{\neg m}$.*

Тогда ***критерий информативности*** *признака m в задаче бинарной классификации с обучающей выборкой $\mathbb{K}_{train}$ определяется для множества объектов A следующим образом:*

$$Q(A, m) = F(A) - \frac{|A_m|}{|A|}F(A_m) - \frac{|A_{\neg m}|}{|A|}F(A_{\neg m}),$$

где $F(X)$ – некоторая функция $F : X \to \mathbb{R}$ с аргументом $X \subseteq 2^{G_{train}}$, а $2^{G_{train}}$ – множество всех подмножеств множества G_{train}.

В зависимости от выбора функции F в определении выше обычно выделяют следующие критерии информативности: ошибка классификации, прирост информации, неопределенность Джини (все три – в задаче классификации) и дисперсионный критерий (в задаче восстановления регрессии). Далее мы рассмотрим прирост информации и неопределенность Джини.

Прирост информации

Если дискретная случайная величина принимает значения $1, \ldots, K$ с вероятностями $p_1, \ldots, p_K$ соответственно, то энтропия этой случайной величины определяется как [Вью13]

$$H(p) \equiv H(p_1, \ldots, p_K) = -\sum_{k=1}^{K} p_k \log_2 p_k.$$

(Энтропийный) прирост информации как критерий информативности разбиения множества объектов A по признаку m в задаче классификации на K классов определяется как

$$Q_H(A, m) = H(p_A) - \frac{|A_m|}{|A|}H(p_{A_m}) - \frac{|A_{\neg m}|}{|A|}H(p_{A_{\neg m}}),$$

где A_m – подмножество объектов из A, обладающих признаком m, $A_{\neg m}$ – соответственно, не обладающих, $p_A = (\frac{|A_1|}{|A|}, \ldots \frac{|A_K|}{|A|})$ – распределение классов для объектов из A, а $A_j \subseteq A$ – подмножество объектов, отнесенных к классу j ($j = 1, \ldots, K$).

Сформулируем определение (энтропийного) прироста информации классифицирующего правила в задаче бинарной классификации.

***Определение* 23.** *Пусть дан классификационный контекст $\mathbb{K} = (G_{train} \cup G_{test}, M \cup \{t\}, I_{train} \cup I_{test})$ и целевой признак t – бинарный. Пусть $B \to_{c,s} t$ – классифицирующее ассоциативное правило (см. Определение 20), где $B \subseteq M$, c,s – достоверность и поддержка правила. Тогда множество объектов G_{train} можно разделить на 4 непересекающихся множества: $G_{train} = G_B^+ \cup G_B^- \cup G_{\neg B}^+ \cup G_{\neg B}^-$, где*

$G_B^+ = B' \cap t'$ *– множество положительных объектов, подходящих под правило $B \to_{c,s} t$;*

$G_B^- = B' \backslash t'$ – *множество отрицательных объектов, подходящих под правило* $B \to_{c,s} t$;

$G_{\neg B}^+ = t' \backslash B'$ – *множество положительных объектов, не подходящих под правило* $B \to_{c,s} t$;

$G_{\neg B}^- = G_{train} \backslash (B' \cup t')$ – *множество отрицательных объектов, не подходящих под правило* $B \to_{c,s} t$.

(Энтропийным) приростом информации *правила* $B \to_{c,s} t$ *называется величина*

$$Q_H(G_{train}, B) = H_0 - \frac{|G_B^+ \cup G_B^-|}{|G_{train}|} H_B - \frac{|G_{\neg B}^+ \cup G_{\neg B}^-|}{|G_{train}|} H_{\neg B},$$

где $H_B = H(\frac{|G_B^+|}{|G_B^+ \cup G_B^-|}, \frac{|G_B^-|}{|G_B^+ \cup G_B^-|})$, $H_{\neg B} = H(\frac{|G_{\neg B}^+|}{|G_{\neg B}^+ \cup G_{\neg B}^-|}, \frac{|G_{\neg B}^-|}{|G_{\neg B}^+ \cup G_{\neg B}^-|})$, *а* $H_0 = H(\frac{|t'|}{|G_{train}|}, \frac{|G_{train} \backslash t'|}{|G_{train}|})$.

Неопределенность Джини

Если дискретная случайная величина принимает значения $1, \ldots, K$ с вероятностями $p_1, \ldots, p_K$ соответственно, то неопределенность Джини (Gini impurity) этой случайной величины определяется как [Вью13]

$$G(p) \equiv G(p_1, \ldots, p_K) = \sum_{k=1}^{K} p_k(1 - p_k).$$

Неопределенность Джини как критерий информативности разбиения множества объектов A по признаку m в задаче классификации на K классов определяется как

$$Q_G(A, m) = G(p_A) - \frac{|A_m|}{|A|} G(p_{A_m}) - \frac{|A_{\neg m}|}{|A|} G(p_{A_{\neg m}}),$$

где A_m – подмножество объектов из A, обладающих признаком m, $A_{\neg m}$ – соответственно, не обладающих, $p_A = (\frac{|A_1|}{|A|}, \ldots \frac{|A_K|}{|A|})$ – распределение классов для объектов из A, а $A_j \subseteq A$ – подмножество объектов, отнесенных к классу j ($j = 1, \ldots, K$).

Сформулируем определение неопределенности Джини классифицирующего правила в задаче бинарной классификации.

Определение* 24.** *В условиях Определения 23* ***неопределенностью Джини *правила* $B \to_{c,s} t$ *называется величина*

$$Q_G(G_{train}, B) = \frac{|G_B^+ \cup G_B^-|}{|G_{train}|} G(\frac{|G_B^+|}{|G_B^+ \cup G_B^-|}, \frac{|G_B^-|}{|G_B^+ \cup G_B^-|}) +$$

$$\frac{|G_{\neg B}^+ \cup G_{\neg B}^-|}{|G_{train}|} G(\frac{|G_{\neg B}^+|}{|G_{\neg B}^+ \cup G_{\neg B}^-|}, \frac{|G_{\neg B}^-|}{|G_{\neg B}^+ \cup G_{\neg B}^-|}),$$

где $G(p_1, p_2) = 1 - p_1^2 - p_2^2$.

1.3.4. Методы классификации по запросу

“Ленивые” деревья решений

Алгоритм построения “ленивых” деревьев решений LazyDT [FKY96] имеет некоторые преимущества по сравнению с обычными деревьями решений. Во-первых, построенные решающие правила получаются намного короче и поэтому лучше интерпретируются. Во-вторых, при ограниченной обучающей выборке многие алгоритмы построения деревьев решений сталкиваются с проблемой сильной фрагментации [PH90]. В алгоритмах типа C4.5 [Qui93] и ID3 [Qui86] на каждом шаге построения дерева выбирается лучшее разбиение на основе среднего улучшения какого-то критерия, например, прироста информации. Поскольку выбор делается на основе усредненного значения критерия, для некоторых дочерних ветвей он может быть и отрицательным. Для объектов, которые “попадают” на такой путь в дереве, дальнейшее разбиение может приводить только к лишней фрагментации данных. В алгоритме 1 построения “ленивых” деревьев решений для каждого тестового объекта строится свой путь дерева решений, что позволяет избежать лишней фрагментации данных. На каждом шаге алгоритма выбирается разбиение, приводящее к максимальному уменьшению энтропии целевого класса.

Algorithm 1 Алгоритм построения “ленивых” деревьев решений LazyDT [FKY96].

Вход: X – обучающая выборка, t – объект из тестовой выборки
Выход: y_t – предсказанная метка целевого класса для объекта t

1. Если все объекты в X имеют одну и ту же метку l, вернуть l
2. В противном случае выбрать признак A, пусть a — значение признака A у объекта t. Пусть X' – подмножество обучающих объектов со значением признака A, равным a.
   Применить алгоритм для X'

Алгоритмы классификации на основе ассоциативных правил

Еще одна альтернатива деревьям решений – классификаторы, построенные по ассоциативным правилам, или просто ассоциативные классификаторы (Eager Associative Classifier, EAC). Как было показано [LHM98], они по качеству классификации часто превосходят деревья решений, поскольку ищут правила с глобальным максимумом критерия (например, прироста информации). Однако число порождаемых ассоциативных правил может быть очень большим.

Суть ассоциативных алгоритмов классификации, основанных на ассоциативных правилах (Eager Associative Classifiers, EAC):

- построить множество всех классифицирующих ассоциативных правил (правил, в заключении которых стоит метка целевого класса). Это можно сделать с помощью немного модифицированных версий классических алгоритмов поиска ассоциативных правил, таких как Apriori [AS94] или FP-growth [HPY00];
- отсортировать правила по некоторому критерию (например, по приросту информации);

– определять метки тестовых объектов с помощью первого “подходящего” правила (то есть с помощью правила, имеющего максимальный прирост информации среди всех правил, посылка которых является подмножеством тестового объекта).

В работе [VMZ06] показывается, что дерево решений может быть выражено в терминах EAC, и что метка каждого тестового объекта определяется с помощью EAC правилом с не меньшим приростом информации, чем соответствующее правило, построенное деревом решений.

Недостатком этого семейства алгоритмов является слишком большое количество построенных классифицирующих правил и, как следствие, высокая вычислительная сложность.

Algorithm 2 Алгоритм классификации на основе ассоциативных правил (Eager Associative Classifier) [VMZ06]

Вход: X – обучающая выборка, t – объект из тестовой выборки
Выход: y_t – предсказанная метка целевого класса для объекта t

1. Найти в X_{train} множество C ассоциативных правил вида $\{\chi \to y_i\}$, где χ – подмножество признаков из объектов X_{train}, y_i – метка целевого класса. Это можно сделать с помощью алгоритма типа Apriori
2. Отсортировать правила C по приросту информации
3. Определить метку y_t как заключение “первого подходящего” правила $\{\chi_i \to y_i\} \in C$, где χ_i – подмножество признаков объекта t

---

Классификация по запросу на основе ассоциативных правил

В отличие от алгоритмов классификации, основанных на ассоциативных правилах, при “ленивом” подходе (Lazy Associative Classification) все множество правил по обучающей выборке не строится. Суть данного подхода (этапы классификации каждого тестового объекта) [VMZ06]:

– построить “проекцию” обучающей выборки на тестовый объект – выборку, составленную из объектов, множество признаков которых имеет ненулевое пересечение с множеством признаков данного тестового объекта (предполагается, что на входе признаки уже дискретизированы);

– построить множество всех классифицирующих ассоциативных правил (правил, в заключении которых стоит метка целевого класса) по “проекции” обучающей выборки на данный тестовый объект;

– отсортировать правила по некоторому критерию (например, по приросту информации);

– определить метку данного тестового объекта с помощью первого “подходящего” правила (то есть с помощью правила, имеющего максимальный прирост информации среди построенных).

В работе [VMZ06] показывается, что “ленивый” (LAC) подход к классификации на основе ассоциативных правил позволяет для заданного множества признаков и заданного порога на минимальную поддержку правила найти все те же правила, что и ассоциативный классификатор, и,

Algorithm 3 Алгоритм классификации по запросу на основе ассоциативных правил (Lazy Associative Classifier) [VMZ06]

Вход: X_{train} – обучающая выборка, X_{test} – тестовая выборка
Выход: y_t – вектор предсказанных меток целевого класса для объекта тестовой выборки Для каждого $t_i \in X_{test}$

1. Пусть X^i_{train} – проекция обучающей выборки X_{train} на признаки объекта t_i
2. Найти в X^i_{train} множество C_{t_i} ассоциативных правил вида $\{\chi \to y_i\}$, где χ – подмножество признаков объекта t_i, y_i – метка целевого класса
3. Отсортировать правила C_{t_i} по приросту информации
4. Определить метку y_{t_i} как посылку правила из C_{t_i} с максимальным приростом информации
5. Добавить метку y_{t_i} в вектор y_t

возможно, какие-то другие, поскольку LAC может найти привила с хорошим приростом информации, которые были отброшены EAC из-за невысокой поддержки. Это во многих случаях приводит к лучшему качеству классификации. При этом в среднем правила получаются “короче” (мощность посылки правил в среднем ниже, чем у правил, полученных при EAC подходе), а значит, лучше интерпретируются.

1.3.5. Замкнутые множества признаков как компактное представление правил

Поиск частых замкнутых множеств признаков (frequent closed itemset mining) – известная задача интеллектуального анализа данных [Pas+99]. Связано это с тем, что поддержка любого генератора (см. Определение 15) замкнутого множества признаков (а это не что иное как содержание формального понятия) равна поддержке самого замкнутого множества признаков. Заметим, что в определениях выше все G^+_B, G^-_B, $G^+_{\bar{B}}$, $G^-_{\bar{B}}$ выражаются только через B' (а также G_{train} и t'). Учитывая свойство $\forall B \subseteq M\ B''' = B'$, справедливы равенства $H(B,\tau) = H(B'',\tau)$ и $G(B,\tau) = G(B'',\tau)$.

То есть значения критериев информативности $H(B,\tau)$ и $G(B,\tau)$ для правил вида $B \to \tau$, посылками которых являются генераторы B содержаний формальных понятий B'', будут такими же, как и для правил $B'' \to \tau$ с посылками в виде самих содержаний. Значит, для нахождения всего множества ассоциативных правил достаточно найти множество замкнутых множеств признаков. В терминах АФП применительно к задаче классификации это значит, что множество классифицирующих ассоциативных правил для обучающего формального контекста $\mathbb{K} = (G, M \cup \tau, I)$ можно представить в виде $\{B_i \to \tau_i\}$, где $B_i \subseteq M$ – замкнутые множества признаков ($B''_i = B_i$), $\tau_i \in \{0{,}1\}$ – значения целевого признака t.

1.4. Классификация на основе Анализа Формальных Понятий

Применение Анализа Формальных Понятий в задачах классификации активно изучается. Основной плюс таких алгоритмов – это вывод в виде интерпретируемых (понятных) человеку набо-

ров правил. Эти плюсы описаны в статье [Kuz04]. В статьях [TMM16; PGO13] даны обзор и систематизация методов классификации на основе Анализа Формальных Понятий. В статье [Каш15] рассмотрен ансамбль алгоритмов, который строится на основе АФП, а [MKK15; KK15; Kas16; KK16a] рассматривают примеры применения алгоритмов на основе АФП в задачах кредитного скоринга и графовой классификации. В статьях [KK16b; MK17] сравниваются алгоритмы на основе деревьев решений и на основе АФП, сравнение делается по качеству классификации и по интерпретируемости результатов в задаче кредитного скоринга. Далее подробней об этом.

1.4.1. Деревья решений в терминах АФП

Пусть решается задача бинарной классификации на два класса, и обучающая выборка задана формальным контекстом $\mathbb{K} = (G_+ \cup G_-, M \cup \tau, I_+ \cup I_-)$, где τ – целевой признак объектов обучающей выборки $G = G_+ \cup G_-$, M – множество признаков объектов обучающей выборки, не включая целевой признак τ, G_+ – множество объектов, обладающих целевым признаком τ (множество положительных объектов), G_- – множество объектов, не обладающих целевым признаком τ (множество отрицательных объектов), I_+ и I_- – бинарные отношения, заданные на $G_+ \times M$ и $G_- \times M$, где $(g,m) \in I_y$ означает, что объект $g \in G_y$ обладает признаком m и $y \in \{+,-\}$.

Пусть также множество признаков M **дихотомизировано** [Kuz96]: $M = M_0 \cup \neg M_0$ и для каждого признака $m \in M_0$ существует признак $\neg m \in \neg M_0$ (“отрицание” признака m): $\forall g \in G\ \neg m \in g' \iff m \notin g'$.

Будем говорить, что подмножество признаков $A \subseteq M$ [Kuz96]:

– **непротиворечиво** если $m \notin A$ или $\neg m \notin A$.

– **полно** если для каждого $m \in M$ имеет место $m \in A$ или $\neg m \in A$.

Построение произвольного дерева решений сводится к последовательному выбору признаков. Сперва мы игнорируем оптимизационный аспект, относящийся к приросту информации.

Последовательность признаков $\langle m_1, \ldots, m_k \rangle$ называется **путем решения** если множество признаков $\{m_1, \ldots, m_k\}$ непротиворечиво и существует пример $g \in G_+ \cup G_-$ такой что $\{m_1, \ldots, m_k\}' \subseteq g'$ (то есть имеется пример с тем же множеством признаков).

– Путь решения $\langle m_1, \ldots, m_i \rangle$ называется **(собственным) подпутем** пути решения $\langle m_1, \ldots, m_k \rangle$ если $i \leq k$ ($i < k$, соответственно).

– Путь решения $\langle m_1, \ldots, m_k \rangle$ называется **полным**, если объекты, обладающие множеством признаков $\{m_1, \ldots, m_k\}$, являются либо положительными либо отрицательными примерами.

– Полный путь решения называется **неизбыточным**, если ни один из его подпутей не является полным путем решения. Множество всех выбранных признаков в полном пути решения можно рассматривать как достаточное условие того, что объект обладает целевым признаком τ.

Деревом решений называется множество полных путей решения.

– **Замыканием пути решения** $\langle m_1, \ldots, m_k \rangle$ называется замыкание соответствующего множества признаков, т.е. $\{m_1, \ldots, m_k\}''$.

– Последовательность понятий с уменьшающимися объемами называется **нисходящей цепью**.

– Цепь, начинающаяся в корневой вершине, называется **корневой**.

Деревья решений и полупроизведения дихотомических шкал

Определение* 25.** *[Kuz96]* ***Полупроизведением *двух контекстов* $\mathbb{K}_1 = (G_1, M_1, I_1)$ *и* $\mathbb{K}_2 = (G_2, M_2, I_2)$ *называется* $\mathbb{K}_1 \between \mathbb{K}_2 := (G_1 \times G_2, M_1 \cup M_2, \nabla)$*, где*

$$(g_1,g_2) \nabla m_j :\iff g_j I_j m, \quad m_j \in M_j, g_j \in G_j, j \in \{1,2\}$$

Определение* 26.** *[Kuz96]* ***Дихотомической шкалой *называется формальный контекст вида* $\mathbb{D} = (\{g_1, g_2\}, \{m, \neg m\}, I)$*, где* $|G| = 2$*,* $g_1 I m, g_2 I \neg m$*.*

***Пример* 3.** *Полупроизведение* $\mathbb{D}_1 \between \mathbb{D}_2 \between \mathbb{D}_3$ *трех дихотомических шкал* $\mathbb{D}_1, D_2$ *и* $\mathbb{D}_3$ *выглядит следующим образом:*

$\mathbb{D}_1$

	a	$\neg a$
g_{11}	×	
g_{12}		×

$\mathbb{D}_2$

	b	$\neg b$
g_{21}	×	
g_{22}		×

$\mathbb{D}_3$

	c	$\neg c$
g_{31}	×	
g_{32}		×

$\mathbb{D}_1 \between \mathbb{D}_2 \between \mathbb{D}_3$

	a	$\neg a$	b	$\neg b$	c	$\neg c$
(g_{11}, g_{21}, g_{31})	×		×		×	
(g_{11}, g_{22}, g_{31})	×			×	×	
(g_{12}, g_{21}, g_{31})		×	×		×	
(g_{12}, g_{22}, g_{31})		×		×	×	
(g_{11}, g_{21}, g_{32})	×		×			×
(g_{11}, g_{22}, g_{32})	×			×		×
(g_{12}, g_{21}, g_{32})		×	×			×
(g_{12}, g_{22}, g_{32})		×		×		×

Рассмотрим контекст $\mathbb{K} = (G, M, I) = \mathbb{D}_1 \between \mathbb{D}_2 \between \ldots D_{|M|/2}$ - полупроизведение $|M|/2$ дихотомических шкал. Обозначим его $\between_M D$, где каждая дихотомическая шкала D_i соответствует паре признаков $(m,\neg m)$.

Отметим, что множество объектов G имеет размер $2^{|M|/2}$, а отношение I таково, что множество объектных содержаний есть в точности множество полных непротиворечивых подмножеств признаков.

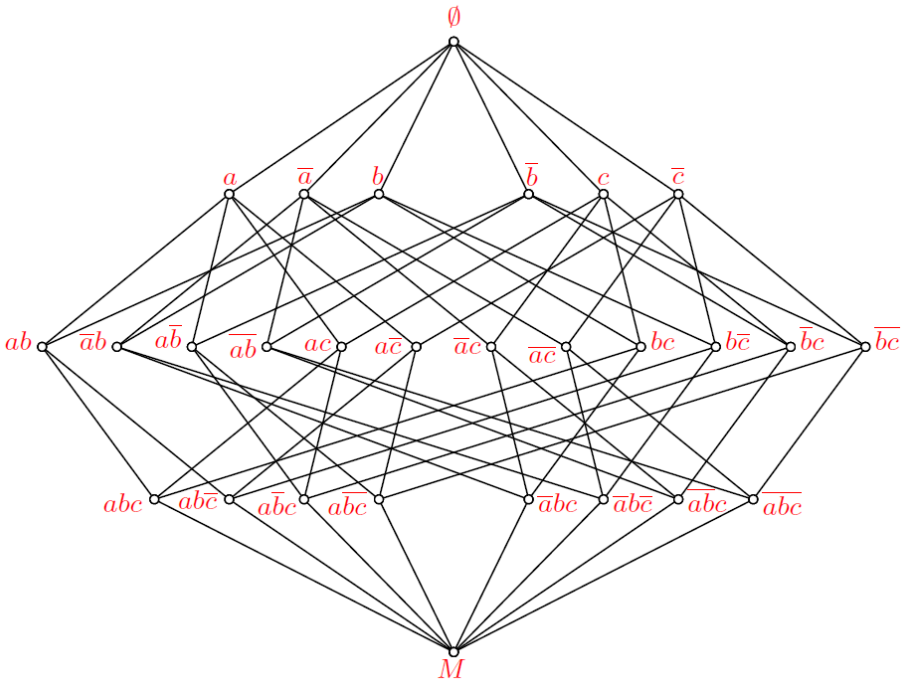


Рисунок 1.4: Решетка понятий полупроизведения трех дихотомических шкал, вершины диаграммы помечены содержаниями.

***Утверждение* 1.** *[Kuz96] Каждый путь решения есть корневая нисходящая цепь в решетке* $\underline{\mathfrak{B}}(\bigtriangledown_M\ D)$ *и каждая нисходящая цепь понятий с непустыми объемами в решетке* $\underline{\mathfrak{B}}(\bigtriangledown_M\ D)$ *есть путь решения.*

Для дихотомизированных признаков прирост информации естественно определять для пары признаков $m, \neg m \in M$.

Для пути решения $\langle m_1, \ldots, m_k \rangle$ имеет место

$$\mathrm{IG}(m, m_1, \ldots, m_k) = -\frac{|A'_m|}{|G|}\mathrm{H}(A_m) - \frac{|A'_{\neg m}|}{|G|}\mathrm{H}(A_{\neg m}),$$

где $A_m := \{m_1, \ldots, m_k, m\}$, $A_{\neg m} := \{m_1, \ldots, m_k, \neg m\}$, и для каждого $A \subseteq M$

$$\mathrm{H}(A) = -\sum_{y \in \{\tau, \neg\tau\}} p(y \mid A) \cdot \log_2 p(y \mid A),$$

где $p(\tau \mid A)$ есть условная выборочная вероятность того, что объект, обладающий множеством признаков A, также обладает целевым признаком τ. Аналогично для $p(\neg\tau \mid A)$.

При замене $\mathrm{H}(A)$ на $\mathrm{G}(A) = -\sum_{y \in \{\tau, \neg\tau\}} p(y \mid A) \cdot (1 - p(y \mid A))$ получаем неопределенность Джини $\mathrm{Gini}(m, m_1, \ldots, m_k)$ для пары признаков $m, \neg m \in M$.

В работе [Kuz96] показано, что прирост информации не изменяется при замыкании множеств признаков. Докажем аналогичное утверждение для неопределенности Джини.

***Утверждение* 2.** *Значение неопределенности Джини для пути решения равно значению неопределенности Джини для замыкания этого пути. Аналогично, значение прироста информации для пути решения равно значению прироста информации для замыкания этого пути.*

Доказательство. Пусть оператор Галуа $(\cdot)'$ связан с контекстом $\mathbb{K} = (G_+ \cup G_-, M \cup \{\tau\}, I_+ \cup I_-)$ и $A \subseteq M$ – подмножество признаков. Тогда для условной выборочной вероятности $p(\tau \mid A)$ того, что объект, обладающий множеством признаков A, также обладает целевым признаком τ, справедливо:

$$p(\tau \mid A) = \frac{|A' \cap G_\tau|}{|A'|} = \frac{|(A'')' \cap G_\tau|}{|(A'')'|} = p(\tau \mid A'')$$

по свойству оператора Галуа $(\cdot)'$: $(A'')' = A'$. Тогда $\mathrm{G}(A) = \mathrm{G}(A'')$ и для пути решения $\langle m_1, \ldots, m_k \rangle$

Это означает, что вместо решетки понятий $\underline{\mathfrak{B}}(\mathbb{K}_M D)$ можно рассматривать решетку понятий $\underline{\mathfrak{B}}(\mathbb{K}_{+-}) = \underline{\mathfrak{B}}(G_+ \cup G_-, M \cup \tau, I_+ \cup I_-)$, которая может быть намного меньше.

***Определение* 27.** *[Kuz96] Пусть T – дерево решений для контекста $\mathbb{K}_{+-} = (G_+ \cup G_-, M_0 \cup \neg M_0 \cup \tau, I_+ \cup I_-)$. Назовем дерево решений T_k, составленное из всех подпутей решения T ограниченной мощности признаков, **поддеревом** дерева решений T глубины k. $T_k = \{P \mid P \in T, |P| \leq k\}$.*

Аналогично будем говорить о глубине признакового CbO-дерева (см. Определение 16) и его поддеревьях глубины k.

***Теорема* 1.** *Пусть решается задача бинарной классификации, и обучающая выборка задана формальным контекстом $\mathbb{K}_{+-} = (G_+ \cup G_-, M \cup \tau, I_+ \cup I_-)$. Пусть также множество признаков дихотомизировано: $M = M_0 \cup \neg M_0$. Пусть для данного формального контекста построено признаковое CbO-дерево T_{CbO}. Для любого пути решения $\langle m_1, \ldots, m_j \rangle$ дерева решений T глубины k ($j \leq k$) с приростом информации $IG(\langle m_1, \ldots, m_j \rangle)$ найдется замкнутое множество признаков, являющееся вершиной CbO-дерева на глубине не более k, а также посылкой классифицирующего правила с не меньшим приростом информации, чем у $\langle m_1, \ldots, m_j \rangle$.*

Доказательство. Обозначим $B_j = \{m_1, \ldots, m_j\}$. По доказанному выше свойству 2 $IG(B_j) = IG(B_j'')$. Значит, надо показать, что $depth_{CbO}(B_j'') \leq k$, где $depth_{CbO}(A)$ – глубина CbO-дерева, на которой расположен элемент A.

Рассмотрим 2 случая, когда множество B_j замкнуто и когда оно не замкнуто.

Случай 1. Пусть $B_j = B_j''$. Значит, B_j является вершиной CbO-дерева. Покажем по индукции, что $\forall A \in T_{CbO}$ справедливо $|A| \geq depth_{CbO}(A)$.

1. Для $|A| = 1$ неравенство выполняется тривиально, так как по построению CbO-дерева в нем на глубине 1 располагаются элементы вида $a'', a \in M$ и $|a''| \geq |a| = 1$.

2. Пусть $|A| \geq depth_{CbO}(A)$ для $\forall A \in T_{CbO}$ такого что $|A| = n$. Покажем, что $\forall suc(A)$ выполнено $|suc(A)| \geq depth_{CbO}(suc(A))$.

Действительно, $suc(A) = (A \cup i)'' \Rightarrow |suc(A)| \geq |A \cup i| = |A| + 1 \geq depth_{CbO}(A) + 1 = depth_{CbO}(suc(A))$.

Итак, мы показали, что мощность любого элемента CbO-дерева не меньше глубины, на которой этот элемент располагается в CbO-дереве.

Применяя это свойство к B_j, получаем $depth_{CbO}(B_j) \leq |B_j| = j \leq k$.

Случай 2. Пусть $B_j \neq B_j''$. $\exists$ перестановка $(i_1, \ldots, i_j)$ чисел $1, \ldots, j$ такая что $\alpha(i_1) < \ldots < \alpha(i_j)$, где α – порядок на признаках, заданный в Определении 16. Тогда в CbO-дереве $\exists$ путь $(m_{i_1})'' \to (m_{i_1} \cup m_{i_2})'' \to \ldots (m_{i_1} \cup m_{i_2} \cup \ldots m_{i_j})'' = B_j''$, длина которого равна j. То есть, на глубине j в CbO-дереве $\exists$ вершина B_j'', или $depth_{CbO}(B_j'') = j \leq k$. □

Забегая вперед и говоря про реализацию алгоритма поиска посылок классифицирующих правил среди формальных понятий, отметим, что доказанная теорема означает, что для любого правила, построенного деревом решений и имеющего мощность посылки k, можно найти правило с не меньшим приростом информации при построении CbO-дерева с глубиной рекурсии k. Легко показать, что аналогичные утверждения верны и для неопределенности Джини в силу доказанного утверждения 2.

***Пример* 4.** *Покажем, что в доказанной выше теореме важно наличие отрицаний признаков в обучающем контексте. Приведем простой контрпример к утверждению выше в случае, когда отрицания признаков не добавляются.*

Рассмотрим пример обучающего формального контекста, представленного Таблицей 1.3. Сначала рассмотрим шкалирование признака "цвет" без отрицаний признаков (то есть имеются бинарные признаки "w" (белый), "y" (желтый), "g" (зеленый), "b" (синий), но нет "$\neg w$", "$\neg y$", "$\neg g$", "$\neg b$"). CbO-дерево построения множества формальных понятий для этого контекста показано на Рис. 1.7. Диаграмма решетки формальных понятий для данного контекста показана на Рис. 1.5. Рассмотрим также тестовый контекст (Таблица 1.4). Дерево решений, построенное по обучающему контексту и классифицирующее все обучающие объекты без ошибок, состоит всего из двух условий и представлено на Рис. 1.6.

Посмотрим, какими правилами классифицируются объекты тестовой выборки деревом решений:

Объект	*Правило*	*Попадает фруктов*	*Попадает не фруктов*	*Не попадает фруктов*	*Не попадает не фруктов*	*Gini*
манго	$\neg w \neg f \to$ "+"	*4*	*0*	*0*	*3*	*0*
мыло	$w \to$ "–"	*1*	*2*	*3*	*1*	*0.4*
шампиньон	$w \to$ "–"	*1*	*2*	*3*	*1*	*0.4*
арбуз	$\neg w f \to$ "–"	*0*	*1*	*4*	*2*	*0.38*

Видим, что объект "манго" классифицируется "идеальным" правилом $\neg w \neg f \to$ "+", которое не совершает ошибок на обучающем контексте. Но можно убедиться, что в CbO-дереве,

показанном на Рис. 1.7, вплоть до глубины 2 нет посылок правил с нулевой средней неопределенностью Джини.

Теперь если добавить к обучающему контексту отрицания признаков w, y, g и b, то есть признаки $\neg w, \neg y, \neg g$ и $\neg b$, уже получается 48 формальных понятий. Диаграмма решетки формальных понятий и CbO-дерево становятся слегка перегруженными и не показаны. Но можно легко убедиться, что в CbO-дереве на глубине 2 как раз будет элемент $\{\neg w, \neg f\}$ (т.к. это множество признаков замкнуто). В таком случае утверждение 1 справедливо.

$G \setminus M$	w	y	g	b	f	$\neg f$	s	$\neg s$	r	$\neg r$	***фрукт***
яблоко		×				×	×		×		+
грейпфрут		×				×		×	×		+
киви			×			×		×		×	+
слива				×		×	×			×	+
кубик			×		×		×			×	−
яйцо	×				×		×			×	−
теннисный мяч	×					×		×	×		−

Таблица 1.3: Пример обучающего формального контекста.

$G \setminus M$	w	y	g	b	f	$\neg f$	s	$\neg s$	r	$\neg r$	***фрукт***
манго		×				×	×			×	*?*
мыло	×				×		×		×		*?*
шампиньон	×					×	×			×	*?*
арбуз			×		×		×		×		*?*

Таблица 1.4: Пример тестового формального контекста.

Итак, неформально говоря, для того чтобы с помощью множества формальных понятий находить классифицирующие правила с качеством как минимум не хуже, чем у дерева решений, надо к обучающему контексту добавить отрицания исходных признаков.

1.4.2. ДСМ-метод

Одной из первых моделей машинного обучения, неявно использовавших системы замыканий и решетки, был ДСМ-метод, предложенный впервые в [Фин83] и являющийся формализацией философского метода сходства Д.С. Милля.

Метод сходства (Первое правило индуктивной логики) [Mil43]

> "Если два или большее число примеров исследуемого явления обладают только одним общим признаком, то ... [этот признак] есть причина (или следствие) данного явления."

В ДСМ-методе гипотезы относительно причины явления ищутся среди пересечений описаний положительных примеров явления.

ДСМ-метод в терминах АФП

В задаче бинарной классификации, согласно Определению 18, объекты классификационного контекста разделяются в зависимости от значения целевого признака t на [Bli+03; Kuz96]:

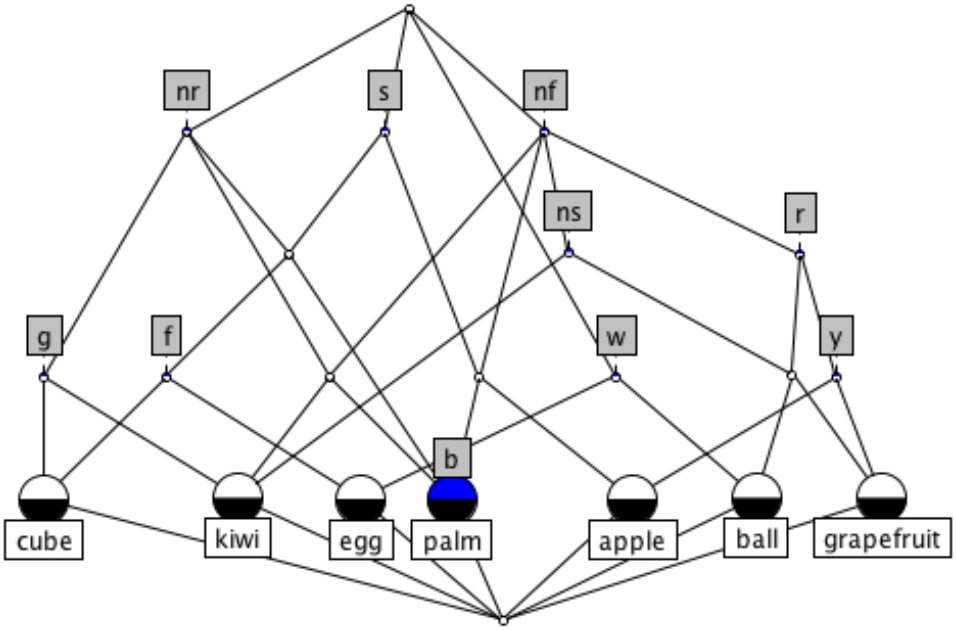


Рисунок 1.5: Диаграмма решетки формальных понятий для контекста из Таблицы 1.3.

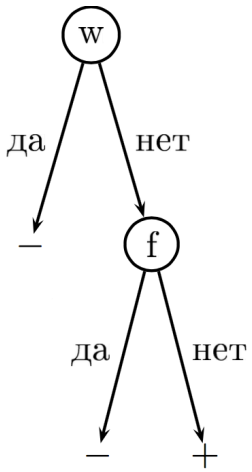


Рисунок 1.6: Дерево решений для контекста из Таблицы 1.3.

– **положительные примеры**: Множество $G_+ \subseteq G_{train}$ объектов, про которые известно, что они обладают целевым признаком t,

– **отрицательные примеры**: Множество $G_- \subseteq G_{test}$ объектов, про которые известно, что они не обладают целевым признаком t,

– **недоопределенные примеры**: Множество $G_\tau = G_{test}$ объектов, про которые не известно, обладают ли они целевым признаком или нет.

Возникают три подконтекста: $\mathbb{K}_\varepsilon := (G_\varepsilon, M, I_\varepsilon), \varepsilon \in \{-, +, \tau\}$.

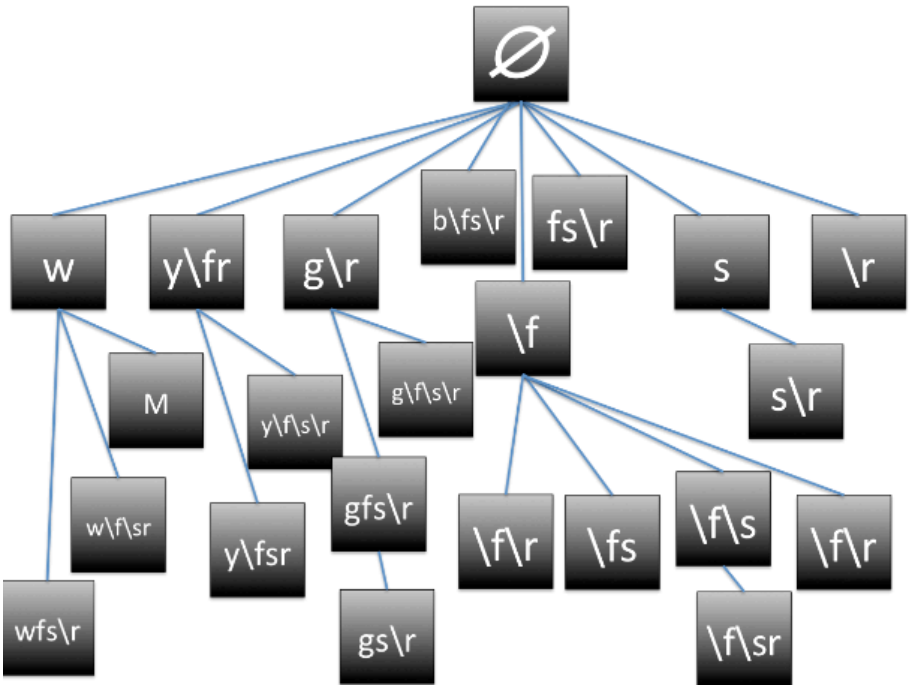


Рисунок 1.7: Признаковое CbO-дерево для контекста, представленного Таблицей 1.3.

В подконтекстах $\mathbb{K}_\varepsilon := (G_\varepsilon, M, I_\varepsilon)$, $\varepsilon \in \{-, +, \tau\}$ операторы Галуа и соответствующие операторы замыкания обозначаются через $(\cdot)^\varepsilon$, $(\cdot)^{\varepsilon\varepsilon}$.

Формальное содержание $H \subseteq M$ контекста $\mathbb{K}_+$ есть **положительная гипотеза**, если H не является подмножеством содержания ни одного отрицательного примера $g \in G_-$:

$$H^{++} = H, \quad \forall g \in G_- \quad H \not\subseteq g^-.$$

Отрицательные гипотезы определяются симметрично (с заменой + на –):

Формальное содержание $H \subseteq M$ контекста $\mathbb{K}_-$ есть **отрицательная гипотеза**, если H не является подмножеством содержания ни одного положительного примера $g \in G_+$:

$$H^{--} = H, \quad \forall g \in G_+ \quad H \not\subseteq g^+.$$

Классификация недоопределенного примера g^τ:

– Если g_τ^τ содержит в качестве подмножества положительную гипотезу и не содержит ни одной отрицательной гипотезы, то g_τ **классифицируется положительно** (предсказывается наличие целевого признака w).

– Если g_τ^τ содержит в качестве подмножества отрицательную гипотезу и не содержит ни одной положительной гипотезы, то g_τ **классифицируется отрицательно** (предсказывается отсутствие целевого признака w).

– Если g_τ^τ содержит в качестве подмножеств гипотезы обоих знаков или

если g_τ^τ вообще не содержит в качестве подмножеств ни положительных ни отрицательных гипотез, то классификация объекта, соответственно, **противоречива** или **недоопределена**.

Как следует из определения, для классификации достаточно иметь множество всех **минимальных** (относительно $\subseteq$) гипотез.

***Пример* 5.** *В Примере 2 классификационного контекста выделяются 3 подконтекста:*

$\mathbb{K}_+ = (G_+, M \cup \{t\}, I_+)$, *где* $G_+ = \{3,4,5,7,9,10\}$;

$\mathbb{K}_- = (G_-, M \cup \{t\}, I_-)$, *где* $G_- = \{1,2,6,8\}$;

$\mathbb{K}_\tau = (G_\tau, M, I_\tau)$, *где* $G_\tau = \{11,12,13,14\}$.

Здесь $M = \{or, oo, os, tc, tm, th, hn, w\}$, $t = play$.

Решетки формальных понятий контекстов $\mathbb{K}_+$ *и* $\mathbb{K}_-$ *изображены на Рисунке 1.8 слева и справа соответственно. Красным обведены минимальные положительные и отрицательные гипотезы:* $H_+ = \{\{oo,th\}, \{os,tc,hn\}, \{or,tm\}\}$ *и* $H_- = \{\{os,tm\}, \{os,th\}, \{or,tc,hn.w\}\}$.

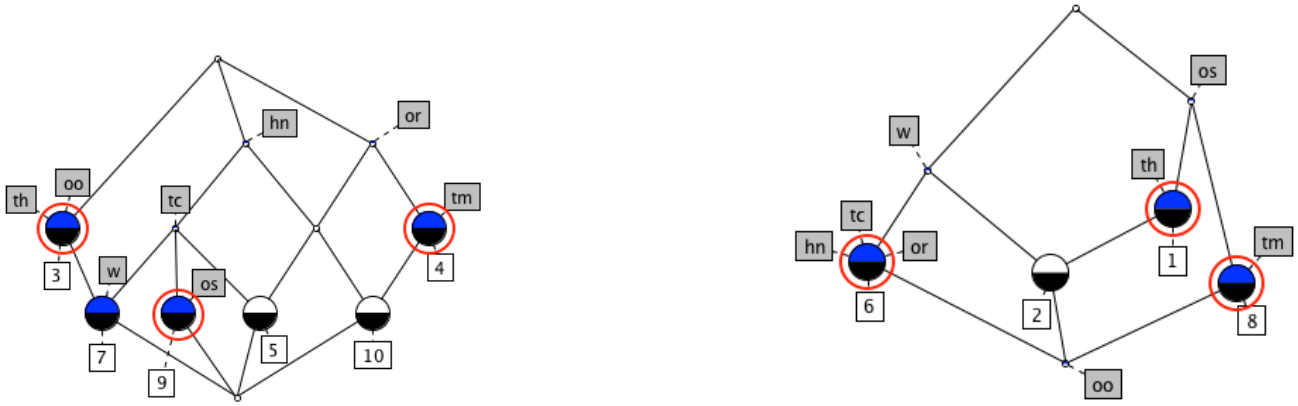


Рисунок 1.8: Решетки формальных понятий положительного (слева) и отрицательного (справа) контекстов Примера 5.

Объекты 11 и 12 из G_τ *классифицируются ДСМ-методом неопределенно, поскольку* $\nexists\, h \in H_+ \cup H_i : h \subseteq 11', 12'$ *(или проще, для них нет "подходящих" гипотез). Объекты 13 и 14 классифицируются положительно (*$\{oo,th\} \subseteq 13', \{or,tm\} \subseteq 14'$*).*

1.5. Заключение

В Главе 1 мы рассмотрели основные термины и понятия, принятые в теории решеток и Анализе Формальных Понятий (Раздел 1.2), а также постановку задачи классификации в машинном обучении (Раздел 1.3). Был представлен обзор методов классификации, основанных на классифицирующих ассоциативных правилах, в том числе деревьев решений и методов классификации по запросу. Также задача классификации была сформулирована в терминах АФП. В Разделе 1.4 были рассмотрены методы классификации, основанные на АФП, в том числе в терминах АФП были

представлены деревья решений. Также было доказано утверждение про то, как с помощью множеств формальных понятий находить классифицирующие правила гарантированно не хуже тех, что строятся деревом решений, по критерию качества типа прироста информации или неопределенности Джини.

Далее в Главе 2 мы перейдем от объектно-признаковых описаний данных в задаче классификации к данным, в которых объекты представляются сложными описаниями. Для этого введем основные термины, связанные с узорными структурами и их проекциями.

Глава 2

Узорные структуры и их проекции

2.1. Введение

В классической постановке Анализа Формальных Понятий входные данные представляются бинарным отношением (формальным контекстом) и на его основе строится решетка формальных понятий. Однако зачастую объекты либо представляются не бинарными признаками (а, скажем, вещественными или строковыми) либо вообще не могут быть описаны вектором признаков. В таких случаях будем говорить, что объект задается сложным описанием. В Разделе 2.2 мы рассмотрим аппарат узорных структур (Pattern Structures) [GK01], который позволяет расширить методы Анализа Формальных Понятий на случай, когда объекты задаются не бинарными признаками, а сложными описаниями. Такими описаниями могут быть интервалы числовых значений, множества последовательностей, строк или графов.

В Разделе 2.3 будут рассмотрены подходы к классификации данных со сложной структурой (последовательности, строки, графы) на основе ядерных функций и метода опорных векторов, а в Разделе 2.5 будет дано описание методов классификации данных со сложной структурой на основе узорных структур и их проекций.

2.2. Узорные структуры

***Определение* 28.** *[GK01] Узорная структура – это тройка* $(G,(D, \sqcap), \delta)$*, где* G *– множество объектов,* $(D, \sqcap)$ *– полная полурешётка всевозможных описаний, а* $\delta\colon G \to D$ *– функция, которая сопоставляет каждому объекту из множества* G *его описание из* D*.*

Соответствие Галуа между подмножествами множества объектов и множеством описаний для узорной структуры $(G, (D, \sqcap), \delta)$ записывается следующим образом:

$$A^\diamond := \prod_{g \in A} \delta(g), \qquad \text{где } A \subseteq G$$

$$d^\diamond := \{g \in G \mid d \sqsubseteq \delta(g)\}, \quad \text{где } d \in D.$$

Здесь $\sqsubseteq$ – это отношение поглощения, однозначно задающееся через полурешёточную операцию как: $a \sqsubseteq b \Leftrightarrow a \sqcap b = a$. Для одного объекта $a \in G$ $a^\diamond$ понимаем как $\{a\}^\diamond = \delta(a)$.

***Определение* 29.** *[GK01] Узорное понятие узорной структуры $(G, (D, \sqcap), \delta)$ – это пара (A, d), в которой $A \subseteq G$ – подмножество множества объектов, $d \in D$ – одно из описаний из полурешётки $(D, \sqcap)$, такие что $A^\diamond = d$ и $d^\diamond = A$. Множество объектов A называется узорным объёмом понятия, а d – его узорным содержанием.*

Как и в классическом случае бинарных признаков, объём понятия – это максимальное множество объектов, разделяющих одно описание, которое не может быть дальше уточнено. Узорные понятия упорядочены отношением $(A_1,d_1) \leq (A_2,d_2) \Leftrightarrow A_1 \subseteq A_2$ (что также равносильно $d_1 \sqsupseteq d_2$) и формируют решётку $\mathcal{L}(G,(D, \sqcap), \delta)$.

Узорная структура может быть построена для произвольных описаний, на множестве которых определено отношение частичного порядка. Во многих задачах такой частичный порядок соответствует отношениям часть–целое или подкласс–класс. При этом суть операций $\sqcap$ и $\sqsubseteq$ – сходство описаний и поглощение одного описания другим.

Например, для данных, описываемых графами, естественный частичный порядок задается отношением изоморфизма подграфу и может быть применен для анализа молекулярных графов на основе их подструктур [KS05]. Для данных, представленных последовательностями некоторых элементов полурешетки, естественно вводится отношение поглощения через понятие подпоследовательности. В этом контексте узорные структуры были применены для анализа последовательностей (траекторий) госпитализации пациентов с целью выявления связи между причинами госпитализации и соответствующими процедурами [Buz+13; Buz+16].

2.2.1. Проекции узорных структур

Количество формальных понятий в решётке, построенной по формальному контексту, может быть экспоненциальным от количества объектов [GW97]. Формальный контекст – это частный случай узорных структур, и поэтому количество узорных понятий в решётке, построенной для некоторой узорной структуры, может быть экспоненциальным от количества объектов в множестве G. Значит, построение полной полурешётки узорных понятий может быть очень вычислительно сложным. Более того, большинство найденных узорных понятий не интересны для дальнейшего исследования, хотя занимают существенную часть времени вычислений. В случае, когда сама полурешёточная операция сходства вычислительно сложна, построение решётки узорных понятий может стать невозможным. Например, в качестве полурешёточной операции сходства на узорной структуре на графах нужно определять изоморфизм подграфу [KS05], что является NP-полной задачей. Для сокращения времени работы алгоритмов построения узорных решёток были введены проекции узорных структур [GK01]. Проекция может быть рассмотрена как способ фильтрации полурешётки описаний с определенными математическими свойствами. Эти свойства позволяют задать связь между понятиями в спроецированной и начальной узорных структу-

рах. К тому же полурешетка, построенная для спроецированной узорной структуры может оказаться значительно меньше исходной, что упрощает ее построение и исследование.

***Определение* 30.** *[GK01] Проекция полурешётки* $(D, \sqcap)$ *– это функция* $\psi : D \to D$*, которая является оператором ядра [GK01], т.е. для любых двух* $x, y \in D$ *верно:*

- $x \sqsubseteq y \Rightarrow \psi(x) \sqsubseteq \psi(y)$ *(монотонность)*
- $\psi(x) \sqsubseteq x$ *(сжимаемость)*
- $\psi(\psi(x)) = \psi(x)$ *(идемпотентность)*

***Определение* 31.** *[GK01] Проекция узорной структуры, полученная из узорной структуры* $(G, (D, \sqcap), \delta)$ *с помощью проекции* ψ *– это такая узорная структура* $(G_\psi, (D_\psi, \sqcap_\psi), \delta_\psi)$*, в которой* $G_\psi = G$*,* $D_\psi = \psi(D) = \{d \in D \mid d = \psi(d)\}$*, с полурешёточной операцией* $\sqcap_\psi$ *такой, что* $\forall x, y \in D\ x \sqcap_\psi y := \psi(x \sqcap y)$*, а* $\delta_\psi = \psi \circ \delta$*.*

В работе [Буз15] показано, что определение 31 корректно, то есть что проекция узорной структуры тоже является узорной структурой в смысле определения 29 ($(D_\psi, \sqcap_\psi)$ – полурешётка). Там же доказываются еще несколько утверждений о связи узорных понятий в исходной и спроецированной узорных структурах.

2.2.2. Интервальные узорные структуры

Для анализа данных с вещественными значениями признаков в Анализе Формальных Понятий вводятся интервальные узорные структуры.

Описания D объектов узорной структуры образуют полную полурешетку $(D, \sqcap)$, где $\sqcap$ – коммутативная, ассоциативная и идемпотентная операция, определенная на описаниях объектов. Интуитивный смысл этой операции – “сходство” описаний. Для интервалов операция сходства $\sqcap$ определяется следующим образом:

***Определение* 32.** *[Kay+11] Пусть* $[a_1,b_1]$ *и* $[a_2,b_2]$ *– два интервала на множестве действительных чисел, расширенном до включения* $\{-\infty, +\infty\}$ *т.е.* $a_1, b_1, a_2, b_2 \in \mathbb{R}^+ = \mathbb{R} \cup \{-\infty, +\infty\}, a_1 \leq b_1, a_2 \leq b_2$*. Тогда операция сходства для двух интервалов определяется как* $[a_1, b_1] \sqcap [a_2, b_2] = [\min(a_1, a_2),\ \max(b_1, b_2)]$.

***Утверждение* 3.** *Операция сходства, заданная на интервалах в Определении 32, удовлетворяет Определению 3 полурешеточной операции.*

Доказательство. Все 4 свойства полурешеточной операции в данном случае очень просто проверить.

- Идемпотентность: $[a, b] \sqcap [a, b] = [a, b]$;
- Коммутативность: $[a_1, b_1] \sqcap [a_2, b_2] = [\min(a_1, a_2),\ \max(b_1, b_2)] = [a_2, b_2] \sqcap [a_1, b_1]$;

– Ассоциативность: $([a_1,b_1] \sqcap [a_2,b_2]) \sqcap [a_3,b_3] = [\min(a_1,a_2),\ \max(b_1,b_2)] \sqcap [a_3,b_3] = [\min(a_1,a_2,a_3),\ \max(b_1,b_2,b_3)] = [a_1,b_1] \sqcap ([a_2,b_2] \sqcap [a_3,b_3])$;

– $\exists\, e = [-\infty,+\infty]\ \forall a,b \in \mathbb{R}^+, a \leq b:\ [a,b] \sqcap e = e$.

□

Заметим, что определенная полурешёточная операция сходства на интервалах не соответствует интуитивному представлению о сходстве интервалов как их пересечении. Результатом такой полурешеточной операции будет больший интервал, соответствующий большему множеству объектов, подобно тому как при уменьшении множества бинарных признаков число объектов, обладающих всеми этими признаками, увеличивается (другими словами, операторы Галуа, определенные на множествах объектов и признаков, обладают свойством экстенсивности).

Частичный порядок на интервалах $\sqsubseteq$ (отношение поглощения) задается через полурешёточную операцию следующим образом (согласно Определению 4):

$$[a_1,b_1] \sqsubseteq [a_2,b_2] \Leftrightarrow [a_1,b_1] \sqcap [a_2,b_2] = [a_1,b_1] \Leftrightarrow a_1 \leq a_2, b_1 \geq b_2.$$

Определения операций $\sqcap$ и $\sqsubseteq$ позволяют применять их покомпонентно к векторам интервалов:

***Определение* 33.** *[Kay+11] Полурешеточная операция (операция сходства) для двух векторов интервалов* $\langle [a_{1i},b_{1i}]\rangle_{i\in[1,m]}$ *и* $\langle [a_{2i},b_{2i}]\rangle_{i\in[1,m]}$ *(где* $m \in \mathbb{N}$*) определяется как покомпонентное сходство составляющих интервалов:*

$$\langle [a_{1i},b_{1i}]\rangle_{i\in[1,m]} \sqcap \langle [a_{2i},b_{2i}]\rangle_{i\in[1,m]} = \langle [a_{1i} \sqcap a_{2i}, b_{1i} \sqcap b_{2i}]\rangle_{i\in[1,m]}.$$

Операция поглощения для двух векторов интервалов задается через покомпонентное поглощение интервалов:

$$\langle [a_{1i},b_{2i}]\rangle_{i\in[1,m]} \sqsubseteq \langle [a_{2i},b_{2i}]\rangle_{i\in[1,m]} \Leftrightarrow [a_{1i},b_{1i}] \sqsubseteq [a_{2i},b_{2i}],\ i \in [1,m].$$

Определение* 34.** *Пусть* G *– некоторое множество, понимаемое как множество объектов,* $(D,\sqcap)$ *– полурешетка описаний объектов, где полурешёточная операция* $\sqcap$ *задается в Определении 33, а каждый элемент* $d \in D$ *– вектор интервалов* $d = \langle [a_i,b_i]\rangle_{i\in[1,m]}, m \in \mathbb{N}$*. Пусть* $\delta : G \to D$ *– отображение. Тогда* $(G,(D,\sqcap),\delta)$ *называется* ***интервальной узорной структурой*.*

***Пример* 6.** *В Таблице 2.1 представлена простая интервальная узорная структура. Здесь признаки объектов не бинарны и не вещественны, а представляются интервалами вещественных чисел. Описаниями объектов являются вектора интервалов.*

Например, $g_2^\diamond = \langle$[5,7],[4,6],[2,5]$\rangle$*. Сходством двух векторов интервалов будет вектор, состоящий из интервалов, каждый из которых будет выпуклой оболочкой двух соответствующих интервалов (согласно Определению 32).*

Например, $g_2^\diamond \sqcap g_3^\diamond = \langle$[5,7],[4,6],[2,5]$\rangle \sqcap \langle$[1,9],[2,7],[6,6]$\rangle = \langle$[1,9],[2,7],[2,6]$\rangle$.

Множество $\{g_2, g_3\}$ *не является замкнутым, так как* $\{g_2,g_3\}^{\diamond\diamond} = \langle[1,9],[2,7],[2,6]\rangle^{\diamond} = \{g_1, g_2, g_3\}$.

	m_1	m_2	m_3
g_1	[1,3]	[3,5]	[2,4]
g_2	[5,7]	[4,6]	[2,5]
g_3	[1,9]	[2,7]	[6,6]

Таблица 2.1: Узорная структура на интервалах [Буз15].

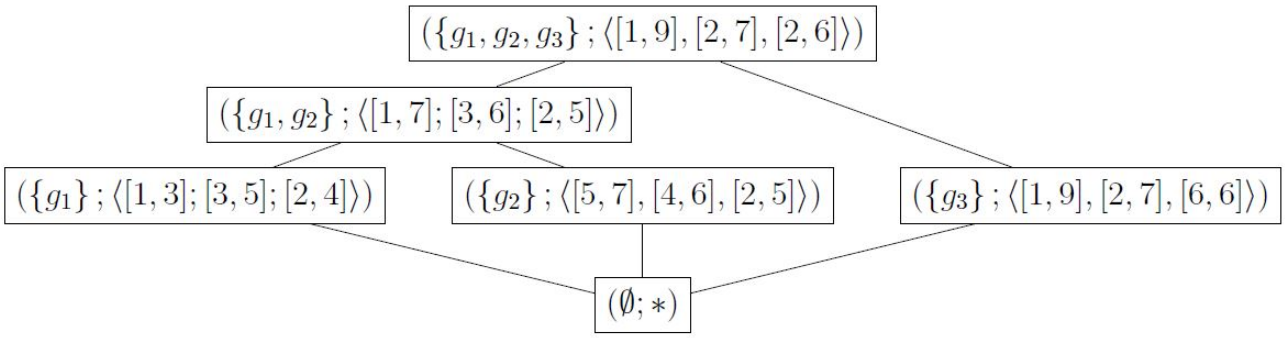


Рисунок 2.1: Решётка узорных понятий для узорной структуры из Таблицы 2.1 [Буз15].

В статье [Kuz09] доказывается наличие изоморфизма между решеткой понятий интервальной узорной структуры и решеткой понятий формального контекста, полученного межпорядковым шкалированием соответствующего многозначного контекста.

***Пример* 7.** *Рассмотрим многозначный контекст, представленный Таблицей 2.2.*

	a
1	*4.6*
2	*4.7*
3	*4.9*
4	*5.0*
5	*5.1*

	$a \leq 4.6$	$a \leq 4.7$	$a \leq 4.9$	$a \leq 5.0$	$a \leq 5.1$	$a \geq 4.6$	$a \geq 4.7$	$a \geq 4.9$	$a \geq 5.0$	$a \geq 5.1$
1	×	×	×	×	×	×				
2		×	×	×	×	×	×			
3			×	×	×	×	×	×		
4				×	×	×	×	×	×	
5					×	×	×	×	×	×

Таблица 2.2: Простой многозначный контекст и межпорядковая шкала.

Решетка формальных понятий, соответствующая контексту, полученному межпорядковым шкалированием, представлена на Рисунке 2.2.

Интервальная узорная структура, соответствующая многозначному контексту в Таблице 2.2 (слева), – это тройка $\{G, (D, \sqcap), \delta\}$, *где*

- $G = \{1,2,3,4,5\}$ – *множество объектов;*
- $D = \{[a,b]\}, a,b \in \mathbb{R} \cup \{-\infty, +\infty\}, a \leq b$ – *множество интервалов (описаний);*
- $\sqcap$ – *полурешёточная операция сходства для интервалов (согласно Определению 32);*

– $\delta : G \to D$ – *функция, сопоставляющая каждому объекту из* G *описание (интервал) из* D. *В примере* $\delta(1) = [4.6,4.6], \ldots, \delta(5) = [5.1,5.1]$.

На Рисунке 2.2 показана решетка узорных понятий, соответствующая описанной выше интервальной узорной структуре. Она изоморфна решетке на Рисунке 2.2, построенной по шкалированному формальному контексту (Таблица 2.2 (слева)).

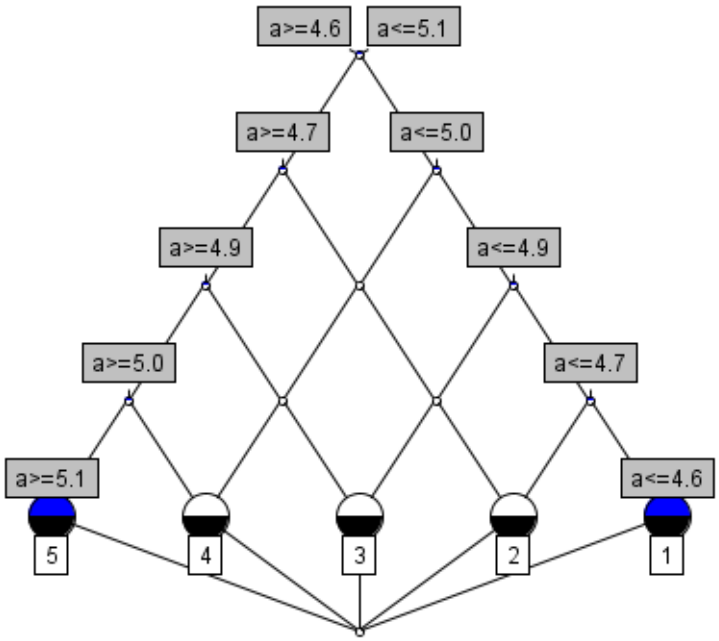


Рисунок 2.2: Решетка формальных понятий, соответствующая контексту справа в Таблице 2.2.

Интервальные узорные структуры были успешно применены для анализа экспрессии генов [Kay+11]. В этой задаче каждый ген описывается степенью своей экспрессии в определённых условиях. Таким образом, задано несколько признаков одного гена, соответствующих условиям и имеющие численные значения. Такая узорная структура может выглядеть, как в Таблице 2.1, где m_1,m_2,m_3 соответствуют условиям, а интервалы – изменчивости экспрессии определенного гена. Полурешеточная операция для такой интервальной узорной структуры задается Определением 33. Решетка, соответствующая узорной структуре из Таблицы 2.1, показана на Рисунке 2.1.

2.2.3. Проекции интервальных узорных структур

Для последующего сравнения деревьев решений и алгоритмов машинного обучения на основе АФП введем специальный вид проекций интервальных узорных структур [KK16b; Каш16].

***Определение* 35.** *Пусть* $(G, (D, \sqcap), \delta)$ *– интервальная узорная структура и* m *– размерность векторов описаний (см. Определение 34). Пусть* $T_i = \{\tau_{i1}, \ldots, \tau_{it_i}\}$ $(\tau_{ij} \in \mathbb{R}^+ = \mathbb{R} \cup \{-\infty, +\infty\}, i =$

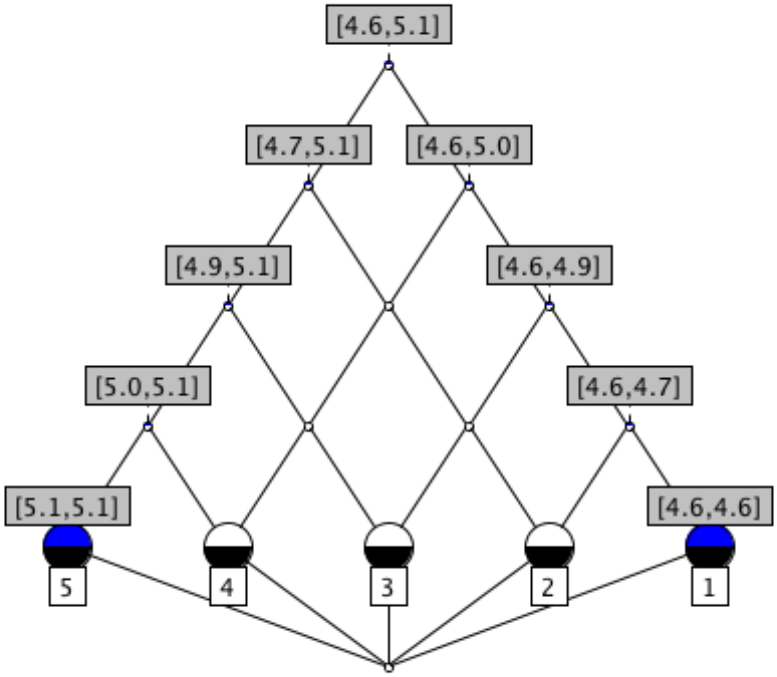


Рисунок 2.3: Решетка узорных понятий, соответствующая интервальной узорной структуре для контекста справа в Таблице 2.2.

$1, \dots m, j = 1, \dots, t_i \;\; i, j, m, t_i \in \mathbb{N}$*) – множества вещественных чисел. Тогда,* $\psi(\langle [a_i,b_i] \rangle_{i \in [1,m]}) = \langle [\max\{\tau \mid \tau \in T_i \cup \{-\infty, +\infty\}, \tau \leq a_i\}, \min\{\tau \mid \tau \in T_i \cup \{-\infty, +\infty\}, \tau \geq b_i\}] \rangle$ *называется* ***дискретизацией*** *интервальной узорной структуры* $(G, (D, \sqcap), \delta)$.

***Утверждение* 4.** *Дискретизация интервальной узорной структуры, введенная в Определении 35, является проекцией по Определению 30.*

Доказательство. Для краткости покажем, что свойства проекций справедливы для $m = 1$ (описания состоят только из одного интервала):
$\psi([a,b]) = [\max\{\tau \mid \tau \in T \cup \{-\infty, +\infty\}, \tau \leq a\}, \min\{\tau \mid \tau \in T \cup \{-\infty, +\infty\}, \tau \geq b\}]$, где $T = \{\tau_i\}_{i \in [1,t]}$.

Обозначим $T^+ = T \cup \{-\infty, +\infty\}$.

– Пусть $[a_1, b_1] \sqsubseteq [a_2, b_2]$. Тогда $\psi([a_1,b_1]) \sqcap \psi([a_2,b_2]) =$
$[\max\{\tau \mid \tau \in T^+, \tau \leq a_1\}, \min\{\tau \mid \tau \in T^+, \tau \geq b_1\}] \sqcap$
$[\max\{\tau \mid \tau \in T^+, \tau \leq a_2\}, \; \min\{\tau \mid \tau \in T^+, \tau \geq b_2\}] =$
$[\max\{\tau \mid \tau \in T^+, \tau \leq min(a_1, a_2)\}, \; \min\{\tau \mid \tau \in T^+, \tau \geq max(b_1, b_2)\}] =$
$[\max\{\tau \mid \tau \in T^+, \tau \leq a_1\}, \; \min\{\tau \mid \tau \in T^+, \tau \geq b_1\}] = \psi([a_1,b_1])$.
Значит, $\psi([a,b])$ монотонна;

– Очевидно, $\max\{\tau \mid \tau \in T^+, \tau \leq a_1\} \leq a_1$ и $\min\{\tau \mid \tau \in T^+, \tau \geq b_1\} \geq b_1$. Значит, $\psi([a,b]) \sqsubseteq [a,b]$ и $\psi([a,b])$ обладает свойством сжимаемости;

– $\psi(\psi([a,b])) = \psi([\max\{\tau \mid \tau \in T^+, \tau \leq a_1\}, \min\{\tau \mid \tau \in T^+, \tau \geq b_1\}]) =$
$\psi([\max\{\tau \mid \tau \in T^+, \tau \leq \max\{\tau \mid \tau \in T^+, \tau \leq a_1\}\},$
$\min\{\tau \mid \tau \in T^+, \tau \geq \min\{\tau \mid \tau \in T^+, \tau \geq b_1\}\}] =$
$\psi([\max\{\tau \mid \tau \in T^+, \tau \leq a_1\}, \min\{\tau \mid \tau \in T^+, \tau \geq b_1\}]) = \psi([a,b])$.
Значит, $\psi([a,b])$ идемпотентна.

Мы показали, что для $m = 1$ функция
$\psi([a,b]) = [\max\{\tau \mid \tau \in T \cup \{-\infty, \infty\}, \tau \leq a\}, \min\{\tau \mid \tau \in T \cup \{-\infty, \infty\}, \tau \geq b\}]$
является монотонной, сжимающей и идемпотентной, то есть проекцией. Для $m > 1$ рассуждения аналогичны.

□

Далее дискретизацию узорной структуры также будем называть дискретизирующей проекцией.

	$a \leq 4.65$	$a \leq 4.95$	$a \geq 4.65$	$a \geq 4.95$
1	×	×		
2		×	×	
3		×	×	
4			×	×
5			×	×

Таблица 2.3: Контекст, полученный дискретизированием признака a из Примера 6 порогами 4.65 и 4.95.

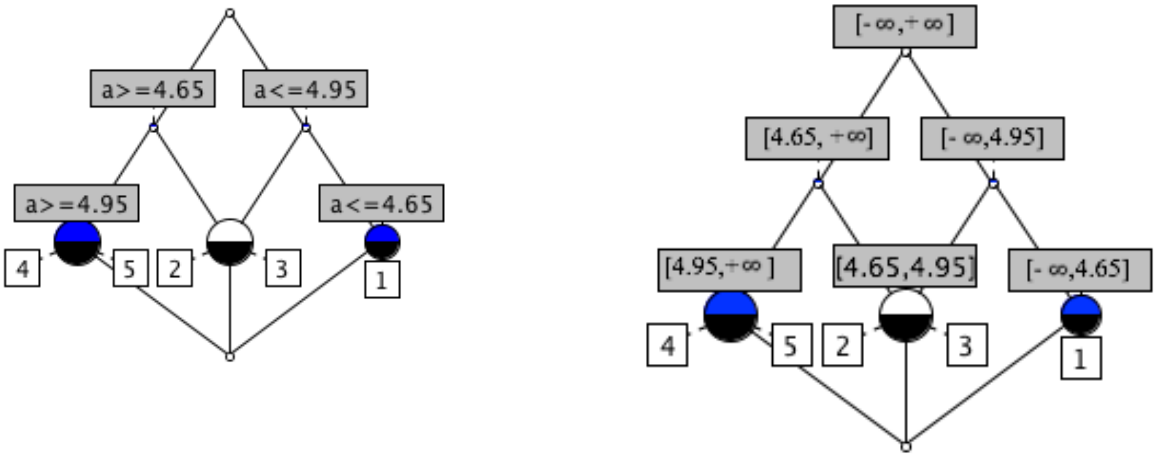


Рисунок 2.4: Решетка формальных понятий для контекста из Таблицы 2.3 и изоморфная ей решетка узорных понятий для узорной структуры из Примера 8.

***Пример* 8.** *Для набора данных из Примера 6 дискретизируем признак a порогами $T = \{4.65, 4.95\}$. Полученный формальный контекст представлен Таблицей 2.3, а соответствующая решетка формальных понятий показана на Рисунке 2.4 (слева).*

$\psi([a,b]) = [\max\{\tau \mid \tau \in T^+, \tau \leq a\}, \min\{\tau \mid \tau \in T^+, \tau \geq b\}]$ с $T^+ = \{-\infty, 1.5, 3.5, +\infty\}$ – это проекция полурешетки из Примера 6, а соответствующая решетка узорных понятий изоморфна решетке формальных понятий дискретизированного контекста (Рис. 2.4 (слева)) и показана на Рис 2.4 (справа).

Проекция ψ сопоставляет каждому узорному понятию из Примера 6 узорное понятие спроецированной узорной структуры.

Графически это можно представить так, как показано на Рисунке 2.5. Здесь иллюстрируется, что $\psi([4.6,4.8]) = [-\infty, 4.95]$. Левой границей результата проекции будет максимальный порог, меньший 4.6 или $-\infty$, если такого нет, а правой – минимальный порог, больший 4.8 или $+\infty$, если такого нет. То есть суть проекции – максимальный по вложению (т.е. минимальный в "привычном" смысле, по длине) интервал, который поглощается данным ($[4.6,4.8]$), и границами которого являются заданные пороги $4.65,4.95$ а также $-\infty$ и $+\infty$.

Полностью для данной узорной структуры дискретизирующая проекция задается Таблицей 2.4.

$[a,b]$	$\psi([a,b])$
$[4.6,4.6]$	$(-\infty, 4.65]$
$[4.7,4.7]$, $[4.9,4.9]$, $[4.7,4.9]$	$[4.65,4.95]$
$[5.0,5.0]$,$[5.1,5.1]$,$[5.0,5.1]$	$[4.95, +\infty)$
$[4.7,5.0]$, $[4.7,5.1]$, $[4.9,5.0]$, $[4.9,5.1]$	$[4.65, +\infty)$
$[4.6,4.7]$, $[4.6,4.9]$	$(-\infty, 4.95]$
$[4.6,5.0]$,$[4.6,5.1]$	$(-\infty, +\infty)$

Таблица 2.4: Значения дискретизирующей проекции ψ.

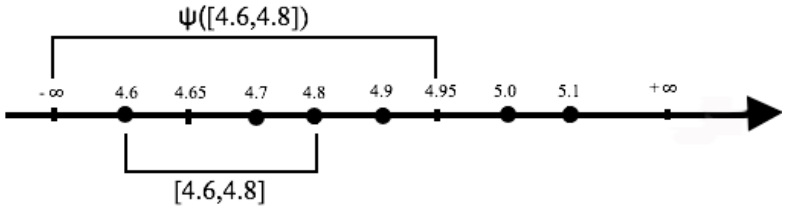


Рисунок 2.5: Пояснение к примеру с дискретизирующей проекцией.

Наконец, дискретизирующая проекция может быть иллюстрирована с помощью Рисунка 2.6 как отображение из одного множества узорных понятий в другое. Слева нарисована решетка узорных понятий исходной интервальной узорной структуры, справа – спроецированная решетка. Зеленые линии задают проекцию.

2.2.4. Постановка задачи классификации для узорных структур

Введенные нами в Главах 1.2 и 1.3 определения задачи бинарной классификации для формального контекста (Определение 17), ассоциативных правил (Определение 19), классифицирую-

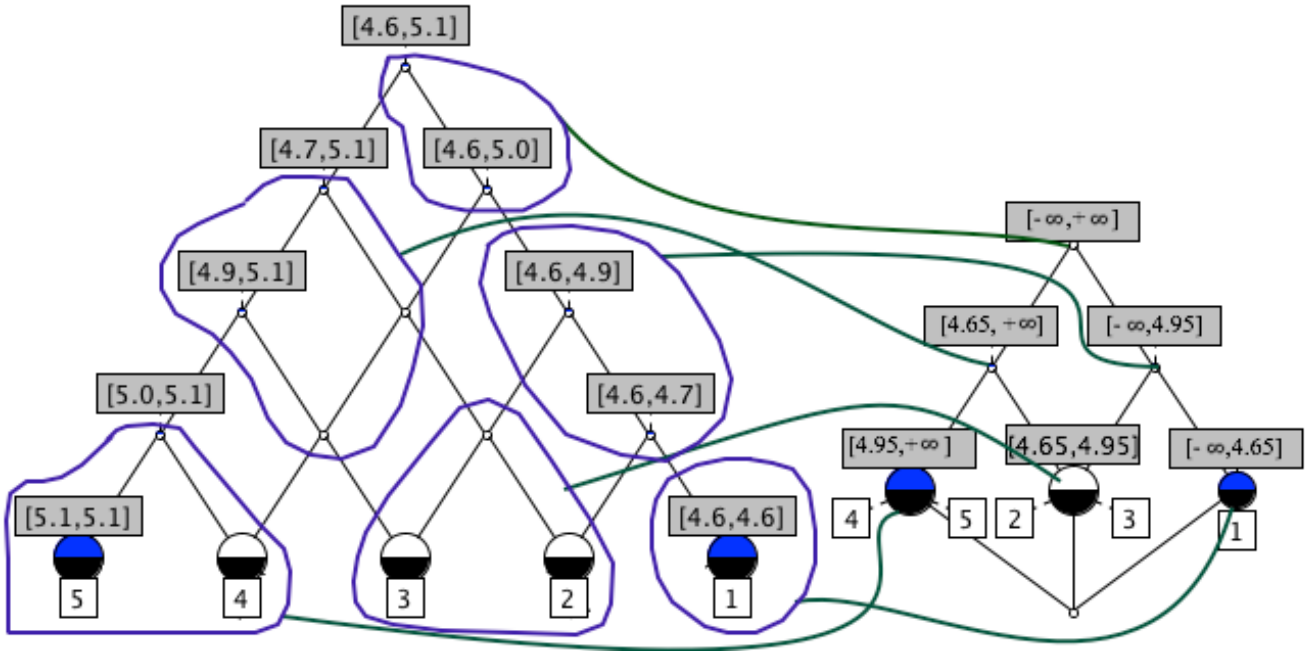


Рисунок 2.6: Дискретизирующая проекция как отображение из одного множества узорных понятий в другое.

щих ассоциативных правил (Определение 20), локальной интерпретируемости множества правил (Определение 21), критерия информативности признака в задаче бинарной классификации (Определение 22), энтропийного прироста информации классифицирующего правила (Определение 23) и CbO-дерева (Определение 16) можно адаптировать и для узорных структур с целью получения Теоремы 2, подобной Теореме 1, только для узорных структур, а не формальных контекстов.

Определение* 36.** *Пусть даны* $(G_{train}, (D, \sqcap), \delta_{train})$ *и* $(G_{test}, (D, \sqcap), \delta_{test})$ *–* ***обучающая *и* ***тестовая*** *узорные структуры. Пусть также задана функция разметки* $\ell : G_{train} \to \mathbb{Y}, |\mathbb{Y}| < \infty$. *Узорную структуру* $(G_{train} \cup G_{test}, (D, \sqcap), \delta_{train} \cup \delta_{test})$ *назовем* ***классификационной****, а область определений функции* ℓ *–* ***целевым*** *признаком* y. *Элементы* $\mathbb{Y}$ *также будем называть метками целевого класса. Задачей* ***классификации*** *для классификационной узорной структуры* $(G_{train} \cup G_{test}, (D, \sqcap), \delta_{train} \cup \delta_{test})$ *называется построение функции* $y^* : G_{train} \cup G_{test} \to y$, *где* $y \in \mathbb{Y}$, *которая каждому объекту* $g \in G_{train} \cup G_{test}$ *ставит в соответствие значение (метку целевого класса) из* $y \in \mathbb{Y}$. *При этом полурешетку* D *будем называть* ***полурешеткой описаний****, а* $g^\diamond$ *для* $\forall g \in G_{train} \cup G_{test}$ *называется* ***описанием*** *объекта* g. *Узорную структуру* $(G_{train}, (D, \sqcap), \delta_{train})$ *также будем называть* ***обучающей выборкой****. Если* $|\mathbb{Y}| = 2$, *то такую задачу классификации назовем задачей* ***бинарной классификации****.*

Определение* 37.** ***Ассоциативное правило *узорной структуры* $(G, (D, \sqcap), \delta)$ *– это выражение вида* $A \to_{c,s} B$, *где* A,B *– множества описаний из* D ($A \subseteq D$, $B \subseteq D$) *и*

- $c, s \in [0, 1]$;
- $c = \frac{|(A \cup B)^\diamond|}{|A^\diamond|}$ – ***достоверность*** *(confidence, conf);*
- $s = \frac{|(A \cup B)^\diamond|}{|G|}$ – ***поддержка*** *(support, supp).*

Множество A называется ***посылкой*** *правила, а B –* ***заключением****.*

Если посылкой правила является множество описаний объектов обучающей выборки, а заключением – метка целевого класса объектов обучающей выборки, то такое ассоциативное правило называется **классифицирующим**:

Определение* 38.** *В условиях Определений 36 и 37* ***классифицирующее ассоциативное правило *узорной структуры $(G, (D, \sqcap), \delta)$ – это выражение вида $A \to_{c,s} y$, где $y \in \mathbb{Y}$, $A \subseteq D$ – подмножество описаний из D, а c,s – достоверность и поддержка правила соответственно, определяющиеся так же, как в Определении 37.*

Чаще всего мы не будем указывать поддержку правил и писать $A \to_c t$ для обозначения классифицирующего правила с достоверностью c, например $\{a,b,c\} \to_{0.8} t$. Также в случае бинарной классификации будем обозначать целевой признак как "+" и писать, например, $\{a,b,c\} \to_{0.8}$ "+".

Определение локальной интерпретируемости 21 остается ровно тем же и для классифицирующих ассоциативных правил, определенных для узорных структур.

***Определение* 39.** *Пусть в условиях Определения 36 A – некоторое множество объектов обучающей выборки в задаче бинарной классификации, $A \in G_{train}$. Пусть A_d – подмножество объектов из A, обладающих описанием d, а $A_{\neg d}$ – подмножество объектов из A, не обладающих описанием d, то есть $\forall\, a \in A_m$ справедливо $d \in \delta(a)$ и $\forall\, a \in A_{\neg m} - d \notin \delta(a)$.*

Тогда ***критерий информативности*** *описания d в задаче бинарной классификации с обучающей выборкой $(G_{train}, (D, \sqcap), \delta_{train})$ определяется для множества объектов A следующим образом:*

$$Q(A, d) = F(A) - \frac{|A_d|}{|A|} F(A_d) - \frac{|A_{\neg d}|}{|A|} F(A_{\neg d}),$$

где $F(X)$ – некоторая функция $F : X \to \mathbb{R}$ с аргументом $X \subseteq 2^{G_{train}}$, а $2^{G_{train}}$ – множество всех подмножеств множества G_{train}.

Наконец, определим энтропийный прирост информации для классифицирующего ассоциативного правила в случае узорных структур.

***Определение* 40.** *Пусть дана классификационная узорная структура $(G_{train} \cup G_{test}, (D, \sqcap), \delta_{train} \cup \delta_{test})$ и целевой признак – бинарный, то есть задана функция $\ell : G_{train} \to \mathbb{Y}$ и $|\mathbb{Y}| = 2$. Пусть, без потери общности, множество уникальных значений $\mathbb{Y} - \{0,1\}$. Тогда G_{train} можно поделить на 2 непересекающихся множества положительных и отрицательных объектов: $G_{train} = G^+_{train} \cup G^-_{train}$ и $\delta(g) = 1\ \forall g \in G^+_{train}, \delta(g) = 0\ \forall g \in G^-_{train}$.*

Пусть $B \to_{c,s} y$ – классифицирующее ассоциативное правило (см. Определение 38), где $B \subseteq D$, c,s – достоверность и поддержка правила. Тогда множество объектов G_{train} можно разделить на 4 непересекающихся множества: $G_{train} = G^+_B \cup G^-_B \cup G^+_{\neg B} \cup G^-_{\neg B}$, где

$G^+_B = B^\diamond \cap G^+_{train}$ – множество положительных объектов, подходящих под правило $B \to_{c,s} y$;

$G_B^- = B^\diamond \backslash G_{train}^+$ – *множество отрицательных объектов, подходящих под правило* $B \to_{c,s} y$;

$G_{\neg B}^+ = G_{train}^+ \backslash B^\diamond$ – *множество положительных объектов, не подходящих под правило* $B \to_{c,s} y$;

$G_{\neg B}^- = G_{train}^- \backslash B^\diamond$ – *множество отрицательных объектов, не подходящих под правило* $B \to_{c,s} y$.

(Энтропийным) приростом информации *правила* $B \to_{c,s} y$ *называется величина*

$$Q_H(G_{train}, B) = H_0 - \frac{|G_B^+ \cup G_B^-|}{|G_{train}|} H_B - \frac{|G_{\neg B}^+ \cup G_{\neg B}^-|}{|G_{train}|} H_{\neg B},$$

где $H_B = H(\frac{|G_B^+|}{|G_B^+ \cup G_B^-|}, \frac{|G_B^-|}{|G_B^+ \cup G_B^-|})$, $H_{\neg B} = H(\frac{|G_{\neg B}^+|}{|G_{\neg B}^+ \cup G_{\neg B}^-|}, \frac{|G_{\neg B}^-|}{|G_{\neg B}^+ \cup G_{\neg B}^-|})$, *а* $H_0 = H(\frac{|G_{train}^+|}{|G_{train}|}, \frac{|G_{train}^-|}{|G_{train}|})$, *и* H *– энтропия, введенная ранее перед Определением 22.*

***Утверждение* 5.** *В условиях определений этой главы значение прироста информации для классифицирующего ассоциативного правила равно значению прироста информации для классифицирующего ассоциативного правила, полученного из исходного заменой посылки на замыкание этой посылки.*

Доказательство. Пусть оператор Галуа $(\cdot)^\diamond$ связан с узорной структурой $(G, (D, \sqcap), \delta)$ и функция разметки $\ell : G_{train} \to \mathbb{Y}$, где $A \subseteq D$ – подмножество описаний. Тогда для условной выборочной вероятности $p(\tau \mid A)$ того, что объект, обладающий множеством описаний A, также обладает целевым признаком τ, справедливо:

$$p(\tau \mid A) = \frac{|A^\diamond \cap G_\tau|}{|A^\diamond|} = \frac{|(A^{\diamond\diamond})^\diamond \cap G_\tau|}{|(A^{\diamond\diamond})^\diamond|} = p(\tau \mid A^{\diamond\diamond})$$

по свойству оператора Галуа $(\cdot)'$: $(A^{\diamond\diamond})^\diamond = A^\diamond$. Тогда $\mathrm{G}(A) = \mathrm{G}(A^{\diamond\diamond})$ и для пути решения $\langle m_1, \ldots, m_k \rangle$

$\mathrm{Gini}(m, m_1, \ldots, m_k) = -\frac{|A_m^\diamond|}{|G|}\mathrm{G}(A_m) - \frac{|A_{\neg m}^\diamond|}{|G|}\mathrm{G}(A_{\neg m}) = -\frac{|(A^{\diamond\diamond})_m^\diamond|}{|G|}\mathrm{G}(A_m^{\diamond\diamond}) - \frac{|(A^{\diamond\diamond})_{\neg m}^\diamond|}{|G|}\mathrm{G}(A_{\neg m}) = \mathrm{Gini}(\{m, m_1, \ldots, m_k\}^{\diamond\diamond})$,

$\mathrm{IG}(m, m_1, \ldots, m_k) = -\frac{|A_m^\diamond|}{|G|}\mathrm{H}(A_m) - \frac{|A_{\neg m}^\diamond|}{|G|}\mathrm{H}(A_{\neg m}) = -\frac{|(A^{\diamond\diamond})_m^\diamond|}{|G|}\mathrm{H}(A_m^{\diamond\diamond}) - \frac{|(A^{\diamond\diamond})_{\neg m}^\diamond|}{|G|}\mathrm{H}(A_{\neg m}) = \mathrm{IG}(\{m, m_1, \ldots, m_k\}^{\diamond\diamond})$.

Здесь G и H – это введенные выше неопределенность Джини и прирост информации соответственно. □

***Определение* 41.** *Пусть дана узорная структура* $(G, (D, \sqcap), \delta)$ *и элементы множества* D *пронумерованы, т.е. для множества* D *задан порядок* $(\alpha(D), <), \forall d \in D\ \alpha(d) \in [1, |D|]$. *Пусть для* $B \subseteq D$ $min(B)$ *выдает элементы из* B *с минимальным номером:*
$min(B) = \{d \mid d \in B, \alpha(d) < \alpha(\tilde{d})\ \forall \tilde{d} \in B \backslash \{d\}\}$.

Обозначим $suc(B)$ *– множество всех наследников множества* B*: узорных понятий с узорным содержанием вида* $(B \cup \{i\})^{\diamond\diamond}$, *таких что* $min((B \cup \{i\})^{\diamond\diamond} \setminus B) = \{i\}$. ***CbO-деревом*** *для узор-*

ной структуры $(G, (D, \sqcap), \delta)$ *называется дерево, состоящее из всевозможных множеств* $suc(B)$, *дуги которого задаются отношением* $(B, suc(B))$.

Теорема 2. *Пусть решается задача бинарной классификации, и обучающая выборка задана интервальной узорной структурой* $\mathbb{PS} = (G^+_{train} \cup G^-_{train}, (D, \sqcap), \delta_{train})$ *с функцией разметки* $\ell : G_{train} \to \{0{,}1\}$, *где* $G_{train} = G^+_{train} \cup G^-_{train}$. *Пусть для данной узорной структуры построено CbO-дерево* T_{CbO}.

1. *Существует многозначный формальный контекст* $\mathbb{K} = (G_{train}, M, W, I)$ *и формальный контекст* $\mathbb{K}_\mathbb{I}$, *полученный из* $\mathbb{K}$ *межпорядковым шкалированием, такие что решетка формальных понятий, построенная для контекста* $\mathbb{K}_\mathbb{I}$, *изоморфна решетке узорных понятий, построенной для узорной структуры* $\mathbb{PS}$.

2. *Можно установить взаимно-однозначное соответствие между вершинами признакового CbO-дерева* $T_{I_{CbO}}$, *построенного по контексту* $\mathbb{K}_\mathbb{I}$, *и вершинами CbO-дерева* T_{CbO}.

3. *Для любого пути решения* $\langle m_1, \ldots, m_j \rangle$ *дерева решений* T *глубины* k $(j \leq k)$ *с приростом информации* $IG(\langle m_1, \ldots, m_j \rangle)$ *найдется замкнутое описание, являющееся вершиной CbO-дерева на глубине не более* k, *а также посылкой классифицирующего правила с не меньшим приростом информации, чем у* $\langle m_1, \ldots, m_j \rangle$.

Доказательство. 1. Напрямую следует из утверждения, доказанного в [Kuz09], про наличие изоморфизма между решеткой понятий интервальной узорной структуры и решеткой понятий формального контекста, полученного межпорядковым шкалированием соответствующего многозначного контекста. Обозначим $\mathbb{K}_\mathbb{I} = (G_{train}, M_I, I)$. Отсюда же следует, что $\forall A \subseteq G_{train}$ мощность A' равна мощности $A^\diamond$, где операция $(\cdot)'$ определена в контексте $\mathbb{K}_\mathbb{I}$, а операция $(\cdot)^\diamond$ – в узорной структуре $\mathbb{PS}$

2. Из п. 1. следует, что любому формальному понятию (A, B) контекста $\mathbb{K}_\mathbb{I}$ можно поставить во взаимно-однозначное соответствие узорное понятие (A, C) узорной структуры $\mathbb{PS}$. Здесь $A \subseteq G_{train}$, $B \subseteq M_I, C \in D$. Тогда для множества B со множеством наследников $suc(B)$ (см. Определение 16 признакового CbO-дерева) можно поставить во взаимно-однозначное соответствие описание C со множеством наследников $suc(C)$ вида $(C \cup \{i\})^{\diamond\diamond}$, таких что $min((C \cup \{i\})^{\diamond\diamond} \setminus C) = \{i\}$ (см. Определение 16 CbO-дерева для узорной структуры).

3. – Пусть $C_j = C_j^{\diamond\diamond}$ –замкнутое описание, о котором идет речь. По доказанному свойству 5 $IG(C_j) = IG(C_j^{\diamond\diamond})$

   – Применяя Теорему 1 к контексту $\mathbb{K}_\mathbb{I}$, получаем, что для любого пути решения $\langle m_1, \ldots, m_j \rangle$ дерева решений T глубины k $(j \leq k)$ с приростом информации $IG(\langle m_1, \ldots, m_j \rangle)$ найдется замкнутое множество признаков B_I, являющееся вершиной CbO-дерева $T_{I_{CbO}}$ на глубине не более k, а также посылкой классифицирующего правила с не меньшим приростом информации, чем у $\langle m_1, \ldots, m_j \rangle$.

– Из п. 2 следует, что этой вершине B_I CbO-дерева $T_{I_{CbO}}$ соответствует некоторая вершина C CbO-дерева T_{CbO}.

– Из п.1 заключаем, что, $|B_I| = |C|$ и тогда проводя рассуждения, аналогичные тем, что сделаны в доказательстве Теоремы 1, получаем, что вершина C располагается в CbO-дерева T_{CbO} на глубине не более k. Приведем эти рассуждения.

Обозначим $C_j = \{d_1, \ldots, d_j\}$, где $d_1 \in D, \ldots d_j \in D$. Надо показать, что $depth_{CbO}(^{\diamond\diamond}_j) \leq k$, где $depth_{CbO}(A)$ – глубина CbO-дерева, на которой расположен элемент A.

Рассмотрим 2 случая, когда множество C_j замкнуто и когда оно не замкнуто.

Случай 1. Пусть $C_j = C_j^{\diamond\diamond}$. Значит, C_j является вершиной CbO-дерева. Покажем по индукции, что $\forall A \in T_{CbO}$ справедливо $|A| \geq depth_{CbO}(A)$.

(a) Для $|A| = 1$ неравенство выполняется тривиально, так как по построению CbO-дерева в нем на глубине 1 располагаются элементы вида $d^{\diamond\diamond}, d \in D$ и $|d^{\diamond\diamond}| \geq |d| = 1$.

(b) Пусть $|A| \geq depth_{CbO}(A)$ для $\forall A \in T_{CbO}$ такого что $|A| = n$. Покажем, что $\forall suc(A)$ выполнено $|suc(A)| \geq depth_{CbO}(suc(A))$.
Действительно, $suc(A) = (A \cup i)^{\diamond\diamond} \Rightarrow |suc(A)| \geq |A \cup i| = |A|+1 \geq depth_{CbO}(A)+1 = depth_{CbO}(suc(A))$.

Мы показали, что мощность любого элемента CbO-дерева не меньше глубины, на которой этот элемент располагается в CbO-дереве. Применяя это свойство к C_j, получаем $depth_{CbO}(C_j) \leq |C_j| = j \leq k$.

Случай 2. Пусть $C_j \neq C_j^{\diamond\diamond}$. $\exists$ перестановка $(i_1, \ldots, i_j)$ чисел $1, \ldots, j$ такая что $\alpha(i_1) < \ldots < \alpha(i_j)$, где α – порядок на описаниях, заданный в Определении 41. Тогда в CbO-дереве $\exists$ путь $(d_{i_1})^{\diamond\diamond} \to (d_{i_1} \cup d_{i_2})^{\diamond\diamond} \to \ldots (d_{i_1} \cup d_{i_2} \cup \ldots m_{i_j})^{\diamond\diamond} = ^{\diamond\diamond}_j$, длина которого равна j. То есть, на глубине j в CbO-дереве $\exists$ вершина $^{\diamond\diamond}_j$, или $depth_{CbO}(^{\diamond\diamond}_j) = j \leq k$.

□

2.3. Порядок на помеченных графах и графовая узорная структура

Приведем основные определения из [Ore62] и [Сам06].

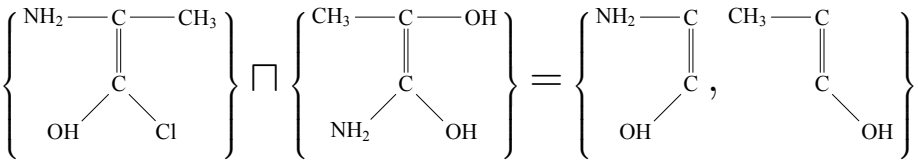
Помеченный граф – это пятерка вида $((V,l_v), (E,l_e), L)$, где V – множество вершин, $E \subseteq V \times V$ – множество ребер, а функции $l_v : V \to L$ и $l_e : V \to L$ сопоставляют метки из множества L вершинам и ребрам соответственно

Помеченный граф $\Gamma_1 := ((V_1,l_{v_1}), (E_1,l_{e_1}), L_1)$ **доминирует** над помеченным графом $\Gamma_2 := ((V_2,l_{v_2}), (E_2,l_{e_2}), L_2)$, или $\Gamma_2 \leq \Gamma_1$, если существует взаимно-однозначное отображение $\varphi\colon V_2 \to V_1$, которое

– учитывает ребра: $(v,w) \in E_2 \Rightarrow (\varphi(v), \varphi(w)) \in E_1$,

– учитывает порядок на метках: $l_{v_2}(v) \leq l_{v_1}(\varphi(v))$, $l_{e_2}(v,w) \leq l_{e_1}(\varphi(v),\varphi(w))$.

Пример:



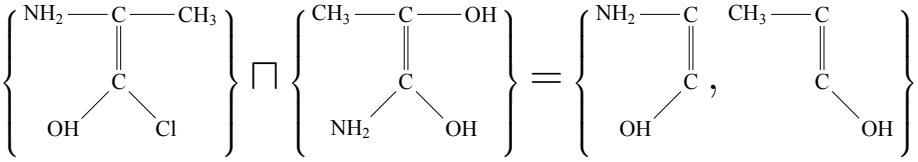
Метки вершин упорядочены (слева), $x \preceq A$ для произвольной вершинной метки $A \in \mathcal{L}$ (справа).

Полурешеточная операция для двух графов задается следующим образом [Сам06]:

$$\{X\} \sqcap \{Y\} := \{Z \mid Z \leq X,Y, \quad \forall Z_* \leq X,Y \quad Z_* \not\geq Z\}$$

–множество всех максимальных общих подграфов графов X и Y.

Пример:

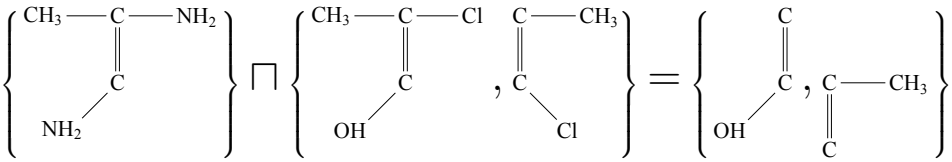


Для множеств графов $\mathbb{X} = \{X_1, \ldots, X_k\}$ и $\mathbb{Y} = \{Y_1, \ldots, Y_k\}$ операция задается следующим образом:

$$\mathbb{X} \sqcap \mathbb{Y} := \text{MAX}\,(\bigcup_{i,j}(\{X_i\} \sqcap \{Y_j\}))$$

Операция $\sqcap$ идемпотентна, коммутативна и ассоциативна [Kuz99], то есть действительно является полурешеточной операцией по Определению 3.

Пример:

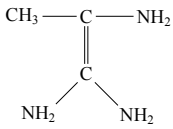
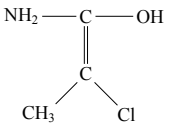
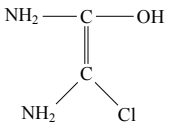


Поскольку операция $\sqcap$ задает полурешетку на множестве помеченных графов, через нее можно определить отношение естественного порядка следующим образом: $\mathcal{G} \sqsubseteq \mathcal{H} \iff \mathcal{G} \sqcap \mathcal{H} = \mathcal{G}$.

Такое определение отношения $\sqsubseteq$ эквивалентно следующему: $\mathcal{G} \sqsubseteq \mathcal{H} \iff \forall g \in \mathcal{G}\ \exists h \in \mathcal{H}$ такой что $g \leq h$.

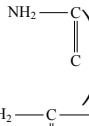
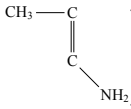
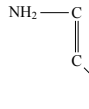
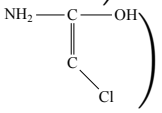
Приведем пример узорной структуры, определенной на множестве объектов, описаниями которых являются помеченные графы.

***Пример* 9.** *Пусть* $\{1,2,3\}$ *– множество,* $\{G_1,G_2,G_3\}$ *– множество их описаний в виде помеченного графа:*

G_1 :  G_2 :  G_3 : 

D – множество всех помеченных графов, $\sqcap$ – операция пересечения на графах, $\underline{D} = (D, \sqcap)$. Множество $\{1,2,3\}$*, их "описаний" (графов)* $D = \{G_1,G_2,G_3\}$ *($\delta(i) = G_i, i = 1,\ldots,3$), и оператор* $\sqcap$ *образуют узорную структуру* $(\{1,2,3\}, \underline{D}, \delta)$.
$\{1,2,3\}^\diamond = \{NH_2 - C = C\}$*, т.к.* $\{NH_2 - C = C\}$ *– единственный граф, изоморфный подграфу каждого из графов* $\{G_1,G_2,G_3\}$*. Аналогично* $\{NH_2 - C = C\}^\diamond = \{1,2,3\}$*, поскольку объекты* $1,2$*, и* 3 *имеют описания, подграфу которых изоморфен граф* $\{NH_2 - C = C\}$.

Все узорные понятия такой узорной структуры:

$\left(\{1,2,3\}\text{ , }\right.$  $\left.\right)$, $\left(\{1,2\}\text{ , }\right.$  $\left.\right)$, $\left(\{1,3\}\text{ , }\right.$  $\left.\right)$,
$\left(\{2,3\}\text{ , }\right.$  $\left.\right)$, $(1,\{G_1\})$, $(2,\{G_2\})$, $(3,\{G_3\})$, $(\emptyset, \{G_1, G_2, G_3\})$.

2.4. Классификация данных со сложной структурой методом ядерных функций

2.4.1. Ядра и ядерный трюк

Ядерные методы обучения работают не с явными описаниями объектов, а только с информацией об их попарном сходстве — ядерной функцией [Mül+01].

Функция $K : X \times X \to \mathbb{R}$, определенная на множестве X, называется **ядром** на $X \times X$, если она:

- Симметрична, т.е. для любых $x \in X, y \in X \to K(x,y) = K(y,x)$;
- Положительно полуопределена, т.е. для любых $x_1, \ldots, x_N \in X$ $(N \geq 1)$ матрица K, такая что $K_{i,j} = K(x_i, x_j)$, положительно полуопределена, а именно: $\sum_{i,j} c_i c_j K_{i,j} \geq 0$, или, эквивалентно, все собственные числа матрицы K неотрицательны.

Если $x \in X$ описывается в виде $\phi(x) = \{\phi_n(x)\}_{n\geq 1}$, и функция K представляется в виде скалярного произведения $K(x,y) =< \phi(x), \phi(y) >= \sum_n \phi_n(x)\phi_n(y)$, то K — ядро. Векторное пространство, полученное с помощью функции $\phi(x)$, называется признаковым.

Ядерные методы работают в признаковом пространстве и ищут линейные зависимости между признаковыми описаниями. В частности, в алгоритме SVM в простейшем случае бинарной классификации ищется оптимальная разделяющая плоскость в признаковом пространстве.

Ядерный трюк ("kernel trick")

Если векторы $\{\phi(x)\}$ используются оптимизационным алгоритмом только в скалярном произве-

дении $< \phi(x_i), \phi(x_j) >$, то алгоритм работает в признаковом пространстве только опосредованно через ядерную функцию [Mül+01].

“Ядерный трюк” в SVM: в признаковом пространстве разделяющая гиперплоскость определяется вектором $w = \sum_i \alpha_i y_i \phi(x_i)$, где x_i — опорные вектора. Для классификации нового примера x надо считать скалярное произведение $< w,\phi(x) >= \sum_i \alpha_i y_i K(x_i,x)$. Таким образом, не надо считать сложное скалярное произведение вида $K(\phi(x),\phi(y))$ в признаковом пространстве, вычисления свелись к подсчету скалярных произведений в пространстве относительно небольшой размерности.

Среди ядер для данных со сложной структурой выделяют: ядра свертки (Convolution kernels), ядра для деревьев (Tree kernels) и графовые ядра (Graph kernels).

Ядра свертки — это ядра на сложных структурах, определенные с помощью более простых ядер на подструктурах описаний объектов.

Пусть $\mathcal{X}$ — пространство объектов, и каждому объекту x соответствует подпространство $\mathcal{X}_§$ пространства $\mathcal{X}$. Пусть также определено ядро $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$. Если имеется отношение $R \subseteq \mathcal{X} \times \mathcal{X}$, то следующее ядро называется **ядром свертки**: $K(x,y) = \sum_{(x',x)\in R} \sum_{(y',y)\in R} k(x',y')$ или $K(x,y) = \sum_{(x',y')\in(\mathcal{X}_§,\mathcal{X}_†)} k(x',y')$, где $\mathcal{X}_§,\mathcal{X}_†$ — конечные подмножества $\mathcal{X}$, определенные объектами x и y.

Среди ядер свертки – ядра отображения (Mapping kernels), строковые ядра (String kernels).

Среди ядер для деревьев выделяют [Mül+01] ядро редакторских расстояний (Tree edit distances kernel), ядро поддеревьев (Subtree kernel), ядро деревьев из подмножеств вершин (Subset tree kernel) и ядро частичного сходства деревьев (Partial tree kernel).

2.4.2. Графовые ядра

Рассмотрим подробней некоторые ядра для графов.

Ядра случайного обхода (Random walk kernels)

Идея ядра произведения графов (product graph kernel) [Vis+10] – вычисление общих путей с одинаковыми метками в двух графах. Хоть число таких путей может быть и бесконечно, тем не менее скалярное произведение в ядре может быть посчитано за полиномиальное время с помощью вычисления произведения двух ядер и предела последовательности степеней матрицы смежности этого графа.

Вычисление таких ядер полиномиально только в случае непомеченных графов. При правильном выборе параметров вычислительную сложность ядра можно понизить до $O(n^3)$, где n — максимальное число вершин в графах.

Прямое произведение двух графов: $P_\times = X \times Y = (V_\times), E_\times$, где $V(X) = \{(x,y) : x \in V(X), y \in V(Y), L_X(x) = L_Y(y)\}$, а $E_\times = \{((x,y),(x',y')) \in V_\times \times V_\times : (x,y) \in E(X), (x',y') \in E(Y), L_X(x,y) = L_Y(x',y')\}$.

$K_\times(X,Y) = \sum_{x,y=0}^{|V_\times|} \sum_{k=0}^{\infty} \left(\lambda_k A_\times^k\right)_{x,y}$, где $A_\times$ – матрица смежности произведения X и Y, $\{\lambda_i\}$ — последовательность весов ($\lambda_i \in \mathbb{R}, \lambda_i \geq 0$).

Предел может быть эффективно подсчитан при правильном выборе $\{\lambda_i\}$ – за $O(n^3)$.

Ядро циклов (Cyclic pattern kernel)

Основная идея [HGW04] – разбить граф на простые цикла и оставшиеся ребра (“мосты”).

$K_{CP} = |C(X) \bigcap C(Y)| + |T(X) \bigcap T(Y)|$, где $C(X) = \{can(c) | \, c \in S(X)\}$,
$S(X)$ — множество простых циклов в графе X, $can(c)$ — каноническое представление цикла c (строка, которая уникально представляет этот цикл). $T(X)$ — множество дуг, полученных удалением простых циклов $S(X)$ из графа X.

Ядро полиномиально по числу вершин и простых циклов в графе. Применяется в химической информатике, где число циклов в графах невелико.

Ядро поддеревьев (Subtree pattern kernel)

Введение ядер поддеревьев [SB09] – попытка побороть такой недостаток ядер случайного обхода, что некоторые пары графов отображаются в одну и то же точку признакового пространства.

Псевдокод алгоритма вычисления ядра:

- Используется базовое ядро на подграфах графов X и Y с небольшим числом вершин (например, ядро всех подграфов);
- Для каждой пары вершин $(x,y) \in V(X) \times V(Y)$:
  - применить преобразование базового ядра к вершинам x и y
  - рекурсивно применить преобразование ядра ко всем вершинам из множеств соседей соседей вершин x и y вплоть до определенной глубины h.

Ядро поддеревьев лучше отражает структурную природу объектов, чем ядра случайного обхода (“более выразительно”), но сложность вычислений растет экспоненциально с ростом глубины “просмотра соседей” h.

Ядро кратчайших путей (Shortest path kernels)

Ядра случайного обхода недостаточно выразительны, а ядра поддеревьев обладают большой вычислительной сложностью. [RG03]. Нужен компромисс. Идея ядер кратчайших путей [BK05] – сравнивать кратчайшие пути между всеми парами вершин в графах.

Известно, что кратчайший путь между двумя вершинами в графе ищется за полиномиальное время, например, алгоритмом Дейкстры для одной вершины (сложность $O(m + nlogn)$, n и m — числа вершин и ребер соответственно) и алгоритмом Флойда-Уоршелла для всех вершин в графе (сложность $O(n^3)$).

Строится “граф кратчайших путей” $S(X)$, включающий все те же вершины, что и граф X, а на каждом ребре ставится метка — длина кратчайшего пути между соответствующими вершинами.

$K_{sp}(X,Y) = \sum_{e \in E(S(X))} \sum_{e' \in E(S(Y))} k^1_{walk}(e,e')$, где k^1_{walk} – ядро случайного обхода на обходах единичной длины 1, то есть ядро на ребрах.

Поскольку сравниваются все пары ребер (которых максимум $O(n^2)$), сложность вычисления такого ядра – $O(n^4)$.

Ядро подграфов фиксированного размера (Graphlet kernel)

Принцип Graphlet-ядра [She+09] – определить сходство графов через количество их общих подграфов с фиксированным максимальным числом вершин.

M-графлетом графа с N вершинами ($M \leq N$), называется его подграф, содержащий не более M вершин.

Графлет-ядро с M-графлетами – это ядро $K_M(X,Y) = \sum_{S \in M_i(X)} \sum_{\hat{S} \in M_i(Y)} \delta(S, \hat{S})$, где $M_i(X)$ – множество всех матриц, полученных из матрицы смежности графа X удалением m строк и соответствующих столбцов, а δ – символ Кронекера, в определение которого в случае графов заложена проблема изоморфизма.

На практике для непомеченных графов множество всех подграфов и изоморфизм считают заранее, что ускоряет классификацию.

Ядра Вайсфайлера-Лемана (Weisfeiler-Lehman kernels)

Основная идея ядер Вайсфайлера-Лемана [She+11] – представить каждый граф последовательностью графов с разными функциями помечивания вершин, удовлетворяющими специальному тесту Вайсфайлера-Лемана на изоморфизм.

$\{X_0, \ldots, X_h\} = \{(V,E,L_0), \ldots, (V,E,L_h)\}$, где L_i — функция помечивания вершин графа, удовлетворяющая тесту Вайсфайлера-Лемана на изоморфизм до глубины i.

$K^h_{WL}(X,Y) = k(X_0, Y_0) + \ldots + k(X_h, Y_h)$.

Ядро Вайсфайлера-Лемана вычисляется за $O(hm)$ для двух графов (m — максимальное число ребер в этих двух графах) и за $O(Nhm + N^2hn)$ для N графов (n — максимальное число вершин графа по всей выборке).

Оно стало “state-of-the-art”-ядром среди графовых ядер благодаря низкой сложности вычисления и хорошим экспериментальным показателям классификации.

2.5. Классификация данных со сложной структурой на основе узорных структур

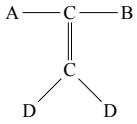
Классификация по запросу для узорных структур

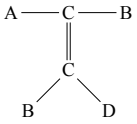
Алгоритм классификации по запросу для данных со сложной структурой на основе аппарата узорных структур был предложен в [Kuz13]. Его основное отличие от алгоритма классификации по запросу, основанного на ассоциативных правилах (рассмотренного в Параграфе 1.3.4), заключается в том, что алгоритм может работать с произвольными типами данных, для которых задано описание объекта (это может быть как множество признаков, так и последовательности, интервалы или графы) и полурешёточная операция сходства этих описаний. То есть алгоритм предназначен для данных со сложной структурой, в которых обучающая выборка может быть представлена узорной структурой. Эта постановка была реализована для интервальных данных в задаче кредитного скоринга [MKK15] и для данных, представленных графами [KK15], в задаче прогнозирования токсичности химических веществ.

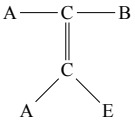
Рассмотрим подробней пример работы алгоритма из [KK15].

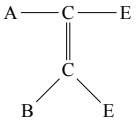
***Пример* 10.** *В задаче предсказания наличия некоторого свойства P химических веществ дана обучающая выборка в виде упрощенной молекулярной структуры 4 положительных веществ (обладающих свойством P) и 3 отрицательных веществ (не обладающих свойством P). Для тестовых объектов необходимо сделать прогноз, обладают ли они свойством P.*

Описания положительных примеров:

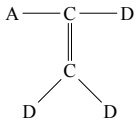
G_1 : 

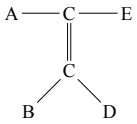
G_2 : 

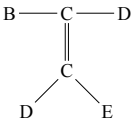
G_3 : 

G_4 : 

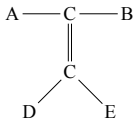
Описания отрицательных примеров:

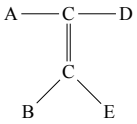
G_5 : 

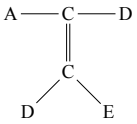
G_6 : 

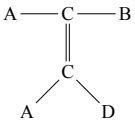
G_7 : 

Описания тестовых примеров:

G_8 : 

G_9 : 

G_{10} : 

G_{11} : 

Все общие 3-подграфы тестовых ($G_8 - G_{11}$) и обучающих примеров показаны в Таблице 2.5.

Таблица 2.6 резюмирует классификацию тестовых примеров. Скажем, "+4" для графов G_1 и G_8 что все общие подграфы графов G_1 и G_8 (то есть, A–C– B, A–C=C, B–C=C и C=C–D) не изоморфны отрицательным примерам G_5,G_6,G_7. Таким образом, пересечение $G_1 \sqcap G_8$ "вносит вклад +4" в положительную классификацию примера G_8. А вот все графы в пересечении $G_4 \sqcap G_8$

	G_8	G_9	G_{10}	G_{11}
G_1	*ACB, ACC, BCC, CCD*	*ACC, BCC, CCD*	*ACC, CCD*	*ACB, ACC, BCC, CCD*
G_2	*ACB, ACC, BCC, CCD*	*ACC, BCC, CCD*	*ACC, CCD*	*ACB, ACC, BCC, CCD*
G_3	*ACB, ACC, BCC, CCE*	*ACC, BCC, CCE*	*ACC, CCE*	*ACB, ACC, BCC*
G_4	*ACC, BCC, CCE*	*ACC, BCC, BCE, CCE*	*ACC, CCE*	*ACC, BCC*
G_5	*ACC, CCD*	*ACC, ACD, CCD*	*ACC, ACD, CCD*	*ACC, ACD, CCD*
G_6	*ACC, BCC, CCD, CCE*	*ACC, BCC, CCD, CCE*	*ACC, CCD, CCE*	*ACC, BCC, CCD*
G_7	*BCC, CCD, CCE, DCE*	*BCC, CCD, CCE*	*CCD, CCE, CDE*	*BCC, CCD*

Таблица 2.5: Общие 3-подграфы тестовых и обучающих примеров.

(A–C=C, B–C=C и C=C–E) в то же время изоморфны подграфу отрицательного примера G_6, поэтому пересечение $G_4 \sqcap G_8$ не "вносит вклад" в положительную классификацию примера G_8.

	G_1	G_2	G_3	G_4	G_5	G_6	G_7	*Итого*	*Класс*
G_7	*+4*	*+4*	*+4*	G_6	G_1	*–4*	*–4*	*+4*	+
G_8	G_6	G_6	G_6	*+4*	*–3*	*–4*	*–3*	*-6*	—
G_{10}	G_5	G_5	G_6	G_6	*–3*	*–3*	*–3*	*-9*	—
G_{11}	*+4*	*+4*	*+3*	G_6	*–3*	G_1	G_1	*+8*	+

Таким образом, примеры G_8 и G_{11} классифицируются как обладающие свойством P, а G_9 и G_{10} – как не обладающие этим свойством.

ДСМ для узорных структур

ДСМ-метод, рассмотренный ранее, естественно адаптируется для классификации данных со сложной структурой. ДСМ-метод для узорных структур описан в [Kuz04].

2.6. Заключение

В Главе 2 были рассмотрены методы классификации данных со сложной структурой, которая не может быть представлена без потерь в виде объектно-признаковой таблицы или формального контекста. Для этого в Разделе 2.2 описан аппарат узорных структур и их проекций в общем виде, а также в частном случае, когда объекты представляются количественными или интервальными признаками. С помощью специального вида проекций интервальных узорных структур показано, как в случае данных с количественными признаками находить классифицирующие правила не хуже, чем правила, построенные деревом решений, по критерию информативности типа прироста информации или неопределенности Джини.

В Разделе 2.3 приведен обзор методов классификации данных со сложной структурой с помощью ядерных функций и метода опорных векторов, а в Разделе 2.5 рассмотрена альтернатива ядерным методам, основанная на узорных структурах.

Глава 3

Алгоритмы классификации данных на основе множеств формальных и узорных понятий

3.1. Введение

В Главах 1 и 2 показано, что Анализ Формальных Понятий предлагает удобный формализм для того чтобы, с одной стороны, выразить на этом языке многие алгоритмы, основанные на классифицирующих ассоциативных правилах, а с другой, чтобы обобщить эти алгоритмы на случай данных со сложной структурой. В Главе 3 мы предлагаем алгоритм классификации произвольных данных со сложной структурой, для которых можно ввести полурешёточную операцию сходства. Отдельно и с подробными примерами рассматриваются частные случаи, когда данные представлены бинарными, количественными и интервальными признаками, а также помеченными графами.

3.2. Классификация данных с бинарными и категориальными признаками на основе множества формальных понятий

Предлагаемый подход в случае бинарных признаков в обучающей и тестовой выборке описан в Алгоритме 4 – CoLiBRi (Concept Lattice-Based Rule-learner, классификация на основе правил с помощью решеток формальных понятий) на основе [Каш16]. Для категориальных признаков предлагается использовать One Hot Encoding, то есть для каждого категориального признака порождать бинарные признаки в количестве, равном уникальному числу значений этого категориального признака.

На вход алгоритму подаются обучающий и тестовый формальные контексты $\mathbb{K}_{train} = (G_{train}, M_0 \cup \overline{M}_0 \cup c_{train}, I_{train})$ и $\mathbb{K}_{test} = (G_{test}, M_0 \cup \overline{M}_0, I_{test})$. Множество признаков M дихотомизировано: $M = M_0 \cup \overline{M}_0$, где $\forall g \in G_{train}, m \in M_0 \; \exists \, \overline{m} \in \overline{M}_0 : g I_{train} m \rightarrow$

$\neg(gI_{train}\overline{m})$. Также алгоритм использует модификацию программной реализации In-Close 2 [1] алгоритма “Замыкай-по-Одному” ($CbO(K, min\_supp)$) [Куз93], в которой выдаются все формальные понятия формального контекста K, поддержки которых ограничены снизу значением параметра min_supp. Для выбора классифицирующих правил используется критерий $inf : M \cup c_{train} \to \mathbb{R}$ типа неопределенности Джини или энтропийного прироста информации (в программной реализации по умолчанию – среднее значение неопределенности Джини). Параметры алгоритма: min_supp и n – минимальная поддержка классифицирующих правил и число правил, используемых для классификации тестового объекта.

Алгоритм состоит из следующих шагов:

1. Инициализировать c_{test} пустым списком, а r_{test} – пустым словарем. В c_{test} будут добавляться предсказанные значения целевого признака для тестовых объектов, а в r_{test} – правила для каждого тестового объекта (ключ в словаре – номер объекта, значение – список правил).

2. Посчитать долю положительных объектов в выборке $c_{pos} = \frac{|c'_{train}|}{|G_{train}|}$.

3. С помощью алгоритма $CbO(K, min\_supp)$ найти все формальные понятия обучающего контекста $\mathbb{K}_{train}$ со значением поддержки не менее min_supp. Параллельно с этим для каждого формального понятия вычислять значение качества соответствующего классифицирующего правила inf. Таким образом, получится словарь $\mathcal{S}$, ключами которого будут содержания формальный понятий, а значениями – соответствующие значения функционала inf.

4. Отсортировать все формальные понятия $\mathcal{S}$ по посчитанным значениям критерия inf в порядке “улучшения”,то есть по возрастанию inf, если малые значения критерия говорят о хороших правилах (как в случае неопределенности Джини) или по убыванию, если, наоборот, большие значения критерия свидетельствуют о хороших правилах (прирост информации, среднее уменьшение Джини).

5. Для каждого тестового объекта $g_t \in G_{test}$:

   – Отобрать n_{rules} “подходящих” содержаний формальных понятий, то есть $\{B_i\}_{i\in[1,n_{rules}]} = \{B \mid (A,B) \in \mathcal{S}, g'_t \subseteq B\}$

   – Для каждого из отобранных содержаний формальных понятий $\{B_i\}_{i\in[1,n_{rules}]}$ определить долю положительных объектов $c_i = \frac{|B'_i \cap c'_{train}|}{|B'_i|}$

   – Сформировать таким образом набор правил $\{B_i \to_{c_i} +\}_{i\in[1,n_{rules}]}$ с достоверностями c_i. Записать его в словарь r_{test} для ключа t (номер объекта g_t).

   – Предсказанное значение целевого признака c_{train_t} определить как индикатор того, что средняя арифметическая достоверность найденных правил превышает долю

[1] http://shura.shu.ac.uk/38/

положительных объектов во всей выборке:

$$c_{train_t} = [\frac{1}{n\_rules} \sum_{i=1}^{n\_rules} c_i \geq c_{pos}].$$

Добавить это значение в c_{test}.

Algorithm 4 Concept Lattice-Based Rule-learner (CoLiBRi) – случай бинарных признаков.

Input: $\mathbb{K}_{train} = (G_{train}, M_0 \cup \overline{M}_0 \cup c_{train}, I_{train})$
$\mathbb{K}_{test} = (G_{test}, M_0 \cup \overline{M}_0, I_{test})$
$min\_supp \in \mathbb{R}^+$, $n_{rules} \in \mathbb{N}$;
$CbO(K, min\_supp) : K \to \mathcal{S}$;
$inf : M \cup c_{train} \to \mathbb{R}$;
$sort(\mathcal{S}, inf) : \mathcal{S} \to \mathcal{S}$

Output: c_{test}, r_{test}

$c_{test} = \emptyset, r_{test} = \emptyset$
$c_{pos} = \frac{|c'_{train}|}{|G_{train}|}$
$\mathcal{S} = \{(A,B) : inf(B, c_{train}) \mid A \subseteq G_{train}, B \subseteq M, A' = B, B' = A, |A| \geq min\_supp\} = CbO(\mathbb{K}_{train}, min\_supp)$
$\mathcal{S} = sort(\mathcal{S}, inf)$
for $g_t \in G_{test}$ **do**
  $\{B_i\}_{i \in [1, n_{rules}]} = \{B \mid (A,B) \in \mathcal{S}, g'_t \subseteq B\}$
  $c_i = \frac{|B'_i \cap c'_{train}|}{|B'_i|}$
  $r_{test}[i] = \{B_i \to_{c_i} +\}_{i \in [1, n_{rules}]}$
  $c_{test}[i] = [\frac{1}{n\_rules} \sum_{i=1}^{n\_rules} c_i \geq c_{pos}]$
end for

***Пример* 11.** *Продемонстрируем суть работы алгоритма для набора данных из Таблицы 3.1. Это тот же пример, что представлен Таблицей 1.2, только теперь с "отрицаниями" признаков. Здесь:*

- $\mathbb{K}_{train} = (G_{train}, M_0 \cup \overline{M}_0 \cup c_{train}, I_{train})$
- $G_{train} = \{1,2,\ldots,10\}$
- $M_0 = \{or,oo,os,tc,tm,th,hn,w\}$ – *множество признаков Outlook=rainy, Outlook=overcast, Outlook=sunny, Temperature=cool, Temperature=mild, Temperature=hot, Humidity=normal, Windy соответственно.*
- $\overline{M}_0 = \{\overline{or},\overline{oo},\overline{os},\overline{tc},\overline{tm},\overline{th},\overline{hn},\overline{w}\}$ – *множество "отрицаний" признаков из* M_0.
- $I_{train} \subseteq G_{train} \times M_0 \cup \overline{M}_0 \cup c_{train}$ – *бинарное отношение, показанное в Таблице 3.1 в строках 1–10.*

$G\backslash M$	**os**	$\neg os$	**oo**	$\neg oo$	**or**	$\neg or$	**th**	$\neg th$	**tm**	$\neg tm$	**tc**	$\neg tc$	$\neg hn$	**hn**	**w**	$\neg w$	**play**
1	×			×		×	×			×		×	×			×	
2	×			×		×	×			×		×	×		×		
3		×	×			×	×			×		×	×			×	×
4		×		×	×			×	×			×	×			×	×
5		×		×	×			×		×	×			×		×	×
6		×		×	×			×		×	×			×	×		
7		×	×			×		×		×	×			×	×		×
8	×			×		×		×	×			×	×			×	
9	×			×		×		×		×	×			×		×	×
10		×		×	×			×	×			×		×		×	×
11	×			×		×		×	×			×		×	×		?
12		×	×			×		×	×			×	×		×		?
13		×	×			×	×			×		×		×		×	?
14		×		×	×			×	×			×	×		×		?

Таблица 3.1: Формальный контекст, полученный из контекста Таблицы 1.2 добавлением признаков $\{\overline{or},\overline{oo},\overline{os},\overline{tc},\overline{tm},\overline{th},\overline{hn},\overline{w}\}$.

– $\mathbb{K}_{test} = (G_{test}, M_0 \cup \overline{M}_0, I_{test})$.

– $G_{test} = \{11,12,13,14\}$

– $I_{test} \subseteq G_{train} \times M_0 \cup \overline{M}_0$ – *бинарное отношение, показанное в Таблице 3.1 в строках 11–14.*

– *Зафиксируем среднее значение неопределенности Джини (Gini gain) как критерий отбора классифицирующих правил* $inf : M \cup c_{train} \to \mathbb{R}$.

– *Выберем параметры алгоритма* $min\_supp = 0.4$ *и* $n = 3$. *Это значит, что каждый тестовый объект будет классифицироваться 3 правилами, посылками которых будут замкнутые множества признаков с относительной поддержкой не менее 0.4.*

Заметим, что в обучающем контексте доля положительных объектов равна 0.6 (6 из 10).

Построим все формальные понятия обучающего контекста $\mathbb{K}_{train}$ *с мощностью объемов не менее 4 (т.к.* $min\_supp * |G_{train}| = 0.4 * 10 = 4$*). Также для всех формальных понятий посчитаем среднее значение неопределенности Джини соответствующего классифицирующего правила.*

Поясним, как это делается, на примере формального понятия $(\{1,3,5,9\},\{\overline{w},\overline{tm}\})$.

– *Составим сводную таблицу по одновременному наличию признаков* $\{\overline{w},\overline{tm}\}$, *а также по наличию признака целевого класса* $play$. *См. Таблицу 3.2.*

	$\{\overline{w},\overline{tm}\}$	
	Yes	***NO***
$play$	*3*	*3*
$\neg play$	*1*	*3*

Таблица 3.2: Таблица сопряженности для $\{\overline{w},\overline{tm}\}$ *и целевого признака* $play$.

– *Поскольку большинство объектов, имеющих признаки* $\{\overline{w},\overline{tm}\}$ *одновременно, положительны (также имеют признак "play"), породим с помощью формального понятия* $(\{1,3,5,9\},\{\overline{w},\overline{tm}\})$ *классифицирующее правило "*$\overline{w},\overline{tm} \to play$*".*

– *Для такого правила среднее значение неопределенности Джини равно* $\frac{1+3}{10} * Gini(\frac{1}{4}, \frac{3}{4}) + \frac{3+3}{10} * Gini(\frac{1}{2}, \frac{1}{2}) = 0.4 * (1 - (\frac{1}{4})^2 - (\frac{3}{4})^2) + 0.4 * (1 - (\frac{1}{2})^2 - (\frac{1}{2})^2) = 0.45$.

	Классифицирующее правило	***Средняя неопределенность Джини***
1	$os, \neg tc, \neg hn \rightarrow_{(1)} +$	*0.171*
2	$\neg os, \neg w \rightarrow_{(1)} +$	*0.267*
3	$\neg oo, \neg tm, w \rightarrow_{(1)} -$	*0.3*
4	$os, \neg tc, \neg hn, \neg w \rightarrow_{(1)} -$	*0.3*
5	$os, th, \neg hn, \rightarrow_{(1)} -$	*0.3*
6	$os \rightarrow_{(0.75)} -$	*0.317*
7	$\neg oo, \neg tc, \neg hn \rightarrow_{(0.75)} -$	*0.317*
8	$\neg or, \neg tc, \neg hn \rightarrow_{(0.75)} -$	*0.317*
9	$\neg os \rightarrow_{(0.83)} +$	*0.317*
10	$or, \neg th, \neg w \rightarrow_{(1)} +$	*0.343*

Таблица 3.3: 10 лучших классифицирующих правил, полученных нахождением формальных понятий контекста из Таблицы 3.1.

Топ-10 классифицирующих правил в порядке возрастания средней неопределенности Джини правила (т.е. в порядке "ухудшения" правил) показаны в Таблице 3.3.

Чтобы определить метки тестового объекта 11, проведем следующие действия согласно Алгоритму 4:

1. *Отбираем среди найденных 3 первые формальные понятия, содержания которых являются подмножествами множества признаков объекта 11 (Outlook=sunny, Temperature=mild, Humidity=normal, Windy=true) –* $\{\bar{or}, \bar{oo}, os, \bar{tc}, tm, \bar{th}, hn, w\}$

2. *Составляем на их основе 3 "лучших" правила, которые показаны в Таблице 3.4.*

3. *Найденные правила определяют значение 0 целевого признака для объекта "Outlook=sunny, Temperature=mild, Humidity=normal, Windy=true", поскольку* $\frac{1}{3}(0.25+0.5+0.5) \approx 0.41 < 0.6$.

Классифицирующее правило	***Средняя неопределенность Джини***
$os \rightarrow_{(0.75)} -$	*0.317*
$\neg oo \rightarrow_{(0.5)} -$	*0.4*
$\neg th, hn \rightarrow_{(0.5)} -$	*0.4*

Таблица 3.4: 3 "лучших" правила для классификации объекта Outlook=sunny, Temperature=mild, Humidity=normal, Windy=true

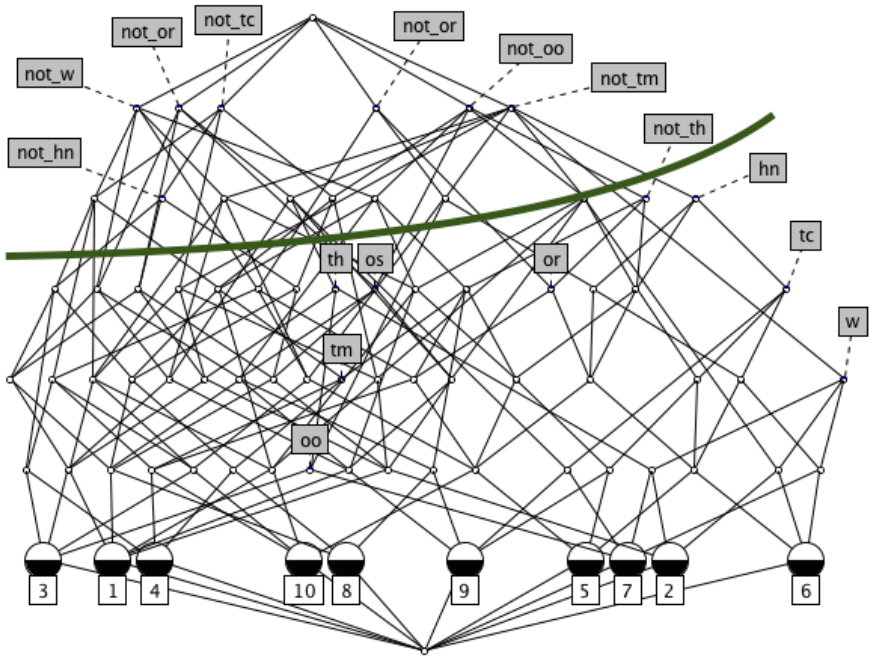


Рисунок 3.1: Решетка формальных понятий, соответствующая обучающему контексту из Примера 11. Выше зеленой линии лежат формальные понятия с минимальной относительной поддержкой 0.4.

3.3. Классификация данных с количественными признаками на основе множества формальных понятий

Рассмотрим, как обобщить Алгоритм 4 на случай решения задач классификации данных с количественными и интервальными признаками.

Предлагаемый подход в случае количественных признаков в обучающей и тестовой выборке описан в Алгоритме 5 на основе [KK16b].

На вход алгоритму подаются обучающий и тестовый многозначные формальные контексты $\mathbb{K}^m_{train} = (G_{train}, M \cup c_{train}, W, I_{train})$ и $\mathbb{K}^m_{test} = (G_{test}, M, W, I_{test})$. Также алгоритм принимает на вход функцию дискретизации признаков $d : M \cup c_{train} \times W \to M_{binary}$, которая возвращает множество бинарных признаков M_{binary}. Прочие параметры алгоритма аналогичны параметрам Алгоритма 4.

Алгоритм состоит из следующих шагов:

1. Инициализировать c_{test} пустым списком, а r_{test} – пустым словарем. В c_{test} будут добавляться предсказанные значения целевого признака для тестовых объектов, а в r_{test} – правила для каждого тестового объекта (ключ в словаре – номер объекта, значение – список правил).

2. Посчитать долю положительных объектов в выборке $c_{pos} = \frac{|c'_{train}|}{|G_{train}|}$.

3. Применить дискретизацию к формальному обучающему контексту и составить новые обучающий и тестовый формальные контексты (уже не многозначные)
$\mathbb{K}_{train} = (G_{train}, d(M,W, c_{train}) \cup c_{train}, I_{train})$ и $\mathbb{K}_{test} = (G_{test}, d(M,W), I_{test})$.

Algorithm 5 Concept Lattice-Based Rule-learner (CoLiBRi) – случай количественных признаков.

Input: $K_{train} = (G_{train}, M \cup c_{train}, W, I_{train})$, $K_{test} = (G_{test}, M, W, I_{test})$, $min\_supp \in \mathbb{R}^+$, $n_{rules} \in \mathbb{N}$;
$CbO(K, min\_supp) : K \to \mathcal{S}$;
$sort(\mathcal{S}, inf) : \mathcal{S} \to \mathcal{S}$
$d : M \times W \to \mathbb{R}$, $inf : M \cup c_{train} \to \mathbb{R}$;
Output: c_{test}, r_{test}

$c_{test} = \emptyset, r_{test} = \emptyset$
$c_{pos} = \frac{|c'_{train}|}{|G_{train}|}$
$\mathcal{S} = \{(A,B) \mid A \subseteq G_{train}, B \subseteq M, A^{\diamond} = B, B^{\diamond} = A, |A| \geq min\_supp\} = CbO(K_{train}, min\_supp)$

$\mathcal{S} = sort(\mathcal{S}, inf)$
for $g_t \in G_{test}$ **do**
$\{B_i\}_{i \in [1, n_{rules}]} = \{B \mid (A,B) \in \mathcal{S}, g_t^{\diamond} \sqsubseteq B\}$
$c_i = \frac{|B'_i \cap c'_{train}|}{|B'_i|}$
$r_{test}[i] = \{B_i \to_{c_i} +\}_{i \in [1, n_{rules}]}$
$c_{test}[i] = [\frac{1}{n\_rules} \sum_{i=1}^{n\_rules} c_i \geq c_{pos}]$
end for

4. С помощью алгоритма $CbO(K, min\_supp)$ найти все формальные понятия обучающего контекста $\mathbb{K}_{train}$ со значением поддержки не менее min_supp. Параллельно с этим для каждого формального понятия вычислять значение качества соответствующего классифицирующего правила inf. Таким образом, получится словарь $\mathcal{S}$, ключами которого будут содержания формальных понятий, а значениями – соответствующие значения функционала inf.

5. Отсортировать все формальные понятия $\mathcal{S}$ по посчитанным значениям критерия inf в порядке "улучшения" (то есть по возрастанию inf, если малые значения критерия говорят о хороших правилах (как в случае неопределенности Джини) или по убыванию, если, наоборот, большие значения критерия свидетельствуют о хороших правилах (прирост информации, среднее уменьшение Джини)).

6. Для каждого тестового объекта $g_t \in G_{test}$:

   – Отобрать n_{rules} "подходящих" содержаний формальных понятий, то есть $\{B_i\}_{i \in [1, n_{rules}]} = \{B \mid (A,B) \in \mathcal{S}, g'_t \subseteq B\}$

   – Для каждого из отобранных содержаний формальных понятий $\{B_i\}_{i \in [1, n_{rules}]}$ определить долю положительных объектов $c_+ = \frac{|B'_i \cap c'_{train}|}{|B'_i|}$

   – Сформировать таким образом набор правил $\{B_i \to_{c_i} +\}_{i \in [1, n_{rules}]}$. Записать его в словарь r_{test} для ключа t (номер объекта g_t).

   – Предсказанное значение целевого признака c_{train_t} определить как индикатор того, что усредненное заключение найденных правил превышает долю положительных объектов во всей выборке:

Id	Pclass	Age	City	Survived
1	3	39	S	1
2	3	16	S	1
3	1	62	C	1
4	3	42	S	0
5	2	30	C	0
6	2	18	C	0
7	2	28	C	?
8	1	47	C	?

Таблица 3.5: Подвыборка набора данных о пассажирах Титаника. Признаки: "Pclass" – класс каюты, "City" – место посадки (в данной подвыборке только Шербур (Cherburg, C) или Саутгемптон (Southampton, S), "Age" – возраст пассажира, "Survived" – выжил ли пассажир в катастрофе Титаника.

Id	2	6	5	1	4	3
Age	16	18	30	39	42	62
Survived	1	0	0	1	0	1

$$c_{train_t} = [\frac{1}{n\_rules} \sum_{i=1}^{n\_rules} c_i \geq c_{pos}].$$

Добавить это значение в c_{test}.

***Пример* 12.** *В Таблице 3.5 представлена подвыборка набора данных о пассажирах Титаника[2]. Покажем, как для такой выборки применить Алгоритм 4 и сделать прогноз для объекта 7 с признаками "Pclass=2, Age=28, City=C". В выборке имеется количественный признак Age. Дискретизируем его с помощью простой процедуры, которая применяется в алгоритме CART [Bre+84]. Отсортируем объекты по признаку Age в порядке возрастания и будем отслеживать целевой признак Survived:*

Видно, что признак Survived меняет значение с 1 на 0 при переходе от значения Age=16 к Age=18 (среднее между ними – 17), а также при переходе от значения Age=39 к Age=42 (среднее между ними – 34.5). Кроме того, признак Survived меняет значение с 0 на 1 при переходе от значения Age=39 к Age=42 (среднее между ними – 40.5) и от Age=42 к Age=62 (среднее между ними – 52). Таким образом, признак Age дискретизируется порогами $T = \{17, 34.5, 40.5, 52\}$*, то есть порождаются 8 новых признаков: "*$Age \leq 17$*", "*$Age \geq 17$*", "*$Age \leq 34.5$*", "*$Age \geq 34.5$*", "*$Age \leq 40.5$*", "*$Age \geq 40.5$*", "*$Age \leq 52$*" и "*$Age \geq 52$*".*

Признак $Pclass$ *– категориальный. Применив к нему One Hot Encoding и добавив отрицания новых признаков, получим 6 признаков: "*$Pclass == 1$*", "*$Pclass \neq 1$*", "*$Pclass == 2$*", "*$Pclass \neq 2$*", "*$Pclass \neq 3$*" и "*$Pclass \neq 3$*".*

[2] https://www.kaggle.com/c/titanic

	Длина чашелистика (petal length, pl)	***Ширина чашелистика (petal width, pw)***	***Длина лепестка (sepal length, sl)***	***Ширина лепестка (sepal width, sw)***	***Вид ириса (+ - versicolor, – - virginica)***
1	*6.2*	*2.9*	*4.3*	*1.3*	*1*
2	*5.1*	*2.5*	*3.0*	*1.1*	*1*
3	*5.7*	*2.8*	*4.1*	*1.3*	*1*
4	*6.3*	*3.3*	*6.0*	*2.5*	*2*
5	*5.8*	*2.7*	*5.1*	*1.9*	*2*
6	*7.1*	*3.0*	*5.9*	*2.1*	*2*
7	*4.9*	*2.5*	*4.5*	*2.7*	***?***
8	*6.6*	*3.0*	*4.4*	*1.4*	***?***

Таблица 3.6: Бинарная классификация на 2 вида цветков ириса.

Признак $City$ – категориальный с двумя уникальными значениями (в данной подвыборке), поэтому заменим его на 2 признака: "$City == C$" и "$City \neq C$".

В итоге с 16 новыми бинарными признаками и одним целевым (Survived) задача классификации сводится к предыдущему случаю, для которого предложен Алгоритм 4. Тестовый объект 7 описывается бинарными признаками
"$Pclass == 1, Pclass \neq 2, Pclass \neq 3, Age \geq 17, Age \leq 34.5, Age \leq 40.5, Age \leq 52, City == C$", и классифицирующие правила будут подбираться соответствующие.

В случае наличия у объектов количественных и интервальных признаков алгоритм в общем остается тем же, что и Алгоритм 4, только для частного случая интервальных узорных структур. Заметим, что на вход Алгоритму 4 можно подавать обучающую и тестовую узорные структуры, являющиеся проекцией некоторой другой узорной структуры. В данном случае они могут получаться с помощью дискретизирующих проекций.

***Пример* 13.** *Рассмотрим пример бинарной классификации для подвыборки цветов ириса Фишера*[3] *– Таблица 3.6.*

Если перейти от многозначного контекста к бинарному, то, согласно Алгоритму 5, надо выбрать пороги $T = \{pl : \{5.75, 5.85, 6.0, 6.25\}, pw : \{2.75, 2.95\}, sl : \{4.7\}, sw : \{1.6\}\}$ для дискретизации исходных признаков, то есть перейти к формальному контексту с признаками $pl \leq 5.75$, $pl \geq 5.75$, $pl \leq 5.85$, . . ., $sw \geq 1.6$. Решетка формальных понятий такого контекста показана на Рис. 3.2. Как было показано ранее, решетка узорных понятий узорной структуры, полученной из исходной с помощью дискретизирующей проекции с порогами T, изоморфна данной.

Получается список классифицирующих правил, представленный Таблицей 3.7. Если для классификации брать одно лучшее правило, то объект 7 классифицируется положительно (как тип ириса versicolor) правилом 1, а объект 8 – отрицательно (как тип ириса virginica) правилом 2.

[3] http://archive.ics.uci.edu/ml/datasets/Iris

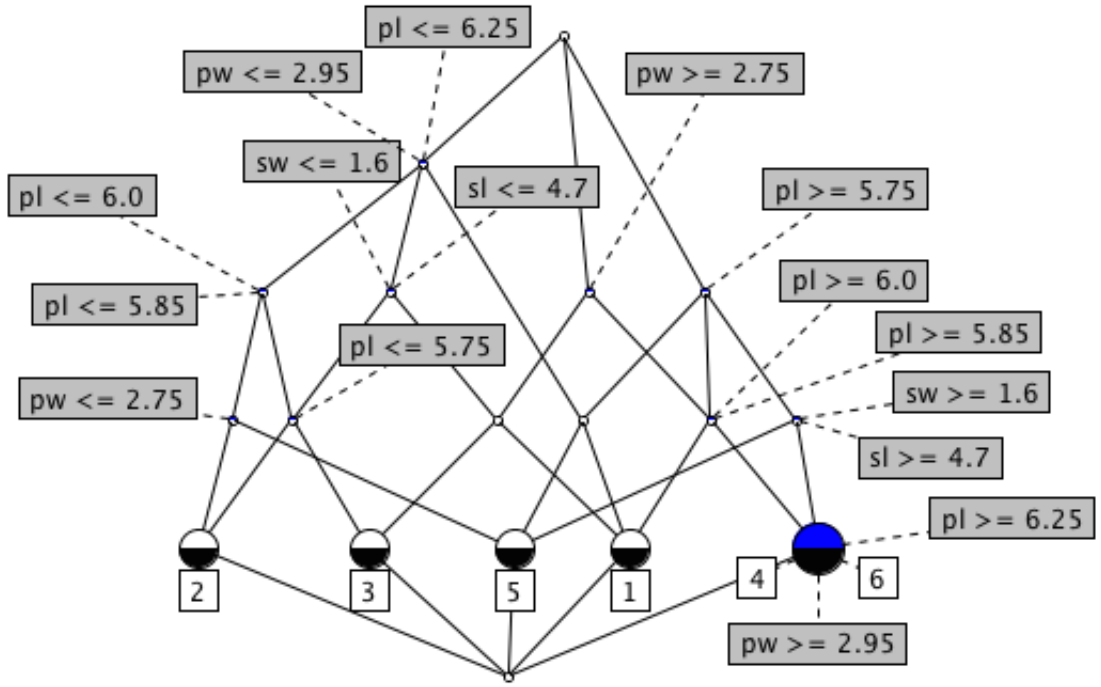


Рисунок 3.2: Диаграмма решетки формальных понятий для контекста, полученного из контекста в Таблице 3.6 дискретизацией с порогами $T = \{pl : \{5.75, 5.85, 6.0, 6.25\}, pw : \{2.75, 2.95\}, sl : \{4.7\}, sw : \{1.6\}\}$.

	Классифицирующее правило	***Объекты***	***Неопределенность Джини***
1	$\langle[6.25, +\infty], [2.95, +\infty],[4.7, +\infty],[1.6, +\infty]\rangle \rightarrow_{(1)} 2$	| 4,5,6	*0*
2	$\langle[-\infty,6.25], [-\infty,2.95],[-\infty,4.7],[-\infty,1.6]\rangle \rightarrow_{()} 1$	1,2,3 |	*0*
3	$\langle[5.75, +\infty], [-\infty, +\infty],[-\infty, +\infty],[-\infty, +\infty]\rangle \rightarrow_{(0.75)} 2$	1 | 4,5,6	*0.25*
4	$\langle[-\infty, 6.25], [-\infty,2.95],[-\infty, +\infty],[-\infty, +\infty]\rangle \rightarrow_{(0.75)} 1$	1,2,3 | 5	*0.25*
5	$\langle[6.0, +\infty], [2.75, +\infty],[-\infty + \infty],[-\infty, +\infty]\rangle \rightarrow\rightarrow_{(0.66)} 2$	1 | 4,6	*0.44*
6	$\langle[-\infty,5.85], [-\infty,2.95],[-\infty, +\infty],[-\infty + \infty]\rangle \rightarrow\rightarrow_{(0.66)} 1$	2,3 | 5	*0.44*

Таблица 3.7: Классифицирующие правила в Примере 13. Символ | отделяет объекты разных классов.

3.4. Классификация данных со сложной структурой на основе множества узорных понятий

Для работы со сложными структурами используется модификация алгоритмов построения решеток узорных понятий. В целом используется тот же подход, что и в алгоритме “Замыкай по-Одному” [Куз93], только теоретико-множественная операция пересечения заменяется на полурешёточную операцию сходства (см. Раздел 2.3), а операция проверки того, что одно множество есть подмножество другого заменяется на проверку поглощения одного элемента полурешетки другим. В Алгоритме 6, демонстрируется псевдокод алгоритма “Замыкай по-Одному”, адаптированного для работы с произвольными замкнутыми описаниями (узорными структурами). Для

каждого узорного понятия находятся все его канонические соседи сверху. Для всех допустимых расширений объема исходного понятия проверяется, является ли допустимым замыкание этого расширения. При этом допустимость объема проверяется в точности как и в оригинальном алгоритме [Куз93].

Algorithm 6 Версия алгоритма “Замыкай по-Одному”, вычисляющая решетку узорных понятий.

Input: $(G, (D, \sqcap), \delta)$, объем Ext и содержание Int некоторого понятия.

Output: Все канонические предки (Ext, Int) в решётке понятий.

Function `CloseByOne`(Ext, Int)

for $S \subseteq G, S \succ Ext$ **do**
  $NewInt \longleftarrow \prod_{g \in S} \delta(g)$
  $NewExt \longleftarrow \{g \in G \mid NewInt \sqsubseteq \delta(g)\}$
  if `IsCanonicExtension`$(Ext, NewExt)$ **then**
    `SaveConcept`$(NewExt, NewInt)$;
    `CloseByOne`$(NewExt, NewInt)$;
  end if
end for
`CloseByOne`$(\emptyset, \top)$;

Ранее в Главе 3.2 мы описывали алгоритм классификации данных с бинарными признаками с помощью формальных понятий – CoLiBRi. Теперь, обсудив, как алгоритм нахождения формальных понятий обобщается для нахождения узорных понятий произвольной узорной структуры, опишем в Алгоритме 7, преложенном в [KK16b], модификацию подхода CoLiBRi для классификации данных со сложной структурой.

На вход алгоритму подаются обучающая и тестовая узорные структуры $PS_{train} = (G_{train}, ((D, \sqcap), c_{train}). \delta_{train})$ и $PS_{test} = (G_{test}, (D, \sqcap), \delta_{test})$. Алгоритм использует модификацию 6 алгоритма “Замыкай по-Одному” ($CbO_{PS}(PS, min\_supp)$) [Куз93], в которой выдаются все узорные понятия узорной структуры PS, поддержки которых ограничены снизу значением параметра min_supp. Для выбора классифицирующих правил используется критерий $inf : D \times c_{train} \to \mathbb{R}$ типа неопределенности Джини или энтропийного прироста информации (в программной реализации по умолчанию – среднее значение неопределенности Джини). Параметры алгоритма: min_supp и n – минимальная поддержка классифицирующих правил и число правил, используемых для классификации тестового объекта.

Алгоритм состоит из следующих шагов:

1. Инициализировать c_{test} пустым списком, а r_{test} – пустым словарем. В c_{test} будут добавляться предсказанные значения целевого признака для тестовых объектов, а в r_{test} – правила для каждого тестового объекта (ключ в словаре – номер объекта, значение – список правил).

2. Посчитать долю положительных объектов в выборке $c_{pos} = \frac{|c'_{train}|}{|G_{train}|}$.

3. С помощью алгоритма $CbO_{PS}(PS, min\_supp)$ найти все узорные понятия обучающей узорной структуры PS_{train} со значением поддержки не менее min_supp. Параллельно с этим для каждого узорного понятия вычислять значение качества соответствующего классифицирующего правила inf. Таким образом, получится словарь $\mathcal{S}$, ключами которого будут содержания узорных понятий, а значениями – соответствующие значения функционала inf.

4. Отсортировать все узорные понятия $\mathcal{S}$ по посчитанным значениям критерия inf в порядке "улучшения" (то есть по возрастанию inf, если малые значения критерия говорят о хороших правилах (как в случае неопределенности Джини) или по убыванию, если, наоборот, большие значения критерия свидетельствуют о хороших правилах (прирост информации, среднее уменьшение Джини)).

5. Для каждого тестового объекта $g_t \in G_{test}$:

   – Отобрать n_{rules} "подходящих" содержаний формальных понятий, то есть $\{d_i\}_{i\in[1,n_{rules}]} = \{d \mid (A,d) \in \mathcal{S}, g_t^{\diamond} \sqsubseteq B\}$

   – Для каждого из отобранных содержаний формальных понятий $\{d_i\}_{i\in[1,n_{rules}]}$ определить долю положительных объектов $c_i = \frac{|d_i^{\diamond} \cap c'_{train}|}{|d_i^{\diamond}|}$

   – Сформировать таким образом набор правил $\{d_i \to_{c_i} +\}_{i\in[1,n_{rules}]}$. Записать его в словарь r_{test} для ключа t (номер объекта g_t).

   – Предсказанное значение целевого признака c_{train_t} определить как индикатор того, что усредненное заключение найденных правил превышает долю положительных объектов во всей выборке:

   $$c_{train_t} = [\frac{1}{n\_rules} \sum_{i=1}^{n\_rules} c_i \geq c_{pos}].$$

   Добавить это значение в c_{test}.

***Пример* 14.** *В задаче предсказания наличия некоторого свойства P химических веществ дана обучающая выборка в виде упрощенной молекулярной структуры 4 положительных веществ и 3 отрицательных веществ. Про положительные объекты известно, что они обладают свойством*

Algorithm 7 Concept Lattice-Based Rule-learner (CoLiBRi) – случай данных со сложной структурой.

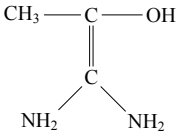
Input: $PS_{train} = (G_{train}, ((D, \sqcap), c_{train}), \delta_{train})$
$PS_{test} = (G_{test}, (D, \sqcap), \delta_{test})$
$min\_supp \in \mathbb{R}^+, n_{rules} \in \mathbb{N}$;
$CbO_{PS}(PS, min\_supp) : PS \to \mathcal{S}$;
$inf : D \times c_{train} \to \mathbb{R}$;
$sort(\mathcal{S}, inf) : \mathcal{S} \to \mathcal{S}$

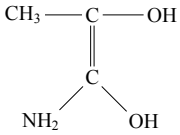
Output: c_{test}, r_{test}

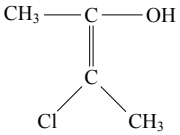
$c_{test} = \emptyset, r_{test} = \emptyset$
$c_{pos} = \frac{|c'_{train}|}{|G_{train}|}$
$\mathcal{S} = \{(A,d) : inf(d, c_{train}) \mid A \subseteq G_{train}, d \in D, A^\diamond = d, d^\diamond = A, |A| \geq min\_supp\} = CbO_{PS}(PS_{train}, min\_supp)$
$\mathcal{S} = sort(\mathcal{S}, inf)$
for $g_t \in G_{test}$ **do**
  $\{d_i\}_{i \in [1, n_{rules}]} = \{d \mid (A,d) \in \mathcal{S}, g_t^\diamond \sqsubseteq d\}$
  $c_i = \frac{|d_i^\diamond \cap c'_{train}|}{|d_i^\diamond|}$
  $r_{test}[i] = \{d_i \to_{c_i} +\}_{i \in [1, n_{rules}]}$
  $c_{test}[i] = [\frac{1}{n\_rules} \sum_{i=1}^{n\_rules} c_i \geq c_{pos}]$
end for

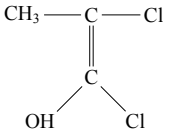
P, про отрицательные известно, что нет. Для тестовых объектов необходимо сделать прогноз, обладают ли они свойством P.

Положительные объекты:

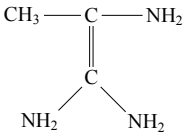
1 : 

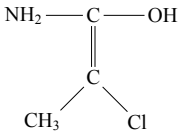
2 : 

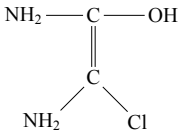
3 : 

4 : 

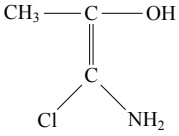
Отрицательные объекты:

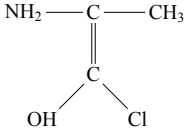
5 : 

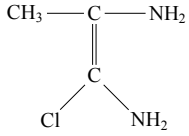
6 : 

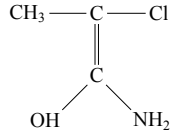
7 : 

Тестовые объекты:

8 : 

9 : 

10 : 

11 : 

Доля положительных объектов в обучающей выборке равна 0.57 (4 из 7).

Найдем все узорные понятия обучающей узорной структуры $PS_{train} = (G_{train}, ((D, \sqcap), c_{train}), \delta_{train})$ *с абсолютной поддержкой не менее 4 (*$min\_supp = \frac{4}{7}$*). Здесь* $G_{train} = \{1, \ldots, 7\}$, D *– множество всех помеченных графов,* $\sqcap$ *– полурешёточная операция для помеченных графов, функция* δ_{train} *задана выше, а* $c_{train} = \{+, +, +, +, -, -, -\}$. *Правила, построенные на основе найденных узорных понятий, указаны в Таблице 3.8. Если делать прогнозы с помощью трех лучших правил (*$n\_rules = 3$*), то объекты 8,9,11 классифицируются положительно (*$\frac{1}{3}(0.8 + 0.4 + 0.66) \approx 0.62 > 0.57$*), а объект 10 – отрицательно (*$\frac{1}{3}(0.4 + 0.66 + 0.5) \approx 0.52 < 0.57$*).*

	Классифицирующее правило	***Объекты***	***Средняя неопределенность Джини***
1	$\{CH_3 - C = C, OH - C = C\} \xrightarrow[(0.8)]{} +$	1,2,3,4 | 6	*0.22*
2	$\{C = C - NH_2\} \xrightarrow[(0.4)]{} +$	1,2 | 5,6,7	*0.34*
3	$\{C = C - CH_3\} \xrightarrow[(0.67)]{} +$	1,2,3,4 | 5,6	*0.38*
4	$\{C = C - OH\} \xrightarrow[(0.67)]{} +$	1,2,3,4 | 6,7	*0.38*
5	$\{CH_3 - C = C - OH\} \xrightarrow[(0.75)]{} +$	2,3,4 | 6	*0.4*
6	$\{CH_3 - C = C - NH_2\} \xrightarrow[(0.5)]{} +$	1,2 | 5,6	*0.47*
7	$\{C = C\} \xrightarrow[(0.57)]{} +$	1,2,3,4 | 5,6,7	*0.49*

Таблица 3.8: Классифицирующие правила в Примере 14. Символом | отделены положительные объекты от отрицательных.

3.5. Заключение

В Главе 3 описан алгоритм классификации данных со сложной структурой, основанный на узорных структурах и их проекциях, а также разобраны примеры применения этих алгоритмов в задачах классификации данных с бинарными признаками (Раздел 3.2), с количественными признаками (Раздел 3.3), а также в случае данных, где объекты представлены описаниями в виде помеченных графов (Раздел 3.4).

Глава 4

Эксперименты с реальными данными

4.1. Введение

В этой главе мы опишем разработанный программный комплекс, реализующий алгоритмы, описанные в Главе 3, затем приведем результаты вычислительных экспериментов с наборами данных репозитория UCI (UC Irvine Machine Learning Repository)[1] – крупнейшего репозитория реальных и модельных задач машинного обучения. Репозиторий содержит данные по прикладным задачам в области биологии, медицины, физики, техники, социологии, и др. Именно эти задачи и наборы данных чаще всего используются научным сообществом для эмпирического анализа алгоритмов машинного обучения.

Также в этой главе приводятся результаты экспериментов с данными, представленными последовательностями и графами.

4.2. Программная реализация алгоритмов классификации на основе множеств формальных и узорных понятий

Структура основных классов программного комплекса CoLiBRi, реализующего Алгоритмы 4, 5 и 7, описанные в Главе 3, представлена на Рис. 4.1. На схеме стрелки синего цвета соответствуют отношению “быть наследником класса”, а стрелки черного цвета – отношению “задействовать”.

Имеются 4 абстрактных класса: DescriptionElement, Description, Concept и CoLiBRi. У каждого из них, в свою очередь по 4 наследника.

Класс BinaryCoLiBRi реализует Алгоритм 4, используя класс BinaryConcept. Каждый экземпляр класса BinaryConcept – это кортеж из множества чисел (номеров объектов) и экземпляра класса BinaryDescription. Каждый экземпляр класса BinaryDescription – это упорядоченное множество экземпляров класса BinaryDescriptionElement (число 0 или 1 в зависимости от того, присутствует определенный признак в описании объекта или нет).

[1] http://archive.ics.uci.edu/ml/

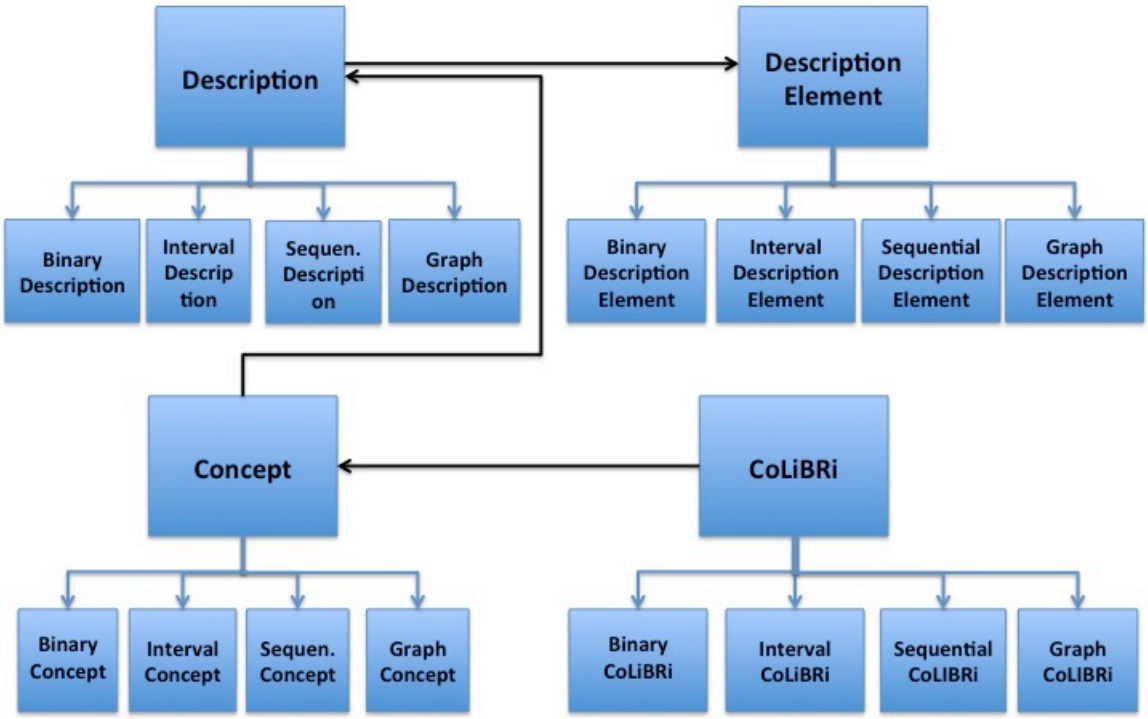


Рисунок 4.1: Структура основных классов программного комплекса CoLiBRi.

Класс IntervalCoLiBRi реализует Алгоритм 5, используя класс IntervalConcept. Каждый экземпляр класса IntervalConcept – это кортеж из множества чисел (номеров объектов) и экземпляра класса IntervalDescription. Каждый экземпляр класса IntervalDescription – это упорядоченное множество экземпляров класса IntervalDescriptionElement (упорядоченная пара двух чисел, соответствующая интервалу).

Класс SequentialCoLiBRi реализует адаптацию Алгоритма 7 для работы с узорными структурами для последовательностей и их проекциями, используя класс SequentialConcept. Каждый экземпляр класса SequentialConcept – это кортеж из множества чисел (номеров объектов) и экземпляра класса SequentialDescription. Каждый экземпляр класса SequentialDescription – это упорядоченное множество экземпляров класса SequentialDescriptionElement (по сути, строк в формате последовательностей SPMF[2]).

Класс GraphCoLiBRi реализует Алгоритм 7, используя класс GraphConcept. Каждый экземпляр класса GraphConcept – это кортеж из множества чисел (номеров объектов) и экземпляра класса GraphDescription. Каждый экземпляр класса GraphDescription – это множество экземпляров класса GraphDescriptionElement.

Общая информация о программном комплексе:

– Комплекс реализован на языках Python, Java и C++

[2] http://www.philippe-fournier-viger.com/spmf/

- Размер кода: ≈ 2200 строк
- Основные классы:
  - AbstractDescriptionElement (наследники Binary*, Interval*, Sequential*, Graph*) – элемент описания. Среди наследников это может быть просто номер признака (BinaryDescriptionElement), вещественный интервал (IntervalDescriptionElement), строка, задающая последовательность признаков (SequentialDescriptionElement), или помеченный граф (GraphDescriptionElement);
  - AbstractDescription (+ наследники) – описание в терминах узорных структур. В реализации – контейнер экземпляров класса AbstractDescriptionElement;
  - AbstractConcept (+ наследники) – узорное понятие;
  - AbstractCoLiBRi (+ наследники) – реализации Алгоритма 4 (BinaryCoLiBRi), Алгоритма 5 (IntervalCoLiBRi) и Алгоритма 7 (SequentialCoLiBRi и GraphCoLiBRi)
  - AbstractJSM (+ наследники) – реализации классического ДСМ-метода классификации и его версии для узорных структур;
  - AbstractLAC (+ наследники) – реализации алгоритма классификации по запросу и его версии для узорных структур;
  - AbstractCbO (+ наследники) – алгоритм “Замыкай-по-Одному” (“Close-by-One”) для бинарных признаков и узорных структур, используется в AbstractJSM и наследниках;
  - BinaryInCloseWrapper – расширение алгоритма In-Close 2.6[3] [And09] для подсчета критериев информативности типа неопределенности Джини и прироста информации для каждого замкнутого множества признаков, удовлетворяющего ограничениям на поддержку, мощность множества признаков и т.д.;
  - SeqCharmWrapper – расширение реализации алгоритма Charm[4] [ZH02] в SPMF для подсчета критериев информативности типа неопределенности Джини и прироста информации для каждого замкнутого множества последовательностей, удовлетворяющего ограничениям на поддержку, число элементов последовательности и т.д.;
  - GraphGastoneWrapper – расширение реализации алгоритма Gaston[5] [NK05] для подсчета критериев информативности типа неопределенности Джини и прироста информации для каждого замкнутого множества графов, удовлетворяющего ограничениям на поддержку, число подграфов и т.д.

[3] https://sourceforge.net/projects/inclose/
[4] http://www.philippe-fournier-viger.com/spmf/index.php?link=download.php
[5] http://liacs.leidenuniv.nl/~nijssensgr/gaston/

4.3. Эксперименты на данных репозитория UCI

4.3.1. Данные с бинарными и категориальными признаками

Версия алгоритма CoLiBRi ("Concept Lattice-Based Rule-learner") для работы с бинарными признаками (Алгоритм 4) была протестирована на 13 наборах данных UCI[6]. Сравнение проводилось с реализациями Scikit-learn [Ped+11] алгоритмов построения деревьев решений CART [Bre+84], случайного леса [Bre01], а также с методом ближайших соседей. Для каждого набора данных решалась задача бинарной классификации, где выделялись самый частый класс и все остальные. Категориальные признаки были преобразованы в бинарные методом One Hot Encoding. Отслеживались значения доли правильных ответов и F1-метрики при 5-кратной кросс-валидации.

Данные	DT acc	RF acc	kNN acc	CoLiBRi acc	DT F1	RF F1	kNN F1	CoLiBRi F1
audiology	0.75	**0.8**	0.63	0.79*	0.71	**0.74**	0.58	**0.74**
breast-cancer	0.63	0.66	**0.76**	0.65	0.58	0.63	**0.75**	0.61
breast-wisc	0.7	0.74	0.73	**0.76**	**0.45**	0.42	0.38	0.44*
car	0.75	0.78*	0.71	**0.79**	0.75	**0.76**	0.71	**0.76**
hayses-roth	0.84*	0.83*	0.49	**0.86**	0.84*	0.82	0.49	**0.85**
lymph	0.8	0.83	**0.86**	0.83	0.77	**0.85**	0.84*	0.84*
mol-bio-prom	0.78	**0.83**	**0.83**	0.82*	0.78	**0.84**	0.8	0.83*
nursery	0.64	0.65	**0.72**	0.65	0.62	0.62	**0.7**	0.62
primary-tumor	0.41	**0.46**	0.41	0.45*	0.37	**0.41**	0.37	0.4*
solar-flare	0.7*	0.7*	0.63	**0.72**	0.67	0.69*	0.6	**0.71**
soybean	0.91*	0.91*	**0.92**	0.91*	0.91*	**0.93**	0.92*	0.91*
spect-train	0.61	0.69	0.68	0.7	0.34	0.36	0.23	0.38
tic-tac-toe	0.79	0.79	**0.85**	0.78	0.82	0.86	**0.89**	0.85

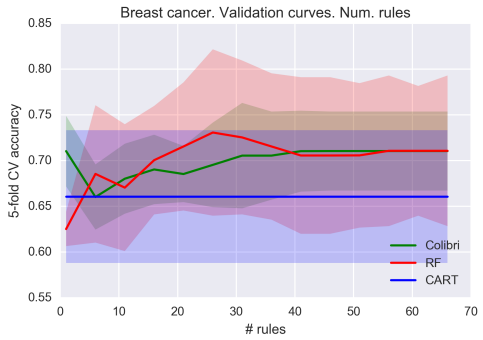
Таблица 4.1: Значения доли правильных ответов и F1-метрики для 13 наборов данных репозитория UCI. "DT acc" и "DT F1" означают средние по 5 запускам доли правильных ответов и F1-метрики алгоритма CART при 5-кратной кросс-валидации , …, "CoLiBRi F1" означает среднее по 5 запускам значение F1-метрики алгоритма CoLiBRi при 5-кратной кросс-валидации. Жирным выделены лучшие значения метрик, звездочками отмечены значения, которые не являются статистически значимо уступающими лучшим.

Мы использовали неопределенность Джини как критерий выбора правил. Значения параметра $min\_samples\_leaf \in [1,10]$ для деревьев и леса, а также $n\_neighbors \in \{1,2,5,15,30,50\}$ для метода ближайших соседей подбирались в процессе 5-кратной кросс-валидации. Для случайного леса каждый раз строилось 10 деревьев.

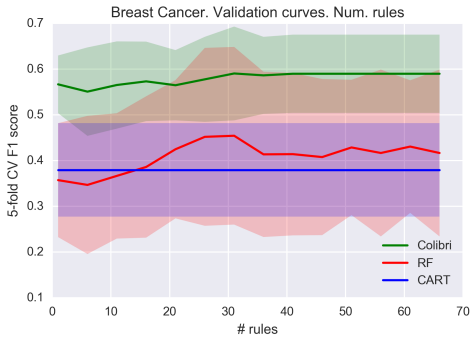
Параметр min_supp для "CoLiBRi" брался равным параметру $min\_samples\_leaf$ алгоритма CART для каждого набора данных. Для определения метки каждого тестового объекта использовалось $n\_rules = 10$ правил.

Результаты представлены в Таблице 4.1. Каждое значение – это усредненные по 5 запускам 5-кратной кросс-валидации значения доли правильных ответов и F1-метрики. Жирным шрифтом выделены лучшие значения метрик, звездочками отмечены значения, которые не являются статистически значимо уступающими лучшим. В качестве статистического критерия использовался

[6] http://repository.seasr.org/Datasets/UCI/csv/

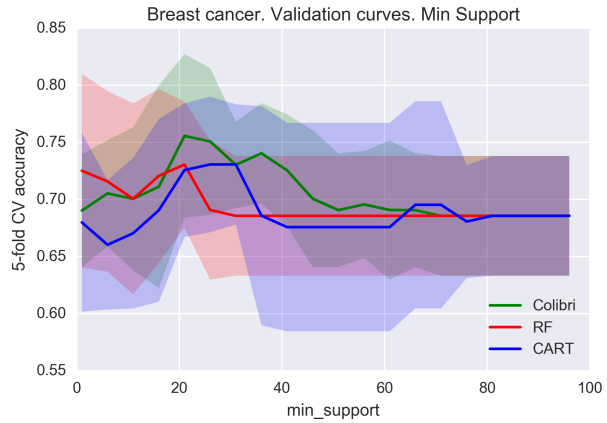


(a) Доля верных ответов

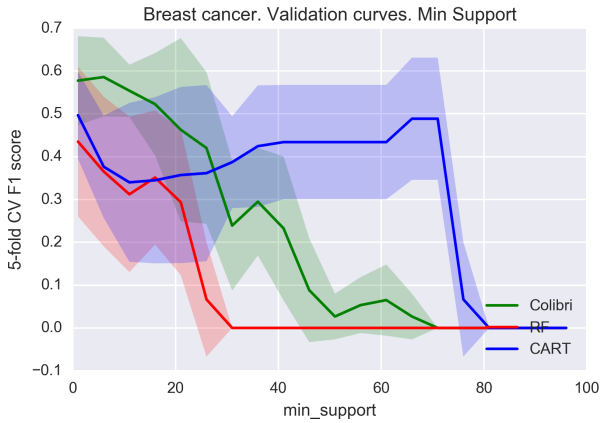


(b) F1

Рисунок 4.2: Кривые валидации по числу правил (для CoLiBRi) или деревьев (для случайного леса) в сравнении с деревом решений CART. 5-кратная стратифицированная кросс-валидация для набора данных Breast Cancer репозитория UCI.



(c) Доля верных ответов



(d) F1

Рисунок 4.3: Кривые валидации по минимальной поддержке для CoLiBRi, случайного леса и дерева решений CART. 5-кратная стратифицированная кросс-валидация для набора данных Breast Cancer репозитория UCI.

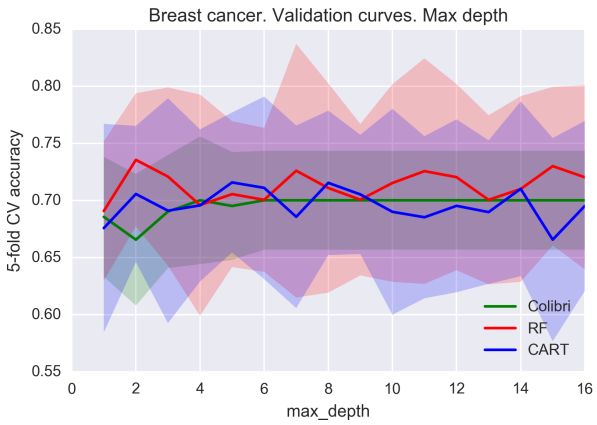
непараметрический критерий знаков для связанных выборок на уровне значимости 0.05. Лучшие значения параметров для каждого алгоритма можно найти в Приложении 4.6 в Таблице 4.12.

По результатам вычислительных экспериментов можно заключить, что для большинства наборов данных CoLiBRi имеет статистически лучшие метрики качества классификации, чем CART. При этом по сравнению со случайным лесом и методом ближайших соседей результаты получаются примерно одинаковыми (статистически не значимо хуже и не значимо лучше).

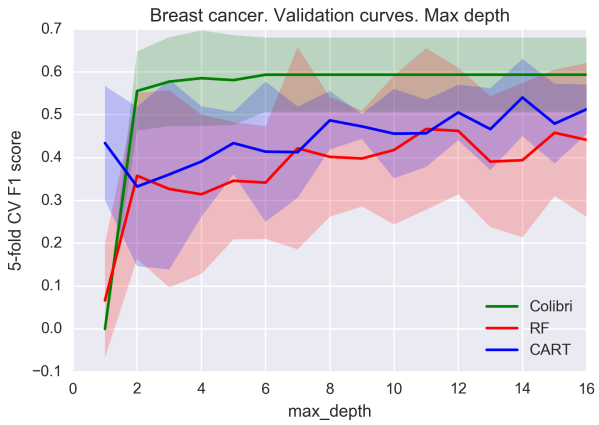
Также изучалась зависимость качества алгоритмов от значений параметров. Для этого были построены кривые валидации по числу правил, минимальной поддержке и максимальной мощности посылки правил для наборов данных репозитория UCI.

Для набора данных по по раку молочной железы репозитория UCI (Breast Cancer[7]) кривые валидации по числу правил представлены на Рис. 4.2a (доля правильных ответов) и 4.2b (F1-метрика), по минимальной поддержке – на Рис. 4.3c (доля правильных ответов) и 4.3d (F1-

[7] https://archive.ics.uci.edu/ml/datasets/Breast+Cancer

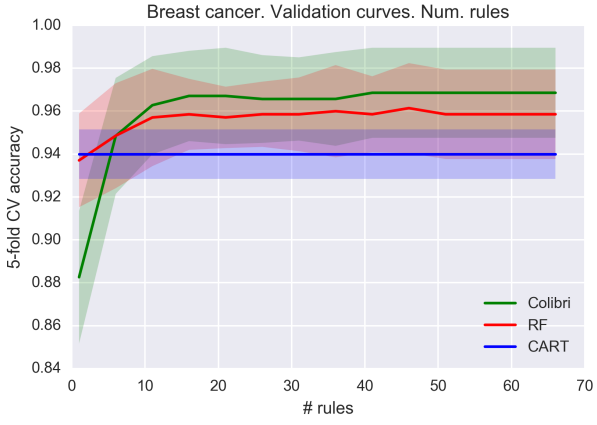


(a) Доля верных ответов

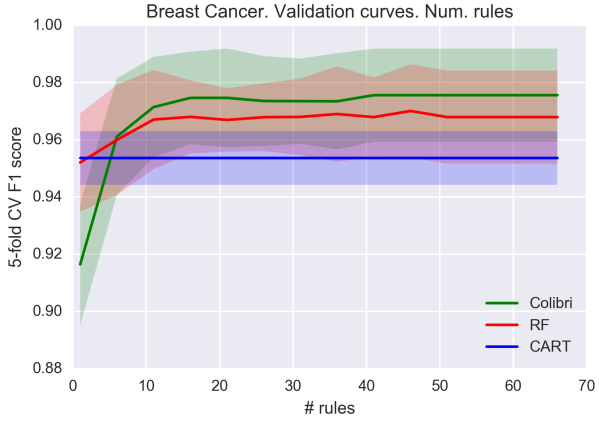


(b) F1

Рисунок 4.4: Кривые валидации по максимальной длине посылки правил для CoLiBRi, случайного леса и дерева решений CART. 5-кратная стратифицированная кросс-валидация для набора данных Breast Cancer репозитория UCI.

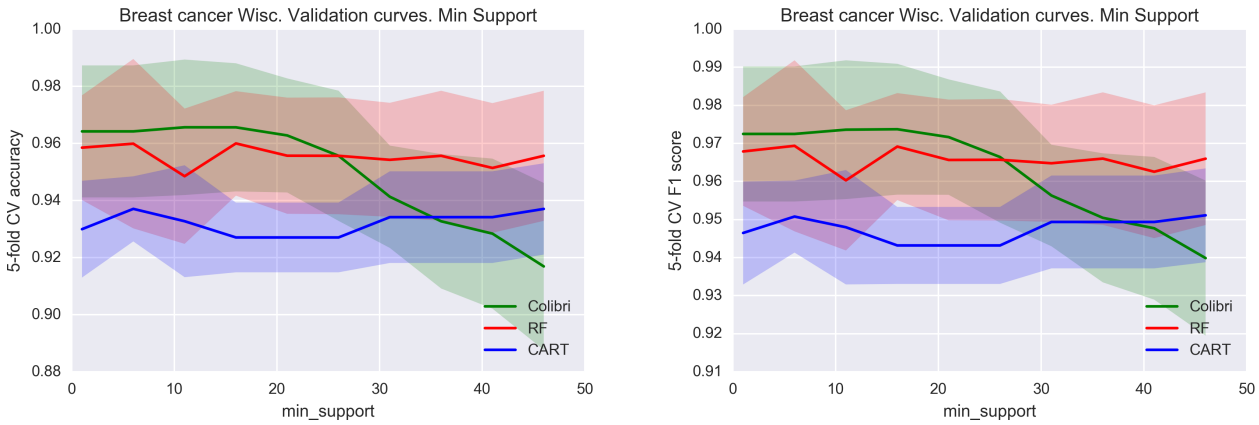


(a) Доля верных ответов



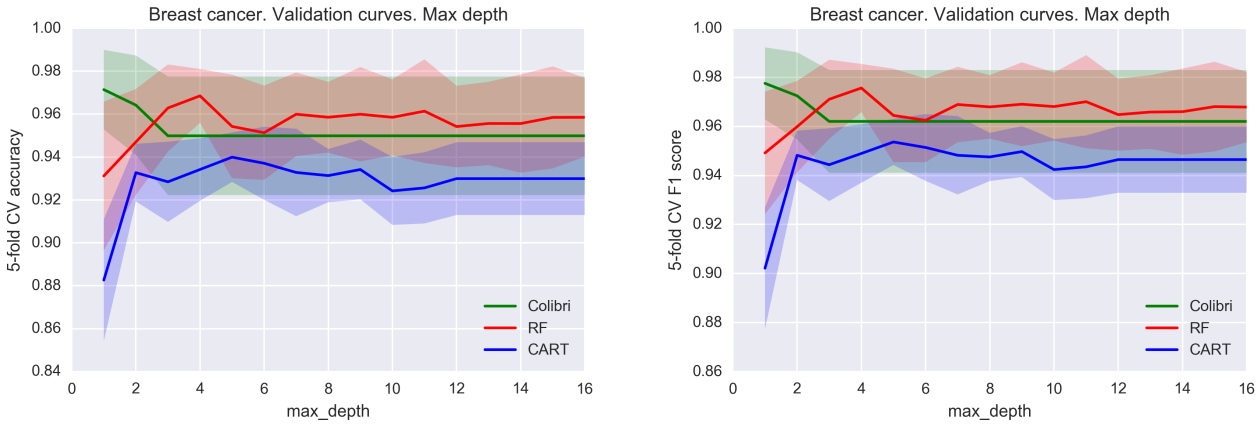
(b) F1

Рисунок 4.5: Кривые валидации по числу правил (для CoLiBRi) или деревьев (для случайного леса) в сравнении с деревом решений CART. 5-кратная стратифицированная кросс-валидация для набора данных Breast Cancer репозитория UCI.



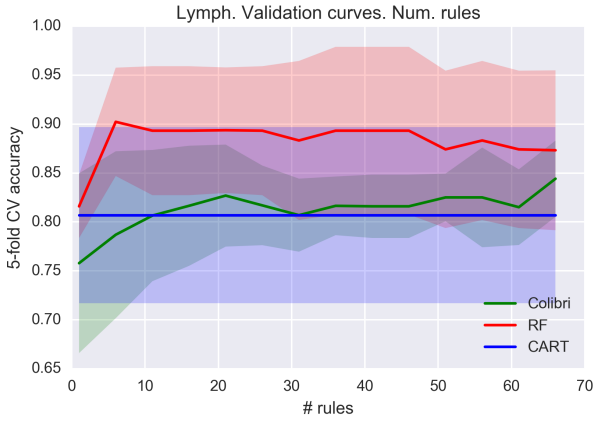
(a) Доля верных ответов (b) F1

Рисунок 4.6: Кривые валидации по минимальной поддержке для CoLiBRi, случайного леса и дерева решений CART. 5-кратная стратифицированная кросс-валидация для набора данных Breast Cancer Wisconsin репозитория UCI.

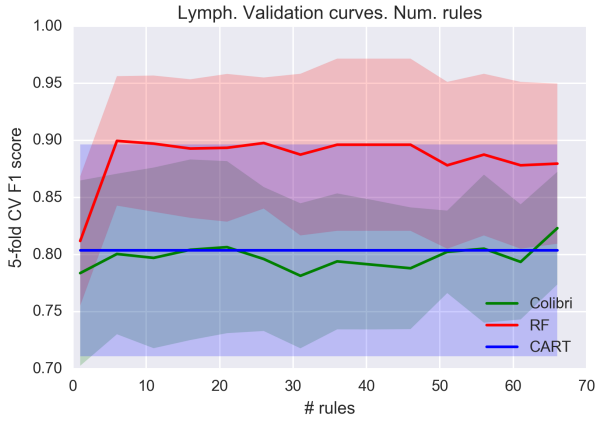


(a) Доля верных ответов (b) F1

Рисунок 4.7: Кривые валидации по максимальной длине посылки правил для CoLiBRi, случайного леса и дерева решений CART. 5-кратная стратифицированная кросс-валидация для набора данных Breast Cancer Wisconsin репозитория UCI.

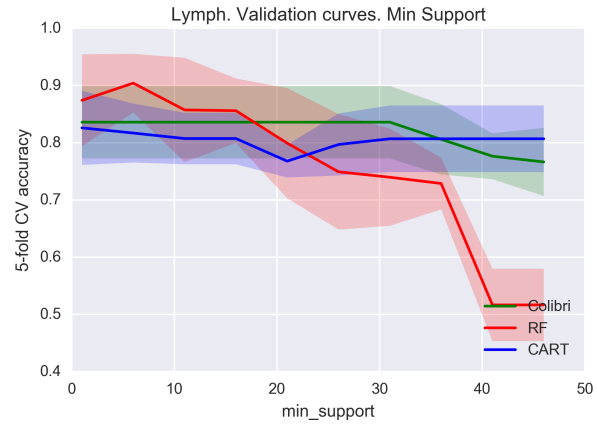


(a) Доля верных ответов

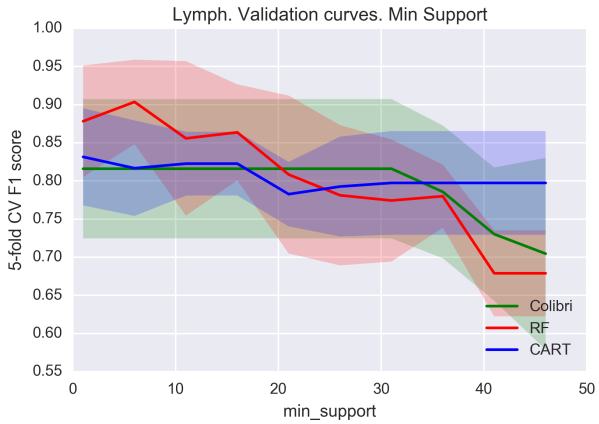


(b) F1

Рисунок 4.8: Кривые валидации по числу правил (для CoLiBRi) или деревьев (для случайного леса) в сравнении с деревом решений CART. 5-кратная стратифицированная кросс-валидация для набора данных Lymph репозитория UCI.

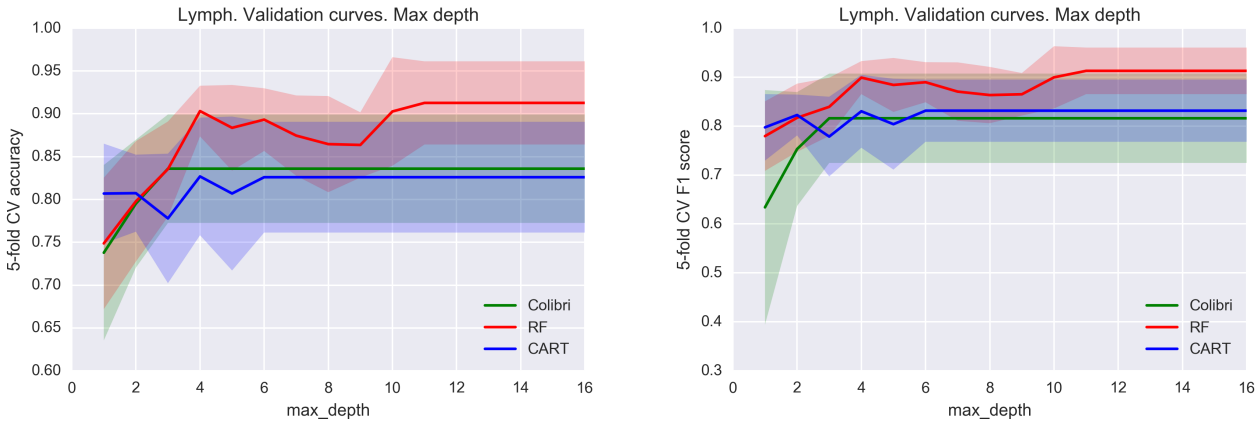


(a) Доля верных ответов



(b) F1

Рисунок 4.9: Кривые валидации по минимальной поддержке для CoLiBRi, случайного леса и дерева решений CART. 5-кратная стратифицированная кросс-валидация для набора данных Lymph репозитория UCI.



(a) Доля верных ответов

(b) F1

Рисунок 4.10: Кривые валидации по максимальной длине посылки правил для CoLiBRi, случайного леса и дерева решений CART. 5-кратная стратифицированная кросс-валидация для набора данных Lymph репозитория UCI.

метрика), а по максимальной мощности посылки правил (для деревьев это максимальная глубина) – на Рис. 4.4a (доля правильных ответов) и 4.4b (F1-метрика).

Для расширенного набора данных по раку молочной железы репозитория UCI (Breast Cancer Wisconsin[8]) кривые валидации по числу правил представлены на Рис. 4.5a (доля правильных ответов) и 4.5b (F1-метрика), по минимальной поддержке – на Рис. 4.6a (доля правильных ответов) и 4.6b (F1-метрика), а по максимальной мощности посылки правил (для деревьев это максимальная глубина) – на Рис. 4.7a (доля правильных ответов) и 4.7b (F1-метрика).

Для набора данных по лимфографии репозитория UCI (Lymph[9]) кривые валидации по числу правил представлены на Рис. 4.8a (доля правильных ответов) и 4.8b (F1-метрика), по минимальной поддержке – на Рис. 4.9a (доля правильных ответов) и 4.9b (F1-метрика), а по максимальной мощности посылки правил (для деревьев это максимальная глубина) – на Рис. 4.10a (доля правильных ответов) и 4.10b (F1-метрика).

Распределения мощностей посылок правил (“длин” правил), которыми определялись метки тестовых объектов для 3 наборов данных UCI и для 3 алгоритмов (CART, RF и CoLiBRi) показаны в виде “ящиков с усами” (boxplots) на Рис. 4.11, 4.12 и 4.13. Средние мощности посылок правил для 13 наборов данных UCI и 3 алгоритмов показаны на Рисунке 4.14.

Средние “длины” правил, которыми определялись метки тестовых объектов для каждого набора данных и для 3 алгоритмов (CART, RF и CoLiBRi) указаны в Таблице 4.2, а также изображены графически на Рис. 4.14. Видно, что в среднем правила, получаемые с CoLiBRi длиннее, чем у CART, но короче, чем у случайного леса, что делает алгоритм CoLiBRi более интерпретируемым, чем случайный лес. Заметим, что длину правил CoLiBRi можно еще сильнее понизить, если для посылки каждого правила считать соответствующий минимальный генератор (см. Определение 15).

[8] https://archive.ics.uci.edu/ml/datasets/Breast+Cancer+Wisconsin+(Original)

[9] https://archive.ics.uci.edu/ml/datasets/Lymphography

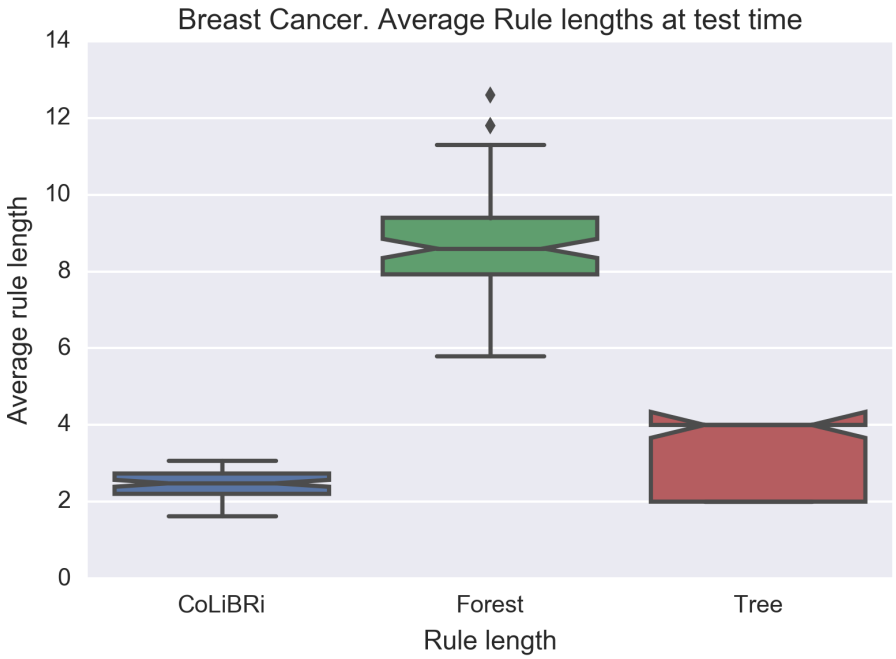


Рисунок 4.11: Средние мощности посылок правил, которыми были классифицированы тестовые объекты набора данных Breast Cancer репозитория UCI, для 3 алгоритмов.

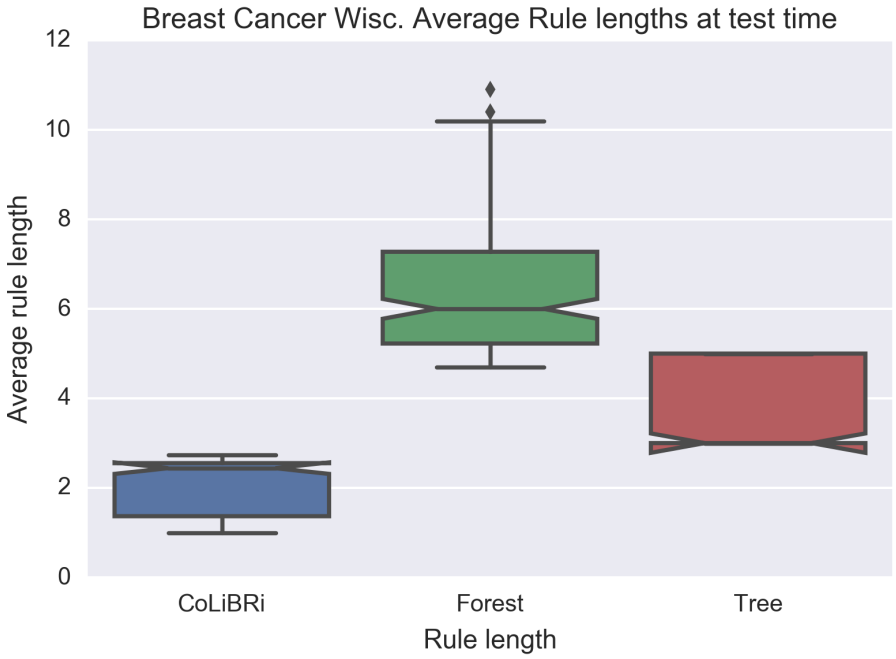


Рисунок 4.12: Средние мощности посылок правил, которыми были классифицированы тестовые объекты набора данных Breast Cancer Wisconsin репозитория UCI, для 3 алгоритмов.

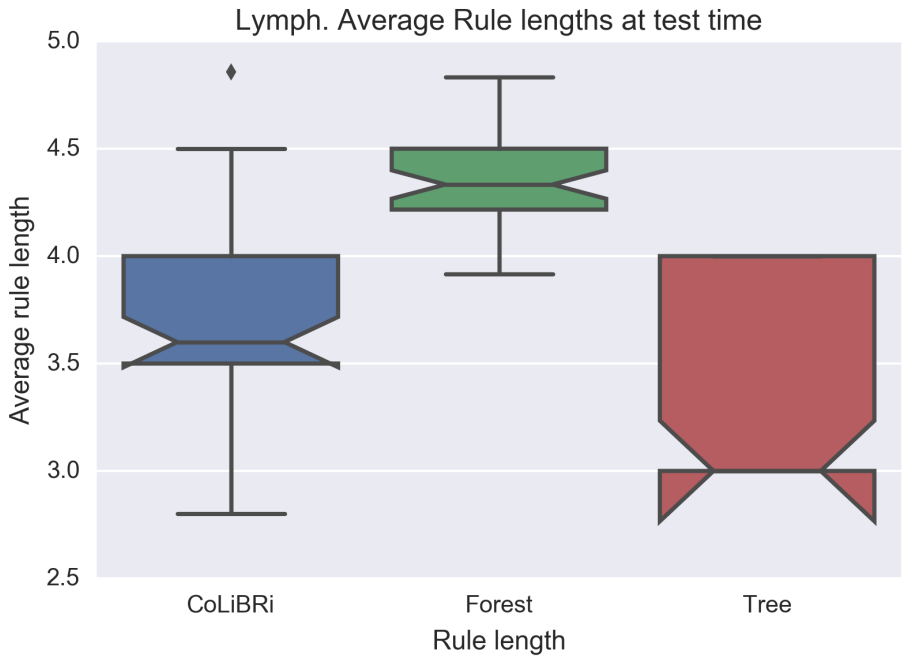


Рисунок 4.13: Средние мощности посылок правил, которыми были классифицированы тестовые объекты набора данных Lymph репозитория UCI, для 3 алгоритмов.

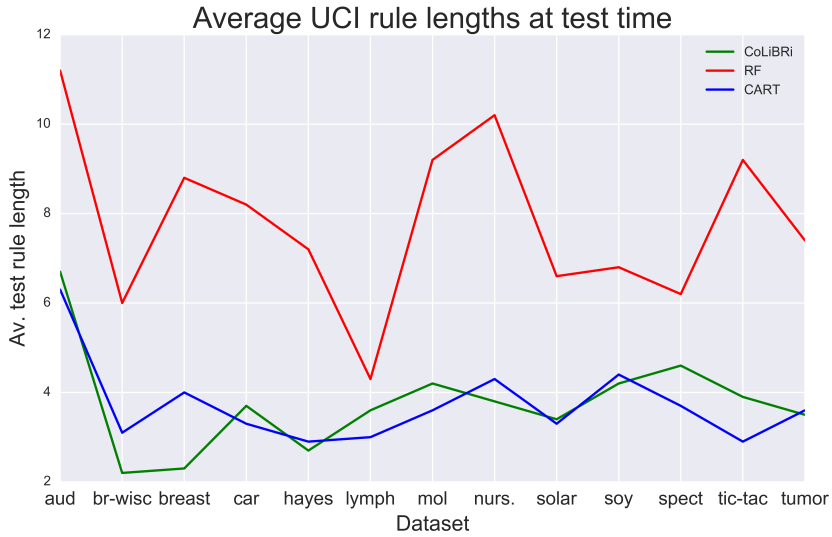


Рисунок 4.14: Средние мощности посылок правил, которыми были классифицированы тестовые объекты, для 3 алгоритмов и 13 наборов данных репозитория UCI (лучше смотреть в цвете).

Алгоритм \ набор данных	aud	br-wisc	breast	car	hayes	lymph	mol	nurs.	solar	soy	spect	tic-tac	tumor
CoLiBRi	6.7	2.2	2.3	3.7	2.7	3.6	4.2	3.8	3.4	4.2	4.6	3.9	3.5
RF	11.2	6.0	8.8	8.2	7.2	4.3	9.2	10.2	6.6	6.8	6.2	9.2	7.4
CART	6.3	3.1	4.0	3.3	2.9	3.0	3.6	4.3	3.3	4.4	3.7	2.9	3.6

Таблица 4.2: Средние мощности посылок правил, которыми были классифицированы тестовые объекты, для 3 алгоритмов и 13 наборов данных репозитория UCI.

4.3.2. Данные с количественными признаками

Версия алгоритма CoLiBRi (“Concept Lattice-Based Rule-learner”) для работы с количественными признаками (Алгоритм 5) была протестирована на 14 наборах данных UCI[10].

Количественные признаки дискретизировались с теми же порогами, как у CART (метод дискретизации один и тот же). Параметр min_supp для CoLiBRi брался равным min_sample_leaf CART, использовались $n\_rules = 5$ правил для классификации каждого тестового объекта.

Результаты представлены в Таблице 4.3. Значения лучших параметров и времена работы алгоритмов указаны в Приложении А в Таблице 4.14. Результаты свидетельствуют, что предлагаемый алгоритм демонстрирует лучшие результаты, чем CART, на большинстве наборов данных. Примечательно, что результаты kNN часто лучше, когда размерность набора данных не очень высока (“проклятие размерности”).

Данные	CART acc	kNN acc	CoLiBRi acc	CART F1	kNN F1	CoLiBRi F1
colic	0.647	0.644	**0.653**	0.619	0.569	**0.664**
heart-h	0.782	**0.837**	0.791	0.664	**0.831**	0.787
heart-statlog	0.804	**0.848**	0.816	0.761	**0.846**	0.823
hepatitis	**0.794**	**0.794**	0.782	**0.867**	0.702	0.755
hypothyroid	**0.975**	0.923	0.968	**0.974**	0.886	0.948
ionosphere	0.9	0.783	**0.924**	0.923	0.757	**0.938**
kr-vs-kp	**0.98**	0.761	**0.98**	0.981	0.756	**0.984**
segment	0.938	0.872	**0.947**	**0.938**	0.869	0.928
sonar	0.697	0.663	**0.73**	0.665	0.658	**0.718**
soybean	0.877	**0.89**	0.88	0.868	0.883	**0.879**
vehicle	**0.708**	0.677	0.692	**0.708**	0.667	0.62
vote	0.956	0.929	**0.968**	0.946	0.929	**0.955**
vowel	0.436	0.405	**0.442**	**0.428**	0.387	0.406
waveform-5000	0.761	**0.834**	0.783	0.761	0.583	**0.774**

Таблица 4.3: Значения доли правильных ответов и F1-метрики для 14 наборов данных репозитория UCI. “DT acc” и “DT F1” обозначают средние по 5 запускам доли правильных ответов и F1-метрики алгоритма CART при 5-кратной кросс-валидации , …, “CoLiBRi F1” обозначают среднее по 5 запускам значение F1-метрики алгоритма CoLiBRi. Жирным выделены лучшие значения метрик.

Также были проведены эксперименты еще с 8 наборами данных репозитория UCI для сравнения с результатами, опубликованными в [VMZ06]. Краткую статистику по этим 8 наборам данных можно найти в Приложении А в Таблице 4.13.

Качество классификации (доля ошибок) предлагаемого алгоритма CoLiBRi (версия с количественными признаками и дискретизацией, как в CART, Алгоритм 5) сравнивается в процессе 10-

[10] http://repository.seasr.org/Datasets/UCI/csv/

кратной кросс-валидации с результатами алгоритмов C4.5 [Qui93], LazyDT [FKY96], EAC (Eager Associative Classifier) и LAC (Lazy Associative Classifier), опубликованными в [VMZ06]. Многоклассовая классификация для 3 наборов данных (iris, wine и zoo) сводилась к бинарной классификации методом “Один против Всех” (подход OneVsAll).

Результаты представлены в Таблице 4.4 и говорят о том, что предлагаемый алгоритм на указанных 8 наборах данных репозитория UCI выдает качество классификации, сравнимое с подходом классификации по запросу с помощью ассоциативных правил (LAC) и лучше, чем у C4.5, LazyDT и EAC.

Набор данных	**C4.5**	**LazyDT**	**EAC**	**LAC**	**CoLiBRi**
heart	18.9	17.7	18.1	16.9	**16.5**
hepatitis	22.6	20.3	17.9	**17.1**	17.2
horse	16.3	17.2	15.4	14.5	**14.2**
ionosphere	8.0	8.0	**7.6**	7.8	7.7
iris	5.3	5.3	4.9	**3.2**	4.5
pima	27.5	25.9	27.5	22.0	**21.6**
wine	7.9	7.9	7.2	**3.4**	4.1
zoo	7.8	7.8	6.6	**6.5**	7.1
В среднем	13.92	13.53	12.9	**11.37**	11.41

Таблица 4.4: Процент ошибок на 8 наборах данных UCI для 5 алгоритмов.

4.4. Прогнозирование оттока клиентов телеком-оператора

Описанный алгоритм CoLiBRi для количественных и категориальных признаков был протестирован в задаче прогнозирования оттока клиентов на данных российского телеком-оператора. Данные выглядят следующим образом (Рис. 4.15).

	Начисления1	**Начисления2**	**Начисления3**	**Начисления4**	**Сервис**	**План**	**Отток**
15	56.59	27.01	7.23	1.46	4	0	1
11	31.91	13.89	8.82	2.46	0	0	0
59	38.98	15.08	8.52	3.24	1	0	0
302	28.97	29.79	13.37	3.02	0	0	1
1245	23.82	19.67	8.46	3.02	2	0	0

Рисунок 4.15: Первые 5 строк обучающей выборки в задаче прогнозирования оттока клиентов телеком-оператора.

Для 2482 клиентов известны 6 признаков: начисления по договору за 4 периода (“Начисления1” – “Начисления4”) и индикаторы подключения тарифного плана (“План”) и объем услуг, оказанных при подключении некоторого сервиса (“Сервис”). Также для этих клиентов известно, определены ли они компанией как ушедшие клиенты или нет (определение компанией оттока

неизвестно). Задача – прогнозировать отток клиентов с высокой долей правильных ответов, которая оценивается с помощью кросс-валидации.

На Рис. 4.16 показано дерево решений, обученное на представленной выборке. Гиперпараметры дерева настроены на 5-кратной стратифицированной кросс-валидации. Среднее значение площади под ROC-кривой (ROC AUC) на кросс-валидации для данной модели – 0.866. На аналогичной кросс-валидации проверялись также случайный лес из 10 деревьев (среднее значение ROC AUC – 0.9) и предлагаемый алгоритм CoLiBRi (среднее значение ROC AUC – 0.875).

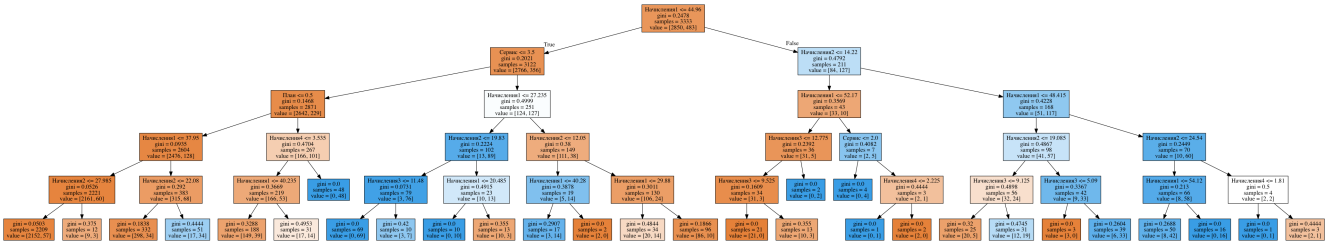


Рисунок 4.16: Дерево решений, построенное для прогнозирования оттока клиентов телеком-оператора.

На Рис. 4.17 показан путь в построенном дереве решений, “объясняющий” классификацию одного из примеров обучающей выборки. На Рис. 4.17 представлены два примера из обучающей выборки и пути в дереве решений, которыми определялись прогнозы для этих двух примеров.

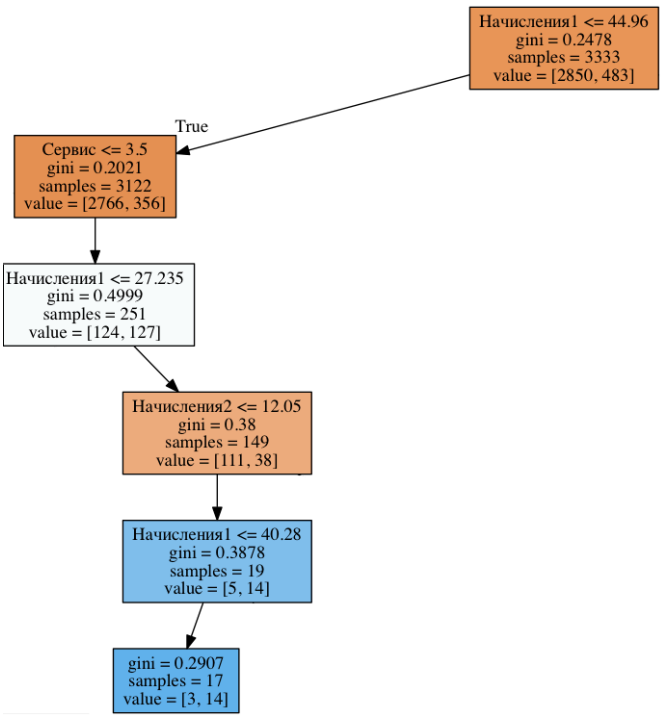


Рисунок 4.17: Путь в дереве, “объясняющий” классификацию конкретного примера в задаче прогнозирования оттока клиентов телеком-оператора.

На Рис. 4.19 изображены статистики распределения длин посылок правил, которыми были классифицированы все примеры обучающей выборке в случае дерева решений, CoLiBRi и случайного леса. Видно, что CoLiBRi строит в среднем более короткие классифицирующие правила, которые, соответственно, проще интерпретировать.

	Начисления1	Начисления2	Начисления3	Начисления4	Сервис	План
17	32.42	18.55	5.83	2.19	3	0
1548	46.09	14.62	7.61	2.78	1	0

Рисунок 4.18: Два примера из обучающей выборки в задаче прогнозирования оттока клиентов телеком-оператора.
Пример 17: {Начисления1 $\leq$ 27.9, План = 0, Сервис $\leq$ 3.5 } $\xrightarrow[(0.026)]{}$ +
Пример 1548: {Начисления1 $\geq$ 44.9, Начисления2 $\geq$ 14.2, Начисления1 $\leq$ 48.9, Начисления2 $\leq$ 19.1, Начисления3 $\leq$ 9.1} $\xrightarrow[(0.2)]{}$ +

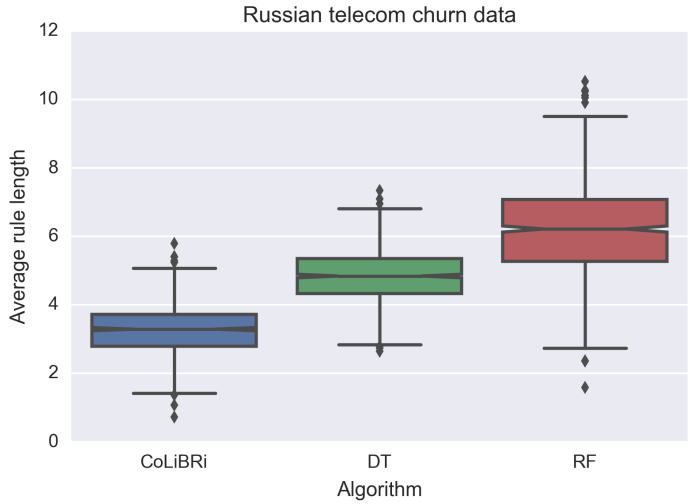


Рисунок 4.19: Средние длины правил, которыми определялся класс тестовых примеров в задаче прогнозирования оттока клиентов телеком-оператора.

4.5. Эксперименты с задачами классификации последовательностей и графов

4.5.1. Эксперименты с задачами классификации последовательностей

Версия алгоритма CoLiBRi (“Concept Lattice-Based Rule-learner”) для работы с описаниями в виде последовательностей (Алгоритм 7) была протестирована в серии экспериментов с данными в виде последовательностей.

Рассматривались 7 наборов данных, краткая статистика по которым приведена в Таблице 4.5. Подробно эти задачи описаны в [MF10].

– ASL-BU (aslbu) – транскрипции американских видеозаписей языка жестов глухонемых людей. Помечены последовательности движений, таких как “движение головы медленное”, “плечи вперед” и т.п. Последовательности принадлежат одному из 7 классов типа “вопрос типа Да-Нет”, “риторический вопрос” и т.п;

– ASL-GT (aslgt) – те же данные, но целевой класс – расшифровка 40 простых слов, а признаки количественные;

– Auslan – транскрипции австралийских видеозаписей языка жестов глухонемых людей. Целевой класс – одно из 10 простых слов;

– Blocks – видео взаимодействия руки человека с предметом. Элементы последовательности – действия человека (какие участки предмета человек трогает), целевой класс – тип взаимодействия (поднятия, опускание) и сценарии (сборка “пирамидки”);

– Context – данные о том, как человек пользуется мобильным телефоном, элементы последовательности – признаки взаимодействия с телефоном (созданы вручную), целевой класс – сценарий использования (встреча, улица и т.д.);

– Pioneer – данные репозитория UCI, целевой класс – 3 вида взаимодействия робота с предметом (захват, толчок, поворот);

– Skating – элементы последовательности – предобработанные признаки временного ряда мускульной активности и позиции ног профессиональных лыжников во время тестирования тренажера. Целевой класс – соответствующий лыжник и его скоростной режим.

Далее в Таблице 4.6 приведены средние доли правильных ответов при 10-кратной кросс-валидации для 7 алгоритмов и 7 задач классификации. Описания алгоритмов даны на следующих ресурсах[11] и в статьях [ZCG13] (CBS, BayesFM, SCII Match и SCII CBA) и [Egh+15] (MiSeRe).

Результаты позволяют утверждать, что качество классификации метода SequentialCoLiBRi достаточно высокое в сравнении с прочими алгоритмами классификации последовательностей.

	Число послед-тей	**Число элементов**	**Число классов**
aslbu	441	140	7
aslgt	3493	47	40
auslan	200	12	10
blocks	210	8	8
context	240	54	5
pioneer	160	92	3
skater	530	41	6

Таблица 4.5: Краткая статистика 7 наборов данных по последовательностям.

4.5.2. Предсказание токсичности химических веществ

Версия алгоритма CoLiBRi (“Concept Lattice-Based Rule-learner”) для работы с описаниями в виде графов (Алгоритм 7) была протестирована в эксперименте Predictive Toxicology Challenge[12]: Описание набора данных и эксперимента [HK03]:

[11] http://misere.co.nf/,http://adrem.ua.ac.be/scii
[12] http://www.predictive-toxicology.org/ptc/

	CBS	BayesFM	SCII Match	SCII CBS	MiSeRe	Binary CoLiBRi	Sequential CoLiBRi
aslbu	0.43	**0.7**	0.57	0.56	**0.7**	0.48	0.62
aslgt	0.23	0.738	0.04	0.04	**0.77**	0.32	0.71
auslan	0.32	0.34	0.04	0.03	0.34	0.33	**0.35**
blocks	**1**	**1**	0.08	0.08	**1**	0.99	**1**
context	0.58	0.896	0.32	0.33	**0.9**	0.74	**0.9**
pioneer	0.79	0.96	0.97	0.95	**1**	0.77	0.97
skater	0.55	**0.87**	0.18	0.18	0.86	0.69	**0.87**

Таблица 4.6: Доля верных ответов при 10-кратной кросс-валидации в задачах классификации последовательностей.

– Обучающая выборка состоит из 417 упрощенных молекулярных структур химических веществ с указанием того, является ли вещество токсичным для представителей одной из четырех групп: $\{\text{mice, rats}\} \times \{\text{male, female}\}$.

– Четыре отдельных набора данных для крыс-самцов (MR, 274 вещества, 117 – токсичны, 157 – нетоксичны), крыс-самок (FR, 281, 86, 195) мышей-самцов (MM, 266, 94, 172) и мышей-самок (FM, 279, 108, 171).

Сравнивались 4 алгоритма:

– CoLiBRi (“Concept Lattice-based Rule-learner”) - предлагаемый Алгоритм 7;

– “GLAC” (“Graphlet-based Lazy Associative Graph Classification”) – алгоритм ленивой классификации для графов на основе их подграфов (графлетов) [KK15];

– SVM с графлет-ядром;

– Метод k ближайших соседей с расстоянием Хэмминга по включению графлетов.

Результаты 5-кратной кросс-валидации для подвыборки самцов крыс Predictive Toxicology Challenge[13] представлены в Таблице 4.7.

Результаты алгоритма CoLiBRi на тестовых наборах данных Predictive Toxicology Challenge представлены в Таблице 4.8, а также на Рис. 4.20 и 4.21, где показатели доли верных и ошибочных классификаций (TPR и FPR соответственно) алгоритма CoLiBRi сравниваются с аналогичными показателями алгоритмов-участников соревнования.

4.5.3. Результаты экспериментов с классификацией данных, представленных графами

Также проводились вычислительные эксперименты еще с 4 алгоритмами и 5 наборами данных, представленных графами.

[13] http://www.predictive-toxicology.org/ptc/

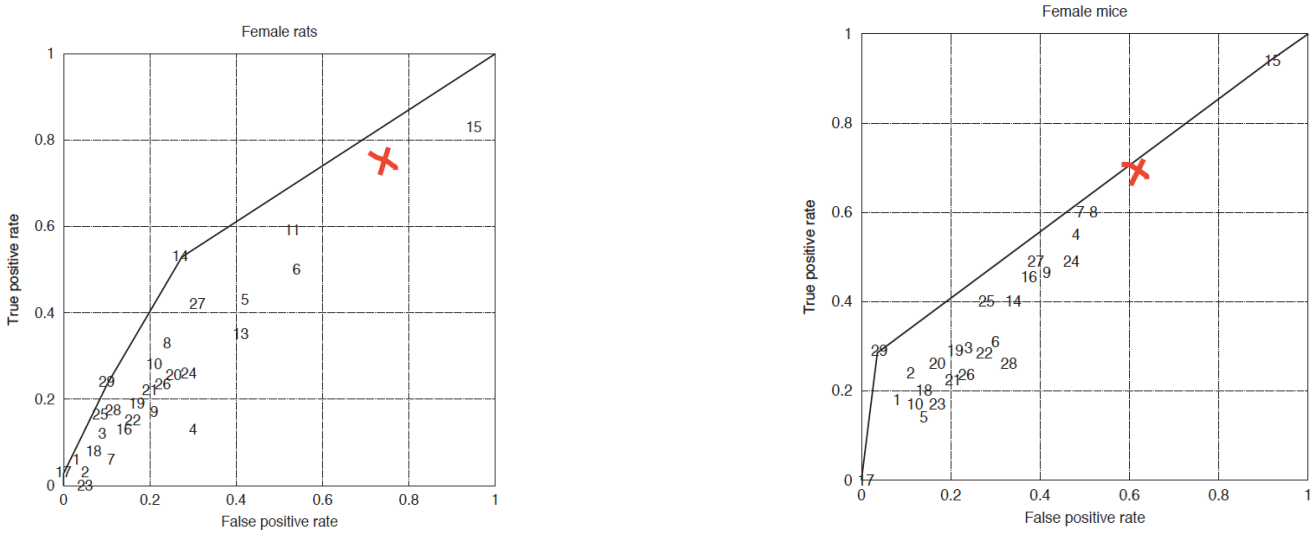


Рисунок 4.20: Сравнение алгоритма CoLiBRi с алгоритмами участников Predictive Toxicology Challenge на тестовых выборках PTC-FM (female mice) и PTC-FR (female rats)

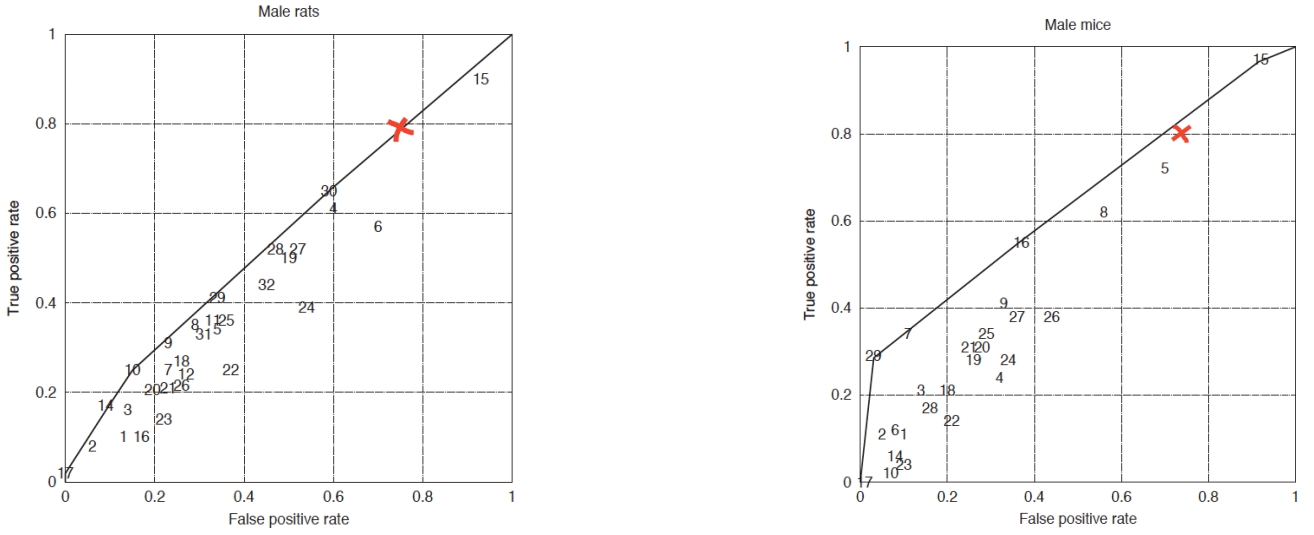


Рисунок 4.21: Сравнение алгоритма CoLiBRi с алгоритмами участников Predictive Toxicology Challenge на тестовых выборках PTC-MM (male mice) и PTC-MR (male rats)

	K	Accuracy	Precision	Recall	F1-score	Time (sec.)
GLAC	2	0.36	0.32	0.33	0.32	1.8
	3	0.68	**0.83**	0.68	**0.75**	3.2
	4	0.59	0.57	0.62	0.59	4
	5	0.55	0.7	0.62	0.66	6.8
SVM	2	0.45	0.15	0.33	0.21	1.5
	3	0.52	0.35	0.35	0.35	2.2
	4	0.41	0.27	0.28	0.28	2.6
	5	0.36	0.24	0.25	0.24	3.2
kNN	2	0.45	0.15	0.33	0.21	0.6
	3	0.34	0.21	0.23	0.22	0.8
	4	0.48	0.31	0.32	0.31	1.2
	5	0.45	0.30	0.31	0.30	2.3
CoLiBRi	2	0.42	0.4	0.36	0.38	7.4
	3	**0.71**	0.78	**0.7**	0.74	11
	4	0.63	0.52	0.68	0.6	25.2
	5	0.57	0.72	0.66	0.69	62.2

Таблица 4.7: Результаты кросс-валидации для группы самцов мышей. “GLAC” означает “Graphlet-based lazy associative classification”, “SVM” – машина опорных векторов с графлет-ядром, “kNN” – метод ближайших соседей с расстоянием Хэмминга.

Набор данных	Accuracy	Precision	Recall	F-score
PTC-FM	0.7	0.71	0.69	0.7
PTC-FR	0.72	0.77	0.75	0.76
PTC-MM	0.73	0.76	0.81	0.78
PTC-MR	0.71	0.74	0.8	0.77

Таблица 4.8: Качество классификации алгоритма CoLiBRi (версия 7) на 4 тестовых выборках набора данных Predictive Toxicology Challenge.

Наборы данных IMDB, MUTAG, NCI, NCI109 и PROTEINS[14] известны тем, что в задачах классификации с этими данными часто проверяются алгоритмы графовой классификации [Ker+16].

Краткое описание задач:

– IMDB – граф отношения совместной съемки в фильме для актеров; фильмы поделены на 2 жанра: романтические и боевики [YV15];

– MUTAG – 188 структур химических веществ, поделенных на 2 класса по мутагенному эффекту, производимому на бактерии [Deb+91];

– NCI, NCI109 – два сбалансированных подмножества наборов данных химических соединений, у которых измерена, соответственно, активность борьбы против немелкоклеточного рака легких и раковых клеток яичников [WWK08];

– PROTEINS – предсказание функциональных классов принадлежности ферментов [Bor+05].

Для всех графов с помощью расширения алгоритма Gaston [NK05] LibGastonForSofia[15] были построены бинарные признаки по включению подграфов до 6 вершин, что заняло от 6 до 42 минут в зависимости от набора данных. Проверялись 4 алгоритма:

[14] https://ls11-www.cs.uni-dortmund.de/staff/morris/graphkerneldatasets
[15] https://github.com/AlekseyBuzmakov/LibGastonForSofia

– CBA – классификация на основе ассоциативных правил (реализация LUCS-KDD[16]);

– DT – дерево решений (sklearn);

– SVM graphlet – линейный метод опорных векторов (sklearn);

– CoLiBRi – предлагаемый алгоритм.

Данные были поделены в пропорции 7/3 на обучающую и проверочную выборку. В Таблице 4.9 указаны доли правильных ответов 4 алгоритмов проверенных на 5 графовых наборах данных. Можно заметить, что в целом SVM справляется лучше остальных алгоритмов, зато остальные алгоритмы –- интерпретируемые, на выходе можно получить набор классифицирующих правил для каждого тестового примера.

	CBA	DT	SVM graphlet	CoLiBRi
IMDB	60.1	55.6	**62.1**	59.3
MUTAG	72.1	68.4	**77.4**	74.6
NCI1	55.1	52.1	**59.6**	58.3
NCI109	56.6	52.8	**59.7**	58.8
PROTEINS	60.5	60.2	66.3	**68.9**

Таблица 4.9: Доли правильных ответов 4 алгоритмов на 5 графовых наборах данных.

В Таблице 4.10 представлены средние мощности посылок правил, участвовавших в классификации тестовых примеров в задачах классификации, результаты которых представлены в Таблице 4.9. Можно сделать вывод, что в данных задачах алгоритм CoLiBRi демонстрирует качество классификации выше, чем CBA и DT, при этом сохраняется интерпретируемость алгоритма (в отличие от случая применения SVM) – мощности посылок правил, участвовавших в классификации тестовых примеров в случае CoLiBRi примерно такие же, как и в случае CBA и DT.

	CBA	DT	CoLiBRi
IMDB	5.1	5.2	5.5
MUTAG	6.8	7.8	7.2
NCI1	8.3	10.5	12.7
NCI109	8.5	11.3	10.5
PROTEINS	7.6	12.2	8.6

Таблица 4.10: Средние мощности посылок правил, участвовавших в классификации тестовых примеров.

В Таблице 4.11 указано время работы рассмотренных алгоритмов классификации в секундах. Отметим, что тут уже был проделан самый затратный этап, построение бинарных признаков по включению подграфов до 6 вершин, и поэтому время работы алгоритмов на таких бинарных признаках невелико, при этом методы, основанные на классифицирующих ассоциативных правилах (CBA и CoLiBRi) работают намного дольше, чем деревья решений и SVM.

[16] http://cgi.csc.liv.ac.uk/~frans/KDD/Software/CBA/cba.html

	CBA	DT	SVM graphlet	CoLiBRi
IMDB	7.2	0.3	2.5	26.7
MUTAG	1.3	0.05	0.8	6.8
NCI1	28.5	1.2	5.2	153.6
NCI109	35.6	1.6	4.6	183.2
PROTEINS	13.5	0.7	2.9	54.3

Таблица 4.11: Время работы алгоритмов в задачах графовой классификации (сек).

4.6. Заключение

Результаты вычислительных экспериментов по решению задачи классификации реальных наборов данных свидетельствуют о том, что предлагаемые алгоритмы имеют лучшее качество классификации, чем у деревьев решений C4.5 и CART и лучшую интерпретируемость (средний размер посылок правил), чем у случайного леса.

В задачах классификации данных, представленными последовательностями и графами, показано, что с помощью предлагаемых алгоритмов можно добиваться высокого качества классификации с помощью коротких классифицирующих правил.

Заключение

В данной работе предложен универсальный подход к классификации данных со сложной структурой. Также предложены модификации для данных с признаками разной природы.

Предложенные вычислительные методы легли в основу программного комплекса, позволяющего решать задачи классификации для данных со сложной структурой. С помощью этого комплекса предлагаемые алгоритмы были протестированы на большом числе данных из разных областей, а также для данных, представленных графами в задачах прогнозирования свойств химических веществ, и в нескольких задачах классификации данных, представленных последовательностями.

Таким образом, основные результаты всей работы могут быть описаны следующим образом:

1. Предложен универсальный подход к классификации данных со сложной структурой на основе решеток замкнутых описаний;
2. В рамках этого подхода предложены алгоритмы для классификации данных, представленных последовательностями и графами, а также числовыми и интервальными признаками;
3. Алгоритмы апробированы в задачах классификации последовательностей и графов и показали высокие значения доли правильных ответов. При этом классификация проводилась с помощью коротких классифицирующих правил;
4. На данных Predictive Toxicology Challenge показаны метрики качества выше, чем у SVM с графлет-ядром, и сравнимые с лучшими из результатов участников соревнования;
5. В вычислительных экспериментах с данными репозитория UCI получены значения метрик качества классификации на кросс-валидации, статистически значимо более высокие, чем у алгоритмов построения деревьев решений;
6. При этом показано, что интерпретируемость полученных правил, понимаемая как средняя мощность посылок правил, которыми определялись метки тестовых объектов, у предлагаемого алгоритма лучше, чем у случайного леса;
7. Методы классификации, основанные на правилах, в том числе деревья решений, представлены с помощью проекций интервальных узорных структур;

8. Предложены и исследованы дискретизирующие проекции для интервальных узорных структур. На их основе предложен способ выбора правил на основе множеств формальных понятий, гарантирующий нахождение правил не хуже, чем построенные деревом решений, по выбранному критерию информативности;

9. Разработан программный комплекс, позволяющий анализировать сложно структурированные данные и решать для них задачи классификации с помощью интерпретируемых наборов правил, подходящих для дальнейшего экспертного анализа.

Список литературы

[Бир84] Биркгоф, Г. *Теория решеток*. М.: Наука, 1984.

[Буз15] Бузмаков, А. В. *Моделирование процессов с состояниями сложной структуры на основе решёток замкнутых описаний*. Диссертация на соискание ученой степени кандидата технических наук: защищена 26.10.15. М., 2015.

[Вор00] Воронцов, К. В. «Оптимизационные методы линейной и монотонной коррекции в алгебраическом подходе к проблеме распознавания». В: *Журнал вычислительной математики и математической физики* 40.1 (2000), с. 166—176.

[Вью13] Вьюгин, В.В. *Математические основы машинного обучения и прогнозирования*. Москва: МЦНМО, 2013.

[Дья05] Дьяконов, А. Г. «Универсальные и локальные ограничения в проблеме коррекции эвристических алгоритмов». В: *Журнал вычислительной математики и математической физики* 45.6 (2005), с. 1134—1145.

[Дюк02] Дюкова, Е. В. «Поиск информативных фрагментов описаний объектов в дискретных процедурах распознавания». В: *Журнал вычислительной математики и математической физики* 42.5 (2002), с. 741—753.

[Жур66] Журавлев, Ю. И. «О математических принципах классификации предметов и явлений». В: *Дискретный анализ* 7 (1966), с. 3—15.

[Жур71] Журавлев, Ю. И. «Алгоритмы распознавания, основанные на вычислении оценок». В: *Кибернетика* 3 (1971), с. 1—11.

[Жур02] Журавлев, Ю. И. «Об алгоритмах распознавания с представительными наборами (о логических алгоритмах)». В: *Журнал вычислительной математики и математической физики* 42.9 (2002), с. 1425—1435.

[Каш15] Кашницкий Ю. С., Игнатов Д. И. «Ансамблевый метод машинного обучения, основанный на рекомендации классификаторов». В: *Интеллектуальные системы. Теория и приложения* 19.4 (2015), с. 37—55.

[Каш16] Кашницкий, Ю. С. «Методы поиска точных и интерпретируемых классифицирующих правил для данных со сложной структурой». В: *Труды Пятнадцатой национальной конференции по искуственному интеллекту с международным участием, г. Смоленск, 03-07 октября 2016 г.* 2016, с. 184—191.

[Куз91] Кузнецов, С. О. «ДСМ-метод как система аксиоматического обучения». В: *Итоги науки и техники*. Интеллектуальные информационные системы 15 (1991), с. 17—53.

[Куз93] Кузнецов, С. О. «Быстрый алгоритм построения всех пересечений объектов из конечной полурешётки». В: *Научно-Техническая Информация* 1 (1993), с. 17—20.

[Про16] Прокофьев, П. А. *Корректное распознавание по прецедентам: построение логических корректоров общего вида и вычислительные аспекты*. Диссертация на соискание ученой степени кандидата физико-математических наук: защищена 06.10.2016 г. : утв. 07.10.2016 г. М., 2016.

[Руд87] Рудаков, К. В. «Универсальные и локальные ограничения в проблеме коррекции эвристических алгоритмов». В: *Кибернетика* 2 (1987). Под ред. Рудаков, К. В., с. 30—35.

[Сам06] Самохин, М. В. *Машинное обучение на узорных структурах*. Диссертация на соискание ученой степени кандидата технических наук: защищена 22.06.06 : утв. 15.04.06. М., 2006.

[Фин83] Финн, В. К. «О машинно-ориентированной формализации правдоподобных рассуждений в стиле Ф. Бэкона-Д.С. Милля». В: *Семиотика и Информатика* 20 (1983), с. 42—63.

[Фин10a] Финн, В. К. «Индуктивные методы Д.С. Милля в системах искусственного интеллекта». В: *Искусственный интеллект и принятие решений* 3 (2010), с. 3—21.

[Фин10b] Финн, В. К. «Об определении эмпирических закономерностей посредством ДСМ-метода автоматического порождения гипотез». В: *Искусственный интеллект и принятие решений* 4 (2010), с. 41—48.

[AS94] Agrawal, R. and Srikant, R. «Fast Algorithms for Mining Association Rules in Large Databases». In: *Proceedings of the 20th International Conference on Very Large Data Bases*. VLDB ’94. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 1994, pp. 487–499.

[And09] Andrews, S. «In-close, a fast algorithm for computing formal concepts». In: (2009).

[Bir40] Birkhoff, G. *Lattice Theory*. American Mathematical Society colloquium publications т. 25,ч. 2. American Mathematical Society, 1940.

[Bli+03] Blinova, V. G. et al. «Toxicology Analysis by Means of the JSM-method». In: *Bioinformatics* 19.10 (2003), pp. 1201–1207.

[BK05] Borgwardt, Karsten M. and Kriegel, Hans-Peter. «Shortest-Path Kernels on Graphs». In: *Proceedings of the Fifth IEEE International Conference on Data Mining (ICDM 2005)*. Washington, DC, USA: IEEE Computer Society, 2005, pp. 74–81.

[Bor+05] Borgwardt, Karsten M. et al. «Protein function prediction via graph kernels». In: *Bioinformatics* 21.1 (2005), p. 47.

[Bre01] Breiman, L. «Random Forests». In: *Machine Learning* 45.1 (Oct. 2001), pp. 5–32.

[Bre+84] Breiman, L. et al. *Classification and Regression Trees*. Statistics/Probability Series. Belmont, California, U.S.A.: Wadsworth Publishing Company, 1984.

[Buz+13] Buzmakov, A. et al. «FCA and pattern structures for mining care trajectories». In: *Proceedings of the International Workshop "What can FCA do for Artificial Intelligence?" (FCA4AI at IJCAI 2013), Beijing, China, August 5, 2013*. Ed. by Kuznetsov, Sergei O., Napoli, Amedeo, and Rudolph, Sebastian. Vol. 1058. CEUR Workshop Proceedings. CEUR-WS.org, 2013, pp. 7–14.

[Buz+16] Buzmakov, A. et al. «On mining complex sequential data by means of FCA and pattern structures». In: *Int. J. General Systems* 45.2 (2016), pp. 135–159.

[CR04] Carpineto, C and Romano, G. *Concept Data Analysis: Theory and Applications*. John Wiley & Sons, 2004.

[CN06] Caruana, R. and Niculescu-Mizil, A. «An Empirical Comparison of Supervised Learning Algorithms». In: *Proceedings of the 23rd International Conference on Machine Learning*. ICML '06. Pittsburgh, Pennsylvania, USA: ACM, 2006, pp. 161–168.

[Coh95] Cohen, W.W. «Fast Effective Rule Induction». In: *In Proceedings of the Twelfth International Conference on Machine Learning*. Morgan Kaufmann, 1995, pp. 115–123.

[CMR08] Cortes, C., Mohri, M., and Rostamizadeh, A. «Learning sequence kernels». In: *2008 IEEE Workshop on Machine Learning for Signal Processing*. Oct. 2008, pp. 2–8.

[CV95] Cortes, C. and Vapnik, V. «Support-Vector Networks». In: *Mach. Learn.* 20.3 (Sept. 1995), pp. 273–297.

[Deb+91] Debnath, Asim Kumar et al. «Structure-activity relationship of mutagenic aromatic and heteroaromatic nitro compounds. Correlation with molecular orbital energies and hydrophobicity». In: *Journal of Medicinal Chemistry* 34.2 (1991), pp. 786–797.

[DJS12] Doerfel, S., Jäschke, R., and Stumme, G. «Formal Concept Analysis: 10th International Conference, ICFCA 2012, Leuven, Belgium, May 7-10, 2012. Proceedings». In: ed. by Domenach, Florent, Ignatov, Dmitry I., and Poelmans, Jonas. Berlin, Heidelberg: Springer Berlin Heidelberg, 2012. Chap. Publication Analysis of the Formal Concept Analysis Community, pp. 77–95.

[Egh+15] Egho, Elias et al. «A Parameter-Free Approach for Mining Robust Sequential Classification Rules». In: *2015 IEEE International Conference on Data Mining, ICDM 2015, Atlantic City, NJ, USA, November 14-17, 2015*. Ed. by Aggarwal, Charu et al. IEEE Computer Society, 2015, pp. 745–750.

[FKY96] Friedman, J. H., Kohavi, R., and Yun, Y. «Lazy Decision Trees.» In: *AAAI/IAAI, Vol. 1*. Ed. by Clancey, William J. and Weld, Daniel S. AAAI Press / The MIT Press, 1996, pp. 717–724.

[Für97] Fürnkranz, J. «Pruning Algorithms for Rule Learning.» In: *Machine Learning* 27.2 (1997), pp. 139–172.

[Für99] Fürnkranz, J. «Separate-and-conquer rule learning». In: *Artificial Intelligence Review* 13 (1999), pp. 3–54.

[GK01] Ganter, B. and Kuznetsov, S.O. «Pattern Structures and Their Projections». In: *Conceptual Structures: Broadening the Base*. Ed. by Delugach, Harry and Stumme, Gerd. Vol. 2120. Lecture Notes in Computer Science. Berlin/Heidelberg: Springer, 2001, pp. 129–142.

[GW97] Ganter, B and Wille, R. *Formal Concept Analysis: Mathematical Foundations*. 1st. Secaucus, NJ, USA: Springer-Verlag New York, Inc., 1997.

[Gan10] Ganter, Bernhard. «Two Basic Algorithms in Concept Analysis». In: *Formal Concept Analysis: 8th International Conference, ICFCA 2010, Agadir, Morocco, March 15-18, 2010. Proceedings*. Ed. by Kwuida, Léonard and Sertkaya, Barış. Berlin, Heidelberg: Springer Berlin Heidelberg, 2010, pp. 312–340.

[GH06] Geng, L. and Hamilton, H. J. «Interestingness Measures for Data Mining: A Survey». In: *ACM Comput. Surv.* 38.3 (Sept. 2006).

[HPY00] Han, J., Pei, J., and Yin, Y. «Mining Frequent Patterns Without Candidate Generation». In: *SIGMOD Rec.* 29.2 (May 2000), pp. 1–12.

[HK03] Helma, C. and Kramer, S. «A Survey of the Predictive Toxicology Challenge 2000-2001.» In: *Bioinformatics* 19.10 (2003), pp. 1179–1182.

[Hel+04] Helma, C. et al. «Data mining and machine learning techniques for the identification of mutagenicity inducing substructures and structure activity relationships of noncongeneric compounds». In: *Journal of Chemical Information and Computer Sciences* 44.4 (2004), pp. 1402–1411.

[Her02] Hereth, J. «Conceptual Structures: Integration and Interfaces: 10th International Conference on Conceptual Structures, ICCS 2002 Borovets, Bulgaria, July 15–19, 2002 Proceedings». In: ed. by Priss, Uta, Corbett, Dan, and Angelova, Galia. Berlin, Heidelberg: Springer Berlin Heidelberg, 2002. Chap. Relational Scaling and Databases, pp. 62–76.

[HSS08] Hofmann, T., Schölkopf, B., and Smola, A.J. «Kernel methods in machine learning». In: *Annals of Statistics* 36.3 (2008), pp. 1171–1220.

[Hol93a] Holte, R. C. «Very Simple Classification Rules Perform Well on Most Commonly Used Datasets». In: *Machine Learning* 11.1 (1993), pp. 63–90.

[Hol93b] Holte, Robert C. «Very Simple Classification Rules Perform Well on Most Commonly Used Datasets». In: *Machine Learning* 11.1 (Apr. 1993), pp. 63–90.

[HGW04] Horváth, T., Gärtner, T., and Wrobel, S. «Cyclic Pattern Kernels for Predictive Graph Mining». In: *Proceedings of the 10t ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD 2004)*. Ed. by Kim, Won et al. Seattle, WA, USA: ACM Press, New York, NY, USA, Aug. 2004, pp. 158–167.

[Ign+15] Ignatov, D.I. et al. «Triadic Formal Concept Analysis and triclustering: searching for optimal patterns». In: *Machine Learning* 101.1-3 (2015), pp. 271–302.

[JM00] Jurafsky, Daniel and Martin, James H. *Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition*. 1st. Upper Saddle River, NJ, USA: Prentice Hall PTR, 2000.

[Kas16] Kashnitsky, Y. S. «Lazy Learning of Succinct Classification Rules for Complex Structure Data». In: *Supplementary Proceedings of the Fifth International Conference on Analysis of Images, Social Networks and Texts (AIST 2016)*. Ed. by al., Ignatov D. I. et. CEUR-WS.org, 2016, pp. 664–673.

[KK16a] Kashnitsky, Y. S. and Kuznetsov, S. O. «Global Optimization in Learning with Important Data: an FCA-Based Approach». In: *Proceedings of the Thirteenth International Conference on Concept Lattices and Their Applications (CLA 2016)*. Ed. by Huchard M., Kuznetsov S. O. Vol. 1624. CEUR-WS.org, 2016, pp. 189–202.

[KK16b] Kashnitsky, Y. and Kuznetsov, S. O. «Interval Pattern Concept Lattice as a Classifier Ensemble». In: *Proceedings of the 5th International Workshop "What can FCA do for Artificial Intelligence"? co-located with the European Conference on Artificial Intelligence, FCA4AI@ECAI 2016, The Hague, the Netherlands, August 30, 2016.* Ed. by Kuznetsov, Sergei O., Napoli, Amedeo, and Rudolph, Sebastian. Vol. 1703. CEUR Workshop Proceedings. CEUR-WS.org, 2016, pp. 105–112.

[KK15] Kashnitsky, Y. and Kuznetsov, S.O. «Lazy associative graph classification». In: vol. 1430. 2015, pp. 63–74.

[Kay+11] Kaytoue, M. et al. «Mining gene expression data with pattern structures in formal concept analysis». en. In: *Information Sciences* 181.10 (May 2011), pp. 1989–2001.

[Ker+16] Kersting, Kristian et al. *Benchmark Data Sets for Graph Kernels*. 2016.

[Kou+09] Kourie, D.G. et al. «An incremental algorithm to construct a lattice of set intersections». In: *Science of Computer Programming* 74.3 (2009), pp. 128–142.

[Kuz96] Kuznetsov, S. O. «Mathematical aspects of concept analysis». In: *Journal of Mathematical Sciences* 80.2 (1996), pp. 1654–1698.

[Kuz99] Kuznetsov, S. O. «Learning of Simple Conceptual Graphs from Positive and Negative Examples.» In: *PKDD*. Ed. by Zytkow, Jan M. and Rauch, Jan. Vol. 1704. Lecture Notes in Computer Science. Springer, 1999, pp. 384–391.

[KS05] Kuznetsov, S. O. and Samokhin, M. V. «Learning Closed Sets of Labeled Graphs for Chemical Applications». English. In: *Inductive Logic Programming*. Ed. by Kramer, Stefan and Pfahringer, Bernhard. Vol. 3625. Lecture Notes in Computer Science. Springer Berlin Heidelberg, 2005, pp. 190–208.

[Kuz13] Kuznetsov, Sergei O. «Scalable Knowledge Discovery in Complex Data with Pattern Structures.» In: *PReMI*. Ed. by Maji, Pradipta et al. Vol. 8251. Lecture Notes in Computer Science. Springer, 2013, pp. 30–39.

[Kuz04] Kuznetsov, S.O. «Machine Learning and Formal Concept Analysis». In: *Concept Lattices, Second International Conference on Formal Concept Analysis, ICFCA 2004, Sydney, Australia, February 23-26, 2004, Proceedings*. Ed. by Eklund, Peter W. Vol. 2961. Lecture Notes in Computer Science. Springer, 2004, pp. 287–312.

[Kuz09] Kuznetsov, S.O. «Pattern Structures for Analyzing Complex Data». In: *Rough Sets, Fuzzy Sets, Data Mining and Granular Computing, 12th International Conference, RSFDGrC 2009, Delhi, India, December 15-18, 2009. Proceedings*. Ed. by Sakai, Hiroshi et al. Vol. 5908. Lecture Notes in Computer Science. Springer, 2009, pp. 33–44.

[KO02] Kuznetsov, S.O and Obiedkov, S.A. «Comparing performance of algorithms for generating concept lattices». In: *J. Exp. Theor. Artif. Intell.* 14.2-3 (2002), pp. 189–216.

[LHM98] Liu, B., Hsu, W., and Ma, Y. «Integrating classification and association rule mining». In: *Proceedings of the 4th international conference on Knowledge Discovery and Data mining (KDD'98)*. AAAI Press, Aug. 1998, pp. 80–86.

[Lod+02] Lodhi, H. et al. «Text Classification Using String Kernels». In: *Journal of Machine Learning Research* 2 (Mar. 2002), pp. 419–444.

[MK17] Masyutin, A. A. and Kashnitsky, Y. S. «Query-Based Versus Tree-Based Classification: Application to Banking Data». In: *Foundations of Intelligent Systems*. Ed. by Kryszkiewicz, Marzena et al. Cham: Springer International Publishing, 2017, pp. 664–673.

[MKK15] Masyutin, A., Kashnitsky, Y., and Kuznetsov, S. «Lazy classification with interval pattern structures: Application to credit scoring». In: vol. 1430. 2015, pp. 43–54.

[Mil43] Mill, J.S. *A System of Logic, Ratiocinative and Inductive: Being a Connected View of the Principles of Evidence and the Methods of Scientific Investigation*. A System of Logic, Ratiocinative and Inductive: Being a Connected View of the Principles of Evidence and the Methods of Scientific Investigation т. 1. John W. Parker, 1843.

[MMF11] Misra, Milind, Martin, Shawn, and Faulon, Jean-Loup. «Graphs: Flexible Representations of Molecular Structures and Biological Networks». In: *Computational Approaches in Cheminformatics and Bioinformatics*. John Wiley & Sons, Inc., 2011, pp. 145–177.

[Mit97] Mitchell, T.M. *Machine Learning*. 1st ed. New York, NY, USA: McGraw-Hill, Inc., 1997.

[MF10] Mörchen, Fabian and Fradkin, Dmitriy. «Robust Mining of Time Intervals with Semi-interval Partial Order Patterns». In: *Proceedings of the SIAM International Conference on Data Mining, SDM 2010, April 29 - May 1, 2010, Columbus, Ohio, USA*. SIAM, 2010, pp. 315–326.

[Mül+01] Müller, K.R. et al. «An introduction to kernel-based learning algorithms». In: *IEEE Transactions on Neural Networks* 12.2 (2001), pp. 181–201.

[Mur97] Murthy, S.K. «Automatic Construction of Decision Trees from Data: A Multi-Disciplinary Survey». In: *Data Mining and Knowledge Discovery* 2 (1997), pp. 345–389.

[Nav14] Navarin, N. «Learning with Kernels on Graphs: DAG-based kernels, data streams and RNA function prediction.» PhD thesis. 2014.

[NK05] Nijssen, Siegfried and Kok, Joost N. «The Gaston Tool for Frequent Subgraph Mining». In: *Electronic Notes in Theoretical Computer Science* 127.1 (2005), pp. 77–87.

[Ore62] Ore, O. *Theory of graphs*. AMS, 1962.

[PH90] Pagallo, Giulia and Haussler, David. «Boolean Feature Discovery in Empirical Learning». English. In: *Machine Learning* 5.1 (1990), pp. 71–99.

[Pas+99] Pasquier, N et al. «Discovering Frequent Closed Itemsets for Association Rules». In: *Proceedings of the 7th International Conference on Database Theory*. ICDT '99. London, UK, UK: Springer-Verlag, 1999, pp. 398–416.

[Ped+11] Pedregosa, F. et al. «Scikit-learn: Machine Learning in Python». In: *Journal of Machine Learning Research* 12 (2011), pp. 2825–2830.

[Poe+13a] Poelmans, J. et al. «Formal concept analysis in knowledge processing: A survey on applications». In: *Expert Syst. Appl.* 40.16 (2013), pp. 6538–6560.

[Poe+13b] Poelmans, J. et al. «Formal Concept Analysis in knowledge processing: A survey on models and techniques». In: *Expert Syst. Appl.* 40.16 (2013), pp. 6601–6623.

[Poe+14] Poelmans, J. et al. «Fuzzy and rough formal concept analysis: a survey». In: *Int. J. General Systems* 43.2 (2014), pp. 105–134.

[PR08] Preisach C. Burkhardt H., Schmidt-Thieme L. and R., Decker. *Data Analysis, Machine Learning and Applications*. 1st ed. Springer-Verlag Berlin Heidelberg, 2008.

[Pri06] Priss, U. «Formal concept analysis in information science». In: *ARIST* 40.1 (2006), pp. 521–543.

[PGO13] Prokasheva, Olga, Gurov, Sergey, and Onishchenko, Alina. «Classification Methods Based on Formal Concept Analysis». In: 2013.

[Qui86] Quinlan, J. R. «Induction of Decision Trees». In: *Mach. Learn.* 1.1 (Mar. 1986), pp. 81–106.

[Qui93] Quinlan, J. R. *C4.5: Programs for Machine Learning*. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 1993.

[RG03] Ramon, J. and Gärtner, T. «Expressivity versus efficiency of graph kernels». In: *Proceedings of the First International Workshop on Mining Graphs, Trees and Sequences*. 2003, pp. 65–74.

[SB09] Shervashidze, Nino and Borgwardt, Karsten. «Fast Subtree Kernels on Graphs». In: *Advances in Neural Information Processing Systems 22 – Proceedings of the 2001 Neural Information Processing Systems Conference NIPS 2009, December 7-10, 2009 Vancouver, British Columbia, Canada*. Ed. by Bengio, Yoshua et al. Neural Information Processing Systems Foundation, 2009, pp. 1660–1668.

[She+09] Shervashidze, Nino et al. «Efficient graphlet kernels for large graph comparison». In: *Journal of Machine Learning Research - Proceedings Track* 5 (2009), pp. 488–495.

[She+11] Shervashidze, Nino et al. «Weisfeiler-Lehman Graph Kernels». In: *J. Mach. Learn. Res.* 12 (Nov. 2011), pp. 2539–2561.

[Smi09] Smith, D. T. «A Formal Concept Analysis Approach to Association Rule Mining: The Quicl Algorithms». AAI3352919. PhD thesis. 2009.

[Sow84] Sowa, J.F. *Conceptual Structures: Information Processing in Mind and Machine*. Boston, MA, USA: Addison-Wesley Longman Publishing Co., Inc., 1984.

[Til04] Tilley, T. «Concept Lattices: Second International Conference on Formal Concept Analysis, ICFCA 2004, Sydney, Australia, February 23-26, 2004. Proceedings». In: ed. by Eklund, Peter. Berlin, Heidelberg: Springer Berlin Heidelberg, 2004. Chap. Tool Support for FCA, pp. 104–111.

[TMM16] Trabelsi, Marwa, Meddouri, Nida, and Maddouri, Mondher. «New Taxonomy of Classification Methods Based on Formal Concepts Analysis». In: *FCA4AI@ECAI*. 2016.

[VMZ06] Veloso, A., Meira Jr., W., and Zaki, M. J. «Lazy Associative Classification». In: *Proceedings of the Sixth International Conference on Data Mining*. ICDM '06. Washington, DC, USA: IEEE Computer Society, 2006, pp. 645–654.

[Vis+10] Vishwanathan, S.V.N. et al. «Graph Kernels». In: *J. Mach. Learn. Res.* 11 (Aug. 2010), pp. 1201–1242.

[WWK08] Wale, Nikil, Watson, Ian A., and Karypis, George. «Comparison of descriptor spaces for chemical compound retrieval and classification». In: *Knowledge and Information Systems* 14.3 (2008), pp. 347–375.

[Wil09] Wille, R. «Restructuring Lattice Theory: an Approach Based on Hierarchies of Concepts.» In: *Proceedings of the 7th International Conference on Formal Concept Analysis*. ICFCA '09. Darmstadt, Germany: Springer-Verlag, 2009, pp. 314–339.

[YV15] Yanardag, Pinar and Vishwanathan, S.V.N. «Deep Graph Kernels». In: *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. KDD '15. Sydney, NSW, Australia: ACM, 2015, pp. 1365–1374.

[ZH02] Zaki, Mohammed Javeed and Hsiao, Ching-Jiu. «CHARM: An Efficient Algorithm for Closed Itemset Mining.» In: *SDM*. Ed. by Grossman, Robert L. et al. SIAM, 2002, pp. 457–473.

[ZCG13] Zhou, Cheng, Cule, Boris, and Goethals, Bart. «Itemset Based Sequence Classification». In: *Machine Learning and Knowledge Discovery in Databases - European Conference, ECML PKDD 2013, Prague, Czech Republic, September 23-27, 2013, Proceedings, Part I*. Ed. by Blockeel, Hendrik et al. Vol. 8188. Lecture Notes in Computer Science. Springer, 2013, pp. 353–368.

Список рисунков

Список таблиц

Приложения

Приложение А.

Данные	DT msl	RF msl	kNN k
audiology	1	1	2
balance-scale	6	1	50
breast cancer	4	3	5
car	3	2	5
hayses-roth	3	1	15
lymph	1	1	5
mol-bio-prom	3	3	5
nursery	3	4	50
primary tumor	4	4	30
solar flare	3	1	30
soybean	1	1	2
spect train	9	5	10
tic-tac-toe	10	3	10

Таблица 4.12: Лучшие найденные значения параметров в процессе кросс-валидации для 13 наборов данных репозитория UCI. “DT msl” и “RF msl” означают параметр $min\_samples\_leaf$ – минимальное число объектов в листе дерева и леса соответственно.

Название	Описание	# obj	# attr	# class
heart	Определение наличия или отсутствия сердечного заболевания у пациента	303	13	2
hepatitis	Предсказание выживания пациента при заболевании гепатитом	155	19	2
horse	Данные о том, проводилось ли лечение колик у лошадей с хирургическим вмешательством или нет	300	27	2
ionosphere	Классификация сигналов от радаров на несущие полезную информацию о структуре ионосферы и не несущие	351	34	2
iris	Данные по длине и ширине чашелистника и лепестка для трех видов цветков ириса	150	4	3
pima	Определение склонности пациентов женского пола к заболеванию диабетом	768	8	2
wine	Результаты химического анализа вина на винограде с 3 плантаций в Италии	178	13	3
zoo	Классификация животных на 7 групп	101	17	7

Таблица 4.13: Информация об используемых в экспериментах наборах данных UCI. Здесь # obj, # attr и # class – числа объектов, признаков и значений целевого класса в обучающей выборке.

Данные	# objects	# attr	CART msl	kNN k	CART time	kNN time	CoLiBRi time
colic	368	59	1	30	0.3	0.52	6.41
heart-h	294	24	2	20	0.3	0.52	0.89
heart-statlog	270	13	5	45	0.3	0.53	3.76
hepatitis	155	285	2	10	0.29	0.55	62.9
hypothyroid	3772	126	7	15	0.63	1.39	298.84
ionosphere	351	34	4	10	0.41	0.54	2.03
kr-vs-kp	3196	38	1	50	0.4	2.03	23.15
segment	2310	19	1	10	1.05	0.83	4.17
sonar	208	60	3	15	0.41	0.53	3.79
soybean	683	98	1	10	0.3	0.73	32.6
vehicle	846	18	4	10	0.62	0.63	1.34
vote	435	32	2	10	0.31	0.53	2.65
vowel	990	26	2	35	0.63	0.63	3.29
waveform-5000	5000	40	5	82	3.79	1.34	40.2

Таблица 4.14: Значения параметров алгоритмов и время их работы в вычислительных экспериметнах с 14 наборами данных репозитория UCI. “CART msl” означает параметр $min\_samples\_leaf$ – минимальное число объектов в листе дерева, “kNN k” означает параметр “число соседей” для метода ближайших соседей.