

МИНИСТЕРСТВО ОБРАЗОВАНИЯ
И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ
НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ
ТОМСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

А.В. Замятин

**ВВЕДЕНИЕ
В ИНТЕЛЛЕКТУАЛЬНЫЙ
АНАЛИЗ ДАННЫХ**

Учебное пособие

Томск
Издательский Дом Томского государственного университета
2016

УДК 519.254

ББК 32.81

3269

Замятин А.В.

3269 Интеллектуальный анализ данных : учеб. пособие. –
Томск : Издательский Дом Томского государственного
университета, 2016. – 120 с.

ISBN 978-5-94621-531-2

В работе рассматриваются вопросы, связанные с набирающей популярность областью интеллектуального анализа данных (англ. *Data Mining*). Изучаются основные технологические тренды, сопровождающие *Data Mining*, вопросы терминологии. Рассматриваются основные методы и инструменты *Data Mining*, связанные с высокопроизводительной интеллектуальной аналитической обработкой данных, направленной на то, чтобы оперативно извлекать из значительных массивов накопленных и поступающих данных ценные экспертные знания, поддерживая эффективную управленческую деятельность.

Для студентов университетов и вузов.

УДК 519.254

ББК 32.81

Рецензенты:

С.П. Сущенко, доктор технических наук, профессор;

Л.Г. Гагарина, доктор технических наук, профессор

ISBN 978-5-94621-531-2 © Замятин А.В., 2016

© Томский государственный университет, 2016

ОГЛАВЛЕНИЕ

ПРЕДИСЛОВИЕ	6
1. АКТУАЛЬНОСТЬ	7
1.1. ЭВОЛЮЦИЯ ИНФОРМАЦИОННО-КОММУНИКАЦИОННЫХ ТЕХНОЛОГИЙ И ПОТЕНЦИАЛ ИНТЕЛЛЕКТУАЛЬНОГО АНАЛИЗА ДАННЫХ	7
1.2. ИНТЕЛЛЕКТУАЛЬНЫЙ АНАЛИЗ ДАННЫХ В БИЗНЕСЕ	11
1.2.1. Розничная торговля	12
1.2.2. Сфера развлечений	13
1.2.3. Маркетинг, страхование, работа с персоналом	13
1.2.4. Примеры применения классификации, кластеризации и прогнозирования	15
1.3. ИНТЕЛЛЕКТУАЛЬНЫЙ АНАЛИЗ ДАННЫХ В РЕШЕНИИ СЛОЖНЫХ ПРИКЛАДНЫХ ЗАДАЧ	17
2. ТЕРМИНОЛОГИЯ	20
2.1. DATA MINING	23
2.2. BIG DATA	26
2.2.1. Основные понятия	28
2.2.2. Свойства Big Data	29
2.3. DATA MINING И BIG DATA	29
2.4. ДЕДУКЦИЯ И ИНДУКЦИЯ	30
3. ОСНОВНЫЕ ЗАДАЧИ И КЛАССИФИКАЦИЯ МЕТОДОВ АНАЛИЗА ДАННЫХ	31
3.1. ЭТАПЫ ИНТЕЛЛЕКТУАЛЬНОГО АНАЛИЗА ДАННЫХ	31
3.2. ОБЩИЕ ТИПЫ ЗАКОНОМЕРНОСТЕЙ ПРИ АНАЛИЗЕ ДАННЫХ	31

3.3. ГРУППЫ ЗАДАЧ АНАЛИЗА ДАННЫХ	32
3.4. КЛАССИФИКАЦИЯ МЕТОДОВ	35
3.5. СРАВНИТЕЛЬНЫЕ ХАРАКТЕРИСТИКИ ОСНОВНЫХ МЕТОДОВ	37
4. ОСНОВНЫЕ МЕТОДЫ АНАЛИЗА И ИНТЕРПРЕТАЦИИ ДАННЫХ	39
4.1. ПРЕДВАРИТЕЛЬНАЯ ОБРАБОТКА ДАННЫХ	39
4.2. ОПТИМИЗАЦИЯ ПРИЗНАКОВОГО ПРОСТРАНСТВА.....	45
4.2.1. С трансформацией пространства признаков.....	46
4.2.2. Без трансформации пространства признаков.....	48
4.3. КЛАССИФИКАЦИЯ.....	50
4.3.1. Постановка задачи классификации	50
4.3.2. Контролируемая непараметрическая классификация.....	54
4.3.3. Контролируемая непараметрическая нейросетевая классификация	56
4.3.4. Классификация по методу машины опорных векторов.....	60
4.3.5. Деревья решений	62
4.3.6. Неконтролируемая классификация	76
4.4. РЕГРЕССИЯ	80
4.4.1. Понятие регрессии.....	80
4.4.2. Основные этапы регрессионного анализа.....	81
4.4.3. Методы восстановления регрессии	81
4.5. АССОЦИАЦИЯ.....	83
4.5.1. Описание алгоритма.....	86
4.5.2. Пример исполнения алгоритма	87
4.6. ПОСЛЕДОВАТЕЛЬНАЯ АССОЦИАЦИЯ.....	89
4.6.1. Алгоритмы семейства «Априори»	90
4.6.2. Алгоритм GSP	93

4.7. ОБНАРУЖЕНИЕ АНОМАЛИЙ	98
4.8. ВИЗУАЛИЗАЦИЯ	100
5. ВЫСОКОПРОИЗВОДИТЕЛЬНАЯ ОБРАБОТКА ДАННЫХ	101
6. ИНСТРУМЕНТЫ DATA MINING.....	105
6.1. ПРОГРАММНЫЕ ИНСТРУМЕНТЫ ДЛЯ ВЫСОКОПРОИЗВОДИТЕЛЬНОЙ ОБРАБОТКИ ДАННЫХ	106
6.1.1. Программная среда.....	106
6.1.2. Базы данных	107
6.1.3. Языки программирования	108
6.2. ПРИМЕРЫ ПРОГРАММНЫХ СИСТЕМ.....	108
6.2.1. Примеры самостоятельных систем	108
6.2.2. Примеры облачных систем	109
ВОПРОСЫ И ТЕМЫ ДЛЯ САМОПРОВЕРКИ.....	110
ЛИТЕРАТУРА.....	112

ПРЕДИСЛОВИЕ

Стремительная технологическая эволюция последних лет в сфере информационно-коммуникационных технологий позволила сформировать существенный задел в части развитой программно-аппаратной инфраструктуры, поддерживающей накопление и постоянное пополнение архивов данных различной природы и назначения.

Обостряющаяся конкурентная борьба в различных областях человеческой деятельности (бизнесе, медицине, корпоративном управлении и др.) и сложность внешней среды делают крайне востребованными подходы к экспертному использованию имеющихся данных для повышения обоснованности и оперативности принятия управленческих решений.

При этом не всегда сегодня возможно непосредственное эффективное применение хорошо проработанного и известного аппарата теории вероятности или математической статистики без учета особенностей конкретной предметной области, компьютерных наук (включая детали хранения и обработки данных, алгоритмов машинного обучения и т.п.), специфики современных информационных технологий.

Именно поэтому относительно недавно стала привлекать особое внимание область, связанная с высокопроизводительной интеллектуальной аналитической обработкой данных, направленная на то, чтобы оперативно извлекать из значительных массивов накопленных и поступающих данных ценные экспертные знания, поддерживая эффективную управленческую деятельность.

Учитывая междисциплинарный характер этой предметной области, ее глубину и ярко выраженную прикладную направленность, до сих пор существует определенный дефицит систематизированных представлений о ней, на устранение которых в некоторой степени направлено данное пособие.

1. АКТУАЛЬНОСТЬ

1.1. ЭВОЛЮЦИЯ ИНФОРМАЦИОННО-КОММУНИКАЦИОННЫХ ТЕХНОЛОГИЙ И ПОТЕНЦИАЛ ИНТЕЛЛЕКТУАЛЬНОГО АНАЛИЗА ДАННЫХ

С 1960-х гг. информационно-коммуникационные технологии (ИКТ) последовательно эволюционировали от простых систем обработки файлов до сложных, мощных систем управления базами данных (БД). Исследования в области БД с 1970-х гг. смещались от ранних иерархических и сетевых баз данных к реляционным системам управления базами данных (СУБД), инструментам моделирования данных, а также к вопросам индексирования и организации данных. Пользователи получили гибкий и удобный интерфейс доступа к данным с помощью языков запросов (типа SQL), пользовательские интерфейсы, управление транзакциями и т.п. При этом создаваемые и поддерживаемые БД имели преимущественно ограниченный регистрирующий характер, поддерживая рутинные операции линейного персонала. Основными требованиями к таким системам были обеспечение транзакционности и оперативность выполнения всех изменений.

Технология баз данных, начиная с середины 1980-х гг., характеризовалась популяризацией, широким внедрением и концентрацией исследовательских усилий на новые, все более мощные СУБД. Появились новые модели данных, такие как объектно-ориентированные, объектно-реляционные, дедуктивные модели. Возникали различные предметно-ориентированные базы данных и СУБД (пространственные, временные, мультимедийные, научные и пр.). Эффективные методы онлайн-обработки транзакций (*on-line transaction processing* – *OLTP*¹) внесли большой вклад в

¹ Способ организации БД, при котором система работает большим потоком с небольшими по размерам транзакциями при минимальном времени отклика системы.

эволюцию и широкое внедрение реляционной технологии в качестве одного из главных универсальных инструментов эффективно-го хранения, извлечения и управления большими объемами структурированных данных реляционных СУБД.

С развитием сети Интернет получили развитие и вопросы построения распределенных баз данных, создания распределенных глобальных информационных систем. Многократно возросла интенсивность формирования и архивирования различных данных, за которыми следовало развитие масштабируемых программно-аппаратных комплексов, дорогостоящих мощных и недорогих пользовательских компьютеров и накопителей данных.

Все это способствовало всплеску развития индустрии ИКТ и сделало огромное количество баз данных доступными для хранения разнородной информации в значительных объемах и управления транзакциями в них. При этом все больше возникала потребность анализа имеющихся данных в разновременном аспекте, с возможностью построения произвольных запросов, при условии обработки сверхбольших объемов данных, полученных, в том числе, из различных регистрирующих БД. Использование для этих задач традиционных регистрирующих систем и БД крайне затруднительно. Например, в регистрирующей системе информация актуальна исключительно на момент обращения к БД, а в следующий момент времени по тому же запросу можно ожидать другой результат. Интерфейс таких систем рассчитан на проведение определенных стандартизованных операций и возможности получения результатов на нерегламентированный произвольный запрос ограничены. Возможности обработки больших массивов данных также могут быть ограничены вследствие ориентации СУБД на нормализованные данные, характерные для стандартных реляционных регистрирующих БД.

Ответом на возникшую потребность стало появление новой технологии организации баз данных – технологии *хранилищ данных* (англ. *Data Warehouse*²), предполагающей некоторую предва-

² Предметно-ориентированная информационная база данных, главным образом предназначенная для поддержки принятия решений с помощью отчетов.

рительную обработку данных и их интеграцию, а также онлайн-овую аналитическую обработку (англ. *On-Line Analytical Processing, OLAP*³).

Несмотря на очевидную пользу такого инструмента анализа данных, он ориентирован на хорошо нормализованные табличные данные и не предполагает использование целого ряда дополнительного аналитического инструментария типа классификации, кластеризации, регрессионного анализа, моделирования, прогнозирования и интерпретации многомерных данных и т.п.

Таким образом, сегодня наблюдается высокий уровень развития масштабируемой аппаратно-программной ИКТ инфраструктуры, позволяющей увеличивать и без того значительные архивы данных. Имеется достаточно существенный задел в области компьютерных наук и информационных технологий, разработаны теория и прикладные аспекты теории вероятности и математической статистики. Однако при этом следует признать, что присутствует заметный *избыток данных*⁴ при *дефиците информации*⁵ и *знаний*⁶. Быстро растущие объемы накопленных и пополняемых (автоматически, а не людьми – как это было когда-то) архивов данных пока существенно превышают способности человека в их практически полезной обработке. Для обострения этого тезиса иногда говорят, что «*большие базы данных стали могилами, которые редко посещаются*». Как следствие, важные решения порой принимаются не на основе аналитических выводов из информативных БД, а на основе интуиции человека, не имеющего подходящих инструментов

³ Технология анализа данных, предполагающая подготовку агрегированной структурированной многомерной информации на основе больших массивов данных (OLAP-куба), используемой в реляционной БД при построении сложных многотабличных запросов.

⁴ Под *данными* будем понимать представление некоторых фактов в формализованном виде, пригодном для хранения, обработки и передачи.

⁵ Под *информацией* будем понимать сведения в любой форме; в отличие от данных, информация имеет некоторый *контекст*.

⁶ Под *знаниями* будем понимать совокупность информации о мире, свойствах объектов, закономерностях процессов и явлений, а также правилах их использования для *принятия решений*.

для извлечения полезных знаний из имеющихся огромных объемов данных.

Поэтому в последние годы стремительное развитие получила область *Data Mining*⁷ (в отечественной литературе наиболее используемая аналогия – *интеллектуальный анализ данных*, ИАД), направленная на поиск и разработку методов извлечения из имеющихся данных *знаний*, позволяющих принимать на их основе конкретные, в высокой степени обоснованные, практически полезные управленческие решения.

На рис. 1 приведен пример обобщенного иерархического представления методологий обработки данных, начиная от интеграции разнородных источников данных и завершая использованием методов *Data Mining* для принятия управленческих решений.



Рис. 1. Пример обобщенного иерархического представления методологий обработки данных при принятии управленческих решений

⁷ Вопросам терминологии посвящена гл. 2.

1.2. ИНТЕЛЛЕКТУАЛЬНЫЙ АНАЛИЗ ДАННЫХ В БИЗНЕСЕ

Наибольший интерес к технологиям интеллектуальной обработки данных, в первую очередь, проявляют компании, работающие в условиях *высокой конкуренции* и имеющие *четкую группу потребителей* (розничная торговля, финансы, связь, маркетинг). Они используют любую возможность для повышения эффективности собственного бизнеса через принятие более эффективных управленческих решений. Эти компании пытаются найти связь между «*внутренними*» (цена, востребованность продукта, компетентность персонала и т.п.) и «*внешними*» (экономические показатели, конкуренция, демография клиентов и т.п.) факторами. Это позволяет им оценить (прогнозировать) уровень продаж и удовлетворенности клиентов, размер доходов, а также сформулировать на основе совокупности всей имеющейся информации практически полезные выводы и рекомендации. Иногда отдача от применения этих инструментов может составлять сотни процентов при сравнительно невысокой стоимости внедрения.

При этом результатом обработки данных должен быть такой *информационный продукт*, который позволяет принять конкретное управленческое действие без избыточного «погружения» лица, принимающего решение (ЛПР) в детали базовых данных или промежуточной аналитики (например, дать рекомендации по покупке / продаже на финансовом рынке, сформировать перечень мероприятий по увеличению производительности или маркетингу продукта и т.п.). Причем на практике возможна ситуация, при которой какое-либо решение в той или иной части необходимо принимать *обязательно* – вопрос только в том, принимается оно на основе объективной информации или интуитивно.

Извлечение своевременной и готовой непосредственно для принятия управленческих решений информации из различных источников предполагает создание некоторых информационных продуктов. Примерами таких информационных продуктов в бизнесе могут быть ответы на вопросы типа:

- Какой из продуктов следует рекламировать больше для увеличения прибыли?
- Как следует усовершенствовать программу модернизации для уменьшения расходов?
- Какой процесс производства изменить, чтобы сделать продукт лучше?

Ключ к ответу на эти вопросы требует глубокого понимания имеющихся данных и их *индуктивного*⁸ анализа.

Рассмотрим некоторые примеры применения методов интеллектуального анализа данных, используемых в бизнес-среде и подтверждающих на практике возрастающую актуальность этой интеллектуальной сферы человеческой деятельности.

1.2.1. Розничная торговля

Используя методы интеллектуального анализа данных, пункт розничной торговли (магазин) может фиксировать информацию обо всех покупках клиента и рассылать таргетировано⁹ рекламные предложения своим клиентам на основе истории их покупок. Анализируя демографическую информацию о клиентах, магазин может предлагать товары и рекламные предложения для конкретного клиентского сегмента.

Всемирно известная торговая сеть США *WalMart* – пионер интеллектуального анализа данных, примененного для модернизации взаимодействия с поставщиками. Компания *WalMart* проанализировала транзакции 2 900 магазинов из 6 стран, сформировав хранилище данных объемом 7,5 Тбайт. При этом потребовалось выполнить к данным более 1 млн сложных запросов. Данные использованы для определения паттернов покупателей при совершенствовании *мерчендайзинговых*¹⁰ стратегий для 3 500 поставщиков.

⁸ От частного к общему.

⁹ Таргетинг (англ. *target* – цель) – рекламный механизм, позволяющий выделить целевую аудиторию для демонстрации ей рекламы.

¹⁰ Мерчендайзинг (англ. – *merchandising*) – искусство сбыта.

Типовыми вопросами, на которые идет поиск ответов при анализе данных в розничной торговле, являются:

- Кто ваш покупатель?
- Как сегментировать клиентов?
- На какую целевую аудиторию сделать акцент?
- Какие факторы влияют на решение о покупке?
- Какова значимость каждого из факторов?
- Какие товары предлагать в совместных акциях?
- Какие существуют зависимости в поведении клиентов?
- На какой объем спроса в будущем ориентироваться?

1.2.2. Сфера развлечений

Интересными могут быть примеры анализа данных в сфере развлечений. Например, компания по продаже контента для видеопросмотра может анализировать историю пользовательских запросов и предлагать в соответствии с ними индивидуальные рекомендации.

В Национальной баскетбольной ассоциации США традиционно используются инструменты анализа данных для оценки перемещений игроков на площадке, помогая тренерам команд в тактической борьбе и выработке стратегий на игру. Еще в 1995 г. такой анализ игры между *New York Knicks* и *Cleveland Cavaliers* выявил, что защитник *Mark Price* позволил забить нападающему *John Williams* из команды соперника лишь один бросок из четырех, в то время как общая статистика за игру по этому показателю для *Cavaliers* была зафиксирована на уровне 49,30%. Учитывая, что видео всех игр сохраняется фрагментарно, тренер с легкостью может отыскать в большом видеомассиве данных любые интересующие моменты и проанализировать причину успеха и неудачи в конкретном игровом эпизоде, без необходимости часами просматривать все видео в поисках нужного фрагмента.

1.2.3. Маркетинг, страхование, работа с персоналом

Наиболее распространенными сегодня примерами использования инструментов *Data Mining* являются различные интернет-магазины и

поисковые системы. На основе ретроспективных запросов пользователи они определяют профиль его интересов. На основе определения профиля возможно предложить товары и услуги, которые в высокой степени также могут заинтересовать пользователя. Многие сталкиваются сегодня с такими примерами в *Google*, *Amazon*, *eBay* и т.п.

С развитием страхового рынка все большую актуальность инструменты анализа данных приобретают для рынка страхования. Эта сфера человеческой деятельности всегда требовала использования математических моделей оценки рисков. С развитием инструментов *Data Mining* появляются новые перспективные возможности оценки рисков на основе большей совокупности факторов, моделирования страховых убытков, исследования связи среднего уровня доходов территории и числа застрахованных, кластерного анализа в автостраховании (например, для избегания случаев мошенничества), оценки распределения размеров убытков, прогнозирования доходов от продажи страховых полисов и др.

Очевидно, инструменты интеллектуального анализа данных могли бы помочь специалистам кадровой службы любой компании. При этом примерами вопросов, на которые ищут ответы такие специалисты, могут быть:

- Каким требованиям должен удовлетворять соискатель?
- Кто и как часто совершает ошибки?
- Сколько компания теряет из-за ошибок сотрудников?
- Кто является мошенником?
- Как оптимизировать штат и нагрузку?
- Как оптимизировать режимы работы оборудования?
- Как сократить число сбоев и простоев по технологическим причинам?

Примерами тем, которые могут искать специалисты в области маркетинга методами *Data Mining*, могут быть:

- целевая аудитория;
- наиболее выгодные типы клиентов;
- маркетинговые каналы;
- взаимодействие с дилерами;
- политика скидок, специальные предложения.

1.2.4. Примеры применения классификации, кластеризации и прогнозирования

Характерной особенностью *Data Mining* является активное использование методов *классификации*, *кластеризации* и *прогнозирования*, применяемых для выявления неявных закономерностей и свойств, присутствующих в данных. Рассмотрим без математической детализации некоторые прикладные примеры применения этих методов при решении практических задач.

Классификация. В общем виде задача *классификации* заключается в том, чтобы определить, к какому классу (типу, категории) относятся те или иные данные в соответствии с некоторым известным набором атрибутов и массивом соответствующих этим атрибутам данных. При этом множество классов, к одному из которых впоследствии можно отнести исследуемый объект, известно. Каждый класс обладает определенными свойствами, которые характеризуют его объекты.

Например, задачу классификации следует решать в банковском секторе при определении степени *достаточной кредитоспособности* клиента. В этом случае банковский служащий оперирует двумя известными ему классами – *кредитоспособный* и *некредитоспособный*. Характеристиками (признаками) исследуемого (классифицируемого) объекта являются возраст, место работы, уровень доходов, семейное положение. Фактически задача сводится к тому, чтобы определить значение одного из параметров объекта анализа (класс «*кредитоспособный*» или класс «*некредитоспособный*») по значениям всех прочих его параметров (признаков). Говорят, что необходимо определить значение *зависимой переменной* «кредитоспособность» (которая может принимать значения «да» или «нет») при известных значениях *независимых переменных* «возраст», «место работы», «уровень дохода», «семейное положение».

Кластеризация. *Кластеризация* – задача, аналогичная предыдущей, но реализуется в случае, когда набор классов не известен заранее. Например, задачу кластеризации решает маркетолог, разделяя всех клиентов некоторого бизнеса на неопределенное коли-

чество групп по характеристикам условного сходства – социальному и географическому положению, основным мотивам покупки, базовым товарам персональной потребительской корзины, размеру чека и т.п. Более четкое определение целевых групп позволяет дифференцировать для них предложения, повысив результативность промо-акций и снизив неэффективные расходы на ее проведение. Например, более детальное социально-демографическое представление об имеющихся сегментах потребителей в розничной торговле позволяет выделить более доходные сегменты и активизировать для них свою рекламную деятельность, а также выделить менее доходные сегменты и существенно скорректировать для них используемые маркетинговые инструменты. В совокупности такой подход позволит более фокусно, целевым образом, расходовать имеющиеся ресурсы, увеличивая объемы продаж.

Прогнозирование. Метод *прогнозирования* – один из наиболее востребованных в бизнесе. Анализируя данные прошлых периодов, можно построить с некоторой точностью прогноз на будущее (которое каждый бизнесмен хотел бы знать). Причем, чем более подробные и точные ретроспективные данные имеются и чем больше этот анализируемый отрезок времени, тем, как правило, точнее получатся результаты прогнозирования.

Этот метод нередко применяется для оценки спроса на услуги и товары, структуры сбыта, характеризующихся сезонными колебаниями, или позволяет оценить ожидаемую потребность в кадрах. Примером применения такого инструментария, конечно, являются фондовые рынки, на которых трейдеры пытаются угадать направления движения тех или иных графиков и определить верные моменты покупки / продажи.

Высокую актуальность и широкое распространение инструменты прогнозирования приобретают в розничной торговле продовольственными товарами. В условиях скоротечности бизнес-процессов заказа, поставки, хранения и продажи скоропортящихся продуктовых товаров, крайне важно выдержать баланс между *удовлетворенностью клиента* (имеющего достаточный выбор на полках магазина и возможность найти желаемое) и *остатками това-*

ра (приобретенного, занимающего место на складе и в торговом зале, но в итоге непроданного), который будет утилизирован по истечению срока годности. Именно в этом обширном сегменте рынка сегодня идет серьезная борьба между производителями соответствующих программных инструментов интеллектуального анализа данных, которые при умелом использовании быстро дают существенный экономический эффект.

Однако насколько эта задача представляется актуальной, настолько непросто до сих пор идет поиск ее более перспективных решений, удачное воплощение которых зависит от множества факторов и достаточно глубокого знания специалиста по интеллектуальному анализу данных деталей предметной области.

1.3. ИНТЕЛЛЕКТУАЛЬНЫЙ АНАЛИЗ ДАННЫХ В РЕШЕНИИ СЛОЖНЫХ ПРИКЛАДНЫХ ЗАДАЧ

Область применения инструментов ИАД не ограничивается исключительно бизнес-сферами, основным показателем эффективности в которых является прибыль. Очевидно, такой инструментарий может найти и находит применение в других областях человеческой деятельности, функционирование которых сопровождается генерированием и анализом различных данных. Одной из таких важнейших сфер, в которых активно адаптируются методы ИАД, является медицина.

Медицина. Например, данные методы применялись для создания алгоритмов диагностики и прогнозирования в онкологии, неврологии, педиатрии, психиатрии, гинекологии и других областях [24, 37, 81]. На основе полученных результатов построены экспертные системы для постановки диагнозов с использованием правил, описывающих сочетания симптомов разных заболеваний. Правила помогают выбирать показания (противопоказания), предсказывать исходы назначенного курса лечения. В молекулярной генетике и геномной инженерии – это определение маркеров, под которыми понимаются генетические коды, контролирующие те или иные фенотипические признаки живого организма. Известно несколько крупных фирм, специализирующихся на применении

ИАД для расшифровки генома человека и растений. В прикладной химии эти методы используют для выяснения особенностей строения химических соединений.

В медицине консолидируют информацию из различных источников – данные из медицинских карт, результаты анализов и проб, выходные показатели диагностирующих тест-систем. Создают и применяют *советующие системы для диагностики заболеваний*, которые выявляют значимые признаки и моделируют сложные зависимости между симптомами и заболеваниями с использованием разнообразных регрессионных моделей, деревьев решений, нейронных сетей и т.д. Также выполняют *оценку диагностических тестов* в сравнении с традиционными методиками, осуществляя подбор оптимальных порогов диагностических показателей и определение чувствительности и пределов применимости конкретной модели. Кроме того, инструменты ИАД (ассоциативные правила и последовательные шаблоны) могут применяться при выявлении связей между *приемом препаратов и побочными эффектами*.

Вместе с тем создание алгоритмического и программного обеспечения систем поддержки медицинской деятельности до сих пор не является тривиальной задачей. Потенциал ИАД может быть раскрыт в эффективном диалоге квалифицированного медицинского специалиста и разработчика интеллектуальных систем. Однако в такой сложной сфере, как медицина, квалифицированный специалист не всегда способен доступно для разработчика объяснить свои «рассуждения»¹¹. Поэтому потенциал применения технологий и методов ИАД в медицине все еще остается в значительной степени нереализованным.

Государственное управление. Большие объемы информации, концентрирующиеся в органах государственного и муниципального управления, обычно хранятся в разрозненном и не всегда готовом для непосредственной автоматизированной обработки виде,

¹¹ Для других областей человеческой деятельности этот тезис часто также справедлив.

содержат значительное количество неточностей и пробелов, что является препятствием для ее эффективного использования в процессе принятия решений. Современные инструменты ИАД используют для *поиска существующих в данных противоречий, дублирования, опечаток и их корректировки*. Также полезным является *реализация различных бюджетных моделей* с возможностями сравнения разных вариантов, условного моделирования, прогнозирования развития ситуации, расчета показателей эффективности. Кроме того, возможны автоматическое *обнаружение отклонений*, их *ранжирование* и *оповещение заинтересованных лиц*. Все это успешно применяют в задачах поиска лиц, уклоняющихся от налогов, или при поиске вероятных источников террористических угроз.

2. ТЕРМИНОЛОГИЯ

Рассматривая вопросы терминологии, описывающей обсуждаемую предметную область интеллектуального анализа данных, логично изучить существующую и наиболее устоявшуюся в мире *англоязычную терминологию* и, уже ориентируясь на нее, обсудить удачные *терминологические аналогии*, используемые в русскоязычных публикациях по данной тематике.

В гл. 1 отмечено, что с развитием ИКТ индустрии, стремительно увеличиваются и возможности генерирования значительных массивов данных, при умелом анализе которых могут быть найдены полезные знания, позволяющие повысить эффективность принятия управленческих решений в бизнесе, медицине или государственном управлении. Область, изучающую эти вопросы, принято называть *Data Mining*, а специалиста этой области – *Data Scientist*. На рис. 2 приведен пример графика, отражающего число вакансий на портале для поиска работы, иллюстрирующий стремительный рост на специалистов *Data Scientist* в последнее время.

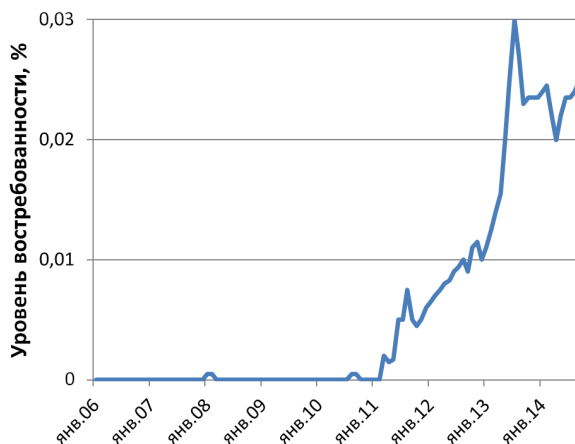


Рис. 2. Иллюстрация востребованности специалистов Data Scientist

Если посмотреть на желаемый профиль специалиста *Data Science*, то видно, насколько разносторонней (междисциплинарной) с точки зрения современного работодателя квалификацией он должен сегодня обладать:

- SQL, 54%
- Python, 46%
- R, 44%
- SAS, 36%
- Hadoop, 35%
- Java, 32%
- optimization, 23%
- C++, 21%
- visualization, 20%
- MATLAB, 18%
- Business Intelligence, 17%
- distributed, 16%
- regression, 16%
- unstructured, 16%
- Hive, 16%
- mobile, 15%

Учитывая продолжающееся интенсивное развитие области анализа данных, встречается отличающаяся терминология, описывающая одно и то же явление или сферу, или один термин, который может трактоваться по-разному.

Например, в англоязычной литературе можно встретить различные термины и их сочетания, описывающие область интеллектуального анализа данных и являющихся достаточно близкими по значению:

- Data Mining;
- Statistical Analysis and Data Mining;
- Machine Learning;
- Deep Learning;
- Predictive Analytics and Data Mining;
- Data Science;
- Data Science and Data Mining;

- Discovery Driven Data Mining;
- Knowledge Discovery in Databases;
- Big Data;
- и др.

Столь разнообразная терминология может содержать множество оттенков, которые определяются порой в каждом конкретном случае в зависимости от контекста. В связи с этим, для большего понимания следует рассмотреть различные аспекты употребления (как видно, все-таки ключевого) понятия *Data Mining*.

2.1. DATA MINING

Существующие массивы данных характеризуются не только значительным объемом и регулярной пополняемостью, но и содержат порой в себе преимущественно тривиальные (неактуальные, ошибочные и т.п.) элементы. Процесс поиска в этих данных чего-то ценного стал сравним с работой на горнорудных предприятиях, где в многотонных завалах руды осуществляется поиск (добыча) драгоценных металлов или камней, полезный выход которых может исчисляться граммами. Учитывая сходную трудоемкость процесса «добычи» (англ. *mining*) знаний из «завалов», данный термин закрепился и для области *Data Mining*.

Формально для *Data Mining* могут быть даны различные определения, не претендующие на исключительную полноту. *Data Mining* – это:

- 1) процесс обнаружения в базах данных нетривиальных и практически полезных закономерностей [4];
- 2) процесс выделения, исследования и моделирования больших объемов данных для обнаружения неизвестных до этого структур (паттернов) с целью достижения преимуществ в бизнесе [31];
- 3) процесс, цель которого – обнаружить новые значимые корреляции, образцы и тенденции в результате просеивания большого объема хранимых данных с использованием методик распознавания образов и других статистических и математических методов [16];
- 4) исследование и обнаружение «машиной» (алгоритмами, средствами искусственного интеллекта) в «сырых» данных скры-

тых знаний, которые ранее не были известны, нетривиальны, практически полезны, доступны для интерпретации человеком [45];

5) процесс обнаружения полезных знаний о бизнесе [38].

Для более глубокого понимания сути явления *Data Mining* только какого-либо одного определения не вполне достаточно. Под *Data Mining* (наиболее устоявшаяся аналогия в русском языке, но не прямой перевод, – *интеллектуальный анализ данных*; по-видимому, это более широкий термин) понимают совокупность методов обнаружения в данных знаний, но обязательно обладающих следующими свойствами:

- ранее неизвестных, неожиданных;
- практически полезных;
- доступных для интерпретации;
- необходимых для принятия решений в различных сферах человеческой деятельности.

В более широком смысле под *Data Mining* понимают концепцию анализа данных, предполагающую, что:

- данные могут быть неточными, неполными (содержать пропуски), противоречивыми, разнородными, косвенными и при этом иметь гигантские объемы; поэтому понимание данных в конкретных приложениях требует значительных интеллектуальных усилий;

- сами алгоритмы анализа данных могут обладать «элементами интеллекта», в частности способностью обучаться по прецедентам, т.е. делать общие выводы на основе частных наблюдений; разработка таких алгоритмов также требует значительных интеллектуальных усилий;

- процессы переработки сырых данных в информацию, а информации в знания уже не могут быть выполнены по старинке «вручную» и требуют порой нетривиальной автоматизации.

Data Mining является *мультидисциплинарной областью*, возникшей и развивающейся на базе достижений прикладной математической статистики, распознавания образов, методов искусственного интеллекта, теории баз данных и др. (рис. 4).

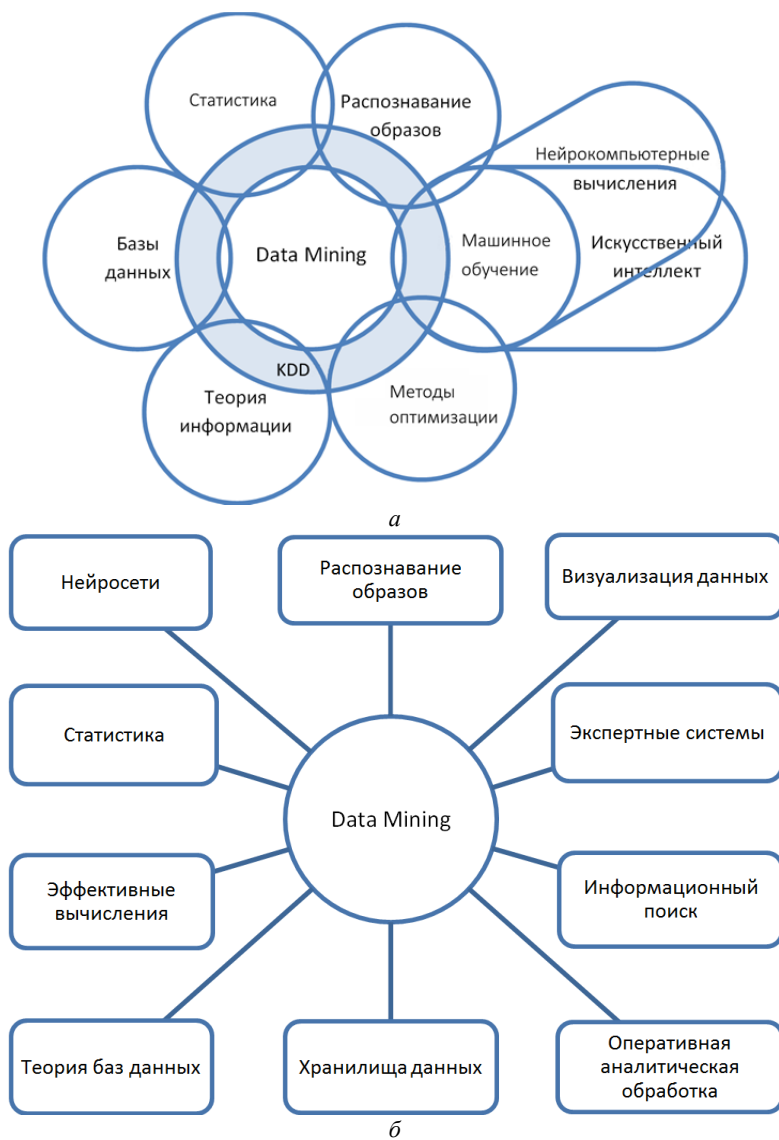


Рис. 4. Иллюстрации Data Mining как междисциплинарной области

Иногда отмечают иной характер мультидисциплинарности *Data Mining* – это объединение *компьютерных наук* (англ. *Computer Science*), *математики* (англ. *Mathematics*) и представлений о *предметной области* (англ. *Domain Expertise*). В этом случае компьютерные науки описывают среду создания информационных продуктов (англ. *data products*), математика выстраивает теоретическую основу для решения поставленных проблем, а представление о предметной области позволяет понять реальность, в которой существует проблемная ситуация.

Именно включение *предметной области* как междисциплинарной компоненты *Data Mining* существенно осложняет практическую интеллектуальную работу в этой сфере, требуя от специалиста (*Data Scientist*) при решении каждой конкретной задачи анализа данных достаточно глубокого погружения в новую, незнакомую ему область человеческой деятельности. Однако очевидно, что без такого погружения сложно найти эффективные решения существующих проблем.

2.2. BIG DATA

Анализ терминологии, достаточно полно описывающей область *Data Mining*, позволяет отметить, что в ней часто в схожем контексте применяется и такой термин, как *Big Data* (рус. *большие данные*). При этом широта его употребления специалистами и неспециалистами порой затрудняет однозначное толкование этого термина, в особенности учитывая обсуждение деталей понятия *Data Mining* выше. Как же следует понимать *Big Data*?

Ежедневно в мире создается более 5 эксабайтов¹² информации. В 2012 г. в мире было сгенерировано около 2,43 зеттабайт (1 ЗБ = около 1 млрд ГБ), что более чем в 2 раза превосходит объем информации в цифровом виде в 2010 г. (1,2 ЗБ). К 2020 г. информационные системы будут иметь дело с количеством данных равным 40 ЗБ (примерно в 57 раз большим, чем количество песчинок на пляжах всей поверхности Земли). Очевидно, появление и

¹² Единица измерения равная 10^{18} , или 2^{60} , байт.

активное тиражирование термина *Big Data* во многом вызвано сопровождением этого объективного процесса накопления сверхбольших объемов данных.

Действительно, обеспечивать соответствие возможностей ИКТ-инфраструктуры, предназначенной лишь для надежного хранения стремительно растущих объемов накапливаемых данных (не говоря о возможностях их интеллектуальной обработки), – непростая и дорогостоящая задача. Это означает перспективы формирования и развития рынка *Big Data* со значительной финансовой емкостью. Аналитики отмечают, что мировой рынок *Big Data* (технологий и сервисов для обработки данных) к 2015 г. [16, 68]:

- составит 16,9 млрд долл.;
- будет расти почти в 7 раз быстрее, чем ИКТ-рынок в целом;
- потребует создания 4,4 млн новых рабочих мест.

Поэтому целесообразно отметить, что основным драйвером продвижения термина *Big Data* во многом являются *рыночные законы маркетинга*¹³. Они позволяют привлечь внимание к проблеме не только ученых, но и государства и бизнеса, и предусмотреть в бюджетах компаний средства на развитие этих технологий (в первую очередь – именно аппаратной ИКТ-инфраструктуры, что объясняет не столь высокую популярность термина *Data Mining*, связанного больше со специализированным алгоритмическим и программным обеспечением интеллектуальной обработки данных). Именно поэтому, порой, дискуссия о границах того, что является «действительно» «большими данными», а что уже не является, сводится к тому, насколько дорогостоящая инфраструктура требуется для их поддержки¹⁴.

¹³ Именно они, главным образом, «реанимировали» традиционные технологии web-хостинга в виде популярных сегодня *облачных сервисов* или способствуют развитию суперкомпьютерной тематики, регулярно продвигая различные рейтинги мощности суперкомпьютеров в мире.

¹⁴ Например, по этой логике компания *Google* или исследовательский коллаيدر *CERN*, конечно, работают в концепции *Big Data*, а вот все менее масштабные предприятия – сомнительно...

2.2.1. Основные понятия

Рассмотрим некоторые наиболее типичные определения и толкования термина *Big Data* [46]:

- данные очень большого объема;
- область управления и анализа больших объемов данных;
- область управления и анализа больших объемов данных, представленных (в отличие от реляционных БД) в слабоструктурированных форматах (веб-журналы, видеозаписи, текстовые документы, машинный код или, например, геопространственные данные и т.п.);
- область работы с информацией огромного объема и разнообразного состава, весьма часто обновляемой и находящейся в разных источниках в целях увеличения эффективности работы, создания новых продуктов и повышения конкурентоспособности;
- область, объединяющая техники и технологии, которые извлекают смысл из данных на экстремальном пределе практичности;
- постоянно растущий объем информации, поступающей в оперативном режиме из социальных медиа, от сетей датчиков и других источников, а также растущий диапазон инструментов, используемых для обработки данных и выявления на их основе важных бизнес-тенденций;
- наборы данных, превосходящие возможности традиционных программно-аппаратных инструментов оперирования данными (например, в случае, когда параметры набора данных превосходят возможности обработки стандартными средствами *MS Excel*).

Очевидно, и данный набор понятий не позволит совершенно четко определить, где построить границу между «действительно» «большими данными» и «просто данными». Просто сравнительно большой объем данных не всегда можно по умолчанию отнести к сфере *Big Data*, так как возможности их анализа зависят не только от объема данных, но и от вычислительной сложности задачи (очевидно, трудно сравнивать по сложности задачу расчета стандартных статистик и задачу построения комплексной оптимизационной модели). Поэтому более интересным выглядит предложение

под *Big* в *Big Data* понимать не какой-то конкретный физический объем данных или другие количественные показатели, а «важные», «ключевые» данные [5].

2.2.2. Свойства Big Data

Дополнительное понимание термину *Big Data* придают четыре свойства, кратко сформулированные по четырем английским словам¹⁵, начинающимся на букву *v*:

- *Volume* – отражает значительный физический объем данных;
- *Variety* – показывает существенное разнообразие типов данных (например, структурированные, частично структурированные, неструктурированные как текст, web-контент, мультимедиа данные), источников данных (внутренние, внешние, общественные) и их детальности;

- *Velocity* – демонстрирует скорость, с которой данные создаются и обрабатываются;

- *Veracity* – определяет варьируемый уровень помех и ошибок в данных.

Как отмечено выше, свойство *Volume* часто наименее важное, и нет какого-либо обязательного требования к минимальному объему обрабатываемых данных в концепции *Big Data*. Существенно более высокой важностью обладают свойства *Variety* и *Velocity*. Так, свойство *Variety* может приносить особенно высокую ценность в данные, скомбинированные из различных источников (например, корпоративные данные, данные социальных сетей и публичная информация), даже на небольших объемах. Наиболее важным является свойство *Veracity*, определяющее качество и корректность данных.

2.3. DATA MINING И BIG DATA

В заключение терминологических пояснений, характеристик и свойств, описывающих области *Data Mining* и *Big Data*, сформу-

¹⁵ *Four Vs* (англ. – четыре буквы «V» по первым буквам использованных слов): объем, многообразие, скорость, достоверность.

лируем более четко отличия одного термина от другого. Итак, наиболее корректным, во избежание путаницы между терминами, представляется целесообразным под *Big Data* здесь и далее понимать некоторый *актив*¹⁶, который при умелом применении *Data Mining* (технологий, методов, способов) позволяет получить (извлечь) практически полезный результат (экономический эффект).

2.4. ДЕДУКЦИЯ И ИНДУКЦИЯ

В интеллектуальном анализе данных обсуждают два основных подхода извлечения практически полезных знаний – *дедуктивный* (на основе некоторой априори сформулированной гипотезы, от общего к частному) и *индуктивный* (на основе известных *паттернов*¹⁷, от частного к общему).

Дедуктивный подход к исследованию данных предполагает наличие некоторой сформулированной гипотезы, подтверждение или опровержение которой после анализа данных позволяет получить некоторые частные сведения.

Индуктивный подход к исследованию данных позволяет сформулировать (скорректировать существующую) гипотезу и найти с ее помощью новые пути аналитических решений.

Для поиска значимых закономерностей порой требуется совместное попеременное использование индуктивного и дедуктивного подходов, при котором формируется такая среда, в которой модели не нужно быть исключительно статической или эмпирической. Вместо этого, модель непрерывно тестируется, модифицируется и улучшается до тех пор, пока не будет достаточно усовершенствована.

¹⁶ В международных стандартах бухгалтерской отчетности – ресурс компании, от которого компания в будущем ожидает экономической выгоды.

¹⁷ Например, некоторое нетривиальное утверждение о структуре данных, имеющих закономерностях, зависимостях между атрибутами и т.п.

3. ОСНОВНЫЕ ЗАДАЧИ И КЛАССИФИКАЦИЯ МЕТОДОВ АНАЛИЗА ДАННЫХ

3.1. ЭТАПЫ ИНТЕЛЛЕКТУАЛЬНОГО АНАЛИЗА ДАННЫХ

Выделяют следующие типовые этапы, сопровождающие решение задач интеллектуального анализа данных:

1. Анализ предметной области, формулировка целей и задач исследования.

2. Извлечение и сохранение данных.

3. Предварительная обработка данных:

– очистка (англ. *cleaning*; исключение противоречий, случайных выбросов и помех¹⁸, пропусков);

– интеграция (англ. *integration*; объединение данных из нескольких возможных источников в одном хранилище);

– преобразование (англ. *transformation*; может включать агрегирование и сжатие данных, дискретизацию атрибутов и сокращение размерности и т.п.).

4. Содержательный анализ данных методами *Data Mining* (установление общих закономерностей или решение более конкретных, частных задач).

5. Интерпретация полученных результатов с помощью их представления в удобном формате (визуализация и отбор полезных паттернов, формирование информативных графиков и / или таблиц).

6. Использование новых знаний для принятия решений.

3.2. ОБЩИЕ ТИПЫ ЗАКОНОМЕРНОСТЕЙ ПРИ АНАЛИЗЕ ДАННЫХ

Как правило, выделяют пять стандартных типов закономерностей, которые позволяют относить используемые методы к методам *Data Mining*:

¹⁸ Если они сами не являются предметом анализа в данном случае.

1. Ассоциация.
2. Последовательность.
3. Классы.
4. Кластеры.
5. Временные ряды.

Ассоциация (англ. *Association*) имеет место в случае, если несколько событий связаны друг с другом. Например, исследование показывает, что 75% покупателей, приобретавших кукурузные чипсы, приобретают и «колу». Это ассоциация позволяет предложить скидку за такой тип продуктового «комплекта» и, возможно, увеличить тем самым объемы продаж.

В случае если несколько событий связаны друг с другом во времени, то имеет место тип зависимости, именуемый *последовательностью* (англ. *Sequential Patterns*). Например, после покупки дома в 45% случаев в течение месяца приобретается и новая кухонная плита, а в пределах двух недель 60% новоселов обзаводятся холодильником.

Закономерность *классы* (англ. *Classes*) появляется в случае, если имеется несколько заранее сформированных классов (групп, типов) объектов. Отнесение нового объекта к какому-либо из существующих классов выполняется путем *классификации*. Закономерность *кластеры* (англ. *Clusters*) отличается тем, что классы (группы, типы) заранее не заданы, а их количество и состав определяются автоматически в результате процедуры *кластеризации*.

Хранимая ретроспективная информация позволяет определить еще одну закономерность, заключающуюся в поиске существующих *временных рядов* (англ. *Time Series*) и прогнозировании динамики значений в них на будущие периоды времени.

3.3. ГРУППЫ ЗАДАЧ АНАЛИЗА ДАННЫХ

Наряду с поиском самых общих типов закономерностей, которые могут присутствовать в данных (§ 3.2), также выделяют группы более конкретных, частных задач анализа данных. Несмотря на обширную сферу применения *Data Mining* в бизнесе, медицине или государственном управлении (§ 1.2, 1.3), подавляющее боль-

шинство этих задач может быть объединено в сравнительно небольшое число групп (табл. 1).

Т а б л и ц а 1

Основные группы задач анализа данных

Группа задач (англ.)	Аналог в отечественной литературе	Пояснение	Пример задачи
<i>Classification and Prediction</i>	Классификация и прогнозирование	Индуктивно разрабатывается обобщенная модель или формулируется некоторая гипотеза, описывающая принадлежность объектов к соответствующим классам	Предсказание роста объемов продаж на основе текущих значений, отнесения претендента на кредит к известным классам кредитоспособности, выявление лояльных или нелояльных держателей кредитных карт, классификация стран по климатическим зонам и т.п.
<i>Clustering</i>	Кластеризация	Выделение некоторого количества групп, имеющих сходные в некотором смысле признаки. Основной принцип – максимизация межклассового и минимизация внутриклассового расстояния	Обнаружение новых сегментов рынка, совершенствование рекламных стратегий для различных групп потребителей
<i>Associations, Link Analysis</i>	Ассоциации, анализ взаимозависимостей	Поиск интересных ассоциаций и / или корреляционных связей	95% покупателей автомобильных шин и автоаксессуаров также приобретали пакет сервисного обслуживания автомобиля, 80% покупателей газировки приобретают и «воздушную» кукурузу
<i>Visualization</i>	Визуализация	С использованием графических методов визуализации информации создается графический образ анализируемых	Визуализация некоторых зависимостей с использованием 2D- и 3D-измерений

Группа задач (англ.)	Аналог в отечественной литературе	Пояснение	Пример задачи
		данных, отражающих имеющиеся в данных интересные закономерности	
<i>Summarization</i>	Подведение итогов	Интегральное (генерализованное) описание конкретных групп объектов из анализируемого набора данных	Суммирование данных сетевого трафика при оценке эффективности каналов связи [10], подготовка краткого реферата по тексту значительного объема, визуализация многомерных данных большого объема
<i>Deviation (Anomaly) Detection, Outlier Analysis</i>	Определение и анализ отклонений и / или выбросов в данных	Обнаружение фрагментов данных, существенно отличающихся от общего множества данных, выявление нехарактерных паттернов (шаблонов)	Применимо при анализе наличия шума / ошибок, а также при выявлении мошеннических действий
<i>Estimation</i>	Оценивание	Предсказание непрерывных значений признака	Оценка производительности процессора на определенных задачах по ряду параметров процессора, оценка числа детей в семье по уровню образования матери, оценка дохода семьи по количеству в ней автомобилей, оценка стоимости недвижимости в зависимости от ее удаленности от бизнес-центра

Группа задач (англ.)	Аналог в отечественной литературе	Пояснение	Пример задачи
<i>Feature Selection, Feature Engineering</i>	Отбор значимых признаков	Применяется при анализе признаков пространств большой размерности путем сокращения размерности и / или выбора значимых признаков с трансформацией признакового пространства или без трансформации	Как правило, применяется как вспомогательный метод на этапе предварительной обработки данных, а также для повышения эффективности методов визуализации в многомерных признаковых пространствах

3.4. КЛАССИФИКАЦИЯ МЕТОДОВ

Существует большое количество различных оснований для стратификации, категоризации, классификации значительного количества существующих и вновь разрабатываемых методов *Data Mining*. Например, можно встретить классификации по принципу работы с исходными обучающими данными (подвергаются они или нет в результате обработки изменениям), по типу получаемого результата (предсказательные и описательные, рис. 5), по видам применяемого математического аппарата (статистические и кибернетические) и др.

Например, по типу используемого математического аппарата, как правило, выделяют следующие основные группы методов *Data Mining*:

1. Дескриптивный анализ и описание исходных данных, предварительный анализ природы статистических данных (проверка гипотез стационарности, нормальности, независимости, однородности, оценка вида функции распределения, ее параметров и т.п.).
2. Многомерный статистический анализ (линейный и нелинейный дискриминантный анализ, кластерный анализ, компонентный анализ, факторный анализ и т.п.).

3. Поиск связей и закономерностей (линейный и нелинейный регрессионный анализ, корреляционный анализ и т.п.).

4. Анализ временных рядов (динамические модели и прогнозирование).



Рис. 5. Иллюстрация примера классификации методов *Data Mining*

Детализируя используемый математический аппарат, являющийся важнейшим компонентом практически любых современных методов *Data Mining*, можно получить существенно более глубокую классификацию существующих методов (табл. 2), многие из которых более подробно изложены в главе 4.

Таблица 2

Пример классификации методов *Data Mining* по математическому аппарату

Раздел	Методы, способы
Метрические методы классификации	Метод ближайших соседей и его обобщения, отбор эталонов и оптимизация метрики
Логические методы классификации	Понятия закономерности и информативности, решающие списки и деревья
Линейные методы классификации	Градиентные методы, метод опорных векторов

Раздел	Методы, способы
Байесовские методы классификации	Оптимальный байесовский классификатор, параметрическое и непараметрическое оценивание плотности, разделение смеси распределений, логистическая регрессия
Методы регрессионного анализа	Многомерная линейная регрессия, нелинейная параметрическая регрессия, непараметрическая регрессия, неквадратичные функции потерь, прогнозирование временных рядов
Нейросетевые методы классификации и регрессии	Многослойные нейронные сети
Композиционные методы классификации и регрессии	Линейные композиции, бустинг, эвристические и стохастические методы, нелинейные алгоритмические композиции
Критерии выбора моделей и методы отбора признаков	Задачи оценивания и выбора моделей, теория обобщающей способности, методы отбора признаков
Ранжирование	
Обучение без учителя	Кластеризация, сети Кохонена, таксономия; поиск ассоциативных правил, задачи с частичным обучением, коллаборативная фильтрация, тематическое моделирование, обучение с подкреплением

3.5. СРАВНИТЕЛЬНЫЕ ХАРАКТЕРИСТИКИ ОСНОВНЫХ МЕТОДОВ

В завершение различных подходов к классификации методов *Data Mining* приведем пример сравнительного анализа наиболее широко используемых методов между собой, применяя в качестве характеристики каждого из атрибутов следующую шкалу оценок: «чрезвычайно низкая, очень низкая, низкая / нейтральная, нейтральная / низкая, нейтральная, нейтральная / высокая, высокая, очень высокая» (табл. 3). Видно, что ни один из методов нельзя признать единственно эффективным, имеющим очевидное превосходство над другими методами.

Т а б л и ц а 3

Пример сравнительного анализа методов *Data Mining*

Характеристика \ Метод	Линейная регрессия	Нейронные сети	Методы визуализации	Деревья решений	К-ближайшего соседа
Точность	Нейтральная	Высокая	Низкая	Низкая	Низкая
Масштабируемость	Высокая	Низкая	Очень низкая	Высокая	Очень низкая
Интерпретируемость	Высокая / нейтральная	Низкая	Высокая	Высокая	Высокая / нейтральная
Пригодность к использованию	Высокая	Низкая	Высокая	Высокая / нейтральная	Нейтральная
Трудоемкость	Нейтральная	Нейтральная	Очень высокая	Высокая	Низкая / нейтральная
Разносторонность	Нейтральная	Низкая	Низкая	Высокая	Низкая
Быстрота	Высокая	Очень низкая	Чрезвычайно низкая	Высокая/нейтральная	Высокая
Популярность	Низкая	Низкая	Высокая / нейтральная	Высокая / нейтральная	Низкая

Это подтверждает тезис о том, что залогом успешного решения задач *Data Mining* является необходимость погружения не только в особенности предметной области, но и в математические основы различных методов обработки и анализа данных.

4. ОСНОВНЫЕ МЕТОДЫ АНАЛИЗА И ИНТЕРПРЕТАЦИИ ДАННЫХ

4.1. ПРЕДВАРИТЕЛЬНАЯ ОБРАБОТКА ДАННЫХ

Практическое применение методов *Data Mining* предполагает многоэтапную процедуру, основные этапы которой изложены в § 3.1. Одним из ключевых этапов этой процедуры, предваряющей, собственно, применение методов *Data Mining*, является этап предварительной обработки данных, включающий различные типы преобразований. Рассмотрим их более подробно.

Одним из ключевых преобразований этапа предварительной обработки данных является «очистка» данных (англ. *Data Cleaning*, *Data Cleansing*, *Data Scrubbing*), предполагающая обнаружение и корректировку / удаление поврежденных элементов данных. Данные, имеющие такие повреждения (неточные, неполные, дублированные, противоречивые, зашумленные), называют «грязными». Источниками «грязных» данных могут быть поврежденные инструменты сбора данных, проблемы во введении исходных данных, «человеческий фактор» в случае неавтоматического варианта формирования данных, проблемы в каналах передачи данных, ограничения технологий передачи данных, использование разных наименований в пределах одной номенклатуры и т.п.

Особую актуальность наличие очистки грязных данных подтверждает известное в информатике выражение – «*Mycop на входе – mycop на выходе*» (англ. *Garbage In – Garbage Out*, *GIGO*¹⁹). Оно означает, что при неверных входных данных будут получены неверные результаты работы, в принципе, верного алгоритма. Действительно, практически полезными результаты применения каких бы то ни было методов *Data Mining* будут только в случае использования ими корректных достоверных данных. Учитывая

¹⁹ В отличие от известной дисциплины обслуживания, *FIFO* встречается не часто, а жаль...

то, что такие данные могут быть доставлены из разных источников и быть достаточно существенными в объеме, задача получения и обработки «чистых» данных может быть крайне непростой.

Более того, следует отметить, что наличие «грязных» данных может быть порой более проблематичным, чем их отсутствие вообще – извлечение полезных знаний из таких данных может потребовать значительного времени, причем безрезультатно. При этом еще более проблематичным будет успешное извлечение из таких данных недостоверных знаний и дальнейшее их практическое использование с трудно предсказуемыми последствиями. Именно поэтому этапу получения «чистых», готовых к анализу данных придают большое значение, а по затратам времени этот этап может быть одним из самых длительных [33].

Сегодня проблемам получения «чистых» данных посвящены отдельные достаточно емкие исследования [28]. В них обсуждается целый спектр различных особенностей этой проблематики, начиная от концептуальных вопросов и завершая деталями современных технологических решений в базах данных и хранилищах данных. Отметим здесь некоторые наиболее принципиальные моменты.

Все проблемы очистки данных разделяют на две группы, вызванные *интеграцией различных источников данных* (англ. *Multi-Source Problems*) или обусловленные проблемами *единственного источника данных* (англ. *Single-Source Problems*). В свою очередь каждая из групп может быть разделена на две другие группы, определяемые либо *несовершенством схем интегрируемых баз данных* (англ. *Schema Level*), либо *несовершенством на уровне собственно элементов данных* (англ. *Instance Level*, записей, объектов и т.п.). Далее каждая из ветвей полученного дерева классификации детализируется конкретным перечнем возможных проблем очистки данных (рис. 6).

В табл. 4 и 5 приведены некоторые примеры «грязных» данных, порожденные на разных уровнях – *Schema Level* и *Instant Level*.

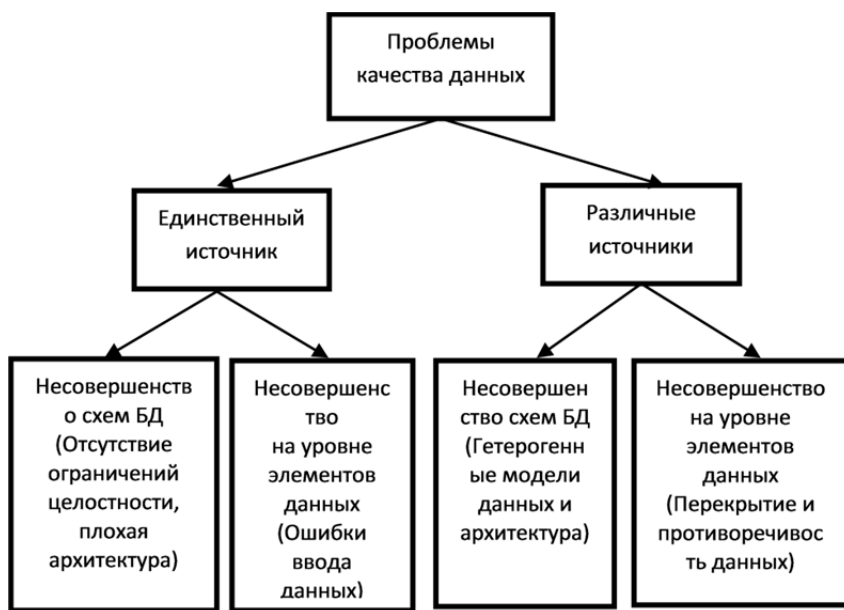


Рис. 6. Пример классификации проблем качества данных в различных источниках

Т а б л и ц а 4

**Примеры «грязных» данных единственного источника
на уровне схемы данных**

	Проблема	«Грязные» данные	Причины
Атрибут	Недопустимые значения	дата рождения=30.13.70	Значение за пределами диапазона
Запись	Нарушение зависимости атрибутов	возраст=22, дата рождения=12.02.70	Возраст = (текущая дата – дата рождения)
Тип записи	Нарушение уникальности	сотр.1=(имя=Иван, SSN=123) сотр.2=(имя=Петр, SSN=123)	SSN должен быть уникальным
Источник	Нарушение ссылочной целостности	сотр.1=(имя=Иван, отд.=789)	Отдела с номером 789 не существует

Таблица 5

Примеры «грязных» данных единственного источника на уровне записей

Причина		«Грязные» данные	Причина
Атрибут	Пропущенное значение	тел.=9999-999999	Недопустимые (некорректные, null и т.п.) значения при вводе
	Орфографические ошибки	город=Тамск город=Москваа	Орфографическая ошибка
	Сокращения и аббревиатуры	должность=А, отдел=ЛТО	
	Объединенные значения	имя=Иван 12.07.70 Томск	Несколько значений в атрибуте
Запись	Нарушение зависимости атрибутов	город=Томск, инд.=666777	Город и индекс не соответствуют друг другу
Тип записи	Дубликаты записей	сотр.1=(имя=Иван, SSN=123) сотр.2=(имя=Иван, SSN=123)	
	Противоречащие записи	сотр.1=(имя=Иван, SSN=123) сотр.1=(имя=Иван, SSN=321)	Записи одного и того же сотрудника с разным SSN
Источник	Неверные ссылки	сотр.=(имя=Иван, отд.=789)	Отдела с номером 789 существует, но указан неверно

Выделяют следующие этапы очистки данных:

1. **Анализ данных** (англ. *Data analysis*). Для того чтобы определить, какие виды ошибок и несоответствий должны быть удалены, требуется детальный анализ данных. В дополнение к инспекции данных или отдельных выборок данных «вручную», следует использовать и метаданные.

2. **Определение способов трансформации потоков данных и правил отображения** (англ. *Definition of transformation workflow and mapping rules*). На данном этапе выполняется оценка количества источников данных, степени их неоднородности и «загрязненности». На основе этой информации создаются схемы потоков

данных, позволяющих преобразовать множество источников данных в один, избегая создания ошибок *Multi-Source* слияния (например, появление дублирующих записей).

3. **Верификация** (англ. *Verification*). Оценка корректности и результативности выполнения предыдущего этапа (например, на небольшой выборке данных). При необходимости производится возврат к этапу 2 для его повторного выполнения.

4. **Трансформация** (англ. *Transformation*). Загрузка данных в единое хранилище с использованием правил трансформации, определенных и отлаженных на этапах 2–3. Очистка данных уровня *Single-Source*.

5. **Обратная загрузка очищенных данных** (англ. *Backflow of cleaned data*). Имея на этапе 4 очищенный набор данных в едином хранилище, целесообразно этими «чистыми» данными заменить аналогичные «грязные» данные в исходных источниках. Это позволит в будущем во многом не выполнять повторно все этапы преобразований по очистке данных.

Реализовать эти этапы можно самыми различными путями с использованием существующих и созданных специально способов и технологий. Рассмотрим наиболее интересные из них.

Этап анализа данных предполагает анализ использования метаданных, которых, как правило, недостаточно для оценки качества данных из имеющихся источников. Поэтому важно анализировать реальные примеры данных, оценивая их характеристики и сигнатуры значений. Это позволяет находить взаимосвязи между атрибутами в схемах данных различных источников. Выделяют два подхода решения этой задачи – *профилирование данных* (англ. *data profiling*) и *извлечение данных* (англ. *data mining*).

Профилирование данных сориентировано на анализ индивидуальных атрибутов, характеризующихся их конкретными свойствами: тип данных, длина, диапазон значений, частота встречаемости дискретных значений, дисперсия, уникальность, встречаемость «*null*» значений, типичная сигнатура записи (например, у телефонного номера). Именно набор подобных свойств (профиль) позволяет оценить различные аспекты качества данных.

Извлечение данных предполагает поиск взаимосвязей между несколькими атрибутами достаточно большого набора данных. Учитывая то, что этот способ получил название *data mining*, здесь используют упоминавшиеся выше (см. табл. 1) методы *кластеризации, подведения итогов, поиска ассоциаций и последовательностей*. Кроме того, для дополнения пропущенных значений, корректировки недопустимых значений или идентификации дубликатов могут быть использованы существующие ограничения целостности (англ. *integrity constraints*), принятые в реляционных базах данных, наложенные дополнительно на бизнес-связи между атрибутами. Например, известно, что «*Total = Quantity × Unit_Price*». Все записи, не удовлетворяющие этому условию, должны быть изучены более внимательно, исправлены или исключены из рассмотрения.

Для разрешения проблем очистки данных в одном источнике (*single-source problems*), в том числе перед его интеграцией с другими источниками данных, реализуют следующие этапы:

– **Извлечение значений из атрибутов свободной формы** (разбиение атрибутов, англ. *Extracting values from free-form attributes (attribute split)*). В данном случае речь может идти о строковых значениях, сохраняющих несколько слов подряд (например, адрес или полное имя человека). В данном случае требуется четкое понимание того, на какой позиции этого значения находится интересующая нас часть атрибута. Возможно, потребуется даже сортировка составных частей такого атрибута.

– **Валидация и коррекция** (англ. *Validation and correction*). Данный этап предполагает поиск ошибок ввода данных и их исправление наиболее автоматическим способом. Например, используя автоматическую проверку правописания во избежание орфографических ошибок и опечаток. Словарь географических названий и почтовых кодов также следует использовать для корректировки значений вводимых адресов. Зависимость атрибутов (дата рождения – возраст, *Total = Quantity × Unit_Price* и т.п.) также способствует избеганию множества ошибок в данных.

– **Стандартизация** (англ. *Standardization*). Этот этап предполагает приведение всех данных к единому универсальному формату.

Примерами таких форматов являются формат написания даты и времени, размер регистра в написании строковых значений. Текстовые поля должны исключать префиксы и суффиксы, аббревиатуры в них должны быть унифицированы, исключены проблемы с различной кодировкой.

Одной из основных проблем, вызванных интеграцией различных источников (*multi-source problems*) данных, является устранение дублирования записей. Этот этап выполняется после подавляющего большинства преобразований и чисток. Он предполагает сначала идентификацию сходных в некотором смысле записей, а затем их слияние с объединением атрибутов. Очевидно, решение этой задачи при наличии у дублирующих записей первичного ключа достаточно просто. Если такого однозначно идентифицирующего признака нет, то задача устранения дубликатов значительно усложняется, требуя применения нечетких (англ. *fuzzy*) подходов сравнения (близости в некотором смысле) записей между собой.

4.2. ОПТИМИЗАЦИЯ ПРИЗНАКОВОГО ПРОСТРАНСТВА

Современные массивы данных, к которым могут быть применены те или иные методы *Data Mining*, могут характеризоваться большим числом признаков, формирующих признаковое пространство большой размерности. Поэтому актуальной является задача снижения размерности такого пространства до размерности, позволяющей без лишних затруднений осуществлять обработку данных и / или их визуализацию. Решение такой задачи называют *оптимизацией признакового пространства* или *поиском значимых признаков* (англ. *Feature Selection*, иногда – *Feature Engineering*). Сегодня это самостоятельная исследовательская задача, которую *решают* различными подходами.

Вместе с тем все эти подходы снижения размерности исходного признакового пространства могут быть разделены на два больших класса.

Первый класс предусматривает трансформацию признакового пространства. Один из наиболее известных и применяемых на практике подходов этого класса – *метод главных компонент*

(МГК, англ. *Principal Component Analysis*) [42, 56]. Рассмотрим кратко его в п. 4.2.1.

Другой класс методов заключается в выборе наиболее информативных, полезных признаков и исключении из рассмотрения неинформативных признаков без трансформации исходного пространства [29, 30]. Здесь применяют различные методы и подходы, с которыми можно подробнее ознакомиться в специальной литературе [43]:

- полного или усеченного перебора;
- ветвей и границ;
- эволюционные;
- со случайным выбором.

Ознакомимся здесь с одним из подходов (п. 4.2.2), который может быть использован при усеченном переборе признаков, в его основе лежит *критерий попарной разделимости Джеффриса-Матуситы* (ДМ) [30].

4.2.1. С трансформацией пространства признаков

Одним из широко используемых традиционных методов решения задачи трансформации исходного пространства признаков в новое является МГК [42]. В основе МГК лежит идея нахождения для исходного набора признаков $\mathbf{x} = \{x_i, i=1, \dots, P\}$ размерности P такого набора скрытых (латентных) переменных $\mathbf{y} = \{y_i, i'=1, \dots, P'\}$ (главных компонент) размерности P' , который бы максимально объяснял дисперсии многомерных переменных \mathbf{x} при выполнении условия $P' < P$.

Главные компоненты представляют собой новое множество исследуемых признаков \mathbf{y} , каждый из которых получен в результате некоторой линейной комбинации исходных признаков \mathbf{x} . Причем полученные в результате преобразования новые признаки \mathbf{y} некоррелированы между собой и упорядочены по степени рассеяния (дисперсии) таким образом, что первый признак обладает наибольшей дисперсией.

В общем случае $[i']$ -й главной компонентой исходного признакового пространства Ω с ковариационной матрицей Σ и вектором

средних μ называется нормированная линейная комбинация компонент исходного P -мерного признакового вектора \mathbf{x} :

$$y_{i'} = l_{i'1}x_1 + l_{i'2}x_2 + \dots + l_{i'P}x_P = L_{i'}\mathbf{x}, \quad (4.1)$$

где $L_{i'} = (l_{i'1}, l_{i'2}, \dots, l_{i'P})$ – $[i']$ -й собственный вектор матрицы Σ пространства Ω .

Геометрическая модель МГК для двумерного случая показана на рис. 7.

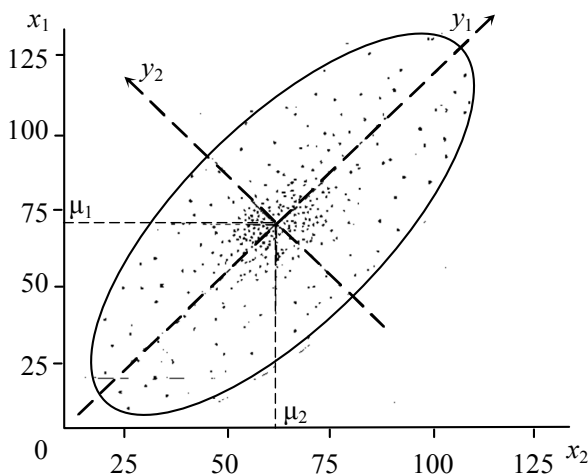


Рис. 7. Пример двумерной модели главных компонент

Множество признаковых векторов $\mathbf{x}^j = (x_1^j, x_2^j)$, $j=1, \dots, N$ располагаются примерно в очертаниях эллипсоида рассеивания, и оси главных компонент y_1 и y_2 проходят вдоль его осей. Обобщенная дисперсия Σ_y и сумма дисперсий $(Dy_1 + Dy_2 + \dots + Dy_{P'})$ главных компонент \mathbf{y} равны обобщенной дисперсии Σ_x и сумме дисперсий $(Dx_1 + Dx_2 + \dots + Dx_P)$ исходных признаков \mathbf{x} . На основании этого выносится решение о том, сколько последних главных компонент P' следует практически без ущерба для информативности изъять из рассмотрения, сократив тем самым размерность исследуемого пространства Ω .

Анализируя изменение относительной доли дисперсии $\Psi(P')$

$$\Psi(P') = \frac{Dy_1 + Dy_2 + \dots + Dy_{P'}}{Dx_1 + Dx_2 + \dots + Dx_P}, (1 \leq P' \leq P), \quad (4.2)$$

вносимой первыми главными компонентами, в зависимости от числа этих компонент, можно разумно определить число компонент P' , которое целесообразно оставить в рассмотрении.

Выбрав предельную величину относительной доли дисперсии $\Psi(P')$, можно задать порог, определяющий количество главных компонент P' , используемых в дальнейшем для классификации.

Несмотря на относительную простоту, МГК обладает двумя существенными недостатками. Во-первых, при использовании МГК предполагается, что распределение исходных многомерных данных подчинено нормальному закону и трансформация происходит относительно многомерного гиперэллипсоида рассеивания (хотя исходные измерения могут быть распределены не в рамках такого гиперэллипсоида). Во-вторых, трансформация исходного признакового пространства может повлечь за собой значительные искажения признакового пространства, что может привести к снижению разделимости в таком новом признаковом пространстве для объектов и снизить итоговое качество классификации.

4.2.2. Без трансформации пространства признаков

Основной идеей метода попарной разделимости ДМ для снижения размерности многомерных данных метода является использование перебора исходного P -мерного набора признаков $\mathbf{x} = \{x_i, i = 1, \dots, P\}$ с целью максимизации заданного критерия информативности и выделения подпространства $\mathbf{y} = \{y_i, i' = 1, \dots, P'\}$ наиболее информативных полезных признаков. При этом в качестве критерия информативности признаков используется критерий интегральной разделимости известного набора классов $\{\omega_i, i = 1, \dots, M\}$, определяемых соответствующими обучающими выборками $\{V_i, i = 1, \dots, M\}$. Интегральная разделимость вычисляется как средневзвешенное значение попарных классовых разделимостей, определяемых расстоянием ДМ (JM) [30], характеризующим среднее расстояние меж-

ду функциями условных плотностей распределения $p(\mathbf{x}|\omega_i)$ и $p(\mathbf{x}|\omega_j)$ соответствующей пары классов ω_i и ω_j . Расстояние JM_{ij} между классами ω_i и ω_j определяется следующим выражением:

$$JM_{ij} = \int_{\mathbf{x}} \{\sqrt{p(\mathbf{x}|\omega_i)} - \sqrt{p(\mathbf{x}|\omega_j)}\}^2 d\mathbf{x}. \quad (4.3)$$

Если в выражении (4.3) использовать предположение о нормальном распределении условных плотностей вероятности распределения признаков \mathbf{x} , то расчет критерия ДМ может быть представлен в виде

$$JM_{ij} = 2(1 - e^{-B}), \quad (4.4)$$

где B – расстояние Бхаттачария [30], зависящее от параметров двух многомерных нормальных распределений $N_i(\mathbf{x}|\mu_i, \Sigma_i)$ и $N_j(\mathbf{x}|\mu_j, \Sigma_j)$, определяемых вектором средних μ и ковариационной матрицей Σ как

$$B = \frac{1}{8(\mu_i - \mu_j)^T \left\{ \frac{\Sigma_i + \Sigma_j}{2} \right\}^{-1} (\mu_i - \mu_j)} + \frac{1}{2 \ln \left\{ \frac{|(\Sigma_i + \Sigma_j)/2|}{|\Sigma_i|^{\frac{1}{2}} |\Sigma_j|^{\frac{1}{2}}} \right\}}. \quad (4.5)$$

Для нахождения интегральной метрики разделимости JM_{ave} всех классов признакового пространства используют вычисления с использованием элементов симметричной матрицы $\|\mathbf{J}\mathbf{M}\|$ попарных разделимостей, расположенных выше главной диагонали:

$$JM_{ave} = \sum_{i=1}^M \sum_{j=i+1}^M p(\omega_i) p(\omega_j) JM_{ij}, \quad (4.6)$$

где $p(\omega_i)$ и $p(\omega_j)$ – весовые коэффициенты (априорные вероятности) классов ω_i и ω_j соответственно. Функция JM_{ij} – расстояния (4.3) асимптотически сходится к значению 2.0. Если элемент матрицы $JM_{ij} = 2.0$, то это означает, что классы ω_i и ω_j абсолютно разделимы и вероятность их перепутывания равна нулю.

Для снятия ограничения на согласованность распределения с гауссовым в распознаваемых классах вместо оценок (4.4) и (4.5) для расчета попарных классовых делимости JM_{ij} используется оценка (4.3). В качестве метода численного интегрирования для нахождения значений оценки (4.3) применяется метод Монте-Карло вычисления кратных определенных интегралов.

В силу высокой вычислительной сложности нахождения значений оценки (4.3) задача полного перебора комбинаций исходных признаков (число сочетаний из P по $P' - C_{P'}^P$) заменяется усеченным перебором, основанном на отыскании информативных поднаборов (1–2 признака) в исходном P -мерном наборе признаков x из условия максимума критерия JM_{ave} (4.6). Каждый найденный поднабор признаков добавляется в информативный набор y и критерий JM_{ave} проверяется для всех попавших в набор y компонент. Усеченный перебор исходных признаков из x завершается по достижению приемлемого значения критерия JM_{ave} (обычно 1.9–2.0) или по завершению перебора всех исходных признаков x .

Несмотря на большую по сравнению с МГК вычислительную сложность, данный метод не накладывает ограничения на вид распределения исходных признаков классифицируемых объектов и не вносит дополнительные искажения в признаковую структуру пространства.

4.3. КЛАССИФИКАЦИЯ

4.3.1. Постановка задачи классификации

Задачу классификации математически в общем случае можно представить следующим образом [56]. Для каждого класса ω_i из исходного алфавита $\{\omega_i, i = 1, \dots, M\}$ (M – количество предопределенных типов) введем понятие вероятности появления класса ω_i в пространстве признаков $\Omega(x)$. Данная вероятность $p(\omega_i)$ называется *априорной вероятностью* класса ω_i . Также предположим, что для

каждого класса ω_i известна многомерная (P -мерная) функция $p(\mathbf{x} | \omega_i)$, описывающая условную плотность распределения (УПР) вектора признаков \mathbf{x} в классе ω_i , для которой справедливо

$$\int_{\omega_i} p(\mathbf{x} | \omega_i) d\mathbf{x} = 1.$$

Априорная вероятность $p(\omega_i)$ и УПР $p(\mathbf{x} | \omega_i)$ являются наиболее полными вероятностными характеристиками класса ω_i .

Таким образом, задача классификации может быть сформулирована в виде задачи статистических решений (испытание M статистических гипотез) с помощью определения *дискриминантной функции* $\phi(\mathbf{x})$, принимающей значение ϕ_i в случае, когда принимается гипотеза $H_i: \mathbf{x} \in \omega_i$. Полагается, что принятие классификатором решения ϕ_i , когда в действительности входной образ принадлежит к классу ω_j , приводит к потере, определяемой *функцией потерь* $L(\phi_i | \omega_j)$. Тогда условный риск $R(\phi_i | \mathbf{x})$ принятия решения ϕ_i в случае $\mathbf{x} \in \omega_j$ находится как

$$R(\phi_i | \mathbf{x}) = \sum_{j=1}^M L(\phi_i | \omega_j) p(\omega_j | \mathbf{x}), \quad (4.7)$$

где $p(\omega_j | \mathbf{x})$ носит название *апостериорной вероятности* события $\mathbf{x} \in \omega_j$ и вычисляется исходя из априорной вероятности $p(\omega_i)$ и условной плотности распределения $p(\mathbf{x} | \omega_i)$ согласно теореме Байеса [56] следующим образом:

$$p(\omega_j | \mathbf{x}) = \frac{p(\omega_j) p(\mathbf{x} | \omega_j)}{\sum_{k=1}^M p(\omega_k) p_k(\mathbf{x} | \omega_k)}. \quad (4.8)$$

Задача классификации сводится к выбору наименьшего условного риска (4.7), т.е.

$$\phi(\mathbf{x}) = \phi_i: (\mathbf{x} \in \omega_i), \text{ если } R(\phi_i | \mathbf{x}) < R(\phi_j | \mathbf{x}), \forall i \neq j. \quad (4.9)$$

Правило классификации (4.9) носит название *байесовского решающего правила классификации*. При использовании в (4.7) нуль-единичной функции потерь

$$L(\phi_i | \omega_j) = \begin{cases} 0, i = j \\ 1, i \neq j \end{cases}, i, j = 1, \dots, M, \quad (4.10)$$

риск, соответствующий такой функции, является средней вероятностью ошибки (ложной классификации) и, исходя из (4.7) и (4.10), определяется как

$$R(\phi_i | \mathbf{x}) = \sum_{j \neq i} p(\omega_j | \mathbf{x}) = 1 - p(\omega_i | \mathbf{x}), i, j = 1, \dots, M. \quad (4.11)$$

Исходя из (4.9) и (4.11) дискриминантная функция $\phi_i(\mathbf{x})$ при использовании нуль-единичной функции потерь (4.10) выглядит следующим образом:

$$\phi_i(\mathbf{x}) = p(\omega_i) p(\mathbf{x} | \omega_i), i, j = 1, \dots, M, \quad (4.12)$$

и согласно (4.8) и (4.12) байесовское решающее правило (4.9) можно записать

$$m(\mathbf{x}): \mathbf{x} \in \omega_i, \text{ если } p(\omega_i) p(\mathbf{x} | \omega_i) > p(\omega_j) p(\mathbf{x} | \omega_j), \forall i \neq j. \quad (4.13)$$

В *параметрическом подходе* к классификации при оценке УПР $p(\mathbf{x} | \omega_i)$ принимается гипотеза о некотором известном виде плотности распределения признаков (например, гауссовом), что позволяет использовать для нахождения $p(\mathbf{x} | \omega_i)$ ее параметрическую оценку. Следует отметить, что УПР называют *правдоподобием*, а такой подход к классификации – методом *максимального правдоподобия* (англ. *maximut likelihood*, *ML*). В случае гауссова распределения используется оценка вида [30]

$$\hat{p}(\mathbf{x} | \omega_i) = (2\pi)^{-P/2} |\Sigma_i|^{-1/2} \exp\left\{-1/2(\mathbf{x} - \hat{\mu}_i)^t \Sigma_i^{-1}(\mathbf{x} - \hat{\mu}_i)\right\}, i = 1, \dots, M, \quad (4.14)$$

где μ_i – выборочный вектор средних типа ω_i ; Σ_i – выборочная ковариационная матрица типа ω_i ; P – количество признаков; $|\Sigma_i|$ –

детерминант выборочной ковариационной матрицы типа ω_i ; Σ_i^{-1} – обратная выборочная ковариационная матрица типа ω_i [56, 96]. При этом вычисление априорной вероятности $p(\omega_i)$ в выражении (4.8) производится с помощью простых способов, в которых $p(\omega_i)$ принимается равной для всех классов либо пропорциональной

$$p(\omega_i) = p(\omega_j), \forall i, j = 1, \dots, M, \quad (4.15)$$

размеру имеющихся обучающих данных

$$p(\omega_i) = \frac{N_i}{\sum_{k=1}^M N_k}, \quad i = 1, \dots, M, \quad (4.16)$$

где N_k – размер обучающей выборки, соответствующий типу ω_k . Классификацию, в основе которой используется байесовское решающее правило (4.8), причем вне зависимости от вида принимаемой оценки УПР (параметрическая или непараметрическая), называют *байесовской* (англ. *Bayes* или *Naive Bayes*²⁰).

Таким образом, основой большинства современных классификаторов является теорема Байеса, а их математический аппарат может быть реализован с использованием различных подходов. Среди них выделяют *непараметрические статистические классификаторы*, включающие простые, такие как классификатор по правилу параллелепипеда, и значительно более сложные, использующие в своей основе непараметрическое оценивание УПР признаков [48, 60, 84, 100]. При классификации оценка УПР в выражении (4.8) на основе непараметрического подхода характеризуется меньшей чувствительностью к статистическим характеристикам данных, эффективна с точки зрения точности классификации при произвольном (неизвестном) распределении признаков, в том числе и отличном от нормального (гауссова). Однако непараметрические классификаторы требуют перебора всех значений обучающей

²⁰ «Наивным» байесовский классификатор называют из-за предположения о независимости признаков вектора \mathbf{x} между собой.

выборки при оценке УПР для каждого x , поэтому их отличают высокие вычислительные затраты. Рассмотрим более подробно этот подход в § 4.3.2.

Также к непараметрическим относят *нейросетевые классификаторы*, основанные на использовании аппарата искусственных нейронных сетей (ИНС) [87]. Однако в этом случае, как правило, не осуществляется оценка УПР. Для краткости изложения такие классификаторы называют *нейросетевыми классификаторами*, а искусственные нейронные сети – просто *нейросетями*. Рассмотрим их более подробно в § 4.3.3.

В последнее время набирают популярность более математически сложные классификаторы с использованием *машины опорных векторов* (англ. *support vector machine* – *SVM*) [6, 30]. Эти классификаторы характеризуются достаточно высокой устойчивостью к статистическим характеристикам исходных векторов признаков [34, 49]. При этом классификаторы *SVM* по сравнению с нейросетевыми классификаторами для задач классификации в пространстве большой размерности выглядят предпочтительнее – при практической реализации они вместо многоэкстремальной задачи (с вероятностью попадания в локальный экстремум) решают задачу квадратичного программирования, имеющую единственное решение. Кроме того, разделяющие поверхности решения классификаторов *SVM* обладают более высокими дифференцирующими (разделяющими) возможностями за счет максимизации ширины разделяющей полосы [34, 49]. Это относит такие классификаторы к перспективным с точки зрения вычислительной эффективности и точности. Детали построения таких классификаторов изложены в п. 4.3.4.

4.3.2. Контролируемая непараметрическая классификация

Как отмечено выше, для проведения классификации признаков, распределение которых априори не известно или не согласовано с нормальным законом распределения, используют различные подходы к *непараметрической оценке плотности распределения*. Среди них наиболее широкое распространение получил подход к

оценке УПР по методу k -го ближайшего соседа (для краткости будем приводить англоязычную аббревиатуру названия метода – k -NN), которая определяется исходя из выражения [100]:

$$p(\mathbf{x} | \omega_i) = \frac{1}{N} \frac{k_p - 1}{V(k_p, N, \mathbf{x})}, i = 1, \dots, M, \quad (4.17)$$

где k_p – параметр близости соседа; N – величина выборки, $V(k_p, N, \mathbf{x})$ – объем множества всех элементов обучающей выборки, расстояние которых до точки \mathbf{x} в P -мерном пространстве меньше или равно R_k^P . В случае использования евклидова расстояния

$$V(k_p, N, \mathbf{x}) = \frac{\pi^{P/2} R_k^P}{|\mathbf{A}|^{1/2} \Gamma[(P+2)/2]}, \quad (4.18)$$

где Γ – гамма-функция, \mathbf{A} – единичная матрица.

В выражении (4.17) величина k_p является параметром, при этом существует ряд методик нахождения ее оптимального значения $k_{\text{опт}}$ [100]. К сожалению, поиск значения $k_{\text{опт}}$ ведет к увеличению вычислительной сложности алгоритмов оценки УПР, что затрудняет их использование при решении практических задач. Поэтому на практике значение k_p часто принимают фиксированным (например, 1, 3, 21, 87, \sqrt{N} , где N – размер выборки [17, 21, 22, 100]). При этом очевидно, что большее значение k_p требует большего количества операций по расчету расстояния R_k^P в (4.18), что ведет к дополнительным вычислительным затратам при классификации. Поэтому, учитывая важность проведения непараметрической классификации с высокой вычислительной эффективностью, это значение принимают фиксированным (например, $k_p = 3$).

Выше отмечено, что более широкому использованию непараметрических подходов к оценке плотности распределения препятствует их низкая вычислительная эффективность, связанная с необходимостью перебора всех значений обучающей выборки для

оценки УПР в точке x P -мерного пространства. Поэтому задача повышения вычислительной эффективности оценки УПР в непараметрических методах оценки плотности для решения практических задач классификации является отдельной, интенсивно разрабатываемой областью исследований [65, 86].

4.3.3. Контролируемая непараметрическая нейросетевая классификация

На сегодняшний день нейросетевой аппарат достаточно широко применяется при решении самых различных задач классификации, прогнозирования, оценки плотности распределения, поэтому приведем лишь некоторые необходимые пояснения, связанные с областью ИНС [6, 87]. Так, широко используется такое понятие, как *формальный нейрон*, представляющий собой упрощенную математическую абстракцию биологического нейрона (рис. 8).

Каждый такой нейрон обладает группой синапсов – односторонних входных связей, соединенных с выходами других нейронов, а также имеет аксон – выходную связь данного нейрона, с которой сигнал может поступать на синапсы других нейронов. Каждый синапс характеризуется величиной синаптической связи или ее весом w_i . Кроме того, важной характеристикой любого формального нейрона является *функция активации* или *пороговая функция* $f(S)$. Наиболее распространенными пороговыми функциями являются сигмоидные функции типа гиперболического тангенса и логистической функции [85].

Наиболее популярными и широко распространенными, в том числе и для решения задач контролируемой классификации, являются нейросети прямого распространения – *многослойные перцептроны* [85]. Они позволяют работать с данными произвольного распределения и учитывать такие закономерности в данных, которые затруднительно учесть другими методами [6].

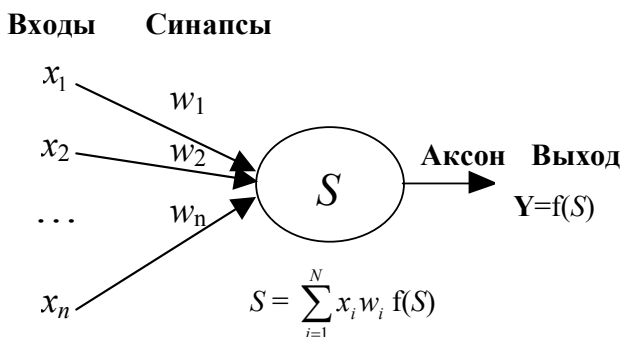


Рис. 8. Схема формального нейрона

Потенциально нейросетевые классификаторы обладают рядом существенных преимуществ по сравнению с традиционными статистическими классификаторами. Так, в качестве входных данных для них можно использовать трудноформализуемые взаимозависимые факторы произвольного распределения. Нейросетевые классификаторы учитывают такие закономерности в данных, которые порой не могут быть учтены никаким другим классификатором [6, 87]. При этом точность классификации нейросетевых классификаторов высока и приближается к байесовской [85].

Несмотря на очевидные достоинства нейросетевого подхода при классификации, существует ряд проблем, требующих решения [6, 87]. Одной из основных проблем является необходимость экспертного обучения нейросети. При этом требуется решения задачи определения оптимальных параметров, задающих структуру нейросети, а также параметров ее обучения. Решение этой задачи для данных различной природы может иметь длительный итерационный характер, что для конечного пользователя может оказаться неприемлемым.

Существуют некоторые сложности практического применения нейросетей при решении практических задач классификации. В частности, применение многослойного персептрона (далее просто нейросети) связано с решением одной из таких задач, заклю-

чающейся в определении его оптимальной топологии – количества слоев и элементов (нейронов) в них. Именно правильно выбранная топология во многом определяет перспективность использования нейросетей в том или ином случае. Минимально необходимое количество элементов нейросети позволит быстро ее обучить и получить точные результаты ее применения. Однако задача определения топологии нейросети является сложной и окончательно до сих пор не решена. В [36] предлагается неравенство для оценки числа синаптических связей N_w в виде

$$\frac{N_y N_p}{1 + \log_2(N_p)} \leq N_w \leq N_y \left(\frac{N_p}{N_x} + 1 \right) (N_x + N_y + 1) + N_y, \quad (4.19)$$

где N_y – размерность выходного вектора, N_p – число примеров обучающей выборки, N_x – размерность входного вектора. Для практически значимого варианта нейросети с одним скрытым слоем, число нейронов N в нем можно определить как

$$N = \frac{N_w}{N_x + N_y}. \quad (4.20)$$

При использовании выражений (4.19) и (4.20), число нейронов скрытого слоя, как правило, получается бóльшим, чем выбранное при решении практических задач эмпирически, поэтому предлагается использовать это число только в качестве рекомендации, а окончательное решение по параметрам нейросети принимать эмпирически.

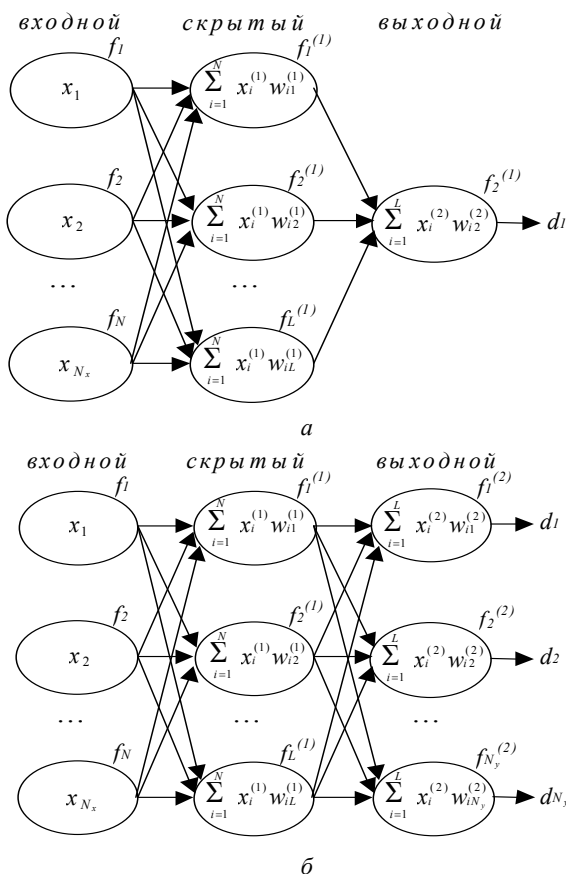


Рис. 9. Нейросетевые топологии: *a* – один выход; *б* – несколько выходов

Отметим два наиболее широко используемых на практике способа определения числа нейронов в выходном слое:

1. Число нейронов равно одному (рис. 9, *a*). Выход интерпретируется как вероятность принадлежности к конкретному типу (классу).

2. Число нейронов более одного. Например, оно может быть равно количеству классов (рис. 9, *б*) или представлять собой неко-

торый вектор, значения которого интерпретируется в более сложной процедуре использования нейросети при оценке плотности распределения [35].

Важным и необходимым этапом практического использования любой нейросети является процесс ее обучения. Обучение нейросети в общем случае представляет собой поиск глобального минимума многомерной целевой функции путем «исследования» нейросетью многомерного пространства выборочных обучающих данных и подстройкой w_i и параметров функций активации f [6, 87].

Обучающие данные представляют собой множество входных векторов $X = \{X_i, i = 1, \dots, N_p\}$ и множество известных выходных векторов $A = \{A_i, i = 1, \dots, N_p\}$, где $X_i = \{x_1, x_2, \dots, x_{N_x}\}$ и $A_i = \{a_1, a_2, \dots, a_{N_y}\}$. В процессе обучения минимизируется значение среднеквадратической ошибки (СКО), вычисляемой согласно выражению [87]

$$E = \frac{1}{2} \sum_{i=1}^M (A_i - Y_i)^2, \quad (4.21)$$

где $Y_i = \{y_1, y_2, \dots, y_{N_y}\}$ – фактические значения, получаемые с выходов нейросети.

При программной реализации моделей нейросетей и их использовании наиболее часто в качестве алгоритма обучения применяют градиентный *алгоритм обратного распространения ошибки* или его модификации [36]. Этот алгоритм обладает устойчивой сходимостью и приемлемыми требованиями к ресурсам вычислительной техники.

4.3.4. Классификация по методу машины опорных векторов

Основная идея метода *машины опорных векторов* (*SVM* классификатора) – отображение исходных векторов в пространство более высокой размерности и поиск разделяющей гиперплоскости с максимальным зазором в этом пространстве [34]. Суть работы стандартного классификатора *SVM* для случая двух классов можно представить с использованием следующего выражения:

$$f(\mathbf{x}, \mathbf{W}) = \text{sign}(g(\mathbf{x}, \mathbf{W})), \text{ где } g(\mathbf{x}, \mathbf{W}) = \langle \mathbf{x}, \mathbf{W} \rangle + b,$$

где параметры \mathbf{W} (вектор весов) и b (свободный коэффициент) определяются процедурой обучения. Границы решения классификатора $g(\mathbf{x}, \mathbf{W}) = 0$ представляют собой гиперплоскость порядка $L - 1$ в L -мерном пространстве (рис. 10).

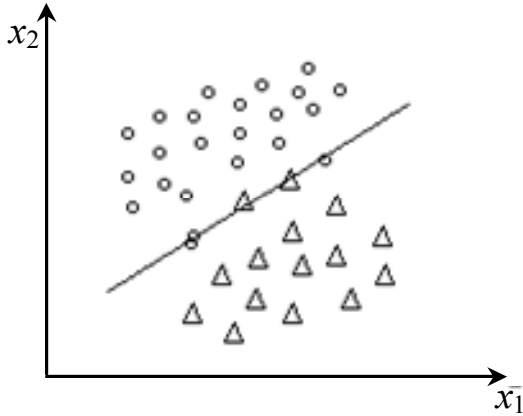


Рис. 10. Иллюстрация работы классификатора *SVM* для двумерного пространства

Для определения параметров \mathbf{W} и b решается задача квадратичной оптимизации:

$$\begin{cases} -L(\lambda) = -\sum_{i=1}^{\ell} \lambda_i + \frac{1}{2} \sum_{i=1}^{\ell} \sum_{j=1}^{\ell} \lambda_i \lambda_j y_i y_j \langle x_i, x_j \rangle \rightarrow \min_{\lambda} \\ 0 \leq \lambda_i \leq C, i = 1, \dots, \ell \\ \sum_{i=1}^{\ell} \lambda_i y_i = 0 \end{cases},$$

где y – вектор меток классов пикселей обучающих выборок $y_i = \{1; -1\}$; C – управляющая константа метода решения задачи; ℓ – размер обучающих выборок. Для ее решения применяют последо-

вательный метод активных ограничений (англ. *incremental active set method, INCAS*) [14]. Решение этой задачи даст возможность найти вектор двойственных переменных $\lambda = (\lambda_1, \dots, \lambda_n)$, что в свою очередь позволяет вычислить параметры алгоритма \mathbf{W} и b как

$$\mathbf{W} = \sum_{i=1}^{\ell} \lambda_i y_i x_i, b = \langle \mathbf{W}, x_i \rangle - y_i, \lambda_i > 0, i = 1 \dots \ell$$

Решение задачи классификации для случая нескольких классов может быть реализовано путем множественной попарной классификации и объединения результатов (например, по мажоритарному правилу) [34]. Классификатор *SVM* отличается достаточно высокой алгоритмической сложностью, но при этом обладает высокой вычислительной эффективностью. Кроме того, его отличает высокая точность и робастность результатов при различных статистических характеристиках обучающих данных.

4.3.5. Деревья решений

Деревья принятий решений (деревья решений, англ. *decision trees*) – еще один распространенный метод контролируемой классификации, позволяющий обеспечивать поддержку принятия решений. В основе этого метода классификации лежит использование ориентированного²¹ дерева как связного ациклического графа²² (рис. 11).

Дерево решений имеет одну вершину (корень), а завершается вершинами с нулевой степенью исхода (из них не исходит ни одна дуга) – *листьями*. При этом принято, что дерево решений растет вниз (а не вверх, как настоящее дерево). Кроме того, дерево решений имеет различные метки:

²¹ Только одна вершина имеет степень захода 0 (в нее не ведут дуги – *корень дерева*), а все остальные вершины имеют степень захода 1 (в них ведет только по одной дуге).

²² *Связность* – наличие путей между любой парой вершин, *ациклическость* – отсутствие циклов и наличие между парами вершин единственного пути.

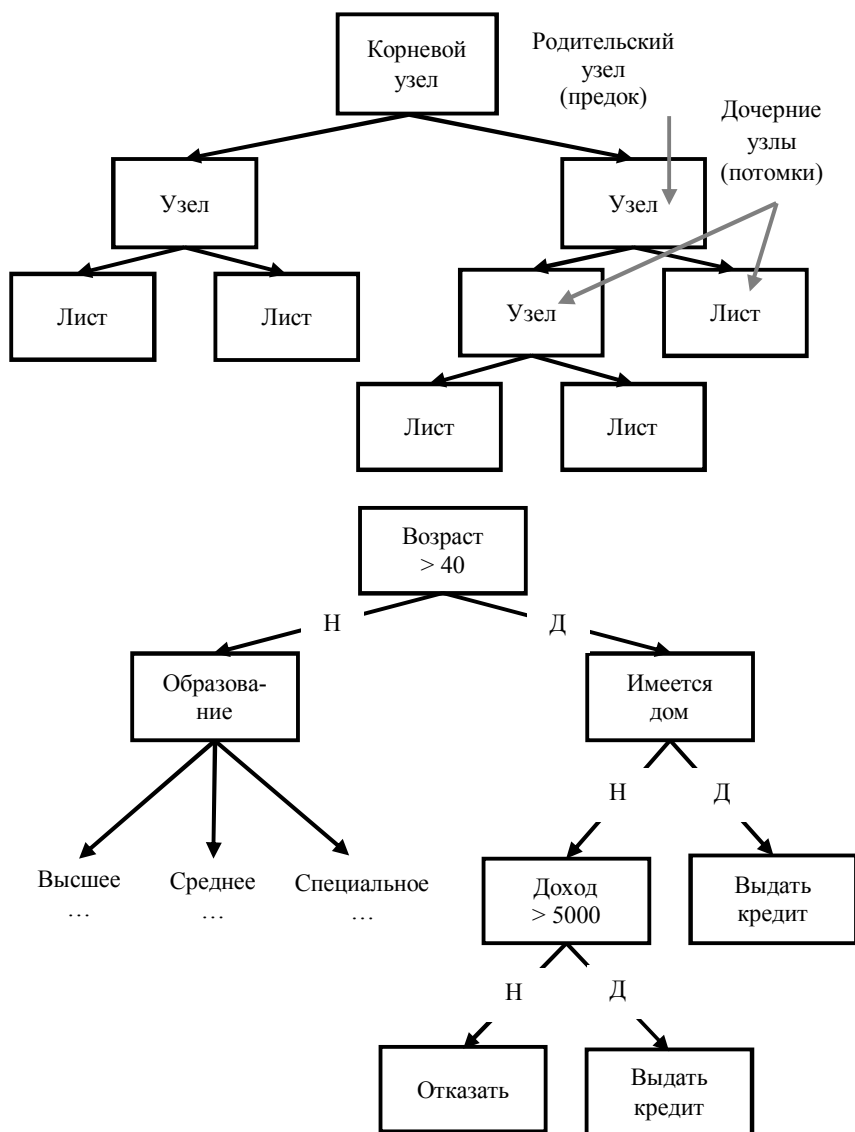


Рис. 11. Примеры ориентированного дерева

– *узлы* (вершины, не являющиеся листьями) – *переменные* набора данных;

– на *дугах* (ветвях) отмечают *атрибуты* (значения переменных), от которых зависит целевая функция;

– в листьях отмечают значения целевой функции.

Если все узлы дерева имеют по две дуги, то такое дерево называют *бинарным*.

В общем случае, для решения задачи классификации необходимо спуститься по дереву от вершины до листа, выполняя соответствующие действия в узлах и выбирая при этом соответствующую дугу. Такая иерархическая структура позволяет реализовать способ представления правил классификации, основанный в каждом узле дерева на логической конструкции «что-если».

Область применения деревьев решений обширна (в банковском деле – при оценке кредитоспособности клиентов, в промышленности – при контроле качества, в медицине – при диагностике заболеваний и т.п.) и может быть разделена на три класса:

– *Описание данных* (деревья не только позволяют сохранять информацию об исходных данных в компактной форме, но и могут быть логически интерпретируемыми).

– *Классификация* (возможна в случае дискретных значений целевой переменной).

– *Регрессия* (если целевая переменная имеет непрерывные значения, может быть установлена ее зависимость от независимых входных переменных; например, в задаче численного прогнозирования).

Построение дерева. Пусть имеется обучающая выборка примеров $T = \{\mathbf{x}_i, f(\mathbf{x}_i) = \omega_i\}$, где \mathbf{x}_i – переменные, каждой из которых соответствуют некоторый набор атрибутов (атрибут – условие перемещения по дуге) $\mathbf{Q}_i = \{\mathbf{Q}_j, j = 1..q\}$, а ω_i – классы, которым принадлежат переменные.

Если переменная, которая проверяется в узле, принимает категориальные значения, то каждому возможному значению соответствует ветвь, выходящая из узла дерева. Если значением переменной является число, то проверяется, больше или меньше это зна-

чение некоторой константы. Иногда область числовых значений разбивают на интервалы и выполняют проверку попадания значения в один из интервалов.

Для классификации методом дерева решений следует разбить множество T на некоторые подмножества. Для этого выбирается один из признаков x , имеющий два и более отличных друг от друга значений x_1, x_2, \dots, x_n . T разбивается на подмножества T_1, T_2, \dots, T_n , где каждое подмножество T_i содержит все примеры, имеющие значение $f(x = x_i)$ для выбранного признака. Эта процедура будет рекурсивно продолжаться до тех пор, пока конечное множество не будет состоять из примеров, относящихся к одному и тому же классу. Фиксируя эти преобразования в виде элементов дерева решений, будет выполнено его построение сверху вниз.

Другими словами, построение дерева решений по известным обучающим данным с использованием указанных обозначений выглядит следующим образом:

1. Выбрать переменную x_i , поместив ее в корень дерева (x_i имеет n значений, что позволяет разбить T на n подмножеств).

2. Далее создаются n потомков корня, каждому из которых поставлено в соответствие свое подмножество, полученное при разбиении T .

3. Повторять рекурсивно для всех i шаги 1 и 2, пока:

- в вершине не окажутся примеры из одного класса (тогда она становится листом, а класс, которому принадлежат ее примеры, будет решением листа);

- вершина оказалась ассоциированной с пустым множеством (тогда она становится листом, а в качестве решения выбирается наиболее часто встречающийся класс у непосредственного предка этой вершины).

Такой процесс построения дерева «сверху вниз» является примером наиболее распространенного поглощающего «жадного» алгоритма. Рассмотрим конкретный пример построения дерева решений для некоторого набора данных.

Предположим, нам известна некоторая статистика игры футбольной команды, а мы на ее основе хотим предсказать исход следующего матча (табл. 6).

Т а б л и ц а 6

Статистика игр футбольной команды

x_1	x_2	x_3	x_4	$f(x)$
<i>Соперник по турнирной таблице</i>	<i>Место проведения</i>	<i>Наличие лидеров команды</i>	<i>Погодные условия</i>	<i>Результат</i>
Выше	Дома	На месте	Осадки	Поражение
Выше	Дома	На месте	Без осадков	Победа
Выше	Дома	Пропускают	Без осадков	Победа
Ниже	Дома	Пропускают	Без осадков	Победа
Ниже	В гостях	Пропускают	Без осадков	Поражение
Ниже	Дома	Пропускают	Осадки	Победа
Выше	В гостях	На месте	Осадки	Поражение
Ниже	В гостях	На месте	Без осадков	?

Построим дерево решений, выбрав в качестве корневого узла переменную x_1 , а остальные атрибуты выберем по порядку упоминания в таблице (рис. 12). Очевидно, чем меньше глубина²³ построенного дерева, чем меньше в нем узлов и ветвей, тем им легче оперировать на практике.

Теперь построим аналогичное дерево решений, но порядок появления узлов зададим иначе (рис. 13).

Очевидно, полученное дерево (рис. 13) существенно проще, а его глубина в два раза меньше, чем дерева на рис. 12. При этом оно позволяет решать ту же задачу прогнозирования результатов матча по известным параметрам. Это означает, что аналогичное дерево решений может быть построено для одних и тех же исходных данных различными способами, выбирая очередность для той или иной переменной x_i и ее атрибутов.

Поэтому именно критерий выбора очередной переменной при построении дерева решений отличает наиболее распространенные алгоритмы построения деревьев решений:

²³ Максимальный уровень листа дерева – длина самого длинного пути от корня к листу.

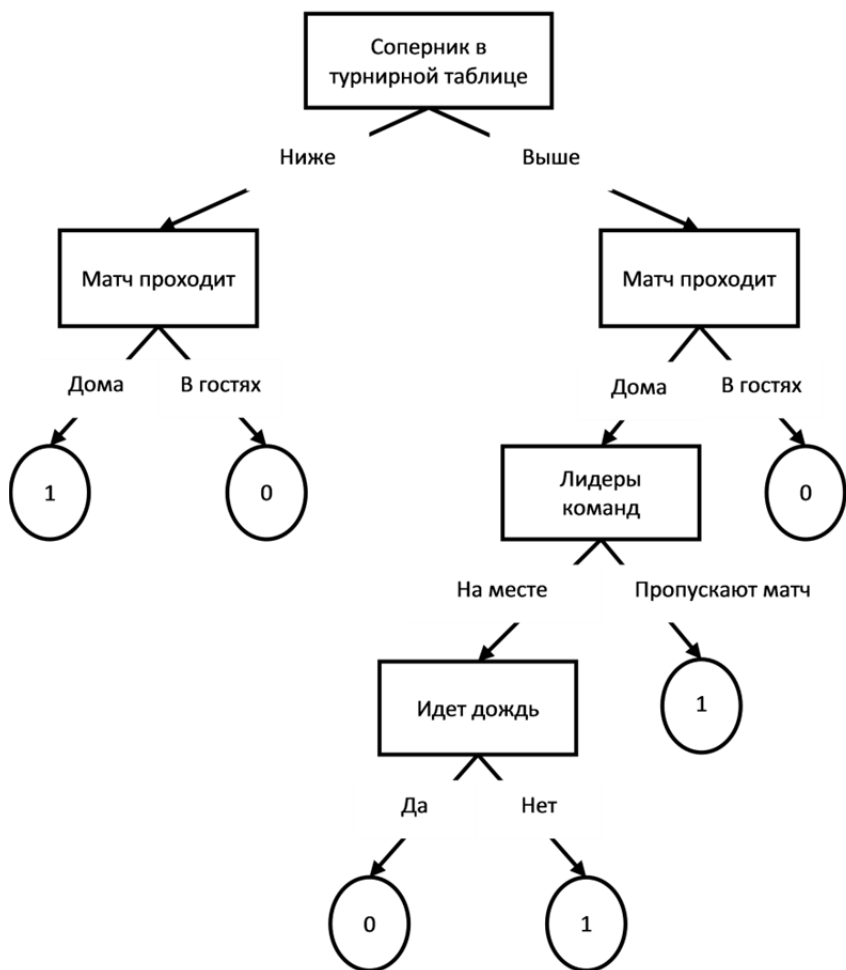


Рис. 12. Пример дерева решения с корнем x_1

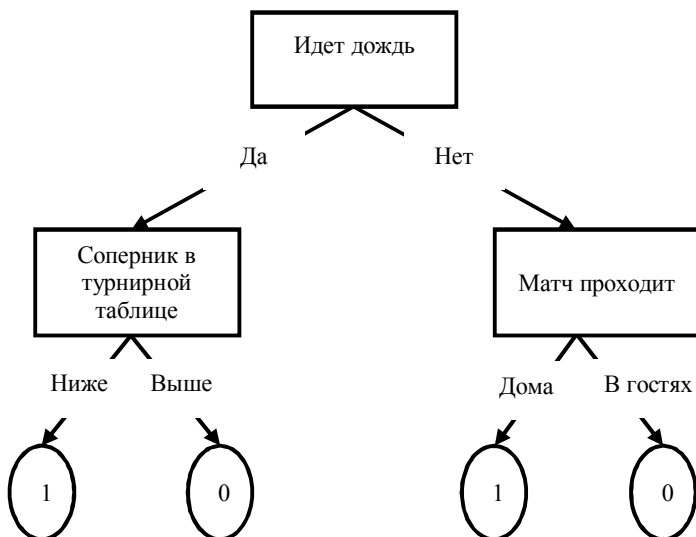


Рис. 13. Пример дерева решения с корнем x_4

– *Алгоритм ID3* (выбор атрибута происходит на основании *прироста информации* (англ. *Gain*) либо на основании индекса Джини (англ. *Gini*)).

– *Алгоритм C4.5* (модифицированная версия алгоритма *ID3*, выбор атрибута происходит на основании *нормализованного прироста информации* (англ. *Gain Ratio*)).

– *Алгоритм CART* (для бинарных деревьев, выбор атрибута зависит от отношений числа примеров в правом и левом потомке к общему числу примеров).

Следует отметить, что задача построения наилучшего, оптимального дерева не может быть решена данными методами и требует построение всех возможных вариантов деревьев решений и полного их перебора (т.е. является *NP*-полной) [19].

Достоинства и недостатки. Использование деревьев решений в сравнении с другими распространенными методами классификации, прогнозирования или регрессионного анализа имеет несколько существенных достоинств:

– *Не требуется предварительная обработка данных.* Например, не требуются нормализация, добавление фиктивных переменных, удаление пропущенных данных. Метод способен работать как с категориальными, так и с интервальными переменными.

– *Имеется возможность интуитивного понимания и интерпретации.* Правила передвижения по узлам деревьев могут иметь четкую логику («белый ящик») и могут быть формализованы даже там, где это затруднительно сделать эксперту. Отметим, что нейросеть такой возможности пользователю, как правило, не предоставляет («черный ящик»).

– *Высокая надежность и точность.* Сравнительно высокая надежность и точность модели могут быть статистически оценены. Отсутствие априорных предположений о законах распределения данных относит их к непараметрическим, устойчивым к данным различных априори неизвестных законов распределения.

– *Высокая вычислительная эффективность.* Метод отличают быстрый процесс обучения (построения дерева), а также высокая вычислительная эффективность обработки значительных объемов данных за счет простоты структуры данных дерева решений.

Вместе с важными достоинствами метода отмечают и его недостатки, некоторые из которых упомянуты выше:

– *Проблематичность построения оптимального дерева решений.* Построение и поиск такого дерева решений являются *NP*-полной задачей, сложно разрешимой на практике. Поэтому практическое построение деревьев решений связано с применением эвристических «жадных» алгоритмов, оптимальных только в каждом узле дерева, но не оптимальных для дерева в целом. При этом требуется обеспечить непростой баланс между точностью и сложностью дерева, уделять внимание опасности переобучения, для чего применять дополнительные алгоритмы регулирования глубины или упрощения дерева (англ. *pruning, reduction*).

– *Вероятность построения избыточно сложного дерева.* Возможны случаи, при которых применение традиционных алгоритмов построения дерева решений приведет к описанию модели «сложным» путем и непомерно большому дереву (например, когда

число возможных атрибутов велико, а не просто «да» / «нет») [18]. В этом случае потребуется дополнительная проработка постановки задачи и формирование иных суждений о предметной области.

– *Вероятность ошибок при построении дерева.* Ключевым элементом алгоритма построения дерева является порядок выбора очередной переменной при построении очередного узла. В случае, если набор исходных данных включает категориальные переменные, больший информационный вес априори присваивается тем переменным, которые имеют большее количество уровней [82].

Таким образом, метод с использованием деревьев решений более применим для задач с дискретными (категориальными) значениями с четким набором отличных атрибутов, а также в тех случаях, где важно понимать логику получения и интерпретации результатов.

Примеры алгоритмов. Выше изложены базовые принципы построения деревьев решений, на которых основано подавляющее большинство применяемых на практике алгоритмов. Основное отличие таких алгоритмов друг от друга заключается в способах определения очередности для той или иной переменной и ее атрибутов при построении очередного узла дерева.

Очевидно, наиболее удачным будет считаться атрибут, который позволит получить такие подмножества данных, которые будут принадлежать в подавляющем большинстве элементов к одному классу.

Алгоритм ID3. Алгоритм ID3 (англ. *Iterative Dichotomizer*, итеративный дихотомайзер), предложенный Д. Куинланом, определяет очередность переменной и ее атрибутов через их информационную значимость (информационную энтропию) [27]. Для этого следует найти энтропию всех неиспользованных признаков и их атрибутов относительно тестовых экземпляров и выбрать тот, для которого энтропия минимальна (а информативность – максимальна).

Энтропию при условии не равновероятных событий p_i находят по известной формуле Шеннона:

$$I = - \sum_i p_i \log_2 p_i,$$

где I – количество информации в битах, которую можно передать, используя m элементов в сообщении при n букв в алфавите, причем $p_i = m/n$.

В случае с деревом решений m – число значений целевой функции $f(\mathbf{x}_i)$ для \mathbf{x}_i (поэтому корректнее использовать запись m_i), n – общее число элементов (записей) исходного набора (множества T). Тогда энтропия множества T по отношению к свойству $S = f(\mathbf{x}_i)$, которое может принимать s значений, может быть найдена как

$$H(T, S) = - \sum_{i=1}^s \frac{m_i}{n} \log_2 \frac{m_i}{n}. \quad (4.22)$$

Если свойство S бинарное (т.е. целевая функция $f(\mathbf{x}_i)$ может принимать только два значения), то запись для энтропии будет выглядеть так:

$$H(T, S) = - \frac{m}{n} \log_2 \frac{m}{n} - \frac{n-m}{n} \log_2 \frac{n-m}{n}.$$

В этом случае, если вероятность появления S равна 0,5 (т.е. равновероятно), то энтропия равна 1 (т.е. нужен 1 бит информации для ее кодирования). Если же появление S не будет равновероятно, то энтропия будет меньше, а значит закодировать информацию из T именно для такого $S = f(\mathbf{x}_i)$ будет более эффективно. Этот пример показывает, что выбор признака \mathbf{x}_i следует осуществлять так, чтобы соответствующая ему энтропия стала минимально возможной.

В общем случае энтропия будет различной для различных потомков узла, поэтому итоговый результат нужно считать с учетом того, сколько исходов осталось в рассмотрении в каждом из потомков. Например, если множество T со свойством $S = f(\mathbf{x}_i)$ классифицировано признаком \mathbf{x}_i с атрибутом Q , имеющим q возможных значений, то прирост информации определяется как

$$\text{Gain}(T, S) = H(T, S) - \sum_{k=1}^q \frac{|T_k|}{|T|} H(T_k, S), \quad (4.23)$$

или, используя более обобщенную запись, имеем выражение

$$\text{Gain}(T, S) = \text{Info}(T) - \text{Info}_S(T),$$

где T_k – множество элементов T , для которых атрибут Q имеет значение k . На каждом шаге алгоритм выбирает тот атрибут Q , для которого это прирост информации максимален.

Результат разницы в выражении (4.23) может быть найден в этом алгоритме альтернативно с использованием *индекса Джини*, упомянутого выше. Индекс Джини начали применять для оценки неравномерности распределения некоторого изучаемого признака (например, годового дохода) для различных социальных групп и широко используют в экономических, социальных и демографических исследованиях [4]. В алгоритме ID3 для его расчета применяется зависимость двух величин (например, $\text{Info}(T)$ и $\text{Info}_S(T)$), упорядоченных по возрастанию и обычно нормированных в процентах (рис. 14). Индекс Джини численно равен площади под кривой Лоренца, которая может принимать значения от 0 до 1.

Алгоритм C4.5. Алгоритм C 4.5 также предложен Д. Куинланом в развитие алгоритма ID3 и расширяет его возможности в части [4]:

- дополнительной функциональности по отсечению ветвей во избежание проблемы переобучения;
- построения дерева из обучающей выборки с пропусками данных (отсутствуют значения для некоторых атрибутов);
- работы не только с дискретными, но и с непрерывными числовыми атрибутами.

Основной недостаток алгоритма ID3 – склонность к переобучению. Действительно, при большом количестве возможных значений признака (возможно, не существенно отличающихся друг от друга), будет построено большое число ветвей с атрибутами этого признака. В крайнем случае, число листьев такого дерева может быть эквивалентно числу имеющихся примеров обучающего множества. Очевидно, для задач классификации такое дерево решений будет бесполезно. Более того, поиск верного признака x_i при построении следующего узла в таком дереве также проблематичен – в (4.22) $\log_2(1) = 0$, поэтому в (4.23) прирост информации для всех x_i максимален.

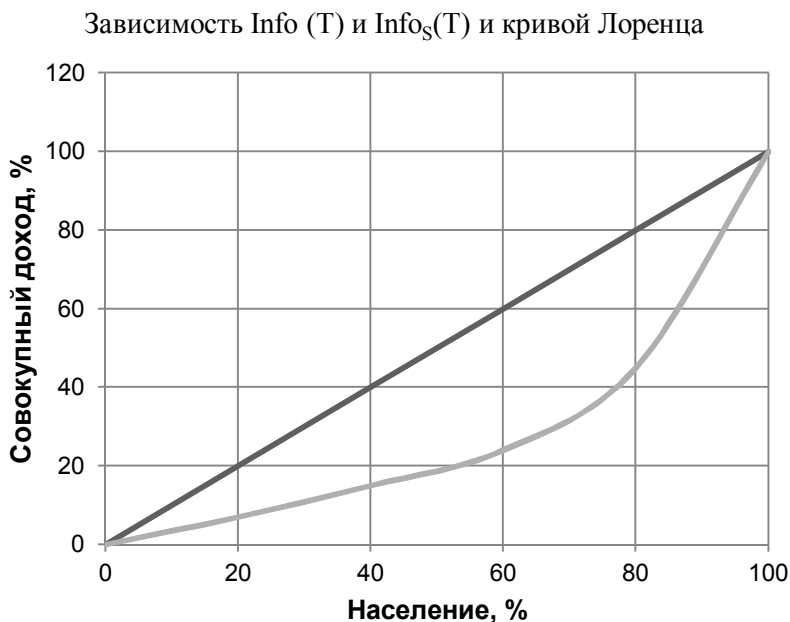


Рис. 14. Иллюстрация зависимости двух переменных $\text{Info}(T)$ и $\text{Info}_S(T)$ и кривой Лоренца, используемых в расчете индекса Джини

Во избежание указанных проблем в алгоритме предусмотрено нормирование при расчете прироста информации путем расчета дополнительного показателя с оценкой потенциальной информации, созданной при разбиении множества T на n подмножеств T_k :

$$\text{Split_Info}(T) = - \sum_{k=1}^n \left[\frac{|T_k|}{|T|} \log_2 \left(\frac{|T_k|}{|T|} \right) \right]. \quad (4.24)$$

Критерий прироста информации (4.23) с использованием выражения (4.25) модифицирован следующим образом:

$$\text{Split_Info}(S) = \frac{\text{Gain}(T, S)}{\text{Split_Info}(T)}. \quad (4.25)$$

С учетом того, что способ расчета критерия прироста информации (4.25) учитывает не только прирост, но и оценку ее потенциальной информативности («полезности»), это способствует выбору более удачного атрибута, построению менее избыточного дерева решений и улучшению классификации.

Еще одной отличительной особенностью алгоритма *C4.5* по отношению к *ID3* является наличие адаптации к присутствию пропусков данных в исходном обучающем множестве.

Вообще, проблема пропусков в данных является практически очень важной, для ее решения на этапе предварительной обработки данных применяют различные приемы и технологии (§ 4.1), позволяющие представить данные для обработки методами *Data Mining* без столь очевидных недостатков.

Тем не менее для полноты представления алгоритма отметим суть вышеуказанной особенности работы с пропусками в данных.

В алгоритме *C4.5* предполагается, что экземпляры обучающего множества с неизвестными значениями имеют статистическое распределение соответствующего признака согласно относительной частоте появления известных значений. Для фиксации этой характеристики введен параметр F , который представляет собой число наблюдений в наборе исходных данных с известным значением данного признака, отнесенное к общему числу наблюдений. Тогда модифицированный для работы с пропущенными значениями критерий прироста информации будет иметь вид

$$\text{Gain}(S) = F \times [\text{Info}(T) - \text{Infos}(T)].$$

Алгоритм CART. CART (англ. *Classification and Regression Tree*, дерево классификации и регрессии) предназначен для построения бинарного дерева решений (каждый узел имеет только две ветви-потомка), предложенный в 1984 г. [9].

Основными отличиями алгоритма *CART* от алгоритмов семейства *ID3* являются:

- бинарное представление дерева решений;
- функция оценки качества разбиения;
- механизм отсечения дерева;

- алгоритм обработки пропущенных значений;
- возможность построения деревьев регрессии.

На каждом шаге построения дерева, правило, формируемое в узле, делит заданное множество примеров на две части – в которых выполняется (первая часть) и не выполняется (вторая часть) решающее правило. При этом производится перебор всех признаков, на основе которых может быть построено разбиение, и выбирается тот, который максимизирует значение некоторого показателя. Например, таким показателем может быть

$$H(T) = 2 P_L P_R \sum_{j=1}^q [P_L(\omega_j) - P_R(\omega_j)],$$

где P_L и P_R – отношение числа примеров в левом и правом потомках к их общему числу в обучающем множестве T , $P_L(\omega_j)$ и $P_R(\omega_j)$ – отношение числа примеров класса ω_j в левом и правом потомках соответственно к их общему числу в каждом из них.

Также в качестве такого показателя применяют выражение, основанное на использовании индекса Джини. Если множество T содержит данные n классов, тогда индекс Джини определяется как

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

где p_i – вероятность (относительная частота) класса ω_i в T . Для узла бинарного дерева с двумя ветвями-потомками после ряда преобразований и упрощений показатель «успешности» разбиения множества рассчитывается как

$$G_{\text{split}} = \frac{1}{L} \sum_{i=1}^n l_i^2 + \frac{1}{R} \sum_{i=1}^n r_i^2,$$

где L, R – число примеров соответственно в левом и правом потомке; l_i и r_i – число экземпляров ω_i -го класса в левом и правом потомке. Лучшим будет то разбиение, для которого величина G_{split} будет максимальной.

Алгоритм CART предусматривает построение не только классификационных деревьев, но и регрессионных. Для этого процесс реализуется аналогично, но вместо меток классов в листьях будут располагаться числовые значения.

4.3.6. Неконтролируемая классификация

Неконтролируемая классификация (кластеризация, англ. *clustering*) позволяет разбить исходный набор данных на конечное количество однородных в некотором смысле кластеров. Неконтролируемая классификация реализуется методами кластерного анализа и позволяет выявлять свойства данных группироваться около некоторых значений (центров). В общем концепцию кластерного анализа многомерных данных можно определить как распределение всех возможных точек (объектов) P -мерного пространства признаков по соответствующим кластерам [59], т.е. разбиение пространства признаков на взаимно непересекающиеся области, каждая из которых соответствует некоторому кластеру C_i с центром μ'_i ($i=1, \dots, M$). При этом объекты одного кластера группируются в признаковом пространстве компактно, т.е. расстояние между объектами одного кластера меньше, чем расстояние между объектами различных кластеров.

Среди методов кластерного анализа можно выделить такие, как *методы итерационной оптимизации (ISODATA метод, англ. Iterative Self Organizing Data Analysis Technique)* [42], *методы иерархической кластеризации* [59], *анализ пиков гистограмм* [90].

Метод *ISODATA* осуществляет итерационную оценку структуры исходных многомерных данных. На каждой итерации определяется новое уточненное пространство кластеров C_i и их центров μ'_i (рис. 15, б, в), исходя из условия минимума расстояния между точками и центрами каждого кластера. Для этого на каждом шаге определяются ближайшие к центрам кластеров элементы изображения и вычисляются новые центры, также осуществляется слияние близких кластеров на основе заданных критериев. В качестве совокупного критерия точности кластеризации метод *ISODATA* использует *суммарную квадратичную ошибку* S , определяемую как

$$S = \sum_{i=1}^{M'} \sum_{x \in C_i} d(x, \mu'_i) \quad (4.26)$$

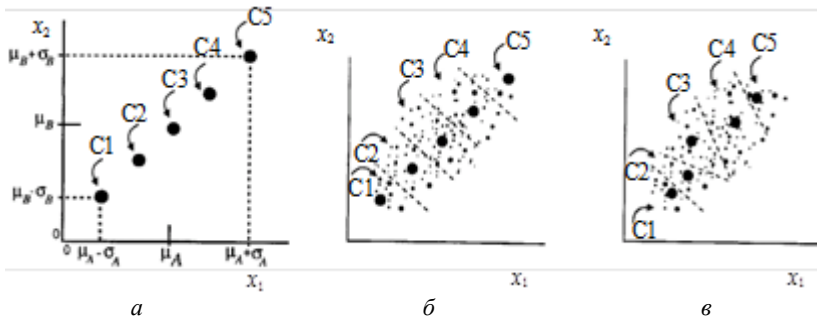


Рис. 15. Иллюстрация процедуры кластерного анализа (итерационный метод *ISODATA*): *а* – начальное размещение центров; *б* – первая итерация; *в* – вторая итерация

Таким образом, качество кластеризации будет наилучшим при наименьшем значении S , т.е. минимальном значении суммы расстояний от всех точек кластеров C_i до их центров μ'_i ($i=1, \dots, M$). Критерием останова итерационного процесса кластеризации методом *ISODATA* являются заданное количество итераций и порог суммарной квадратичной ошибки S (4.26). При этом в качестве меры близости между точками кластера C_i и его центром μ'_i могут быть использованы $L1$ или Евклидова метрики расстояния. $L1$ расстояние между P -мерным вектором \mathbf{x} и центром μ'_i определяется выражением

$$d^{L1}(\mathbf{x}, \mu'_i) = \sum_{v=1}^P |x_v - \mu'_{iv}|, \quad (4.27)$$

а Евклидово расстояние – выражением

$$d^E(\mathbf{x}, \mu'_i) = \sqrt{\sum_{v=1}^P (x_v - \mu'_{iv})^2}. \quad (4.28)$$

Отмечают, что метрика (4.28) более предпочтительна по критерию точности по сравнению с метрикой (4.27), поэтому ее использование при кластеризации более целесообразно.

Алгоритм *k-means* (алгоритм *k*-внутригрупповых средних, *k*-средних). Основан на минимизации функционала Q суммарной выборочной дисперсии, характеризующего разброс элементов относительно центров кластеров:

$$Q = \sum_i |X_i| \sum_{x \in X_i} d(x, C_i) \rightarrow \min,$$

где

$$C_i = \frac{1}{|X_i|} \sum_{x \in X_i} x - \text{центр кластера } X_i.$$

Алгоритм выполняется итерационно (рис. 16). На каждой итерации находятся центры кластеров, а также производится разбиение выборки на кластеры. Вычисления продолжаются, пока функционал Q не перестанет уменьшаться.

Порядок выполнения алгоритма следующий:

1. Выделяются начальные центры кластеров $C_1^{(0)}, \dots, C_m^{(0)}$, $k=0$.
2. Выборка разбивается на m кластеров по принципу ближайшего соседства и получаются некоторые кластеры $X_1^{(k)}, \dots, X_m^{(k)}$.
3. Находим новые центры кластеров как

$$C_i^{(k+1)} = \frac{1}{|X_i^{(k)}|} \sum_{x \in X_i^{(k)}} x.$$

4. Если не выполняется условие $c_i^{(k+1)} = c_i^{(k)}$ для всех $k=1, \dots, m$, то переходим на шаг 2.

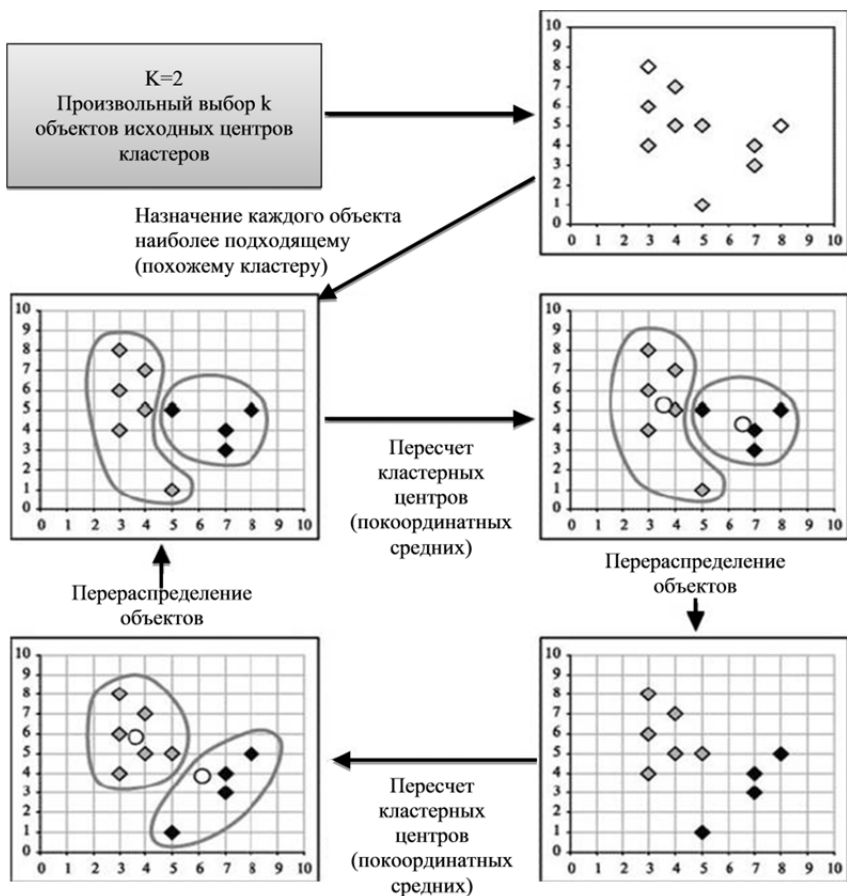


Рис. 16. Графическая иллюстрация работы алгоритма k -средних

Преимуществом алгоритма k -средних являются быстрота и простота реализации. К его недостаткам можно отнести неопределенность выбора числа и исходных центров кластеров, неверный выбор которых может вести к неудовлетворительным результатам работы итерационной процедуры или чрезмерному количеству необходимых итераций. Кроме того, алгоритм может быть чув-

ствителен к выбросам, которые могут существенно исказить среднее. В этом случае может быть применена модификация алгоритма с использованием k -медианы, однако имеющая бóльшую вычислительную сложность, препятствующая работе с большими наборами данных.

4.4. РЕГРЕССИЯ

4.4.1. Понятие регрессии

Термин *регрессия* (англ. *regression*, обратное движение) введен в 1886 г. антропологом Ф. Гальтоном при изучении статистических закономерностей наследственности роста, которые позволили выявить *линейную связь* между средним ростом отцов и их сыновей. Причем рост отцов в среднем оказался выше, чем средний рост сыновей, что позволило сделать исследователю вывод о «*регрессии к посредственности*», т.е. о снижении роста в популяции к ее среднему значению [15]. Вместе с подобным «прикладным» появлением термина *регрессия* справедливо отмечают и еще одно дополнительное принципиальное основание. Термин *регрессия* также отражал новизну в очередности этапов исследования – сначала собраны данные, а затем по ним угадана модель зависимости. В то время как традиционно данные использовались для проверки априори построенных теоретических моделей. Позднее термин *регрессия* закрепился как канонический, отражающий методы восстановления зависимостей между переменными.

Сегодня, под *регрессией* принято понимать зависимость среднего значения какой-либо величины от некоторой другой величины или от нескольких величин, а под *регрессионным анализом* – методы исследования взаимосвязи переменных. Если пространство объектов обозначить как X и множество возможных ответов Y , то существует неизвестная целевая зависимость $y^* : X \rightarrow Y$, значения которой известны только на объектах обучающей выборки $X^l = (x_i, y_i)_{i=1}^l$, $y_i = y^*(x_i)$. Требуется построить алгоритм, который принято называть *функцией регрессии* $f: X \rightarrow Y$, аппроксимирующий целевую зависимость y^* . Наконец, задачу обучения

по прецедентам (x_i, y_i) , позволяющую найти регрессионную зависимость y^* , называют *восстановлением регрессии* [55].

Следует отметить, что регрессионный подход позволяет не только выявлять зависимости между признаками, но и решает задачи прогнозирования (например, временного ряда на основе ретроспективных данных), а также задачи классификации (например, путем использования кривой регрессии в качестве разделяющей плоскости между классами), примеры которых рассмотрены в п. 1.2.4.

4.4.2. Основные этапы регрессионного анализа

Выделяют следующий обобщенный многоэтапный подход к регрессионному анализу:

1. **Формулировка задачи.** На этом этапе формируются предварительные гипотезы о зависимости исследуемых явлений.

2. **Определение зависимых и независимых (объясняющих) переменных.**

3. **Сбор статистических данных.** Данные должны быть собраны для каждой из переменных, включенных в регрессионную модель.

4. **Формулировка гипотезы о форме связи** (простая или множественная, линейная или нелинейная).

5. **Определение функции регрессии** (заключается в расчете численных значений параметров уравнения регрессии).

6. **Оценка точности регрессионного анализа.**

7. **Интерпретация полученных результатов.** Полученные результаты регрессионного анализа сравниваются с предварительными гипотезами. Оцениваются корректность и правдоподобие полученных результатов.

8. **Предсказание неизвестных значений зависимой переменной.**

4.4.3. Методы восстановления регрессии

Существует целый ряд методов восстановления регрессии. Примерами таких методов являются непараметрическая регрессия

с ядерным сглаживанием, линейная и нелинейная регрессия, опорные векторы, регрессионные деревья решений, многомерные адаптивные регрессионные сплайны (англ. *Multivariate Adaptive Regression Splines*, *MARS*), мультилинейная интерполяция (англ. *Multilinear Interpolation*), радиальные базисные функции (англ. *Radial Basis Functions*), робастная регрессия (англ. *Robust Regression*), каскадная корреляция (англ. *Cascade Correlation*) и многие другие способы и подходы. Детали каждого из этих подходов могут быть найдены в специальной литературе.

Вместе с тем большинство исследуемых на практике зависимостей может быть аппроксимировано стандартными нелинейными математическими функциями, для построения которых широко используют *метод наименьших квадратов* (МНК). Его суть заключается в следующем.

Пусть задана модель регрессии – параметрическое семейство функций $g(x, \alpha)$, где $\alpha \in \mathbb{R}^p$ – вектор параметров модели. Определим функционал качества аппроксимации целевой зависимости на выборке X^ℓ как сумму квадратов ошибок:

$$Q(\alpha, X^\ell) = \sum_{i=1}^{\ell} (g(x_i, \alpha) - y_i)^2.$$

Обучение МНК состоит в том, чтобы найти вектор параметров α^* , при котором достигается минимум среднего квадрата ошибки на заданной обучающей выборке X^ℓ :

$$\alpha^* = \arg \min_{\alpha \in \mathbb{R}^p} Q(\alpha, X^\ell).$$

Стандартный способ решения этой оптимизационной задачи – воспользоваться необходимым условием минимума. Если функция $g(x, \alpha)$ достаточное число раз дифференцируема по α , то в точке минимума выполняется система p уравнений относительно p неизвестных

$$\frac{\partial Q}{\partial \alpha}(\alpha, X^\ell) = 2 \sum_{i=1}^{\ell} (g(x_i, \alpha) - y_i) \frac{\partial g}{\partial \alpha}(x_i, \alpha) = 0.$$

Решение системы таких уравнений (методами Гаусса или Крамера) позволяет найти необходимые коэффициенты α в уравнении регрессии.

4.5. АССОЦИАЦИЯ

Еще одним методом, который выделяют в *Data Mining*, является *ассоциация* – выявление закономерностей между связанными событиями. Следует отметить, что в *Data Mining* методы *ассоциаций*, *поиска ассоциативных правил* (англ. *association rule induction*) принято выделять в отдельный класс, хотя, по сути они являются частью более общих методов неконтролируемой классификации [61]. Примером ассоциативной закономерности (*ассоциативного правила*) служит правило, указывающее, что из события *X* следует событие *Y*.

Впервые эта задача апробирована в торговой отрасли при нахождении типичных шаблонов покупок, совершаемых в супермаркетах, поэтому иногда ее называют *анализом рыночной корзины* (англ. *market basket analysis*). *Ассоциативные правила* помогают выявлять группы товаров, как правило, приобретаемые совместно. Знание этих правил позволяет соответствующим образом размещать товары на прилавках, стимулируя интенсивность их продаж. Задача поиска ассоциативных правил актуальна и в сфере обслуживания, где интерес представляет то, какими услугами клиенты предпочитают пользоваться в совокупности. В медицине интересной представляется возможность выявлять наиболее сочетаемые болезни и симптомы, требующиеся для определения диагноза.

Наиболее известным и широко используемым (в ритейле и их консалтинговых компаний) примером практической реализации поиска ассоциативных правил является алгоритм «*Априори*» (англ. *Apriori*, *A Priori*) [2].

Продemonстрируем работу алгоритма «*Априори*» на примере задачи анализа рыночной корзины, оперируя привычными на практике табличными записями. Одним из наиболее распространенных типов данных в этой задаче являются *транзакционные данные* (англ. *transaction data*), отражающие в таблице базы данных взаимодействие клиентов и магазина (табл. 7).

Таблица 7

Пример транзакционных данных клиентов и магазина

ID	Яблоки	Пиво	Сыр	Финики	Яйца	Рыба	Конфеты
Условное обозначение	<i>a</i>	<i>b</i>	<i>c</i>	<i>d</i>	<i>e</i>	<i>f</i>	<i>g</i>
1	1	1		1			1
2			1	1	1		
3		1	1			1	
4		1				1	
5					1		1
6						1	
7	1			1			
8						1	
9			1		1		
10		1					1
11					1		1
12	1						
13			1			1	
14			1			1	
15							
16				1			
17	1					1	
18	1	1	1	1			
19	1	1		1			1
20					1		

В примере представлены 20 записей, полученные при продаже 7 видов товаров. В большом супермаркете аналогичные реальные данные могут быть представлены миллионами записей в месяц, а число групп товаров может достигать тысячи и более.

При анализе рыночной корзины оперируют утверждениями типа:

«Если корзина содержит *X* и *Y*, то она также содержит и *Z*». (4.29)

Подобное правило сопровождается двумя метриками:

1. *Достоверность* или *точность* (англ. *confidence* или *accuracy*). Отражает, как часто при справедливости выражения в части «если», оно также справедливо в части «то».

2. *Покрывтие* или *поддержка* (англ. *coverage* или *support*). Отражает долю выражений в части «если» в базе данных.

Для утверждения (4.29) в случае $X = \text{пиво}$, $Y = \text{сыр}$, $Z = \text{рыба}$ в рассматриваемом примере достоверность составляет $1/2 = 50\%$, а поддержка $2/20 = 10\%$.

Какие же утверждения конструкции «если / то» следует признать интересными и достойными внимания? Логично, к таким утверждениям отнести те, которые справедливы чаще, чем случайно.

Например, статистический анализ потребительской корзины показывает, что 10% в ней занимает хлеб, а 4% – стиральный порошок. Это означает, что вероятность встретить в такой корзине хлеб $P(\text{хлеб}) = 0.1$, а порошок – $P(\text{порошок}) = 0.04$. Какова же вероятность встретить эти товары в корзине совместно? Очевидно, $P(\text{хлеб}|\text{порошок}) = P(\text{хлеб}) \times P(\text{порошок}) = 0.1 \times 0.04 = 0.004$. На практике, анализируя, например, тысячу таких потребительских корзин, лишь в четырех из них ожидается увидеть хлеб и порошок. Поэтому если в действительности такую комбинацию удастся обнаружить в 20 или 30 случаях из тысячи, это будет сюрпризом, достойным внимательного анализа (между товарами в магазине или торговой сети есть какая-то неочевидная связь).

Каким же образом в базе данных можно найти наиболее интересные для анализа правила (ассоциации)? Очевидно, такие правила будут отличаться не только более высокой *точностью*, но и должны иметь значительную *поддержку* (обеспечивая воспроизводимость найденного правила на данном наборе данных). Однако задача поиска таких ассоциативных правил на практике осложняется высокой вычислительной сложностью. Можно оперировать 500 000 000 возможными правилами типа (4.29) и если база данных состоит из 20 000 000 записей, то число возможных вычислительных операций может доходить до 10^{16} . Проверка уровней *точности* и *поддержки* в таком массиве – нетривиальная задача.

Поэтому для решения подобных задач был предложен несложный, вычислительно эффективный алгоритм «*Априори*», позволяющий находить интересные ассоциативные связи в данных. Алгоритм оперирует не собственно ассоциативными правилами типа (4.29), а *множествами* (англ. *itemsets*), элементы которых определенным образом ассоциированы между собой.

4.5.1. Описание алгоритма

Рассмотрим описание алгоритма «*Априори*», основываясь на утверждении, что любое множество, содержащееся в некотором часто встречающемся множестве (ЧВМ), является ЧВМ. Другими словами, если

$$Y \subseteq X \text{ и } P(X) \geq c, \text{ то } P(Y) \geq c. \quad (4.30)$$

При этом k -множеством будем называть множество, состоящее из k элементов.

Обозначим через L_k множество всех часто встречающихся k -множеств. Объединение L_k по всем k дает все искоемое множество ЧВМ. Построение L_k выполняется по шагам.

Сначала находится L_1 (множество одноэлементных ЧВМ). Затем для каждого фиксированного $k \geq 2$, используя найденное множество L_{k-1} , определяется L_k . Процесс завершается, как только k станет больше максимального количества элементов.

Определение L_k при известном L_{k-1} выполняется в два шага:

- 1) генерируются множества – кандидаты C_k ;
- 2) затем из этого множества исключаются лишние элементы.

Полученное таким образом множество и будет равно L_k .

Генерация множества кандидатов. Множество кандидатов C_k составляется путем *слияний* всех *допустимых пар* $l_1, l_2 \in L_{k-1}$.

Сокращение. Множество кандидатов C_k содержит все множества из L_k , но содержит дополнительно и лишние множества, не являющиеся ЧВМ. Чтобы получить L_k , необходимо лишь исключить такие множества. Для этого необходимо для каждого набора из C_k посчитать количество его повторений в базе данных и исключить это множество, если число повторений меньше заданного порога (уровня поддержки). Но такой подсчет – довольно трудоемкая процедура, так как C_k может иметь очень большой размер. Поэтому рекомендуется сначала произвести его предварительную очистку следующим образом.

Пусть l – некоторое множество из C_k (следовательно, он состоит из k элементов). Если l – ЧВМ, то в соответствии с утверждением

(4.30), все подмножества l , состоящие из $k - 1$ элементов, должны быть также ЧВМ, т.е. принадлежать множеству L_{k-1} . Поэтому, если хотя бы одно множество, полученное из l удалением одного элемента, не принадлежит L_{k-1} , то l не может являться ЧВМ и должно быть исключено из C_k .

Допустимая пара и их слияние. Пусть $l_1, l_2 \in L_{k-1}$ – два множества из множества L_{k-1} . Обозначим через $l_i[j]$ j -й элемент в множестве l_i . Например, $l_1[k-2]$ – это предпоследний элемент в l_1 . Предполагается, что на исходном множестве элементов задано некоторое отношение порядка ' $>$ ' (например, по номерам элементов), и в наборе l_i элементы отсортированы в соответствии с данным отношением порядка.

Пара l_1, l_2 – допустимая для слияния, если

$$(l_1[1] = l_2[1]) \wedge (l_1[2] = l_2[2]) \wedge \dots \wedge (l_1[k-2] = l_2[k-2]) \wedge (l_1[k-1] < l_2[k-1]).$$

Условие $(l_1[k-1] < l_2[k-1])$ гарантирует, что дубликатов в множестве C_k не будет. Слиянием $u(l_1, l_2)$ допустимых наборов l_1, l_2 будет множество, состоящее из элементов $l_1[1], l_1[2], \dots, l_1[k-1], l_2[k-1]$.

4.5.2. Пример исполнения алгоритма

Для демонстрации работы алгоритма «*Априори*» приведем его краткое пошаговое описание:

Шаг 1. Найти все 1-элементные ЧВМ множества

Шаг 2. For ($k = 2$; while $L_{k-1} \neq \emptyset$; $k++$)

Шаг 3. $C_k = \text{apriori_gen}(L_{k-1})$

Шаг 4. Для каждого c в C_k , $c.\text{count} = 0$

Шаг 5. Для всех записей r в БД

Шаг 6. $C_r = \text{subset}(C_k, r)$; For each c in C_r , $c.\text{count}++$

Шаг 7. Set $L_k :=$ all c in C_k whose $\text{count} \geq \text{minsup}$

Шаг 8. } // возвращаем все множества L_k .

Обозначим с помощью minsup – параметр минимального уровня поддержки. Пусть $\text{minsup} = 4$ (т.е. 20%).

Шаг 1. Находим все 1-элементные ЧВМ с заданным *minsup* (т.е. записи, представленные в БД минимум 4 раза):

Результат шага: $L_1 = \{\{a\}, \{b\}, \{c\}, \{d\}, \{e\}, \{f\}, \{g\}\}$.

Шаг 2–3. Задаем $k = 2$ и запускаем процедуру *apriori_gen* для формирования из L_1 множества кандидатов C_2 , отсортированных по алфавиту.

Результат: $C_2 =$

$$\begin{aligned} &\{\{a, b\}, \{a, c\}, \{a, d\}, \{a, e\}, \{a, f\}, \{a, g\}, \\ &\quad \{b, c\}, \{b, d\}, \{b, e\}, \{b, f\}, \{b, g\}, \\ &\quad \{c, d\}, \{c, e\}, \{c, f\}, \{c, g\}, \\ &\quad \{d, e\}, \{d, f\}, \{d, g\}, \\ &\quad \{e, f\}, \{e, g\}, \\ &\quad \{f, g\}\}. \end{aligned}$$

Шаг 4. Инициализируем счетчик *c.count* нулевым значением.

Шаг 5–6. Перебираем все записи БД и находим те, которые содержатся в C_2 .

Первая запись $r1 = \{a, b, d, g\}$. Элементами C_2 , содержащими элементы из $r1$ будут: $C_{r1} = \{\{a, b\}, \{a, d\}, \{a, g\}, \{a, d\}, \{a, g\}, \{b, d\}, \{b, g\}, \{d, g\}\}$. Также для каждого из C_r для этих множеств признаков инкрементируем счетчик *c.count*++.

Вторая запись $r2 = \{c, d, e\}$, а $C_{r2} = \{\{c, d\}, \{c, e\}, \{d, e\}\}$.

И так далее, для всех записей из БД. После анализа 20 записей БД, проверяем, для каких множеств-кандидатов значения счетчика *c.count*, не ниже заданного уровня *поддержки* – т.е. $\geq \text{minsup}$ (4): $\{a, b\} \{a, c\} \{a, d\} \{c, d\} \{c, e\} \{c, f\}$.

Результат: $L_2 = \{\{a, b\} \{a, c\} \{a, d\} \{c, d\} \{c, e\} \{c, f\}\}$

Затем осуществляется переход к шагу 2 алгоритма, $k = 3$ и выполняется процедура *apriori_gen* по данным L_2 . Формируются следующие пары значений: $\{a, b\}:\{a, c\}$ $\{a, c\}:\{a, d\}$ $\{c, d\}:\{c, e\}$ $\{c, d\}:\{c, f\}$ $\{c, e\}:\{c, f\}$. Множества из 3 элементов будут выглядеть следующим образом: $\{a, b, c\}, \{a, c, d\}, \{c, d, e\}, \{c, d, f\}, \{c, e, f\}$.

- $\{a, b, c\}$ исключается из рассмотрения, так как $\{b, c\}$ не в L_2 ;
- $\{c, d, e\}$ исключается из рассмотрения, так как $\{d, e\}$ не в L_2 ;
- $\{c, d, f\}$ исключается из рассмотрения, так как $\{d, f\}$ не в L_2 ;
- $\{c, e, f\}$ исключается из рассмотрения, так как $\{e, f\}$ не в L_2 .

Таким образом, остается $C_3 = \{a, c, d\}$. Выполняются шаги 5–7 с подсчетом количества появлений C_3 в исходной БД. Их число составляет 4, поэтому $L_3 = \{a, c, d\}$.

Затем осуществляет очередной переход к шагу 2, $k = 4$ и выполняется процедура *apriori_gen* по данным L_3 . В данном случае $L_4 = \{\}$, выполнение алгоритма завершается, возвращая все полученные не пустые множества.

Результат: $L_s = \{\{a\}, \{b\}, \{c\}, \{d\}, \{e\}, \{f\}, \{g\}, \{a, c\}, \{a, d\}, \{c, d\}, \{c, e\}, \{c, f\}, \{a, c, d\}\}$.

Каждый из 13 элементов множества данных L_s может представлять бизнес-интерес, интерпретируя которые можно составить несложные, но практически полезные ассоциативные правила.

4.6. ПОСЛЕДОВАТЕЛЬНАЯ АССОЦИАЦИЯ

При анализе ассоциаций, которым посвящен § 4.5, может быть полезной не только найденная в совокупности транзакций базы данных связь между переменными, но и последовательность появления таких транзакций в базе данных во времени. В таком случае появляются анализ *последовательности* (англ. *sequence*) и поиск *последовательных ассоциаций* (англ. *sequential association*), которые могут содержать интересную информацию. Фактически, *ассоциация* является частным случаем *последовательной ассоциации* с временным лагом, равным нулю.

При наличии закономерностей в таких последовательностях возможно с некоторой долей вероятности предсказывать появление событий в будущем и принимать на основе этого более обоснованные решения. Такую разновидность поиска ассоциативных правил называют *сиквенциальным анализом* (англ. *Sequential Pattern Mining, SPM*), отличающимся учетом отношения порядка между исследуемыми наборами. Данное отношение может быть определено разными способами. При анализе последовательности событий, происходящих во времени, объектами таких наборов являются события, а отношение порядка соответствует хронологии их появления. В этом случае, примером закономерности (шаблоном, паттерном) служит правило, указывающее, что из события X спустя время t последует событие Y .

Сиквенциальный анализ широко используется, например, в телекоммуникационных компаниях, для анализа данных об авариях на различных узлах сети [101]. Информация о последовательности совершения аварий может помочь в обнаружении неполадок и предупреждении новых аварий. Например, если известна типичная последовательность сбоев некоторой телекоммуникационной среды $\langle e_5, e_2, e_7, e_{13}, e_6, e_1, \dots \rangle$, где e_i – сбой с кодом i , то на основании факта появления сбоя e_2 можно сделать вывод о скором появлении сбоя e_7 . Это позволяет априори предпринять профилактические меры, устраняющие причины возникновения сбоя. Кроме того, если дополнительно обладать и знаниями о времени между сбоями, то можно предсказать не только факт его появления, но и время.

При анализе поведения пользователей интернет-сайтов полезным может быть определение профиля пользователя по закономерностям в последовательности навигации по тем или иным разделам (*web*-страниц) сайта. В биоинформатике такие ассоциации могут быть информативны при поиске каркасов белковых последовательностей.

В маркетинге и менеджменте при управлении циклом работы с клиентом (англ. *Customer Lifecycle Management*) эти подходы крайне востребованы, позволяя предугадать связь приобретения одного товара / услуги с вероятностью приобретения других товаров / услуг позднее. Например, после покупки квартиры большинство в течение двух недель приобретают холодильник, а в течение двух месяцев – телевизор. Или может существовать последовательная связь между покупкой кровати и постельных принадлежностей для нее.

4.6.1. Алгоритмы семейства «Априори»

Первыми алгоритмами, разработанными для решения задач сиквенциального анализа, были алгоритмы, построенные на базе алгоритма «*Априори*» (4.5), но отличающиеся учетом дополнительного параметра – времени совершения транзакции (например, *AprioriAll*, *AprioriSome*, *DynamicSome*) [1, 101]. Эти алгоритмы используют подход генерации и отбора кандидатов часто встречающихся последова-

тельностью (ЧВП), а также свойство, заключающееся в том, что каждая подпоследовательность ЧВП должна также быть ЧВП.

Опишем кратко суть этих алгоритмов сиквенциального анализа.

Пусть имеется база данных, в которой каждая запись представляет собой клиентскую транзакцию – *идентификатор клиента, дата / время транзакции, набор приобретенных товаров*. При этом есть условие – клиент не может иметь две и более транзакций, совершенных в один момент времени.

Введем несколько основных понятий, требуемых для описания сути алгоритмов.

Предметное множество (англ. *itemset*) – непустой набор предметов (товаров), появившихся в одной транзакции, т.е. приобретенные одновременно. Для обозначения используем фигурные скобки – $I = \{i_1, i_2, \dots, i_m\}$, где i_j – предмет.

Последовательность – упорядоченное предметное множество. Для обозначения используем треугольные скобки – $S = \langle I_1, I_2, \dots, I_m \rangle$, где I_i – предметное множество. *Длиной* последовательности будем называть количество предметов в этой последовательности, а последовательность длины k – *k-последовательностью*.

Последовательность S_1 содержится в последовательности S_2 , если все предметные множества S_1 содержатся в предметных множествах S_2 , при этом порядок надмножеств из S_2 соответствует порядку предметных множеств S_1 . Например, последовательность $\langle \{3\}, \{4,5\}, \{8\} \rangle$ содержится в последовательности $\langle \{7\}, \{3,8\}, \{9\}, \{4,5,6\}, \{8\} \rangle$, поскольку $\{3\}$ содержится в $\{3,8\}$, $\{4,5\}$ – в $\{4,5,6\}$ и $\{8\}$ – в $\{8\}$.

Однако, $\langle \{3\}, \{5\} \rangle$ не содержится в $\langle \{3,5\} \rangle$ (и наоборот), поскольку в первой последовательности предметы 3 и 5 приобретены один за другим, а во второй – совместно.

Все транзакции одного клиента могут быть показаны в виде последовательности, в которой они упорядочены по дате, времени или номеру визита. Такие последовательности будем называть *клиентскими*. Формально это записывается следующим образом. Пусть клиент совершил несколько упорядоченных во времени транзакций T_1, T_2, \dots, T_k . Тогда каждый предметный набор в тран-

закции T_i обозначим $I(T_i)$, а каждую клиентскую последовательность для данного клиента запишем как $\langle I(T_1), I(T_2), \dots, I(T_k) \rangle$.

Последовательность S называется *поддерживаемой* клиентом, если она содержится в клиентской последовательности данного клиента. Тогда *поддержка* последовательности определяется как число клиентов, поддерживающих данную последовательность (аналогично *поддержке* при ассоциативном анализе в § 4.5).

Для базы данных клиентских транзакций задача поиска шаблонов заключается в обнаружении последовательностей, имеющих *поддержку* выше заданного порогового значения. Каждая такая последовательность является *шаблоном* (*паттерном*) последовательных событий. Последовательность, удовлетворяющую ограничению минимальной поддержки, будем называть ЧВП.

Последовательность S называется *максимальной*, если она не содержится в какой-либо другой последовательности.

Упомянутые выше алгоритмы сиквенционального анализа решают задачу поиска последовательных шаблонов и состоят в общем виде из следующих этапов:

Этап 1. Сортировка. Заключается в перегруппировке записей в таблице транзакций. Сначала записи сортируются по уникальному ключу покупателя, а затем по времени внутри каждой группы.

Этап 2. Отбор кандидатов. В исходном наборе данных производится поиск всех ЧВП. В частности, на этом этапе происходит поиск всех одноэлементных шаблонов.

Этап 3. Трансформация. Производится для ускорения процесса проверки присутствия последовательностей в наборе транзакций покупателей. Трансформация заключается в замене каждой транзакции списком ЧВП, которые в ней содержатся. При этом если в транзакции отсутствуют частые предметы, то данная транзакция не учитывается. Аналогичным образом не учитываются предметы, не являющиеся частыми, а также последовательности, транзакции которых не содержат ЧВП.

Этап 4. Генерация последовательностей. Из полученных на предыдущих этапах последовательностей строятся более длинные шаблоны последовательностей.

Этап 5. Максимизация (опционально). Среди имеющихся последовательностей происходит поиск тех, которые не входят в более длинные последовательности.

Все упомянутые алгоритмы реализованы аналогично (различаются в деталях реализации этапа (4) и обладают невысокой вычислительной эффективностью).

4.6.2. Алгоритм GSP

Для существенного повышения вычислительной эффективности (до 20 раз) сиквенционального анализа предложена модификация алгоритма *AprioriAll*, названная *GSP* (англ. *Generalized Sequential Pattern*, *обобщенный сиквенциональный паттерн*), учитывающая ограничения по времени между соседними транзакциями [1, 32].

В случае с алгоритмом *GSP* требуется учитывать дополнительные условия, чтобы определить, содержит ли последовательность указанную последовательность (подпоследовательность).

Введем такие параметры, как минимальное и максимальное допустимое время между транзакциями (*min_gap* и *max_gap*), а также понятие скользящего окна, размера *win_size*. Допускается, что элемент последовательности может состоять не из одной, а из нескольких транзакций, если разница во времени между ними меньше, чем размер окна.

Последовательность $d = \langle d1...dm \rangle$ содержит последовательность $s = \langle s1...sm \rangle$, если существуют такие целые числа $l1 \leq u1 < l2 \leq u2 < ... < ln \leq un$, что:

1. si содержится в объединении dk , где $li \leq k \leq ui$, $1 \leq i \leq n$.
2. $t_{\text{транзакции}}(dl[i]) - t_{\text{транзакции}}(du[i-1]) \leq win_size$, $1 \leq i \leq n$.
3. $min_gap \leq t_{\text{транзакции}}(dl[i]) - t_{\text{транзакции}}(du[i-1]) \leq max_gap$, $2 \leq i \leq n$.

Выполнение алгоритма *GSP* предусматривает несколько проходов по исходному набору данных. При первом проходе вычисляется поддержка для каждого предмета и из них выделяются частые. Каждый подобный предмет представляет собой одноэлементную последовательность. В начале каждого последующего прохода имеется некоторое число ЧВП, выявленных на предыдущем шаге

алгоритма. Из них будут формироваться более длинные последовательности-кандидаты.

Каждый кандидат представляет собой последовательность, длина которой *на один больше* чем у последовательностей, из которых кандидат был сформирован. Таким образом, число элементов всех кандидатов одинаково. После формирования кандидатов происходит вычисление их поддержки. В конце шага определяется, какие кандидаты являются ЧВП. Найденные ЧВП послужат исходными данными для следующего шага алгоритма. Работа алгоритма завершается тогда, когда не найдено ни одной новой ЧВП в конце очередного шага, или когда невозможно сформировать новых кандидатов.

Таким образом, в работе алгоритма можно выделить следующие основные этапы:

1. Генерация кандидатов.

- 1.1. Объединение.

- 1.2. Упрощение.

2. Подсчет поддержки кандидатов.

Рассмотрим эти операции более подробно.

Этап 1. Генерация кандидатов. Пусть L_k содержит все частые k -последовательности, а C_k – множество кандидатов из k -последовательностей. В начале каждого шага имеем L_{k-1} – набор из $(k-1)$ ЧВП. На их основе необходимо построить набор всех k ЧВП.

Введем понятие *смежной подпоследовательности*.

При наличии последовательности $s = \langle s_1 s_2 \dots s_n \rangle$ и подпоследовательности c , c будет являться *смежной последовательностью* s , если соблюдается одно из условий:

- c получается из s при удалении предмета из первого $\{s_1\}$ или последнего $\{s_n\}$ предметного множества;

- c получается из s при удалении одного предмета из предметного множества s_i , если в его составе не менее двух предметов;

- c – смежная подпоследовательность c' , где c' – смежная подпоследовательность s .

Например, дана последовательность $s = \langle \{1,2\}, \{3,4\}, \{5\}, \{6\} \rangle$. Последовательности $\langle \{2\}, \{3,4\}, \{5\} \rangle$, $\langle \{1,2\}, \{3\}, \{5\}, \{6\} \rangle$ и $\langle \{3\}, \{5\} \rangle$ являются смежными подпоследовательностями s , а последовательности $\langle \{1,2\}, \{3,4\}, \{6\} \rangle$ и $\langle \{1\}, \{5\}, \{6\} \rangle$ таковыми не являются.

Если некоторая последовательность содержит последовательность s , то она также содержит и все смежные подпоследовательности s .

Генерация кандидатов происходит в два этапа (условно назовем эти этапы функцией *candidate_gen_SPM()*).

Этап 1.1. Объединение (англ. *join*). Создаем последовательности-кандидаты путем объединения двух последовательностей L_{k-1} и L_{k-1} . Последовательность s_1 объединяется с s_2 , если подпоследовательность, образуемая путем удаления первого предмета из s_1 , будет та же, что и в случае удаления последнего предмета из s_2 . Объединение последовательностей происходит путем добавления к s_1 соответствующего предмета из последнего предметного множества s_2 . При этом возможны два варианта:

- если последний предмет из s_2 составлял одноэлементное предметное множество, то при объединении он будет добавлен к s_1 как новое предметное множество;
- в противном случае, он будет включен в последнее предметное множество s_1 как его элемент.

При объединении L_1 с L_1 нужно добавить предмет к s_2 как отдельное предметное множество, а также в качестве дополнительного предмета в предметное множество последовательности s_1 . Так, объединение $\langle \{x\} \rangle$ с $\langle \{y\} \rangle$ дадут как $\langle (x, y) \rangle$, так и $\langle (x), (y) \rangle$. При этом x и y упорядочены.

Этап 1.2. Упрощение (англ. *prune*). Удаляем последовательности-кандидаты, которые содержат смежные $(k-1)$ -последовательности, чья поддержка меньше минимально допустимой.

Этап 2. Подсчет поддержки кандидатов. Сканируя набор последовательностей, обрабатываем их по очереди. Для тех кандидатов, которые содержатся в обрабатываемой последовательности, увеличиваем значение поддержки на единицу. Для уменьшения количества кандидатов, требующих проверки вхождения в обраба-

тываемые последовательности, а также для увеличения скорости проверки используется хэш-дерево [3].

Проиллюстрируем основные этапы работы алгоритма *GSP* на примере данных в табл. 8–10.

Т а б л и ц а 8

Пример таблицы транзакций магазина

ID клиента	Время транзакции	Транзакция
1	20 июля 2015	30
1	25 июля 2015	90
2	9 июля 2015	10, 20
2	14 июля 2015	30
2	20 июля 2015	40, 60, 70
3	25 июля 2015	30, 50, 70
4	25 июля 2015	30
4	29 июля 2015	40, 70
4	2 августа 2015	90
5	12 июля 2015	90

Т а б л и ц а 9

Пример таблицы последовательностей на основе БД транзакций

ID клиента	Последовательность данных
1	$\langle \{30\} \{90\} \rangle$
2	$\langle \{10, 20\} \{30\} \{40, 60, 70\} \rangle$
3	$\langle \{30, 50, 70\} \rangle$
4	$\langle \{30\} \{40, 70\} \{90\} \rangle$
5	$\langle \{90\} \rangle$

Т а б л и ц а 10

Пример итогового набора последовательностей различной длины с заданным уровнем поддержки

	Последовательные паттерны с поддержкой $\geq 25\%$
1-последовательности	$\langle \{30\} \rangle, \langle \{40\} \rangle, \langle \{70\} \rangle, \langle \{90\} \rangle,$
2-последовательности	$\langle \{30\} \{40\} \rangle, \langle \{30\} \{70\} \rangle, \langle \{30\} \{90\} \rangle, \langle \{40\} \{70\} \rangle$
3-последовательности	$\langle \{30\} \{40, 70\} \rangle$

Формальная запись алгоритма *GSP* может быть представлена следующим образом:

Алгоритм *GSP(S)*:

Шаг 1. $C_1 \leftarrow \text{init_pass}(S)$; // первый проход через *S*

Шаг 2. $F_1 \leftarrow \{(\{f\}) \mid f \in C_1, f.\text{count}/n \geq \text{minsup}\}$; // *n* – число последовательностей в *S*

Шаг 3. **for** ($k = 2$; $F_{k-1} \neq \emptyset$; $k++$) **do** // формирование подпоследовательностей из *S*

Шаг 4. $C_k \leftarrow \text{candidate_gen_SPM}(F_{k-1})$;
// однократное сканирование набора данных

Шаг 5. **for** (для каждой последовательности данных $s \in S$) **do**

Шаг 6. **for** (для каждого кандидата $c \in C_k$) **do**

Шаг 7. **if** c содержится в s **then**

Шаг 8. $c.\text{count}++$; // инкремент счетчика поддержки

Шаг 9. **end**

Шаг 10. **end**

Шаг 11. $F_k \leftarrow \{c \in C \mid c.\text{count} / n \geq \text{minsup}\}$

Шаг 12. **end**

Шаг 13. **return** $\cup_k F_k$;

Детали реализации функции генерирования кандидата-последовательности *candidate_gen_SPM()* изложены выше.

Пример генерирования кандидатов-последовательностей приведен в табл. 11.

Т а б л и ц а 11

Пример работы этапа генерирования кандидатов-последовательностей

3-последовательности	4-последовательности	
	После присоединения	После упрощения
$\langle \{1, 2\} \{4\} \rangle$	$\langle \{1, 2\} \{4, 5\} \rangle$	$\langle \{1, 2\} \{4, 5\} \rangle$
$\langle \{1, 2\} \{5\} \rangle$	$\langle \{1, 2\} \{4\} \{6\} \rangle$	
$\langle \{1\} \{4, 5\} \rangle$		
$\langle \{1, 4\} \{6\} \rangle$		
$\langle \{2\} \{4, 5\} \rangle$		
$\langle \{2\} \{4\} \{6\} \rangle$		

Недостатками подобных алгоритмов, существенно снижающими вычислительную эффективность обработки данных, являются:

- большое количество обращений к базе данных, соответствующее длине максимального кандидата-последовательности;
- большое число генерируемых кандидатов-последовательностей.

4.7. ОБНАРУЖЕНИЕ АНОМАЛИЙ

Еще одним крайне полезным направлением научно-практической деятельности в *Data Mining* принято считать обнаружение аномалий (англ. *Anomaly Detection*). Это направление, по сути, можно рассматривать как противоположное направлению кластеризации (п. 4.3.6), так как в данном случае необходимо решить обратную задачу – найти такие экземпляры данных, которые являются нетипичными, редкими для некоторого обычного (типичного, традиционного) профиля исследуемого набора данных. Очевидно, такие аномалии (явления с низкой частотой возникновения) могут появиться и быть выявленными только в случае значительного объема накопленных данных с «типичным профилем».

Учитывая высокую практическую ценность, порой заложенную в такие аномалии, сегодня присутствует целый ряд областей приложения методов обнаружения аномалий:

- обнаружение фактов мошенничества со счетами кредитных карт, подозрительных сделок в страховании и т.п.;
- обнаружение фактов несанкционированного вторжения в информационно-коммуникационные сети и среды;
- обнаружение ошибок.

На рис. 17 представлен пример двумерной диаграммы с данными признаков X и Y , позволяющей визуальнo оценить выборки без аномалий (N_1 и N_2), а также выборку (O_3) и отдельные экземпляры с аномалиями (o_1 , o_2).

Для решения задач обнаружения аномалий применяют огромное количество подходов, методов, алгоритмов и их модификаций, основанных на анализе разновременных данных различным математическим аппаратом.

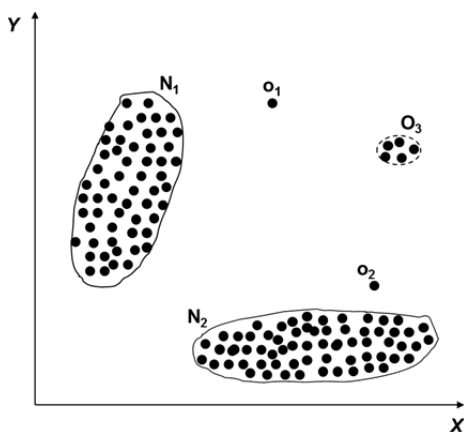


Рис. 17. Пример двумерной диаграммы данных признаков X и Y , отражающей наличие и отсутствие аномалий

Приводят следующую классификацию методов обнаружения аномалий, позволяющую в некоторой степени систематизировать массив имеющихся подходов и способов:

1. Основаны на графическом представлении (англ. *Graphical-based*).

В этом случае используют различные диаграммы и scatter-граммы 1D–3D-мерные. Основные недостатки: трудоемкость построения и субъективность.

2. Основаны на статистических методах (англ. *Statistical-based*).

Требуют предположения о наличии смеси распределений «типичных» данных и данных отклонения, а также гипотез о законах их распределения. К недостаткам относят то, что обычно распределение данных не известно, а в случае многомерной среды еще и затруднительно достаточно точно оценить статистическую плотность распределения выборки.

3. Основаны на использовании различных метрик расстояния (англ. *Distance-based*). В этом случае данные представлены в виде набора векторов признаков, а для их анализа используют три основных подхода – ближайшего соседа, оценки плотности и кластеризации.

4. Основанных на использовании моделей (англ. *Model-based*). В этом случае общий алгоритм обнаружения аномалий может быть представлен в виде трех основных этапов:

- прогнозирование значений наблюдаемых временных рядов $x_i(t)$ на один шаг в соответствии с построенной моделью (x_i – интересующий параметр системы или среды, t – время);

- измерение отклонений между прогнозными значениями модели и фактическими значениями среды;

- механизм (набор решающих правил и процедур), определяющий, является ли значение (или последовательность значений) «слишком» отклоняющимся от прогнозного (соответствующего профилю «нормальной» работы сети).

Примерами конкретных методов для решения задач обнаружения отклонений могут быть авторегрессионная модель, модель скользящего среднего, комбинированная модель авторегрессии и скользящего среднего, фильтр Калмана, *SVM*, нейросети, байесовские сети и др., некоторые из которых подробно изложены выше. Активно развиваются новые научно-исследовательские методы и появляются примеры коммерческого применения, требующие отдельного внимательного рассмотрения и изучения [41, 47, 101].

4.8. ВИЗУАЛИЗАЦИЯ

Еще один метод *Data Mining*, который упоминают как имеющий самостоятельную ценность, – *визуализация* (англ. *Visualization*). Его суть заключается в том, чтобы обеспечить для экспертного рассмотрения данные в таком визуальном представлении, которое позволит более контрастно показать имеющиеся в данных закономерности, связи и исключения (паттерны), неочевидные при ином рассмотрении.

Для решения этой задачи используют имеющийся инструментальный графического представления данных (варьирование размерности данных, типа диаграмм и графиков, используемых цветов, форм и других характеристик отображаемых элементов), который может (а порой и должен) быть применен совместно с другими методами *Data Mining*.

5. ВЫСОКОПРОИЗВОДИТЕЛЬНАЯ ОБРАБОТКА ДАННЫХ

Применение сложных, в том числе интеллектуальных, подходов к обработке данных, представленных мега-, гига- и терабайтами, требует особого внимания к обеспечению их высокопроизводительной обработки. Выше отмечено, что в условиях стремительного наращивания объемов данных, многие стандартные вычислительные средства и традиционные алгоритмические подходы для такой сложной и затратной обработки в значительной степени не пригодны. Сформулируем несколько основных принципов организации высокопроизводительных вычислений, которые следует учитывать в практическом применении методов *Data Mining*.

Первый принцип организации высокопроизводительных вычислений заключается в увеличении производительности обработки данных за счет совершенствования вычислительной эффективности алгоритмов, позволяя применять достаточно сложную обработку данных, но в некоторых случаях доступную даже для ПЭВМ со стандартными характеристиками.

Сегодня доступны различные вычислительные мощности. Вычислительные кластеры организуют как на базе стандартных недорогих ПЭВМ, объединенных в локальной вычислительной сети, так и в специализированных центрах, оснащенных суперкомпьютерной многопроцессорной техникой [50]. Поэтому *второй принцип* определяет целесообразность использования таких подходов к обработке данных, которые применимы как на дорогостоящей суперкомпьютерной технике, так и на недорогих ПЭВМ, объединенных в сети. Причем в этом случае подразумевается, что высокопроизводительная обработка достигается двумя способами, которые могут быть использованы совместно.

Первый способ предполагает обработку исключительно за счет применения более высокопроизводительной вычислительной техники. Второй – адаптацию существующих подходов для возмож-

ности не только канонического *параллельного*, но и так называемого *распределенно-параллельного* исполнения. Обширный класс *параллельных вычислений* характеризуется возможностью одновременно-го решения одной вычислительной задачи путем ее декомпозиции [50]. Очевидно, параллельные вычисления могут быть реализованы на одной ПЭВМ в многопроцессном режиме под управлением многозадачной операционной системы [63]. Параллельные вычисления на нескольких вычислительных узлах (например, нескольких ПЭВМ, объединенных в локальной вычислительной сети или нескольких процессорах суперкомпьютера) терминологически относят к *распределенным вычислениям*. Именно поэтому, здесь и далее применяя термин *распределенно-параллельные вычисления*, будем иметь ввиду вычислительный процесс, реализуемый не только как параллельный, но и как распределенный.

Практическая организация высокопроизводительных распределенно-параллельных вычислений с использованием указанных выше принципов требует ряда важных пояснений.

Организация распределенных вычислений возможна с использованием множества различных подходов и архитектур [52]. Однако при всем многообразии возможных вариантов, принципиально отличаются два различных класса построения вычислительных систем – *системы с общей (разделяемой) памятью* и *системы с распределенной памятью* [50].

К первому варианту построения вычислительных систем относят системы с симметричной архитектурой и симметричной организацией вычислительного процесса – *SMP-системы* (англ. *Symmetric MultiProcessing*). Поддержка *SMP*-обработки данных присутствует в большинстве современных ОС при организации вычислительных процессов на многопроцессных (многоядерных) ПЭВМ [63]. Однако разделяемая память требует решения вопросов синхронизации и исключительного доступа к разделяемым данным, а построение высокопроизводительных систем в этой архитектуре затруднено технологическими сложностями объединения большого числа процессоров с единой оперативной памятью [50, 63].

Второй вариант предполагает объединение нескольких вычислительных узлов (ВУ) с собственной памятью в единой коммуникационной среде, взаимодействие между узлами в которой осуществляется путем пересылки сообщений. Причем такими ВУ могут быть как стандартные недорогие ПЭВМ, так и процессоры дорогостоящего суперкомпьютера. Такая архитектура построения вычислительной системы характеризуется большими возможностями построения высокопроизводительных вычислительных систем и значительного масштабирования доступных вычислительных мощностей. В отличие от *SMP*-систем, здесь более высокая *латентность* (существенные накладные коммуникационные расходы), негативно влияющая на производительность системы и требующая более внимательной организации распределенно-параллельной обработки [50]. Причем, в случае построения вычислительного кластера на базе ПЭВМ такая латентность, как правило, существенно выше, чем при построении кластера на базе суперкомпьютера (за счет значительно более развитой коммуникационной среды).

В соответствии с классификацией архитектур вычислительных систем М. Флинна, наибольшее распространение на практике получили две модели параллельных вычислений – *Multiple Process / Program-Multiple Data (MPMD)*, множество процессов / программ, множество данных) и *Single Process / Program-Multiple Data (SPMD)*, один процесс / программа, множество данных) [50]. Модель *MPMD* предполагает, что параллельно выполняющиеся процессы исполняют различные программы (процессы, потоки) на различных процессорах. Модель *SPMD* обуславливает работу параллельно выполняющихся программ (процессов, потоков), но исполняющих идентичный код при обработке отличных (в общем случае) массивов данных.

Очевидно, организацию распределенных вычислений целесообразно реализовать с использованием программ с параллельной обработкой данных на основе модели *SPMD*. Во-первых, это более практично из-за необходимости разрабатывать и отлаживать лишь одну программу, а во-вторых, такие программы, как правило, при-

менимы и при традиционном последовательном исполнении, что расширяет их области практической применимости.

Кроме вышеизложенных особенностей организации высокопроизводительных вычислений в различных архитектурах и на основе различных моделей, при разработке алгоритмического обеспечения параллельной обработки необходимо:

- определить фрагменты вычислений, которые могут быть исполнены параллельно;
- распределить данные по модулям локальной памяти отдельных ВУ;
- согласовать распределение данных с параллелизмом вычислений.

Важным является выполнение всех перечисленных условий, так как иначе значительные фрагменты вычислений не удастся представить как параллельно исполняемые, и реализация алгоритма в распределенно-параллельной архитектуре не позволит добиться роста производительности. Задачи распределения данных по модулям памяти и задача согласования распределения данных важны для обеспечения низкой латентности между ВУ и обеспечения возможностей масштабирования системы.

Таким образом, повышение производительности обработки данных может быть реализовано путем:

- наращивания мощности аппаратной инфраструктуры;
- применения распределенно-параллельных подходов к организации вычислений на кластерах различной стоимости и конфигурации, различных моделей параллельных вычислений и архитектур организации вычислительного процесса;
- увеличения вычислительной эффективности алгоритмов за счет оптимизации, использования приближенных эвристик вместо трудоемких расчетов и т.п.

6. ИНСТРУМЕНТЫ DATA MINING

Стремительное развитие информационных технологий, включая нарастающую интенсивность процессов сбора, хранения и обработки данных (рассмотренные подробно в гл. 1), порождает новый рынок специализированного программного обеспечения для решения задач *Data Mining*. Причем здесь справедливо выделить два различных класса такого программного обеспечения.

Один класс представляет собой прикладные пользовательские программные системы (как коммерческие, так и свободно распространяемые), как правило, реализованные в виде самостоятельных приложений или модулей (англ. *standalone application*) и предназначенные для решения конкретных предметных задач в той или иной области человеческой деятельности. Учитывая достаточно высокую сложность и междисциплинарность *Data Mining*, широкий набор потенциально доступных средств и методов (лишь основные из которых возможно рассмотреть здесь – в главах 1–4), такие системы позволяют только в некоторой степени закрыть потребность в квалифицированном анализе данных.

Учитывая эти обстоятельства, значительную популярность приобретают программно-аналитические системы, реализованные с использованием «облачной» архитектуры. В этом случае пользователю предоставляется лишь «тонкий» клиент (как правило, *web*-клиент), а мощная интеллектуальная вычислительная поддержка реализуется сервером поставщика услуг. Очевидно, это требует адаптации такой программной системы для задач пользователя со стороны экспертов. Такое привлечение профессиональной экспертизы при постановке задачи и ее адаптации к программно-вычислительной среде, а также потенциально более мощное методическое и вычислительное наполнение «облака», позволяет существенно увеличить эффективность ее решения. Рассмотрим кратко некоторые примеры таких систем в п. 6.2.

Другой класс программных систем охватывает специализированные инструменты, которые при квалифицированном примене-

нии позволяют решать задачи *Data Mining*. Рассмотрим их кратко в п. 6.1.

6.1. ПРОГРАММНЫЕ ИНСТРУМЕНТЫ ДЛЯ ВЫСОКОПРОИЗВОДИТЕЛЬНОЙ ОБРАБОТКИ ДАННЫХ

В гл. 2 приведен приоритет перечня компетенций, которыми должен обладать специалист в области *Data Mining* с точки зрения современного работодателя. Заметно, что набор компетенций, навыков и инструментария состоит не только из традиционных позиций для современного разработчика программного обеспечения (англ. *Software Engineer*), но и дополнен некоторыми отличительными элементами, требующимися при распределенно-параллельной высокопроизводительной работе с *Big Data* методами *Data Mining*. Рассмотрим кратко некоторые наиболее распространенные из них, как правило, относящиеся к свободно распространяемому программному обеспечению.

6.1.1. Программная среда

Для повышения эффективности и упрощения процессов распределенно-параллельной обработки данных, фондом *Apache Software Foundation* реализуется проект по разработке программной системы *Apache Hadoop* (часто просто *Hadoop*) разработки высокопроизводительных приложений. Данная среда характеризуется возможностью горизонтальной масштабируемости кластера путем добавления недорогих вычислительных узлов, без использования дорогостоящих суперкомпьютерных мощностей.

По состоянию на 2014 г. этот программный сервис представлен четырьмя модулями:

- *Hadoop MapReduce* (платформа программирования и выполнения распределенных *MapReduce*-вычислений);
- *HDFS* (*Hadoop Distributed File System*, распределенная файловая система);
- *YARN* (система для планирования заданий и управления ресурсами кластера);

– *Hadoop Common* (набор инфраструктурных программных библиотек и утилит, используемых для других модулей и родственных проектов типа *Mahout*, *Cassandra*, *Spark* и др.).

6.1.2. Базы данных

Важным элементом вычислительной среды обработки данных являются БД и СУБД. Необходимость обработки «больших данных», особенности которых рассмотрены в п. 2.1–2.3, диктует и новые требования к «распределенным» принципам построения и функционирования БД и систем управления ими, отличными от принятых для них реляционных аналогов. Отличают три основных свойства данных (согласованность, доступность и устойчивость к разделению), которые являются противоречивыми и добиться выполнения в одинаковой степени одновременно можно только двух из них. Современные нереляционные БД и СУБД направлены на попытку решения именно этой задачи.

Одним из примеров нереляционной распределенной БД стала БД с открытым исходным кодом *HBase*, реализуемая в развитие проекта *Apache Hadoop*. Эта БД функционирует совместно с распределенной файловой системой *HDFS* и обеспечивает отказоустойчивый способ хранения больших объемов разреженных данных.

Еще одним аналогичным, распространенным на практике примером нереляционной БД является *MongoDb* – документно-ориентированной БД, не имеющей строгой схемы данных, позволяющей добиваться высокой скорости записи и чтения, масштабируемости, но уступающей в сохранности и целостности данных.

Отметим, что в подобных БД присутствует примат масштабируемости и доступности данных над их согласованностью. В таких БД оперировать данными приходится не только использованием стандартного структурированного языка запросов *SQL*, принятого в реляционных БД, но и с помощью так называемого *NoSQL* (англ. *not only SQL*, не только *SQL*), который обеспечивает доступность и масштабируемость, но не высокую степень согласованности данных.

6.1.3. Языки программирования

Анализ многочисленных языков программирования, сопровождающих разработку и развитие программных систем анализа данных, вряд ли позволяет выявить очевидного лидера в этой области. Как и в любой другой области, выбор конкретного языка программирования, интегрированной среды разработки или компилятора зависит от множества специфических факторов. Вместе с тем нельзя не отметить возрастающую популярность языка программирования *Python*, приобретающего все большую востребованность вместе с традиционно распространенными в среде *Data Scientist* инструментами типа языка *R*, *Mathlab* или *Ruby*. Подробнее об этих и других языках программирования – в [11, 92].

6.2. ПРИМЕРЫ ПРОГРАММНЫХ СИСТЕМ

6.2.1. Примеры самостоятельных систем

В качестве примеров универсальных, полнофункциональных, распространенных статистических пакетов называют *SAS Enterprise Miner* (компания *SAS Institute*), *SPSS (SPSS Modeler Professional и SPSS Modeler Premium)*, *Statistica (StatSoft)* и др.

Большинство современных СУБД также включают поддержку функциональности *Data Mining*:

- Microsoft SQL Server Analysis Services (Microsoft Corp.);
- Oracle Business Intelligence (Oracle Corp.);
- IBM DB2 Intelligent Miner (IBM).

Кроме того, существует целый ряд систем, основанных, главным образом, на какой-то одной группе методов *Data Mining*:

- нейронные сети (BrainMaker (CSS), NeuroShell (Ward Systems Group), OWL (HyperLogic));
- деревья решений (See5/C5.0 (RuleQuest), Clementine (Integral Solutions), SIPINA (University of Lyon), IDIS (Information Discovery), KnowledgeSeeker (ANGOSS);
- генетические алгоритмы (*GeneHunter*, *Ward Systems Group*);

- алгоритмы ограниченного перебора (*WizWhy* от компании *WizSoft*);
- системы рассуждений на основе аналогичных случаев (англ. *Case Based Reasoning*; *KATE tools* (Acknosoft), *Pattern Recognition Workbench* (Unica));
- Визуализация многомерных данных (*DataMiner 3D* компании *Dimension5*).

Очевидно, упомянутыми программными системами рынок современного программного обеспечения *Data Mining* не исчерпывается. Более того, в каждой из упомянутых групп регулярно появляются новинки, направленные на снижение требований к квалификации пользователя таких систем, повышение адекватности в результатах решения задач и т.п.

6.2.2. Примеры облачных систем

Примерами «не коробочных» программных систем *Data Mining*, набирающими популярность, являются системы, реализованные в «облачной» архитектуре. Число таких примеров сегодня наиболее велико на высоко конкурентных рынках, где значима ценность предсказательной аналитики, – рынках США и Западной Европы.

Интересным примером здесь является компания *Blue Yonder*, развивающая линейку продуктов в парадигме *SaaS* (англ. *Software as a Service*, «ПО как сервис») [7]. Основным ядром создаваемого ПО является нейросетевой алгоритм оценки условной плотности распределения вероятности, разработанный в результате научных исследований на адронном коллайдере проект ЦЕРН (фр. *CERN – Conseil Européen pour la Recherche Nucléaire*, Европейский совет по ядерным исследованиям) [13]. Сегодня на основе этой разработки компания реализует целый ряд решений, предварительно адаптированных для сфер ритейла, производства, телекоммуникаций, энергетики, финансов, медиа. Заявляется, что предлагаемые решения отличаются более высокой точностью, достигаемой глубокой научной проработкой применяемых алгоритмов.

ВОПРОСЫ И ТЕМЫ ДЛЯ САМОПРОВЕРКИ

1. Какие тренды информационно-коммуникационных технологий способствовали развитию *Data Mining*?
2. Приведите примеры применения методов *Data Mining* для решения практических задач.
3. Какие области человеческой деятельности наиболее и наименее подходят для их анализа методами *Data Mining*?
4. Что понимается под *Data Mining* и *Big Data*? Почему возникла такая терминология?
5. В чем состоит суть индуктивных и дедуктивных подходов в *Data Mining*?
6. Каковы основные этапы интеллектуального анализа данных?
7. Какие классификации методов *Data Mining* существуют? Приведите примеры.
8. В чем заключается предварительная обработка данных и какова ее цель? Какие подходы при этом применяются?
9. В чем заключается оптимизация признаковового пространства? Какие методы с трансформацией и без трансформации пространства применяют и в чем их отличия?
10. В чем заключается метод классификации? Какие подходы для его реализации могут быть использованы и в чем их суть?
11. Что такое неконтролируемая классификация и какие методы применяют для ее реализации?
12. В чем заключается суть метода машины опорных векторов и в чем его преимущество перед аналогами?
13. Как работают деревья принятия решений? Какие их разновидности существуют? Каковы пределы применимости этого метода?
14. Что такое регрессия? Какие подходы применяют для ее реализации?
15. Как работают ассоциативные алгоритмы?
16. Как работают алгоритмы последовательной ассоциации?

17. Что такое обнаружение аномалий? Приведите примеры применения этого подхода и методы его реализации.

18. Что такое визуализация и какие инструменты ее реализации существуют?

19. Какие инструменты, модели и технологии существуют сегодня для реализации высокопроизводительных вычислений? Какие критерии эффективности при этом используют?

20. Приведите примеры коммерческих многофункциональных систем и свободно распространяемых решений, реализующих инструментарий *Data Mining*. Их сравнительные характеристики.

21. Архитектуры и особенности функционирования информационных систем, реализующих методы *Data Mining* как сервис.

ЛИТЕРАТУРА

1. Agrawal R., Srikant R. Mining Sequential Patterns // Proc. of the 11th Int'l Conference on Data Engineering. 1995. P. 3–14.
2. Agrawal R., Srikant R. Fast algorithms for mining association rules in large databases // Proceedings of the 20th International Conference on Very Large Data Bases. VLDB, Santiago, Chile, 1994. P. 487–499.
3. Ayres J., Flannick J., Gehrke J., Yiu T. Sequential Pattern Mining using a Bitmap Representation // ACM SIGKDD Conference. 2002. P. 429–435.
4. BaseGroup Labs. Технологии анализа данных. URL: <http://www.base-group.ru> (дата обращения: 03.03.2015).
5. Big Data Analytics Methodological Training in Statistical Data Science. URL: <http://www.statoo.com/dm> (дата обращения: 03.03.2015).
6. Bishop C.M. Neural Networks for Pattern Recognition. Oxford Univ. Press, 1995. 508 p.
7. Blue Yonder. URL: <http://www.blue-yonder.com> (дата обращения: 03.03.2015).
8. Boulding K.E. General Systems Theory – The Skeleton of Science // Management Science. 1956. № 2. P. 197–208.
9. Breiman L., Friedman J.H., Olshen R.A., Stone C.T. Classification and Regression Trees. Wadsworth, Belmont, California, 1984. 358 p.
10. Chandola V., Kumar V. Summarization – compressing data into an informative representation // Knowledge and Information Systems. New York : Springer-Verlag, 2007. Vol. 12, is. 3. P. 355–378.
11. Data Mining Community Top Resource. URL: <http://www.kdnuggets.com/> (дата обращения: 03.03.2015).
12. Deng H., Runger G., Tuv E. Bias of importance measures for multi-valued attributes and solutions // Proceedings of the 21st International Conference on Artificial Neural Networks (ICANN). 2011. P. 293–300.
13. Feindt M. A Neural Bayesian Estimator for Conditional Probability Densities, 2004. URL: <http://arxiv.org/abs/physics/0402093> (дата обращения: 03.03.2015).
14. Fine S., Scheinberg K. INCAS: An incremental active set method for SVM. Technical Report, IBM Research Labs, Haifa, 2002.
15. Galton F. Regression Towards Mediocrity in Hereditary Stature // Journal of the Anthropological Institute. 1886. № 15. P. 246–263.
16. GartnerGroup. URL: www.gartner.com (дата обращения: 03.03.2015).

17. Giacinto G., Roli F. Dynamic Classifier Selection Based on Multiple Classifier Behaviour // Pattern Recognition. 2001. № 34 (9). P. 179–181.
18. Horvath T., Yamamoto A. (eds.) Inductive Logic Programming. Series: Lecture Notes in Computer Science. Springer, 2003. Vol. 2835. P. 215–232.
19. Hyafil L., Rivest R. Constructing Optimal Binary Decision Trees is NP-complete // Information Processing Letters. 1976. № 5 (1). P. 15–17.
20. Ian H. Witten, Eibe Frank, Mark A. Hall, Morgan Kaufmann. Data Mining: Practical Machine Learning Tools and Techniques. 3rd ed. Elsevier, 2011. 629 p.
21. Jain A., Zongker D. Feature Selection: Evaluation, Application, and Small Sample Performance // IEEE Transactions on Pattern Analysis and Machine Intelligence. 1997. № 2 (19). P. 153–158.
22. Kaneko I.S., Igarashi S. Combining Multiple k-Nearest Neighbour Classifiers Using Feature Combinations // J. IECI. 2000. № 3 (2). P. 23–31.
23. Knowledge Discovery Through Data Mining: What Is Knowledge Discovery? Tandem Computers Inc., 1996.
24. Kuznetsova A.V., Sen'ko O.V., Matchak G.N., Vakhotsky V.V., Zabotina T.N., Korotkova O.V. The Prognosis of Survivance in Solid Tumor Patients Based on Optimal Partitions of Immunological Parameters Ranges // Journal Theoretical Medicine. 2000. Vol. 2. P. 317–327.
25. Petrushin V.A., Khan L. Multimedia Data Mining and Knowledge Discovery. New York : Springer-Verlag, 2006.
26. Piatetsky-Shapiro G. Knowledge Discovery in Real Databases: A Report on the IJCAI-89 Workshop // AI Magazine. 1991. № 11 (5). P. 68–70.
27. Quinlan J.R. Induction of Decision Trees // Machine Learning, 1986. № 1 (1). P. 81–106.
28. Rahm E., Do H.H. Data Cleaning: Problems and Current Approaches // IEEE Bulletin on Data Engineering. 2000. № 4 (23). P. 3–13.
29. Raymer M.L., Punch W.F., Goodman E.D., Kuhn L.A., and Jain L.C. Dimensionality reduction using genetic algorithms // IEEE Trans. on Evolutionary Computation. 2000. № 4 (2). P. 164–171.
30. Richards J.A., Xiuping Jia. Remote Sensing Digital Image Analysis: An Introduction. Berlin : Springer, 1999. 363 p.
31. SAS Institute. URL: <http://www.sas.com> (дата обращения: 03.03.2015).
32. Srikant R., Agrawal R. Mining Sequential Patterns: Generalizations and Performance Improvements // EDBT. Springer Berlin Heidelberg, 1996. P. 1–17.

33. Stanton J.M. Introduction to Data Science, Third Edition. iTunes Open Source eBook. 2012. URL: <https://itunes.apple.com/us/book/introduction-to-data-science/id529088127?mt=11> (дата обращения: 03.03.2015).
34. Wang L. (ed.). Support vector machines: theory and applications. Springer Science & Business Media, 2005. Vol. 177. 434 p.
35. Weigend A.S., Srivastava A.N. Predicting Conditional Probability Distributions: A Connectionist Approach // International Journal of Neural Systems. 1995. № 2 (6). P. 109–118.
36. Widrow B., Lehr M.A. 30 Years of Adaptive Neural Networks: Perceptron, Madaline, and Backpropagation // Proceedings of the IEEE. 1990. № 9 (78). P. 1415–1442.
37. Zhirnova I.G., Kuznetsova A.B., Rebrova O.Yu., Labunsky D.A., Komelkova L.V., Poleshchuk V.V., Sen'ko O.V. Logical and Statistical Approach for the Analysis of Immunological Parameters in Patients with Wilson's Disease // Russian Journal of Immunology. 1998. № 2 (3). P. 174–184.
38. Абдикеев Н.М. Когнитивная бизнес-аналитика. М. : ИНФРА-М, 2011. 510 с.
39. Абдикеев Н.М., Данько Т.П., Ильдеменов С.В. и др. Реинжиниринг бизнес-процессов. Курс МВА. М. : Эксмо, 2005. 592 с.
40. Абдикеев Н.М., Киселев А.Д. Управление знаниями в корпорации и реинжиниринг бизнеса. М. : Инфра-М, 2011. 382 с.
41. Аграновский А.В., Репалов С.А., Хади Р.А., Якубец М.Б. О недостатках современных систем обнаружения вторжений // Информационные технологии. 2005. № 5. С. 39–43.
42. Айвазян С.А., Вежаева З.И., Староверов О.В. Классификация многомерных наблюдений. Статистика, 1974. 240 с.
43. Айвазян С.А., Бухштабер В.М., Енюков И.С., Мешалкин Л.Д. Прикладная статистика: Классификация и снижение размерности. М.: Финансы и статистика, 1989. 607 с.
44. Асеев М.Г., Баллюзек М.Ф., Дюк В.А. Разработка медицинских экспертных систем средствами технологий Data Mining. URL: <http://www.datadiver.nw.ru> (дата обращения: 03.03.2015).
45. Барсегян А.А., Куприянов М.С., Степаненко В.В., Холод И.И. Методы и модели анализа данных: OLAP и Data Mining, СПб. : БХВ-Петербург, 2007. 336 с.
46. Большие данные (Big Data). URL: <http://www.tadviser.ru> (дата обращения: 03.03.2015).

47. Еремеев В.Б. Разработка математического и программного обеспечения активного мониторинга вычислительной сети // Автоматизация и информатика. Вести высших учебных заведений Черноземья. 2008. № 4 (14). URL: http://www.stu.lipetsk.ru/files/materials/2408/2008_04_013.pdf (дата обращения: 03.03.2015).
48. Вапник В.Н. Восстановление зависимостей по эмпирическим данным. М. : Наука, 1979. 488 с.
49. Вапник В.Н., Червоненкис А.Я. Теория распознавания образов. Статистические проблемы обучения. М. : Наука, 1974. 416 с.
50. Воеводин В.В., Воеводин Вл.В. Параллельные вычисления. СПб. : БХВ-Петербург, 2004. 608 с.
51. Гик Дж., ван. Прикладная общая теория систем. М. : Мир, 1981. 731 с.
52. Головкин Б.А. Параллельные вычислительные системы. М. : Наука, 1980. 520 с.
53. Давыдов А.А. Системная социология: анализ мультимедийной информации в Интернете. URL: http://www.isras.ru/files/File/Publication/Multimedia_Information_DavydovA.pdf (дата обращения: 03.03.2015).
54. Доровских И.В., Кузнецова А.В., Сенько О.В., Реброва О.Ю. Прогноз динамики депрессивных синдромов в остром периоде сотрясения головного мозга по показателям первичного обследования (с использованием логико-статистических методов) // Социальная и клиническая психиатрия. 2003. № 4. С. 18–24.
55. Дрейпер Н., Смит Г. Прикладной регрессионный анализ. Множественная регрессия. 3-е изд. М. : Диалектика, 2007. 912 с.
56. Дуда Р., Харт П. Распознавание образов : пер. с англ. М. : Наука, 1981. 450 с.
57. Дюк В.А. Обработка данных на ПК в примерах. СПб. : Питер, 1997. 240 с.
58. Дюк В., Самойленко А. Data Mining : учеб. курс. СПб. : Питер, 2001. 386 с.
59. Дюран Б., Оделл П. Кластерный анализ : пер. с англ. М. : Статистика, 1977. 128 с.
60. Епанечников В.А. Непараметрическая оценка многомерной плотности вероятности // Теория вероятностей и ее применения. 1969. Т. 14, вып. 1. С. 156–161.
61. Загоруйко Н.Г. Прикладные методы анализа данных и знаний. Новосибирск : Изд-во ИМ СО РАН, 1999. 268 с.
62. Загоруйко Н.Г. Прикладные методы анализа данных и знаний. Новосибирск : Изд-во ИМ СО РАН, 1999. 270 с.

63. Замятин А.В. Операционные системы : учеб. пособие. Томск : Изд-во Том. политехн. ун-та, 2010. 167 с.
64. Замятин А.В., Марков Н.Г. Анализ динамики земной поверхности с использованием данных дистанционного зондирования Земли. М. : Физматлит, 2007. 176 с.
65. Замятин А.В., Марков Н.Г., Напрюшкин А.А. Адаптивный алгоритм классификации с использованием текстурного анализа для автоматизированной интерпретации аэрокосмических изображений // Исследование Земли из космоса. 2004. № 2. С. 32–40.
66. Киселев М., Соломатин Е. Средства добычи знаний в бизнесе и финансах // Открытые системы. 1997. № 4. С. 41–44.
67. Кислова О.Н. Интеллектуализация информационных технологий как фактор развития интеллектуального анализа социологических данных // Методологія, теорія та практика соціологічного аналізу сучасного суспільства. Збірник наукових праць. Харків : Видавничий центр ХНУ імені В. Н. Каразіна, 2009. С. 318–324.
68. Консалтинговая компания IDC. URL: <http://idc-group.ru> (дата обращения: 03.03.2015).
69. Костюкова Н.И. Применение технологии Data Mining для решения задач оптимизации проектирования сложных технических систем // Альманах современной науки и образования. Тамбов : Грамота, 2010. № 5 (36). С. 60–61.
70. Костюкова Н.И. Принятие решений в условиях риска // Приложение к журналу «Открытое образование». 2010. С. 90–93.
71. Костюкова Н.И. Создание новой технологии в среде C++, JAVA на базе вычисления группы, допускаемой дифференциальными уравнениями // Альманах современной науки и образования. Тамбов : Грамота, 2010. № 7 (38). С. 59–61.
72. Костюкова Н.И. Технология Data Mining в задачах исследования сетевого трафика // Приложение к журналу «Открытое образование». 2010. С. 148–149.
73. Костюкова Н.И., Залевский А.А., Москвин Н.В. Разработка системы поддержки принятия решений // Альманах современной науки и образования. Тамбов : Грамота, 2010. № 5 (36). С. 59–60.
74. Костюкова Н.И., Кудинов А.Е. Математические модели лечения с учетом эффективности // Альманах современной науки и образования. Тамбов : Грамота, 2010. № (34). С. 17–21.

75. Костюкова Н.И. Система принятия решений в области медицинской диагностики и выбора оптимальных решений по технологии Data Mining // Приложение к журналу «Открытое образование». 2010. С. 145–146.
76. Костюкова Н.И. Создание автоматизированной системы анализа технологии добычи данных для обнаружения сетевого вторжения // Приложение к журналу «Открытое образование». 2010. С. 149–151.
77. Костюкова Н.И., Кудинов А.Е. Автоматизация научных исследований в области медицины с применением технологии Data Mining // Альманах современной науки и образования. Тамбов : Грамота, 2010. № 3 (34), ч. 1. С. 22–24.
78. Костюкова Н.И., Родин Е.В. Система поддержки принятия решений для отраслей, связанных с риском // Альманах современной науки и образования. Тамбов : Грамота, 2010. № 7 (38). С. 41–44.
79. Костюкова Н.И., Кудинов А.Е. Статистические методы в медицине // Альманах современной науки и образования. Тамбов : Грамота, 2011. № 4 (47). С. 100–107.
80. Кречетов Н. Продукты для интеллектуального анализа данных // Рынок программных средств. 1997. № 14–15. С. 32–39.
81. Кузнецов В.А., Сенько О.В., Кузнецова А.В. и др. Распознавание нечетких систем по методу статистически взвешенных синдромов и его применение для иммуногематологической нормы и хронической патологии // Химическая физика. 1996. Т. 15, № 1. С. 81–100.
82. Кузнецова А.В., Сенько О.В. Возможности использования методов Data Mining при медико-лабораторных исследованиях для выявления закономерностей в массивах данных // Врач и информационные технологии. 2005. № 2. С. 49–56.
83. Кузнецова А.В. Диагностика и прогнозирование опухолевого роста по иммунологическим данным с помощью методов синдромного распознавания : автореф. дис. ... канд. биол. наук. М., 1995. 23 с.
84. Лапко А.В., Ченцов С.В. Непараметрические системы обработки информации : учеб. пособие. М. : Наука, 2000. 350 с.
85. Назаров Л.Е. Применение многослойных нейронных сетей для классификации земных объектов на основе анализа многозональных сканерных изображений // Исследование Земли из космоса. 2000. № 6. С. 41–50.
86. Напрюшкин А.А. Алгоритмическое и программное обеспечение системы интерпретации аэрокосмических изображений для решения за-

- дач картирования ландшафтных объектов : дис. ... канд. техн. наук. Томск, 2002. 168 с.
87. Нейроинформатика / А.Н. Горбань, В.Л. Дунин-Барковский, А.Н. Кирдин и др. Новосибирск : Наука, 1998. 296 с.
 88. Нейронные сети. Statistica Neural Networks : пер. с англ. М. : Горячая линия-Телеком, 2000. 182 с.
 89. Прикладная статистика: Классификации и снижение размерности : справ. изд. / С.А. Айвазян, В.М. Бухштабер, И.С. Енюков, Л.Д. Мешалкин ; под ред. С.А. Айвазяна. М. : Финансы и статистика, 1989. 607 с.
 90. Прэйт У. Цифровая обработка изображений : пер. с англ. М. : Мир, 1982. Кн. 2. 480 с.
 91. Реброва О.Ю. Статистический анализ медицинских данных. Применение пакета прикладных программ STATISTICA. М. : Медиа Сфера, 2002. 305 с.
 92. Рейтинг языков программирования для data mining. URL: <http://computerscience.ru/posts/48> (дата обращения: 03.03.2015).
 93. Рекрутинговая компания. URL: www.indeed.com (дата обращения: 03.03.2015).
 94. Татарова Г.Г. Методология анализа данных в социологии (введение) : учеб. для вузов. М. : Nota Bene, 1999. 224 с.
 95. Толстова Ю.Н. Анализ социологических данных. Методология, дескриптивная статистика, изучение связей между номинальными признаками. М. : Научный мир, 2000. 352 с.
 96. Ту Д., Гонсалес Р. Принципы распознавания образов : пер. с англ. М. : Мир, 1978. 412 с.
 97. Финн В.К. Об интеллектуальном анализе данных // Новости искусственного интеллекта. 2004. № 3. С. 3–18.
 98. Что такое Data Mining. URL: <http://www.iso.ru> (дата обращения: 03.03.2015).
 99. Чубукова И.А. Курс Data Mining. URL: <http://www.intuit.ru/departments/database/datamining> (дата обращения: 03.03.2015).
 100. Шапиро Е.И. Непараметрические оценки плотности вероятности в задачах обработки результатов наблюдений // Зарубежная радиоэлектроника. 2000. № 2. С. 3–22.
 101. Якубец М.Б. Обнаружение сетевых атак методом поиска аномалий на основе вероятностного и верификационного моделирования // Искусственный интеллект. 2006. № 3. С. 816–823.

Учебное издание

Александр Владимирович Замятин

**ВВЕДЕНИЕ В ИНТЕЛЛЕКТУАЛЬНЫЙ
АНАЛИЗ ДАННЫХ**

Учебное пособие

Редактор Ю.П. Готфрид
Оригинал-макет А.И. Лелоюр
Дизайн обложки Л.Д. Кривцовой

Подписано к печати 24.02.2016 г. Формат 60×84¹/₁₆.

Бумага для офисной техники. Гарнитура Times.

Усл. печ. л. 6,9.

Тираж 30 экз. Заказ № 1540.

Отпечатано на оборудовании
Издательского Дома
Томского государственного университета
634050, г. Томск, пр. Ленина, 36
Тел. 8+(382-2)–53-15-28
Сайт: <http://publish.tsu.ru>
E-mail: rio.tsu@mail.ru

ISBN 978-5-94621-531-2

