O'REILLY®

Практическая статистика для специалистов Data Science

50 ВАЖНЕЙШИХ ПОНЯТИЙ

Питер Брюс и Эндрю Брюс

bhv®

Practical Statistics for Data Scientists

50 Essential Concepts

Peter Bruce and Andrew Bruce

Beijing • Boston • Farnham • Sebastopol • Tokyo O'REILLY®

Питер Брюс, Эндрю Брюс

Практическая статистика для специалистов Data Science

50 ВАЖНЕЙШИХ ПОНЯТИЙ

Санкт-Петербург
«БХВ-Петербург»
2018

УДК 004.6+519.2
ББК 32.81+22.172
Б89

Брюс, П.

Б89 Практическая статистика для специалистов Data Science: Пер. с англ. / П. Брюс, Э. Брюс. — СПб.: БХВ-Петербург, 2018. — 304 с.: ил.

ISBN 978-5-9775-3974-6

Книга рассчитана на специалистов в области Data Science, обладающих некоторым опытом работы с языком программирования R и имеющих предварительное понятие о математической статистике. В ней в удобной и легкодоступной форме представлены ключевые понятия из статистики, которые относятся к науке о данных, а также объяснено, какие понятия важны и полезны с точки зрения науки о данных, какие менее важны и почему. Подробно раскрыты темы: разведочный анализ данных, распределения данных и выборок, статистические эксперименты и проверка значимости, регрессия и предсказание, классификация, статистическое машинное обучение и обучение без учителя.

Для аналитиков данных

УДК 004.6+519.2
ББК 32.81+22.172

Группа подготовки издания:

Руководитель проекта	*Евгений Рыбаков*
Зав. редакцией	*Екатерина Капалыгина*
Компьютерная верстка	*Ольги Сергиенко*
Оформление обложки	*Марины Дамбиевой*

"БХВ-Петербург", 191036, Санкт-Петербург, Гончарная ул., 20.

ISBN 978-1-491-95296-2 (англ.)
ISBN 978-5-9775-3974-6 (рус.)

Оглавление

Мы хотим посвятить эту книгу памяти наших родителей,
Виктора Г. Брюса и Нэнси С. Брюс, которые воспитали в нас
страсть к математике и точным наукам,
а также нашим первым учителям, Джону У. Тьюки и Джулиану Саймону,
и нашему верному другу, Джеффу Уотсону, который вдохновил нас на то,
чтобы мы посвятили свою жизнь статистике

Об авторах

Питер Брюс основал и расширил Институт статистического образования Statistics.com, который теперь предлагает порядка 100 курсов в области статистики, из которых примерно половина предназначена для аналитиков данных. Нанимая в качестве преподавателей ведущих авторов и шлифуя маркетинговую стратегию для привлечения внимания профессиональных аналитиков данных, Питер развил широкое представление о целевом рынке и свои собственные экспертные знания для его завоевания.

Эндрю Брюс имеет более чем 30-летний стаж работы в области статистики и науки о данных в академической сфере, правительстве и бизнесе. Он обладает степенью кандидата наук в области статистики Вашингтонского университета и опубликовал несколько работ в рецензируемых журналах. Он разработал статистико-ориентированные решения широкого спектра задач, с которыми сталкиваются разнообразные отрасли, начиная с солидных финансовых фирм до интернет-стартапов, и располагает глубоким пониманием практики науки о данных.

Предисловие

Книга рассчитана на аналитика данных, обладающего некоторым опытом работы с языком программирования R и имеющего предшествующий (возможно, обрывочный или сиюминутный) контакт с математической статистикой. Мы оба, авторы этой книги, пришли в мир науки о данных из мира статистики, и поэтому у нас есть определенное понимание того вклада, который статистика может привнести в науку о данных, как прикладную дисциплину. В то же время мы хорошо осведомлены об ограничениях традиционного статистического обучения: статистика как дисциплина насчитывает полтора столетия, и большинство учебников и курсов по статистике отягощены кинетикой и инерцией океанского лайнера.

В основе настоящей книги лежат две цели:

- представить в удобной, пригодной для навигации и легкодоступной форме ключевые понятия из статистики, которые относятся к науке о данных;
- объяснить, какие понятия важны и полезны с точки зрения науки о данных, какие менее важны и почему.

Чего ожидать

Ключевые термины

Наука о данных — это сплав многочисленных дисциплин, включая статистику, информатику, информационные технологии и конкретные предметные области. В результате при упоминании конкретной идеи могут использоваться несколько разных терминов. Ключевые термины и их синонимы в данной книге будут выделяться в специальной выноске, такой как эта.

Условные обозначения, принятые в книге

В книге используются следующие условные обозначения.

- *Курсив* указывает новые термины.
- **Полужирный шрифт** — URL-адреса, адреса электронной почты.
- `Моноширинный шрифт` используется для распечаток программ, а также внутри абзацев для ссылки на элементы программ, такие как переменные или имена функ-

ций, базы данных, типы данных, переменные окружения, операторы и ключевые слова.

- *Моноширинный шрифт курсивом* показывает текст, который должен быть заменен значениями пользователя либо значениями, определяемыми по контексту.

Данный элемент обозначает подсказку или совет.

Данный элемент обозначает общее замечание.

Данный элемент обозначает предупреждение или предостережение.

Использование примеров кода

Дополнительный материал (примеры кода, упражнения и пр.) доступен для скачивания по адресу **https://github.com/andrewgbruce/statistics-for-data-scientists**.

Эта книга предназначена для того, чтобы помочь вам решить ваши задачи. В целом, если код примеров предлагается вместе с книгой, то вы можете использовать его в своих программах и документации. Вам не нужно связываться с нами с просьбой о разрешении, если вы не воспроизводите значительную часть кода. Например, написание программы, которая использует несколько фрагментов кода из данной книги, официального разрешения не требует.

Адаптированный вариант примеров в виде электронного архива вы можете скачать по ссылке **ftp://ftp.bhv.ru/9785977539746.zip**, которая доступна также со страницы книги на сайте **www.bhv.ru**.

Благодарности

Авторы выражают признательность всем, кто помог воплотить эту книгу в реальность.

Герхард Пилхер (Gerhard Pilcher), генеральный директор консалтинговой фирмы Elder Research в области глубинного анализа данных, был свидетелем ранних черновиков книги и предоставил нам подробные и полезные поправки и комментарии. Так же как и Энья Макгверк (Anya McGuirk) и Вей Сяо (Wei Xiao), специалисты

в области статистики в SAS, и Джэй Хилфигер (Jay Hilfiger), член авторского коллектива O'Reilly, которые предоставили полезные рекомендации на первоначальных стадиях работы над книгой.

В издательстве O'Reilly Шэннон Катт (Shannon Cutt) в дружеской атмосфере сопровождал нас в течение всего процесса публикации и в меру подстегивал нашу работу, в то время как Кристен Браун (Kristen Brown) гладко провела нашу книгу через производственную стадию. Рэйчел Монагэн (Rachel Monaghan) и Элайю Сассмэн (Eliahu Sussman) бережно и терпеливо проводили корректировку и улучшение нашего текста, тогда как Эллен Траутмэн-Зайг (Ellen Troutman-Zaig) подготовила предметный указатель. Мы также благодарим Мари Богуро (Marie Beaugureau), которая инициировала наш проект в O'Reilly, и Бена Бенгфорда (Ben Bengfort), автора O'Reilly и преподавателя в statistics.com, представившего нас издательству O'Reilly.

Мы извлекли большую пользу из многих бесед, которые Питер провел за последние годы с Галитом Шмуели (Galit Shmueli), соавтором других книжных проектов.

Наконец, мы хотели бы особо поблагодарить Элизабет Брюс (Elizabeth Bruce) и Дебору Доннелл (Deborah Donnell), чьи терпение и поддержка сделали это начинание возможным.

Комментарий переводчика

Прилагаемый к настоящей книге программный код протестирован в среде Windows 10 с использованием действующих версий программных библиотек (время перевода книги — октябрь-ноябрь 2017 г.). При тестировании исходного кода за основу взята среда R версии 3.4.2.

Адаптированный и скорректированный исходный код примеров лучше всего разместить в подпапке домашней папки пользователя. Например:

/home/r_projects/statistics-for-data-scientists-master

или

C:\Users\[*ИМЯ_ПОЛЬЗОВАТЕЛЯ*]\r_projects\statistics-for-data-scientists-master

Структура папки с прилагаемыми примерами такова:

data	Наборы данных, используемые в книге и в сценариях R
figures	Графики, полученные в результате выполнения сценариев R
src	Исходный код примеров в виде сценариев R

Разведочный анализ данных

Статистика как дисциплина получила свое развитие главным образом в прошлом столетии. Теория вероятностей — математический фундамент статистики — разрабатывалась с XVII по XIX в. на основе работ Томаса Байеса (Thomas Bayes), Пьера-Симона Лапласа (Pierre-Simon Laplace) и Карла Гаусса (Carl Gauss). В отличие от чисто теоретической природы вероятности, статистика является прикладной наукой, занимающейся анализом и моделированием данных. Современная статистика как строгая научная дисциплина восходит корнями к концу 1800-х гг. — Фрэнсису Гальтону (Francis Galton) и Карлу Пирсону (Karl Pearson). Р. А. Фишер (R. A. Fisher) в начале XX в. был ведущим новатором современной статистики, который ввел в употребление такие ключевые понятия, как *планирование эксперимента* и *оценка максимального правдоподобия*. Эти и многие другие статистические понятия в основном находятся в отдаленных уголках науки о данных. Главная цель настоящей книги состоит в том, чтобы помочь высветить эти понятия и разъяснить их важность — или ее отсутствие — в контексте науки о данных и больших данных.

В данной главе основное внимание уделяется первому шагу в любом проекте науки о данных: разведке данных. *Разведочный анализ данных* (exploratory data analysis, EDA) — это сравнительно новая область статистики. Классическая статистика фокусировалась почти исключительно на *статистическом выводе*, т. е. иногда сложном наборе процедур для получения выводов о популяциях (или генеральных совокупностях) на основе небольших выборок. В 1962 г. Джон У. Тьюки (John W. Tukey, рис. 1.1) в своей концептуальной статье "Будущее анализа данных" [Tukey-1962] призвал к преобразованию статистики. Он предложил новую научную дисциплину под названием *анализ данных*, которая включала статистический вывод в качестве всего лишь одного из компонентов. Тьюки наладил связи с инженерным и вычислительным сообществами (он ввел термины *"бит"* — от англ. *binary digit*, и *"программное обеспечение"*), а его исходные принципы оказались удивительно прочными и формируют часть фундамента науки о данных. Область разведочного анализа данных появилась благодаря книге Тьюри *"Анализ результатов наблюдений"* [Tukey-1977], теперь уже ставшей классической.

Благодаря доступности вычислительных мощностей и выразительному программному обеспечению для анализа данных разведочный анализ данных эволюционировал далеко за пределы своей исходной области. Ключевыми факторами этой дисциплины явились быстрая разработка новой технологии, доступ к более разнообразным и бóльшим по объему данным и более широкое применение количественного анализа во множестве дисциплин. Дэвид Донохо (David Donoho), препода-

ватель статистики в Стэнфордском университете и прежний выпускник Тьюки, написал превосходную статью "50 лет науки о данных" на основе своей презентации на семинаре в честь 100-летия Тьюки, проходившем в Принстоне, шт. Нью-Джерси [Donoho-2015]. В ней Донохо прослеживает возникновение науки о данных к новаторскому вкладу Тьюки в анализ данных.

Рис. 1.1. Джон Тьюки, выдающийся статистик, чьи идеи, разработанные более чем 50 лет назад, формируют фундамент науки о данных

Элементы структурированных данных

Данные поступают из многих источников: показаний датчиков, событий, текста, изображений и видео. *Интернет вещей* (Internet of Things, IoT) извергает потоки информации. Значительная часть этих данных не структурирована: изображения представляют собой набор пикселов, при этом каждый пиксел содержит информацию о цвете в формате RGB (красный, зеленый, синий). Тексты состоят из последовательностей словарных и несловарных символов, часто разбитых на разделы, подразделы и т. д. Потоки нажатий клавиш представляют собой последовательности действий пользователя, взаимодействующего с приложением или веб-страницей. По сути дела, основная задача науки о данных состоит в том, чтобы переработать этот поток сырых данных в информацию, полезную в практической деятельности. Для применения рассмотренных в этой книге статистических понятий неструктурированные сырые данные должны быть обработаны и помещены в структурированную форму — подобно той, которая может появляться из реляционной базы данных — либо быть собранными для статистического исследования.

Ключевые термины

Непрерывные данные (continuous)

Данные, которые могут принимать любое значение в интервале.

Синонимы: интервал, число с плавающей точкой, числовое значение.

Дискретные данные (discrete)

Данные, которые могут принимать только целочисленные значения, такие как количественные значения.

Синонимы: целое число, количество.

Категориальные данные (categorical)

Данные, которые могут принимать только определенный набор значений, в частности набор возможных категорий.

Синонимы: перечисления, перечислимые данные, факторы, именованные данные, полихотомические данные.

Двоичные данные (binary)

Особый случай категориальных данных всего с двумя категориями значений (0/1, истина/ложь).

Синонимы: дихотомический, логический, флаг, индикатор, булево значение.

Порядковые данные (ordinal)

Категориальные данные с явно выраженной упорядоченностью.

Синонимы: порядковый фактор.

Существует два основных типа структурированных данных: числовой и категориальный. Числовые данные поступают в двух формах: *непрерывной*, как например скорость ветра или продолжительность времени, и *дискретной*, как например количество возникновений события. *Категориальные* данные принимают только фиксированный набор значений, например тип экрана телевизора (плазма, LCD, LED и т. д.) или название штата (Алабама, Аляска и т. д.). *Двоичные* данные представляют собой важный особый случай категориальных данных. Эти данные принимают только одно из двух значений, таких как 0/1, да/нет или истина/ложь. Еще один полезный тип категориальных данных представлен *порядковыми* данными, в которых категории упорядочены; их примером является числовая характеристика (1, 2, 3, 4 или 5).

Зачем заморачиваться таксономией типов данных? Оказывается, что в целях анализа данных и предсказательного моделирования тип данных играет важную роль для определения типа визуального отображения, анализа данных либо статистической модели. По сути дела, в программных системах для науки о данных, таких как R и Python, эти типы данных используются для улучшения вычислительный производительности. А более важно, что тип данных переменной определяет то, каким образом программная система будет обращаться с вычислениями для этой переменной.

У разработчиков программного обеспечения (ПО) и программистов баз данных (БД) может возникнуть вопрос: зачем в аналитике нужны понятия *"категориальные"* и *"порядковые"* данные? В конце концов, категории являются просто набором текстовых (либо числовых) значений, и основная БД автоматически работает с их внутренним представлением. Однако четкая идентификация данных как категориальных, в отличие от текстовых, действительно предлагает некоторые преимущества.

- Знание, что данные категориальные, может служить сигналом для программной системы о том, каким образом должны вести себя статистические процедуры, такие как создание графика или подгонка модели. В частности, в R и Python порядковые данные могут быть представлены как порядковый фактор `ordered.factor`, сохраняя определенную пользователем упорядоченность в графиках, таблицах и моделях.
- Могут быть оптимизированы хранение и индексация данных (как в реляционной базе данных).
- Возможные значения, которые принимает конкретная категориальная переменная, реализуются в ПО (как, например, перечисление `enum`).

Третье "преимущество" может привести к непреднамеренному или неожиданному поведению: по умолчанию функции импорта данных в R (например, read.csv) ведут себя таким образом, что автоматически преобразуют столбец текста в `factor`. Последующие операции на этом столбце будут исходить из предположения, что единственно допустимыми значениями для этого столбца являются значения, которые были импортированы первоначально, и присвоение нового текстового значения выдаст предупреждение и сообщение об отсутствии значения NA (not available).

Ключевые идеи для структурированных данных

- В программной системе данные обычно классифицируются по типу.
- Тип данных может быть непрерывным, дискретным, категориальным (который включает двоичный тип) и порядковым.
- Типизация данных в программной системе сигнализирует программной системе, каким образом обрабатывать данные.

Дополнительные материалы для чтения

- Типы данных могут вызывать путаницу, поскольку одни и те же данные можно отнести к разным типам, и таксономия в одной программной системе может отличаться от таксономии в другой. Веб-сайт R-tutorial знакомит с таксономией языка R (**http://www.r-tutor.com/r-introduction/basic-data-types**).
- В БД применяется более подробная классификация типов данных, включая учет уровней прецизионности, фиксированную либо переменную длину полей и мно-

гое другое (см. руководство по SQL W3Schools guide for SQL — **http://www.w3schools.com/sql/sql_datatypes_general.asp**).

Прямоугольные данные

В науке о данных типичной опорной конструкцией для анализа является объект с *прямоугольными данными* наподобие электронной таблицы или таблицы базы данных.

Ключевые термины

Кадр данных (data frame)

Прямоугольные данные (как в электронной таблице) — это типичная структура данных для статистических и машинно-обучаемых моделей.

Признак (feature)

Столбец в таблице принято называть *признаком.*

Синонимы: атрибут, вход, предиктор, переменная.

Исход (outcome)

Многие проекты науки о данных сопряжены с предсказанием *исхода* — нередко в формате да/нет (например, в табл. 1.1 это ответ на вопрос "были ли торги состязательными или нет?"). Для предсказания *исхода* в эксперименте или статистическом исследовании иногда используются *признаки.*

Синонимы: зависимая переменная, отклик, цель, выход.

Записи (records)

Строку в таблице принято называть *записью*.

Синонимы: случай, образец, прецедент, экземпляр, наблюдение, шаблон, паттерн, выборка.

Прямоугольные данные по существу представляют собой двумерную матрицу, в которой строки обозначают записи (случаи), а столбцы — признаки (переменные). Исходно данные поступают в такой форме не всегда: неструктурированные данные (например, текст) необходимо обработать и привести к такому виду, чтобы их можно было представить как набор признаков в прямоугольных данных *(см. разд. "Элементы структурированных данных" ранее в этой главе)*. Данные в реляционных БД должны быть извлечены и помещены в единственную таблицу для большинства аналитических и модельных задач.

В табл. 1.1 показана смесь измерительных и количественных данных (например, длительность и цена) и категориальных данных (например, категория и валюта). Как уже упоминалось ранее, специальной формой категориальной переменной является двоичная переменная (да/нет или 0/1), которую можно увидеть в самом правом столбце табл. 1.1 — это индикаторная переменная, показывающая, были ли торги состязательными.

Таблица 1.1. *Типичный формат данных*

Категория	Валюта	Рейтинг продавца	Длительность	Дань закрытия	Цена закрытия	Цена открытия	Конкурентноспособность?
Music/Movie/Game	US	3249	5	Mon	0,01	0,01	0
Music/Movie/Game	US	3249	5	Mon	0,01	0,01	0
Automotive	US	3115	7	Tue	0,01	0,01	0
Automotive	US	3115	7	Tue	0,01	0,01	0
Automotive	US	3115	7	Tue	0,01	0,01	0
Automotive	US	3115	7	Tue	0,01	0,01	0
Automotive	US	3115	7	Tue	0,01	0,01	1
Automotive	US	3115	7	Tue	0,01	0,01	1

Кадры данных и индексы

Традиционные таблицы БД имеют один или несколько столбцов, называемых *индексом*. Он может значительно повысить эффективность определенных SQL-запросов. В Python при использовании библиотеки `pandas` основной прямоугольной структурой данных является объект `DataFrame`, содержащий таблицу данных. По умолчанию для `DataFrame` создается автоматический целочисленный индекс, который основывается на порядке следования строк. В программной библиотеке `pandas` для повышения эффективности определенных операций также можно задавать многоуровневые/иерархические индексы.

В R основной прямоугольной структурой данных является объект `data.frame`, кадр данных. Объект `data.frame` тоже имеет неявный целочисленный индекс на основе порядка следования строк. Хотя посредством атрибута `row.names`[1] можно создать пользовательский ключ, нативный (родной) для R, объект `data.frame` не поддерживает задаваемые пользователем или многоуровневые индексы. Для преодоления этого недостатка широкое распространение получили два новых программных пакета: `data.table` и `dplyr`. Оба пакета поддерживают многоуровневые индексы и обеспечивают значительное ускорение в работе с объектом `data.frame`.

Различия в терминологии

Терминология для прямоугольных данных может вызывать путаницу. В статистике и науке о данных используются разные термины, которые говорят об одном и том же. Для статистиков в модели существуют предикторные переменные, которые используются для предсказания отклика либо зависимой переменной. В отличие от них для аналитика данных существуют признаки,

[1] Атрибут кадра данных `row.names` — это символьный вектор длиной, которая соответствует числу строк в кадре данных, без дубликатов и пропущенных значений. — *Прим. пер.*

которые применяются для предсказания целевой переменной. В особенности сбивает с толку один синоним: специалисты по информатике используют термин "выборка" для обозначения одиночной строки, тогда как для статистика выборка означает набор строк.

Непрямоугольные структуры данных

Помимо прямоугольных данных существуют и другие структуры данных.

Временно́й ряд содержит последовательные данные измерений одной и той же переменной. Эти данные представляют собой сырой материал для статистических методов предсказания, и они также являются ключевым компонентом данных, производимых устройствами — Интернет вещей.

Пространственные структуры данных, которые используются в картографической и геопространственной аналитике, более сложны и вариативны, чем прямоугольные структуры данных. В их *объектном* представлении центральной частью данных являются объект (например, дом) и его пространственные координаты. В *полевой* проекции, в отличие от него, основное внимание уделяется небольшим единицам пространства и значению соответствующего метрического показателя (яркости пиксела, например).

Графовые (или сетевые) структуры данных используются для представления физических, социальных и абстрактных связей. Например, граф социальной сети, такой как Facebook или LinkedIn, может представлять связи между людьми в сети. Соединенные дорогами центры распределения являются примером физической сети. Графовые структуры широко применяются в определенных типах задач, таких как оптимизация сети и рекомендательные системы.

В науке о данных каждый из этих типов данных имеет свою специализированную методологию. В центре внимания этой книги находятся прямоугольные данные — основополагающий структурный элемент в предсказательном моделировании.

Графы и графики в статистике

В информатике и информационных технологиях термин "граф" (graph) обычно обозначает описание связей среди объектов и основную структуру данных. В статистике термин "график" (graph) используется для обозначения самых разных графиков и визуализаций, а не только связей между объектами, и этот термин применяется исключительно для обозначения визуализаций, а не структуры данных.

Ключевые идеи для прямоугольных данных

- В науке о данных базовой структурой данных является прямоугольная матрица, в которой строки — это записи, а столбцы — переменные (признаки).
- Терминология может вызывать путаницу, поскольку существует множество синонимов, вытекающих из разных дисциплин, которые вносят свой вклад в науку о данных (статистика, информатика и информационные технологии).

Дополнительные материалы для чтения

- Документация по кадрам данных в R (**https://stat.ethz.ch/R-manual/R-devel/library/base/html/data.frame.html**).
- Документация по кадрам (таблицам) данных в Python (**http://pandas.pydata.org/pandas-docs/stable/dsintro.html#dataframe**).

Оценки центрального положения

Переменные с измерительными или количественными данными могут иметь тысячи различных значений. Основной этап исследования данных состоит в получении "типичного значения" для каждого признака (переменной): оценки того, где расположено большинство данных (т. е. их центральной тенденции).

Ключевые термины

Среднее (mean)

Сумма всех значений, деленная на количество значений.

Синоним: среднее арифметическое.

Среднее взвешенное (weighted mean)

Сумма произведений всех значений на их веса, деленная на сумму весов.

Синоним: среднее арифметическое взвешенное.

Медиана (median)

Такое значение, при котором половина сортированных данных находится выше и ниже данного значения.

Синоним: 50-й процентиль.

Медиана взвешенная (weighted median)

Такое значение, при котором половина суммы весов находится выше и ниже сортированных данных.

Среднее усеченное (trimmed mean)

Среднее число всех значений после отбрасывания фиксированного числа предельных значений.

Синоним: обрезанное среднее.

Робастный (robust)

Не чувствительный к предельным значениям.

Синоним: устойчивый.

Выброс (outlier)

Значение данных, которое сильно отличается от большинства данных.

Синоним: предельное значение.

На первый взгляд обобщить данные довольно тривиально: просто взять *среднее арифметическое* данных *(см. разд. "Среднее" далее в этой главе)*. На самом деле, несмотря на то, что среднее вычисляется довольно просто и его выгодно использовать, оно не всегда бывает лучшей мерой центрального значения. По этой причине в статистике были разработаны и популяризированы несколько альтернативных оценок среднего значения.

Метрические и оценочные показатели

В статистике термин *"оценки"* часто используется для значений, вычисляемых из данных, которые находятся под рукой, чтобы отличить то, что мы видим, исходя из этих данных, от теоретически истинного или точного положения дел. Аналитики данных и бизнес-аналитики с большей вероятностью будут называть такие значения *метрическими показателями*, или *метриками*. Эта разница отражает подходы, принятые в статистике, в отличие от науки о данных: учитывается неопределенность, которая лежит в основе статистики, тогда как центром внимания науки о данных являются конкретные деловые или организационные цели. Следовательно, статистики оценивают, а аналитики измеряют.

Среднее

Самой элементарной оценкой центрального положения является среднее значение, или *среднее арифметическое*. Среднее — это сумма всех значений, деленная на число значений. Рассмотрим следующий ряд чисел: {3, 5, 1, 2}. Среднее составит $(3+5+1+2)/4 = 11/4 = 2,75$. Вы часто будете встречать символ $\bar{x}$ (произносится "x с чертой"), который обозначает среднее значение выборки из популяции, или генеральной совокупности. Формула среднего значения для ряда из n значений $x_1, x_2, ..., x_n$ следующая:

$$\text{Среднее} = \bar{x} = \frac{\sum_{i}^{n} x_i}{n}.$$

N (или *n*) обозначает общее число записей или наблюдений. В статистике это обозначение используется с заглавной буквы, если оно относится к популяции, и строчной, если оно относится к выборке из популяции. В науке о данных это различие не является принципиальным, и поэтому можно увидеть и то и другое.

Разновидностью среднего является *среднее усеченное*, которое вычисляется путем отбрасывания фиксированного числа сортированных значений с каждого конца последовательности и затем взятия среднего арифметического оставшихся значений. Если представить сортированные значения как $x_{(1)}, x_{(2)}, ..., x_{(n)}$, где $x_{(1)}$ — самое маленькое значение, а $x_{(n)}$ — самое большое, то формула для вычисления усеченного среднего с пропуском p самых малых и самых больших значений будет следующей:

$$\text{Среднее усеченное} = \bar{x} = \frac{\sum_{i=p+1}^{n-p} x_{(i)}}{n-2p}.$$

Среднее усеченное устраняет влияние предельных значений. Например, в международных состязаниях по прыжкам в воду верхние и нижние баллы пяти судей отбрасываются, и итоговым баллом является среднеарифметический балл трех оставшихся судей [Wikipedia-2016]. Такой подход не дает одному судье манипулировать баллом, возможно, чтобы оказать содействие спортсмену из своей страны. Усеченные средние получили широкое распространение и во многих случаях предпочтительны вместо обычного среднего *(см. разд. "Медиана и робастные оценки" далее в этой главе)*.

Еще один вид среднего значения — это *среднее взвешенное*, которое вычисляется путем умножения каждого значения данных x_i на свой вес w_i и деления их суммы на сумму весов. Формула среднего взвешенного выглядит так:

$$\text{Среднее взвешенное} = \bar{x}_w = \frac{\sum_{i=1}^{n} w_i x_i}{\sum_{i}^{n} w_i}.$$

Существует два главных побудительных мотива для использования среднего взвешенного.

- Некоторые значения внутренне более переменчивы, чем другие, и сильно переменчивым наблюдениям придается более низкий вес. Например, если мы берем среднее данных, поступающих от многочисленных датчиков, и один из датчиков менее точен, тогда вес данных от этого датчика можно понизить.
- Собранные данные не одинаково представляют разные группы, которые мы заинтересованы измерить. Например, в зависимости от того, каким образом проводится онлайновый эксперимент, у нас может не быть набора данных, который точно отражает все группы в базе пользователей. Для того чтобы это исправить, можно придать более высокий вес значениям из тех групп, которые были представлены недостаточно.

Медиана и робастные оценки

Медиана — это число, расположенное в сортированном списке данных ровно посередине. Если число данных четное, срединным значением является то, которое не находится в наборе данных фактически, а является средним арифметическим двух значений, которые делят сортированные данные на верхнюю и нижнюю половины. По сравнению со средним, в котором используются абсолютно все наблюдения, медиана зависит только от значений в центре сортированных данных. Хотя это может выглядеть как недостаток, поскольку среднее значение намного более чувствительно к данным, существует много примеров, в которых медиана является

лучшим метрическим показателем центрального положения. Скажем, мы хотим взглянуть на типичные доходы домохозяйств в округах вокруг озера Вашингтон в Сиэтле. При сравнении округа Медина с округом Уиндермир использование среднего значения дало бы совершенно разные результаты, потому что в Медине живет Билл Гейтс. Если же мы станем использовать медиану, то уже не будет иметь значения, насколько богатым является Билл Гейтс — позиция срединного наблюдения останется той же.

По тем же самым причинам, по которым используется среднее взвешенное, можно вычислить и *медиану взвешенную.* Как и с медианой, мы сначала выполняем сортировку данных, несмотря на то, что с каждым значением данных связан вес. В отличие от срединного числа медиана взвешенная — это такое значение, в котором сумма весов равна для нижней и верхней половин сортированного списка. Как и медиана, взвешенная медиана устойчива к выбросам.

Выбросы

Медиана называется *робастной* оценкой центрального положения, поскольку она не находится под влиянием *выбросов* (предельных случаев), которые могут исказить результаты. Выброс — это любое значение, которое сильно удалено от других значений в наборе данных. Точное определение выброса несколько субъективно, несмотря на то, что в различных сводных данных и графиках используются определенные правила *(см. разд. "Процентили и коробчатые диаграммы" далее в этой главе)*. Выброс как таковой не делает значение данных недопустимым или ошибочным (как в предыдущем примере с Биллом Гейтсом). Однако выбросы часто являются результатом ошибок данных, таких как смешивание данных с разными единицами измерения (километры против метров) или плохие показания датчика. Когда выбросы являются результатом неправильных данных, среднее значение приводит к плохой оценке центрального положения, в то время как медиана будет по-прежнему допустимой. В любом случае выбросы должны быть идентифицированы и обычно заслуживают дальнейшего обследования.

Обнаружение аномалий

В отличие от типичного анализа данных, где выбросы иногда информативны, а иногда — досадная помеха, в *обнаружении аномалий* целевыми объектами являются именно выбросы, и значительный массив данных преимущественно служит для определения "нормы", с которой соразмеряются аномалии.

Медиана не единственная робастная оценка центрального положения. На самом деле, для того чтобы предотвратить влияние выбросов, широко используется и среднее усеченное. Например, усечение нижних и верхних 10% данных (общепринятый выбор) обеспечит защиту от выбросов во всех, кроме самых малых, наборах данных. Среднее усеченное может считаться компромиссом между медианой и средним: оно устойчиво к предельным значениям в данных, но использует больше данных для вычисления оценки центрального положения.

Другие робастные метрические показатели центрального положения

В статистике было разработало множество других инструментов оценки, так называемых оценщиков, или эстиматоров, центрального положения преимущественно с целью разработки более робастных инструментов оценки, чем среднее, и более *эффективных* (т. е. способных лучше обнаруживать небольшие разницы в центральном положении между наборами данных). Эти методы потенциально полезны для небольших наборов данных. Вместе с тем они едва дают дополнительные выгоды в условиях крупных или даже умеренно размерных наборов данных.

Пример: оценки центрального положения численности населения и уровня убийств

В табл. 1.2 показаны первые несколько строк из набора данных, содержащего данные о численности населения и уровне убийств (в единицах убийств на 100 тыс. человек в год) по каждому штату.

Таблица 1.2. *Несколько строк данных* `data.frame` *о численности населения и уровне убийств по штатам*

№	Штат	Население	Уровень убийств
1	Alabama	4 779 736	5,7
2	Alaska	710 231	5,6
3	Arizona	6 392 017	4,7
4	Arkansas	2 915 918	5,6
5	California	37 253 956	4,4
6	Colorado	5 029 196	2,8
7	Connecticut	3 574 097	2,4
8	Delaware	897 934	5,8

Вычислим среднее, среднее усеченное и медиану численности населения, используя R:

```
> state <- read.csv(file="/Users/andrewbruce1/book/state.csv")
> mean(state[["Population"]])
[1] 6162876
> mean(state[["Population"]], trim=0.1)
[1] 4783697
> median(state[["Population"]])
[1] 4436370
```

Среднее больше среднего усеченного, которое больше медианы.

Это вызвано тем, что среднее усеченное исключает самые большие и самые малые пять штатов (`trim=0.1` отбрасывает по 10% с каждого конца). Если мы захотим вычислить среднестатистическое количество убийств в стране, то должны использовать среднее взвешенное или медиану, чтобы учесть разную численность населения в штатах. Поскольку базовый R не имеет функции для взвешенной медианы, то мы должны установить программный пакет, в частности `matrixStats`:

```
> weighted.mean(state[["Murder.Rate"]], w=state[["Population"]])
[1] 4.445834
> library("matrixStats")
> weightedMedian(state[["Murder.Rate"]], w=state[["Population"]])
[1] 4.4
```

В этом случае среднее взвешенное и медиана почти одинаковы.

Ключевые идеи для оценок центрального положения

- Основным метрическим показателем центрального положения является среднее, но оно может быть чувствительным к предельным значениям (выбросам).
- Другие метрические показатели (медиана, среднее усеченное) более робастны.

Дополнительные материалы для чтения

- Майкл Левин (Michael Levine, Университет Пердью) разместил несколько полезных слайдов, посвященных основным расчетам мер центрального положения (**http://www.stat.purdue.edu/~mlevins/STAT511_2012/Lecture2standard.pdf**).
- "Анализ результатов наблюдений" [Tukey-1977] — классическая книга Джона Тьюки, которая по-прежнему пользуется спросом.

Оценки вариабельности

Центральное положение — это всего одна из размерностей в обобщении признака. Вторая размерность, *вариабельность*, именуемая также *дисперсностью*, показывает, сгруппированы ли значения данных плотно, или же они разбросаны. В основе статистики лежит вариабельность: ее измерение, уменьшение, различение произвольной вариабельности от реальной, идентификация разных источников реальной вариабельности и принятие решений в условиях ее присутствия.

Ключевые термины

Отклонения (deviations)

Разница между наблюдаемыми значениями и оценкой центрального положения.

Синонимы: ошибки, остатки.

Дисперсия (variance)

Сумма квадратических отклонений от среднего, деленная на $n - 1$, где n — число значений данных.

Синонимы: среднеквадратическое отклонение, среднеквадратическая ошибка.

Стандартное отклонение (standard deviation)

Квадратный корень из дисперсии.

Синонимы: норма l_2, евклидова норма.

Среднее абсолютное отклонение (mean absolute deviation)

Среднее абсолютных значений отклонений от среднего[2].

Синонимы: норма l_1, манхэттенская норма.

Медианное абсолютное отклонение от медианы (median absolute deviation from the median)

Медиана абсолютных значений отклонений от медианы.

Размах (range)

Разница между самым большим и самым малым значениями в наборе данных.

Порядковые статистики (order statistics)

Метрические показатели на основе значений данных, отсортированных от самых малых до самых больших.

Синоним: ранг.

Процентиль (percentile)

Такое значение, что P процентов значений принимает данное значение или меньшее и $(100 - P)$ процентов значений принимает данное значение или большее.

Синоним: квантиль.

Межквартильный размах (interquartile range)

Разница между 75-м и 25-м процентилями.

Синонимы: МКР, IQR.

Так же как и в случае центрального положения, которое можно измерить разными способами (среднее, медиана и т. д.), существуют различные способы измерить вариабельность.

[2] Абсолютным оно является потому, что суммируются отклонения по модулю, т. к. в противном случае сумма всех разбросов будет равна нулю. — *Прим. пер.*

Стандартное отклонение и связанные с ним оценки

Наиболее широко используемые оценки вариабельности основаны на разницах, или *отклонениях*, между оценкой центрального положения и наблюдаемыми данными. Для набора данных {1, 4, 4}, среднее равняется 3, и медиана — 4. Отклонения от среднего представляют собой разницы: $1-3=-2$, $4-3=1$, $4-3=1$. Эти отклонения говорят о том, насколько данные разбросаны вокруг центрального значения.

Один из способов измерить вариабельность состоит в том, чтобы оценить типичное значение этих отклонений. Усреднение самих отклонений мало, поэтому отрицательные отклонения нейтрализуют положительные. Фактически сумма отклонений от среднего как раз равна нулю. Вместо этого простой подход заключается в том, чтобы взять среднее абсолютных значений отклонений от среднего значения. В предыдущем примере абсолютное значение отклонений равно {2, 1, 1}, а их среднее — $(2+1+1)/3=1{,}33$. Это и есть среднее абсолютное отклонение, которое вычисляется по следующей формуле:

$$\text{Среднее абсолютное отклонение} = \frac{\sum_{i=1}^{n} |x_i - \bar{x}|}{n},$$

где $\bar{x}$ — среднее значение в выборке, или выборочное среднее.

Самыми известными оценками вариабельности являются *дисперсия* и *стандартное отклонение*, которые основаны на квадратических отклонениях. Дисперсия — это среднее квадратических отклонений, а стандартное отклонение — квадратный корень из дисперсии.

$$\text{Дисперсия} = s^2 = \frac{\sum (x-\bar{x})^2}{n-1};$$

$$\text{Стандартное отклонение} = s = \sqrt{\text{Дисперсия}}.$$

Стандартное отклонение интерпретируется намного проще, чем дисперсия, поскольку оно находится на той же шкале измерения, что и исходные данные. Однако, учитывая его более сложную и интуитивно менее понятную формулу, может показаться странным, что в статистике стандартному отклонению отдается предпочтение по сравнению со средним абсолютным отклонением. Такое преобладание обязано статистической теории: математически работа с квадратическими значениями намного более удобна, чем с абсолютными, в особенности со статистическими моделями.

Степени свободы и *n* или *n* – 1?

В книгах по статистике всегда так или иначе обсуждается вопрос, почему в формуле дисперсии у нас в знаменателе $n-1$, вместо n, который приводит к понятию *степеней свободы*. Это различие не является важным, поскольку n обычно настолько велико, что уже не имеет большого значения, будет ли деление выполняться на n или $n-1$. Однако в случае если вам интересно, то вот объяснение. Оно основывается на предпосылке, что вы хотите получить оценки популяции исходя из вынутой из нее выборки.

Если в формуле дисперсии применить интуитивно понятный знаменатель n, то истинное значение дисперсии и стандартного отклонения в популяции будет недооценено. Это называется *смещенной* оценкой. Однако если поделить на $n-1$ вместо n, то стандартное отклонение становится *несмещенной* оценкой.

Полное объяснение, почему использование n приводит к смещенной оценке, сопряжено с понятием степеней свободы, которое принимает во внимание число ограничений при вычислении оценки. В данном случае существуют $n-1$ степеней свободы, поскольку существует одно ограничение: стандартное отклонение зависит от вычисления среднего в выборке. В большинстве задач аналитикам данных не нужно беспокоиться по поводу степеней свободы, но в отдельных случаях это понятие имеет особое значение *(см. разд. "Выбор K" главы 6)*.

Ни дисперсия и стандартное отклонение, ни среднее абсолютное отклонение не устойчивы к выбросам и предельным значениям (*см. разд. "Медиана и робастные оценки" ранее в этой главе*, где обсуждаются робастные оценки центрального положения). Дисперсия и стандартное отклонение чувствительны к выбросам больше всего, поскольку они основаны на квадратических отклонениях.

Робастной оценкой вариабельности является *медианное абсолютное отклонение от медианы* (MAO — median absolute deviation, MAD):

$$\text{Медианное абсолютное отклонение} =$$
$$= \text{Медиана}\left(\left|x_1 - m\right|,\ \left|x_2 - m\right|,\ ...,\ \left|x_N - m\right|\right),$$

где m — это медиана. Как и в случае с медианой, MAO не находится под влиянием предельных значений. Можно также вычислить усеченное стандартное отклонение по аналогии со средним усеченным *(см. разд. "Среднее" ранее в этой главе)*.

Дисперсия, стандартное отклонение, среднее абсолютное отклонение и медианное абсолютное отклонение от медианы не являются эквивалентными оценками, даже в случае, когда данные поступают из нормального распределения. На деле стандартное отклонение всегда больше среднего абсолютного отклонения, которое в свою очередь больше медианного абсолютного отклонения. Иногда медианное абсолютное отклонение умножается на постоянный поправочный коэффициент (который часто сводится к 1,4826), чтобы в случае нормального распределения привести MAO к той же шкале измерения, что и стандартное отклонение.

Оценки на основе процентилей

Другой подход к оценке дисперсности основывается на рассмотрении разброса сортированных данных, или их спреда. Статистические показатели на основе сортированных (ранжированных) данных называются *порядковыми статистиками*. Элементарная мера — это *размах*, т. е. разница между самым большим и самым малым числом. Минимальные и максимальные значения как таковые полезно знать, поскольку они помогают идентифицировать выбросы, но размах чрезвычайно чувствителен к выбросам и не очень полезен в качестве общей меры дисперсности в данных.

Для того чтобы предотвратить чувствительность к выбросам, можно обратиться к размаху данных после отбрасывания значений с каждого конца. Эти типы оценок формально основываются на разницах между *процентилями.* В наборе данных P-й процентиль является таким значением, что, по крайней мере, P процентов значений принимает это значение или меньшее и, по крайней мере, $(100 - P)$ процентов значений принимает это значение или большее. Например, для нахождения 80-го процентиля надо отсортировать данные. Затем, начиная с самого малого значения продолжить 80% вверх к самому большому значению. Отметим, что медиана — это то же самое, что и 50-й процентиль. Процентиль по существу аналогичен *квантилю*, при этом квантили индексируются долями (так, квантиль 0,8 — это то же самое, что и 80-й процентиль).

Общепринятой мерой вариабельности является разница между 25-м и 75-м процентилями, которая называется *межквартильным размахом* (interquartile range, IQR). Вот простой пример: 3, 1, 5, 3, 6, 7, 2, 9. Эти числа мы сортируем, получив 1, 2, 3, 3, 5, 6, 7, 9. 25-й процентиль находится в 2,5, и 75-й процентиль — в 6,5, поэтому межквартильный размах будет $6,5 - 2,5 = 4$. Программная система может иметь немного другие подходы, которые дают отличающиеся ответы (см. приведенное далее примечание); как правило, эти отличия небольшие.

Для очень больших наборов данных расчет точных процентилей может быть вычислительно очень затратным, поскольку он требует сортировки всех значений данных. В программных системах для машинного обучения и статистического анализа используются специальные алгоритмы, такие как [Zhang-Wang-2007], которые получают приблизительный процентиль, вычисляя его очень быстро и гарантированно обеспечивая определенную точность.

Процентиль: точное определение

Если имеется четное число данных (n — четное), то исходя из предыдущего определения процентиль неоднозначен. На деле можно взять любое значение между порядковыми статистиками $x_{(j)}$ и $x_{(j+1)}$, где j удовлетворяет:

$$100 \cdot \frac{j}{n} \le P \le 100 \cdot \frac{j+1}{n}.$$

В формальном плане процентиль — это средневзвешенное значение:

$$\text{Процентиль } (P) = (1-w)x_{(j)} + wx_{(j+1)}$$

для некоторого веса w между 0 и 1. В статистических программных системах содержатся слегка отличающиеся подходы к выбору значения w. На самом деле, R-функция `quantile` предлагает девять разных способов вычисления квантиля. За исключением небольших наборов данных, вам, как правило, не придется беспокоиться по поводу точного метода, которым вычисляется процентиль.

Пример: оценки вариабельности населения штатов

В табл. 1.3 (для удобства взятой повторно из ранее приведенной табл. 1.2) показаны первые несколько строк из набора данных, содержащего численность населения и уровни убийств для каждого штата.

Таблица 1.3. *Несколько строк из кадра* `data.frame` *с данными о численности населения и уровне убийств по каждому штату*

№	Штат	Население	Уровень убийств
1	Alabama	4 779 736	5,7
2	Alaska	710 231	5,6
3	Arizona	6 392 017	4,7
4	Arkansas	2 915 918	5,6
5	California	37 253 956	4,4
6	Colorado	5 029 196	2,8
7	Connecticut	3 574 097	2,4
8	Delaware	897 934	5,8

Используя встроенные функции R для стандартного отклонения (`sd`), межквартильного размаха (`IQR`) и медианного абсолютного отклонения из медианы (`mad`), можно вычислить оценки вариабельности данных о населении штатов:

```
> sd(state[["Population"]])
[1] 6848235
> IQR(state[["Population"]])
[1] 4847308
> mad(state[["Population"]])
[1] 3849870
```

Стандартное отклонение почти вдвое больше MAO (в R по умолчанию показатель MAO корректируется, чтобы быть на той же шкале измерения, что и среднее). И это не удивительно, поскольку стандартное отклонение чувствительно к выбросам.

Ключевые идеи для оценок вариабельности

- Дисперсия и стандартное отклонение — наиболее широко распространенные и в рутинном порядке регистрируемые статистики вариабельности.
- Оба показателя чувствительны к выбросам.
- Более робастные метрические показатели включают среднее абсолютное отклонение, медианное абсолютное отклонение от медианы и процентили (квантили).

Дополнительные материалы для чтения

- Онлайновый статистический ресурс Дэвида Лэйна (David Lane) содержит раздел по процентилям (**http://onlinestatbook.com/2/introduction/percentiles.html**).
- Кевин Дэйвенпорт (Kevin Davenport) на R-bloggers предлагает полезный пост, посвященный отклонениям от медианы и их робастным свойствам (**http://www.r-bloggers.com/absolute-deviation-around-the-median/**).

Обследование распределения данных

Все рассмотренные нами оценки обобщают данные в одном числе с целью описания центрального положения либо вариабельности данных. Помимо этого, также полезно обследовать характер распределенных данных в целом.

Ключевые термины

Коробчатая диаграмма (boxplot)

График, введенный в употребление Тьюки, в качестве быстрого способа визуализации распределения данных.

Синоним: диаграмма типа "ящик с усами".

Частотная таблица (frequency table)

Сводка количеств числовых значений, которые разбиты на серию частотных интервалов (корзин, бинов).

Гистограмма (histogram)

График частотной таблицы, где частотные интервалы откладываются на оси *x*, а количества (или доли) — на оси *y*.

График плотности (density plot)

Сглаженная версия гистограммы, часто на основе *ядерной оценки плотности*.

Процентили и коробчатые диаграммы

В *разд. "Оценки на основе процентилей" ранее в этой главе* мы рассмотрели, каким образом процентили могут использоваться для измерения разброса данных. Процентили также важны для обобщения всего распределения в целом. Общепринято сообщать о квартилях (25-й, 50-й и 75-й перцентили) и децилях (10-й, 20-й, ..., 90-й процентили). Процентили особенно важны для обобщения *хвостов* (внешнего размаха) распределения. Массовая культура ввела в обиход термин *"однопроцентовики"*, который относится к людям в верхнем 99-м процентиле богатства.

В табл. 1.4 показаны некоторые процентили уровня убийств по штатам. В R их можно получить при помощи функции `quantile`:

```
quantile(state[["Murder.Rate"]], p=c(.05, .25, .5, .75, .95))
   5%   25%   50%   75%   95%
1.600 2.425 4.000 5.550 6.510
```

Таблица 1.4. *Процентили уровня убийств по штатам*

5%	25%	50%	75%	95%
1,60	2,42	4,00	5,55	6,51

Медиана равна 4 убийствам на 100 тыс. человек, несмотря на то, что присутствует довольно большая вариабельность: 5-й процентиль составляет всего 1,6, тогда как 95-й процентиль — 6,51.

Коробчатые диаграммы, введенные в употребление Тьюки [Tukey-1977], основаны на процентилях и обеспечивают быстрый способ визуализации распределения данных. На рис. 1.2 представлена коробчатая диаграмма населения по штатам, полученная в R:

```
boxplot(state[["Population"]]/1000000, ylab="Население, млн человек")
```

Верх и низ коробки представляют собой соответственно 75-й и 25-й процентили. Медиана показана в коробке горизонтальной линией. Пунктирные линии, называемые *усами*, выходят из верха и низа и говорят о размахе основной части данных. Существует много вариантов коробчатой диаграммы; например, обратитесь к документации по R-функции `boxplot` [R-base-2015]. По умолчанию данная функция R простирает усы к самой далекой точке вне коробки, за исключением того, что она не выходит за пределы межквартильного размаха (МКР или IQR), умноженного на 1,5 (в других программных системах могут использоваться иные правила). Все данные за пределами усов отображаются как одиночные точки.

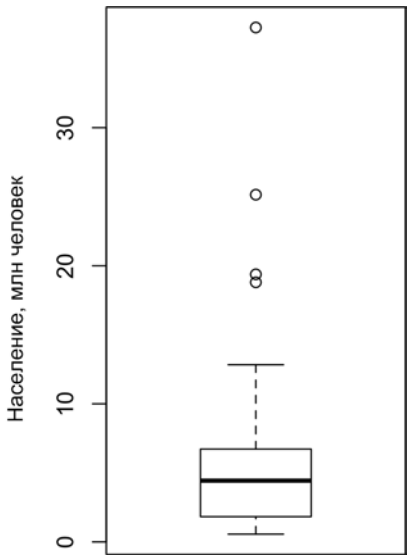


Рис. 1.2. Коробчатая диаграмма населения штатов

Частотная таблица и гистограммы

Частотная таблица переменной делит диапазон переменной на равноотстоящие сегменты и сообщает о том, сколько значений попадает в каждый сегмент. В табл. 1.5 показана частотная таблица численности населения по штатам, вычисленная в R:

```
breaks <- seq(from=min(state[["Population"]]),
              to=max(state[["Population"]]), length=11)
pop_freq <- cut(state[["Population"]], breaks=breaks,
              right=TRUE, include.lowest = TRUE)
table(pop_freq)
```

Наименее густонаселенным штатом является Вайоминг с населением 563 626 человек (согласно переписи 2010 г.), а самым густонаселенным — Калифорния с населением 37 253 956 человек. Это дает нам размах $37\,253\,956 - 563\,626 = 36\,690\,330$, который мы должны разделить на равные частотные интервалы — скажем, 10 интервалов. При 10 равноразмерных интервалах каждый частотный интервал будет иметь ширину 3 669 033, таким образом, первый интервал будет простираться от 563 626 до 4 232 658. В отличие от него верхний интервал, 33 584 923–37 253 956, имеет всего один штат: Калифорнию. Два интервала, которые идут непосредственно ниже штата Калифорния, пусты, пока мы не достигнем штата Техас. Важно учитывать пустые интервалы; тот факт, что в этих интервалах значения отсутствуют, является полезной информацией. Может также быть полезным поэкспериментировать с разными размерами интервалов. Если они слишком большие, то важные признаки распределения могут быть затушеваны. Если же они слишком малы, то результат будет слишком детальным, и возможность наблюдать общую картину будет потеряна.

Таблица 1.5. *Частотная таблица численности населения по штатам*

№ интервала	Диапазон интервала	Количество	Штаты
1	563 626– 4 232 658	24	WY,VT,ND,AK,SD,DE,MT,RI,NH,ME,HI,ID,NE,WV, NM,NV,UT,KS,AR,MS,IA,CT,OK,OR
2	4 232 659– 7 901 691	14	KY,LA,SC,AL,CO,MN,WI,MD,MO,TN,AZ,IN,MA,WA
3	7 901 692– 11 570 724	6	VA,NJ,NC,GA,MI,OH
4	11 570 725– 15 239 757	2	PA,IL
5	15 239 758– 18 908 790	1	FL
6	18 908 791– 22 577 823	1	NY
7	22 577 824– 26 246 856	1	TX
8	26 246 857– 29 915 889	0	
9	29 915 890– 33 584 922	0	
10	33 584 923– 37 253 956	1	CA

И таблицы частот, и процентили обобщают данные за счет создания частотных интервалов. В общем и целом квартили и децили будут иметь одинаковое количество в каждом интервале (равноколичественные интервалы), но размеры интервалов будут различаться. Частотная таблица, в отличие от них, будет иметь разные количества в интервалах (равноразмерные интервалы).

Гистограмма — это способ визуализации частотной таблицы, где частотные интервалы откладываются на оси *x*, а количество данных — на оси *y*. Чтобы создать гистограмму, соответствующую табл. 1.5 на языке R, используется функция `hist` с аргументом `breaks`:

```
hist(state[["Population"]], breaks=breaks)
```

В результате получится гистограмма, которая показана на рис. 1.3. В целом гистограммы отображаются таким образом, что:

- пустые интервалы включены в график;
- интервалы имеют равную ширину;
- число интервалов (или, что то же самое, размер интервала) задается пользователем;
- столбцы гистограммы непрерывны — пробелы между ними отсутствуют, если нет пустого интервала.

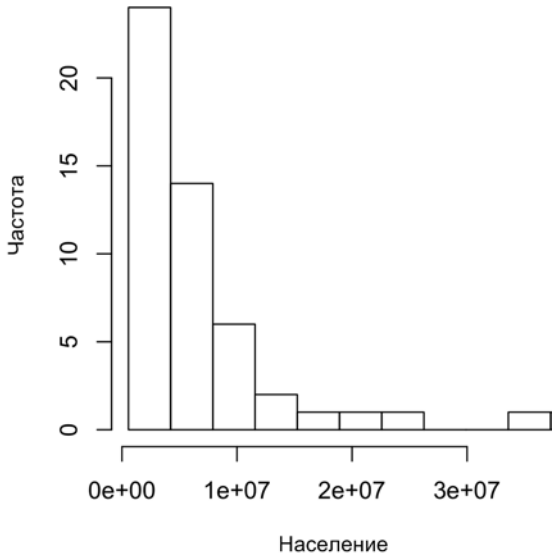


Рис. 1.3. Гистограмма численности населения штатов

Статистические моменты

В статистической теории центральное положение и вариабельность (колеблемость, изменчивость) упоминаются как моменты распределения первого и второго порядка. Моменты третьего и четвертого порядка — это *асимметрия* и *эксцесс*. Под асимметрией понимается смещение данных к бóльшим или меньшим значениям, а под эксцессом — склонность данных к предельным значениям. Для измерения асимметрии и эксцесса, как правило, метрические показатели не используются; вместо этого они идентифицируются при визуальном отображении, таком как на рис. 1.2 и 1.3.

Оценки плотности

С гистограммой связан график плотности, который показывает распределение значений данных в виде сплошной линии. График плотности можно рассматривать как сглаженную гистограмму, несмотря на то, что он обычно вычисляется непосредственно из данных с помощью *ядерной оценки плотности* (см. [Duong-2001] по поводу краткого руководства). На рис. 1.4 представлена оценка плотности, наложенная на гистограмму. В R оценка плотности вычисляется при помощи функции `density`:

```
hist(state[["Murder.Rate"]], freq=FALSE)
lines(density(state[["Murder.Rate"]]), lwd=3, col="blue")
```

Ключевое отличие от гистограммы, показанной на рис. 1.3, состоит в шкале оси *y*: график плотности соответствует отображению гистограммы как доли, а не количества (в R она задается при помощи аргумента `freq=FALSE`).

Оценка плотности

Оценивание плотности является обширной темой с долгой историей в статистической литературе. Фактически было опубликовано свыше 20 программных пакетов R, которые предлагают функции оценки плотности. [Deng-Wickham-2011] дает всесторонний анализ пакетов R, особо выделяя программные пакеты `ASH` и `KernSmooth`. Для многих задач науки о данных нет необходимости беспокоиться по поводу различных типов оценок плотности; достаточно использовать базовые функции.

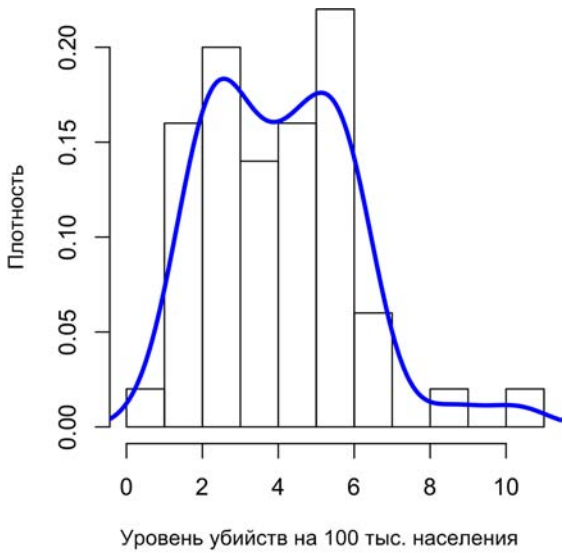


Рис. 1.4. Плотность уровня убийств в штатах

Ключевые идеи для исследования распределения данных

- Частотная гистограмма показывает частоты на оси *y* и значения переменных — на оси *x*; она дает визуальное представление о распределении данных.
- Частотная таблица является табличной версией расчета частот, которые можно найти на гистограмме.
- Коробчатая диаграмма — диаграмма, в которой верх и низ коробки находятся соответственно в 75-м и 25-м процентилях — также дает быстрое представление о распределении данных; она часто используется на парных графиках с целью сравнения распределений.
- График плотности — это сглаженная версия гистограммы; для оценки графика на основе данных требуется специальная функция (разумеется, возможны многочисленные оценки).

Дополнительные материалы для чтения

- Преподаватель Университета Освего, штат Нью-Йорк, предлагает пошаговое руководство по созданию коробчатой диаграммы (**http://www.oswego.edu/~srp/stats/bp_con.htm**).
- Оценка плотности в R рассматривается в работе Генри Денга (Henry Deng) и Хэдли Уикхэма (Hadley Wickham) под тем же названием (**http://vita.had.co.nz/papers/density-estimation.pdf**).
- На R-bloggers есть полезный пост по поводу гистограмм в R, включая такие элементы настройки, как разбиение на интервалы (breaks) (**http://www.r-bloggers.com/basics-of-histograms/**).
- На R-bloggers также имеются аналогичные посты в отношении коробчатых диаграмм в R (**http://www.r-bloggers.com/box-plot-with-r-tutorial/**).

Обследование двоичных и категориальных данных

Если говорить о категориальных данных, то представление о них дают простые доли или процентные соотношения.

Ключевые термины

Мода (mode)

Наиболее часто встречающаяся категория или значение в наборе данных.

Математическое ожидание (expected value)

Когда категории связаны с числовыми значениями, этот показатель дает среднее значение на основе вероятности появления категории.

Синоним: ожидаемое значение.

Столбчатые диаграммы (bar charts)

Частота или доля каждой категории, отображаемая в виде прямоугольника.

Круговые диаграммы (pie charts)

Частота или доля каждой категории, отображаемая в виде сектора круга.

Получение сводной информации о двоичной либо категориальной переменной с несколькими категориями представляет собой довольно простую задачу: мы просто выясняем долю единиц (1) либо важных категорий. Например, в табл. 1.6 показан процент задержанных рейсов из-за проблем в аэропорту Даллас–Форт-Уэрт, начиная с 2010 г. Задержки классифицированы в силу факторов, связанных с перевозчиком (Carrier), системными задержками при управлении воздушным движением (ATC), погодных условий (Weather), соображений безопасности (Security) или опоздания прибывающего воздушного судна (Inbound).

Таблица 1.6. *Процент задержек в аэропорту Даллас–Форт-Уэрт*

Carrier	ATC	Weather	Security	Inbound
23,02	30,40	4,03	0,12	42,43

Столбчатые диаграммы — это общепринятый визуальный инструмент для отображения одиночной категориальной переменной, который можно часто встретить в популярной прессе. Категории перечислены на оси *x*, а частоты или доли — на оси *y*. На рис. 1.5 показаны годичные задержки авиарейсов из-за проблем в аэропорту Даллас–Форт-Уэрт; график создан при помощи R-функции `barplot`:

```
barplot(as.matrix(dfw)/6, cex.axis=.5)
```

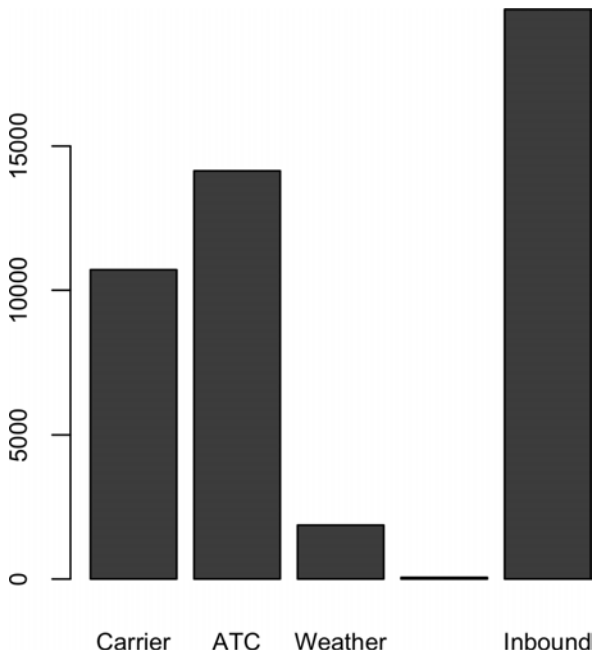


Рис. 1.5. Столбчатая диаграмма задержек авиарейсов из-за проблем в аэропорту Даллас–Форт-Уэрт

Отметим, что *столбчатая диаграмма* напоминает гистограмму; в столбчатой диаграмме ось *x* представляет разные категории факторной переменной, в то время как на гистограмме ось *x* представляет значения одиночной переменной на числовой шкале. На гистограмме прямоугольники обычно изображаются впритык друг к другу, а разрывы указывают на то, что значения в данных отсутствовали. На столбчатой диаграмме прямоугольники отображаются отдельно друг от друга.

Круговые диаграммы являются альтернативой столбчатым диаграммам, хотя специалисты в области статистики и эксперты по визуализации данных обычно сторонятся круговых диаграмм, как менее визуально информативных (см. [Few-2007]).

Числовые данные как категориальные данные

В *разд. "Частотная таблица и гистограммы" ранее в этой главе* мы рассмотрели частотные таблицы на основе разбивки данных на частотные интервалы, в результате чего числовые данные неявно преобразуются в порядковый фактор. В этом смысле гистограммы и столбчатые диаграммы подобны, за одним исключением — категории на оси *x* в столбчатой диаграмме не упорядочены. Преобразование числовых данных в категориальные является важным и широко используемым этапом в анализе данных, поскольку эта процедура уменьшает сложность (и размер) данных. Она помогает обнаруживать связи между признаками, в особенности на начальных стадиях анализа.

Мода

Мода — это значение (или значения в случае, если они одинаковы), которое появляется в данных чаще других. Например, модой задержек из-за проблем в аэропорту Даллас–Форт-Уэрт является "опоздание прибывающего воздушного судна" (Inbound). В другом примере в большинстве уголков США модой религиозных предпочтений будет христианство. Мода представляет собой простую сводную статистическую величину, или статистику, для категориальных данных и обычно для числовых данных не используется.

Математическое ожидание

Особым типом категориальных данных являются данные, в которых категории представлены или могут быть сопоставлены с дискретными значениями на одинаковой шкале измерения. Маркетолог новой "облачной" технологии, например, продает два уровня веб-служб: один по цене 300 долларов в месяц, а другой по цене 50 долларов в месяц. При этом он предлагает бесплатные вебинары для формирования списка потенциальных клиентов, и фирма полагает, что 5% посетителей подпишутся на веб-службы за 300 долларов, 15% на веб-службы за 50 долларов и 80% не подпишутся совсем. Эти данные можно обобщить для проведения финансовых расчетов в одном "математическом ожидании", являющемся формой среднего взвешенного, в котором весами выступают вероятности.

Математическое ожидание вычисляется следующим образом:

1. Умножить каждый исход на вероятность его наступления.
2. Просуммировать эти значения.

В примере облачной службы математическое ожидание посетителя вебинара, таким образом, составит 22,50 долларов в месяц, которое получено так:

$$EV = 0,05 \cdot 300 + 0,15 \cdot 50 + 0,80 \cdot 0 = 22,5.$$

Математическое ожидание, по сути дела, является формой среднего взвешенного: оно привносит понятие будущих ожиданий и весов вероятности, которые зачастую основаны на субъективном суждении. Математическое ожидание является фундаментальным понятием в оценке бизнеса и составлении бюджета долгосрочных расходов — например, математическое ожидание для пятилетних прибылей от нового

приобретения или ожидаемое сокращение затрат от новой программной системы учета пациентов в клинике.

Ключевые идеи для обследования двоичных и категориальных данных

- Категориальные данные, как правило, обобщаются в долях, и их можно визуализировать на столбчатой диаграмме.
- Категории могут представлять отличающиеся объекты (яблоки и апельсины, мужчин и женщин), уровни факторной переменной (низкий, средний и высокий) либо числовые данные, которые были разбиты на частотные интервалы.
- Математическое ожидание — это сумма значений, умноженных на вероятность их возникновения, которое часто используется для обобщения уровней факторных переменных.

Дополнительные материалы для чтения

Ни один курс статистики не будет полным без урока по дезориентирующим графикам, под которыми часто подразумеваются столбчатые и круговые диаграммы (**http://passyworldofmathematics.com/misleading-graphs/**).

Корреляция

Разведочный анализ данных во многих проектах моделирования (в науке о данных либо в статистическом исследовании) сопряжен с изучением корреляции среди предикторов и между предикторами и целевой переменной. Говорят, что переменные X и Y (каждая с измерительными данными) коррелируют положительно, если высокие значения X сопровождаются высокими значениями Y, а низкие значения X сопровождаются низкими значениями Y. Если высокие значения X сопровождаются низкими значениями Y, и наоборот, то переменные коррелируют отрицательно.

Ключевые термины

Коэффициент корреляции (correlation coefficient)

Метрический показатель, который измеряет степень, с какой числовые переменные связаны друг с другом (в диапазоне от -1 до $+1$).

Корреляционная матрица (correlation matrix)

Таблица, в которой строки и столбцы — это переменные, и значения ячеек — корреляции между этими переменными.

Диаграмма рассеяния (scatterplot)

График, в котором ось x является значением одной переменной, а ось y — значением другой.

Рассмотрим эти две переменные, идеально коррелированные в том смысле, что каждая идет параллельно от низкого значения до высокого:

v1: {1, 2, 3}

v2: {4, 5, 6}

Векторная сумма произведений равняется $4+10+18=32$. Теперь попробуем перетасовать одну из них и вычислить повторно — векторная сумма произведений больше не будет выше 32. Поэтому данная сумма произведений может использоваться в качестве метрического показателя; т. е. наблюдаемую сумму, равную 32, можно сравнивать с многочисленными произвольными перетасовками (по сути дела, эта идея касается оценки на основе повторного отбора: *см. разд. "Перестановочный тест" главы 3*). Производимые этим метрическим показателем значения, тем не менее, не особо содержательны, кроме как с опорой на распределение повторных выборок.

Более полезный стандартизированный вариант — это *коэффициент корреляции*, дающий оценку корреляции между двумя переменными, которые всегда находятся на одинаковой шкале измерения. Для того чтобы вычислить *коэффициент корреляции Пирсона*, мы умножаем отклонения от среднего для переменной 1 на то же самое для переменной 2, а затем делим результат на произведение стандартных отклонений:

$$r = \frac{\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{(n-1)s_x s_y}.$$

Отметим, что мы делим на $n-1$, а на n (*см. врезку "Степени свободы и n или n – 1?" ранее в этой главе* для получения дополнительной информации). Коэффициент корреляции всегда находится между $+1$ (идеальная положительная корреляция) и -1 (идеальная отрицательная корреляция); 0 свидетельствует об отсутствии корреляции.

Переменные могут иметь нелинейную связь, и в этом случае коэффициент корреляции может оказаться бесполезным метрическим показателем. Связь между налоговыми ставками и поступлениями, полученными за счет налогов, служит примером: когда налоговые ставки увеличиваются от 0, полученные поступления тоже увеличиваются. Однако, как только налоговые ставки достигают высокого уровня и приближаются к 100%, уклонение от уплаты налогов увеличивается, и налоговые поступления в итоге уменьшаются.

В табл. 1.7, которая называется *корреляционной матрицей*, показана корреляция между ежедневными доходностями акций телекоммуникационных компаний с июля 2012 по июнь 2015 г. Из таблицы видно, что Verizon (VZ) и ATT (T) имеют самую высокую корреляцию. Инфраструктурная компания Level Three (LVLT) имеет самую низкую корреляцию. Обратите внимание на диагональ из единиц (корреляция акции с самой собой равна 1) и избыточность информации выше и ниже диагонали.

Таблица 1.7. *Корреляция между ежедневными доходностями акций телекоммуникационных компаний*

	T	CTL	FTR	VZ	LVLT
T	1,000	0,475	0,328	0,678	0,279
CTL	0,475	1,000	0,420	0,417	0,287
FTR	0,328	0,420	1,000	0,287	0,260
VZ	0,678	0,417	0,287	1,000	0,242
LVLT	0,279	0,287	0,260	0,242	1,000

Таблица корреляций, аналогичная табл. 1.7, широко используется с целью отобразить связь между многочисленными переменными. На рис. 1.6 показана корреляция между ежедневными доходностями крупнейших биржевых инвестиционных фондов (exchange traded funds, ETF). Она легко создается в R при помощи пакета `corrplot`:

```
etfs <- sp500_px[row.names(sp500_px)>"2012-07-01",
                 sp500_sym[sp500_sym$sector=="etf", 'symbol']]
library(corrplot)
corrplot(cor(etfs), method = "ellipse")
```

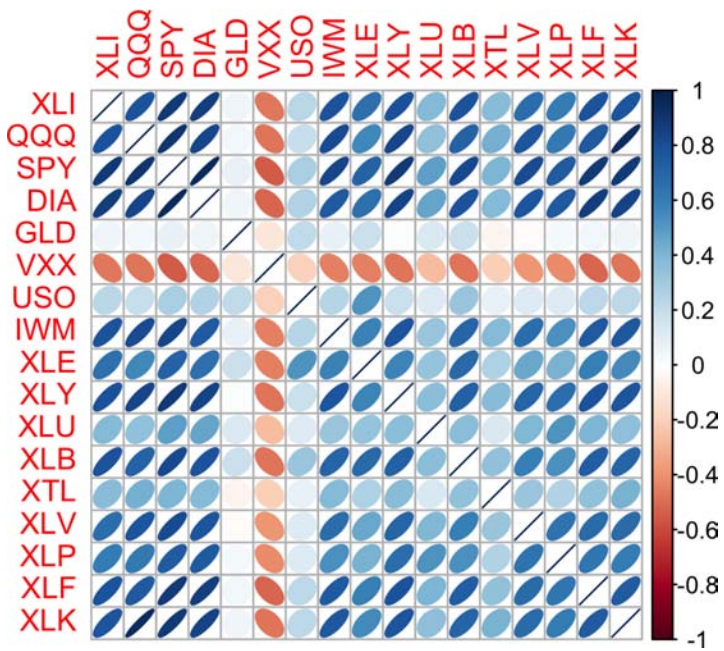


Рис. 1.6. Корреляция между доходностями фондов ETF

Биржевые инвестиционные фонды для индексов S&P 500 (SPY) и Доу Джонса (Dow Jones, DIA) имеют высокую корреляцию. Аналогичным образом, фонды QQQ и XLK, состоящие главным образом из технологических компаний, коррелированы положительно. Защитные биржевые инвестиционные фонды, такие, которые отслеживают цены на золото (GLD), цены на нефть (USO) или волатильность рынка (VXX), имеют тенденцию отрицательно коррелироваться с другими ETF. Ориентация эллипса говорит о том, коррелируют ли две переменные положительно (эллипс повернут вправо) или отрицательно (эллипс повернут влево). Заливка и ширина эллипса свидетельствуют о силе связи: более тонкие и темные эллипсы соответствуют более сильным связям.

Так же как среднее значение и стандартное отклонение, коэффициент корреляции чувствителен к выбросам в данных. В программных пакетах предлагаются робастные альтернативы классическому коэффициенту корреляции. Например, R-функция `cor` имеет аргумент `trim`, аналогичный тому, который используется для вычисления усеченного среднего (см. [R-base-2015]).

Другие оценки корреляции

В статистике давно были предложены другие типы коэффициентов корреляции, такие как коэффициент ранговой корреляции Спирмена ρ (ро) или коэффициент ранговой корреляции Кендалла τ (тау). Эти коэффициенты корреляции основаны на ранге данных, т. е. номерах наблюдений в наборе. Поскольку они работают с рангами, а не со значениями, эти оценки устойчивы к выбросам и могут справляться с определенными типами нелинейности. Однако аналитики данных в целях разведочного анализа могут обычно придерживаться коэффициента корреляции Пирсона и его робастных альтернатив. Ранговые оценки привлекательны главным образом в случае небольших наборов данных и определенных проверок статистических гипотез.

Диаграммы рассеяния

Стандартным методом визуализации связи между двумя переменными с измерительными данными является диаграмма рассеяния, чья ось *x* представляет одну переменную, ось *y* — другую, а каждая точка на графике — это запись. Посмотрим на рис. 1.7 с графиком, содержащим ежедневные доходности акций ATT и Verizon. График создан в R следующей командой:

```
plot(telecom$T, telecom$VZ, xlab="T", ylab="VZ")
```

Доходности имеют сильную положительную связь: в большинстве торговых дней обе акции идут вверх или вниз в тандеме. Существует буквально несколько дней, когда цена на одну акцию значительно падает, в то время как цена на другую акцию растет (и наоборот).

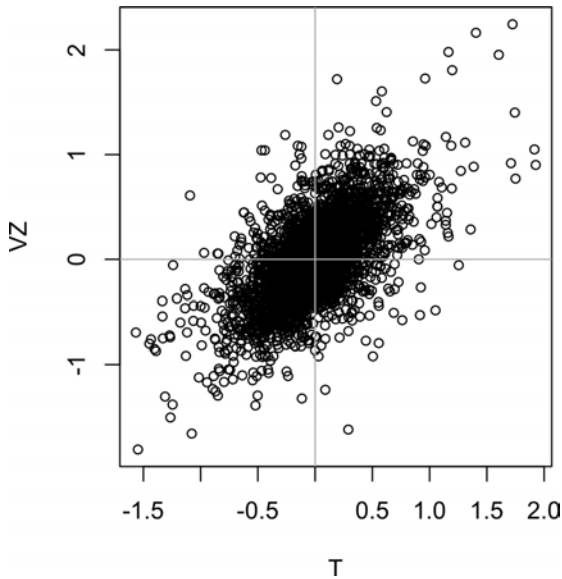


Рис. 1.7. Диаграмма рассеяния между доходностями акций ATT и Verizon

Ключевые идеи для корреляции

- Коэффициент корреляции измеряет степень, с которой две переменные между собой связаны.
- Когда высокие значения v1 сопровождаются высокими значениями v2, v1 и v2 связаны положительно.
- Когда высокие значения v1 связаны с низкими значениями v2, v1 и v2 связаны отрицательно.
- Коэффициент корреляции — это стандартизированный метрический показатель, который всегда колеблется от −1 (идеальная отрицательная корреляция) до +1 (идеальная положительная корреляция).
- Коэффициент корреляции 0 говорит об отсутствии корреляции, но следует учитывать, что произвольные перестановки данных будут порождать как положительные, так и отрицательные значения коэффициента корреляции по чистой случайности.

Дополнительные материалы для чтения

Книга "Статистика" Дэвида Фридмана, Роберта Пизани и Роджера Парвеса (Freedman D., Pisani R., Purves R. Statistics. — 4th edition. — W. W. Norton, 2007) предлагает превосходное обсуждение корреляции.

Исследование двух или более переменных

Уже знакомые нам инструменты оценки, такие как оценки среднего значения и дисперсии, рассматривают по одной переменной за раз *(одномерный анализ)*. Корреляционный анализ *(см. разд. "Корреляция" ранее в этой главе)* — это важный метод, который сравнивает две переменные *(двумерный анализ)*. В этом разделе мы обратимся к дополнительным оценкам, графикам и более чем к двум переменным *(многомерному анализу)*.

Ключевые термины

Таблицы сопряженности (contingency tables)

Сводка количеств двух или более категориальных переменных.

Графики с шестиугольной сеткой (hexagonal binning)

График двух числовых переменных, в котором записи сгруппированы в шестиугольниках.

Контурные графики (contour plots)

График, показывающий плотность двух числовых переменных в виде топографической карты.

Скрипичные графики (violin plots)

График, аналогичный коробчатой диаграмме, но показывающий оценку плотности.

Двумерный анализ подобно одномерному анализу связан с вычислением сводных статистик и их визуализацией. Соответствующий тип двумерного или многомерного анализа зависит от природы данных: числовых либо категориальных.

Шестиугольная сетка и контуры (отображение числовых данных против числовых)

Диаграммы рассеяния прекрасны, когда имеется относительно небольшое количество значений данных. График доходности акций на рис. 1.7 включает всего порядка 750 точек. Для наборов данных с сотнями тысяч и миллионов записей диаграмма рассеяния будет слишком плотной, и поэтому требуется другой способ визуализации связи. В качестве иллюстрации рассмотрим набор данных `kc_tax`, который содержит налогооблагаемую стоимость жилой недвижимости в округе Кинг, штат Вашингтон. Для того чтобы сосредоточиться на главной части данных, мы исключим очень дорогое жилье, а также жилье очень малых или очень больших размеров при помощи функции `subset`:

```
kc_tax0 <- subset(kc_tax, TaxAssessedValue < 750000 &
                  SqFtTotLiving>100 & SqFtTotLiving<3500)
nrow(kc_tax0)
[1] 432733
```

На рис. 1.8 показан график с *шестиугольной сеткой* для отображения связи между общей площадью в квадратных футах[3] и налогооблагаемой стоимостью жилья в округе Кинг. Вместо того чтобы отображать точки, которые появятся как монолитное темное облако, мы сгруппировали записи в шестиугольные сегменты и отобразили шестиугольники в цвете, указывающим на число записей в конкретном сегменте. На этой диаграмме положительная связь между квадратными футами и налогооблагаемой стоимостью жилой недвижимости очевидна. Интересной особенностью графика является тень второго облака над главным облаком, указывая на дома, которые имеют одинаковую площадь в квадратных футах, что и те, которые расположены в главном облаке, но с более высокой налогооблагаемой стоимостью.

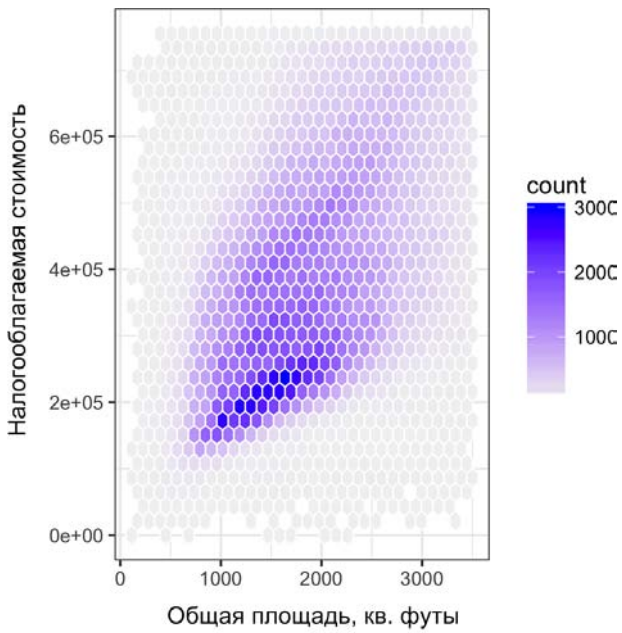


Рис. 1.8. График с шестиугольной сеткой для налогооблагаемой стоимости против общей площади в квадратных футах

Рис. 1.8 был сгенерирован мощным программным R-пакетом `ggplot2`, разработанным Хэдли Уикхэмом (Hadley Wickham) [ggplot2]. Пакет `ggplot2` — это одна из нескольких новых программных библиотек, предназначенных для продвинутого разведочного визуального анализа данных *(см. разд. "Визуализация многочисленных переменных" далее в этой главе)*.

[3] Общая площадь жилой недвижимости (finished square footage) — в американском налогообложении и среди торговцев недвижимостью за исключением межпотолочных и межстенных пространств и сводчатых перекрытий это практически вся площадь жилья, находящаяся выше и ниже уровня земли. — *Прим. пер.*

```
ggplot(kc_tax0, (aes(x=SqFtTotLiving, y=TaxAssessedValue))) +
  stat_binhex(colour="white") +
  theme_bw() +
  scale_fill_gradient(low="white", high="black") +
  labs(x="Общая площадь, кв. футы", y="Налогооблагаемая стоимость")
```

На рис. 1.9 используются контуры, наложенные на диаграмму рассеяния с целью визуализации связи между двумя числовыми переменными. Контуры подобны топографической карте, соответствующей двум переменным; каждая полоса контура представляет соответствующую плотность точек, увеличиваясь по мере того, как она приближается к "пику". Этот график говорит о том же, что и график на рис. 1.8: имеется вторичный пик "к северу" от главного пика. Этот график тоже был построен при помощи пакета `ggplot2` и встроенной функции `geom_density2d`.

```
ggplot(kc_tax0, aes(SqFtTotLiving, TaxAssessedValue)) +
  theme_bw() +
  geom_point( alpha=0.1) +
  geom_density2d(colour="white") +
  labs(x="Общая площадь, кв. футы", y="Налогооблагаемая стоимость")
```

Другие типы диаграмм используются для отображения связи между двумя числовыми переменными, в том числе *тепловые карты*. Тепловые карты, графики с шестиугольной сеткой и контурные графики — все они дают визуальное представление о двумерной плотности. В этом смысле они являются естественными аналогами гистограмм и графиков плотности.

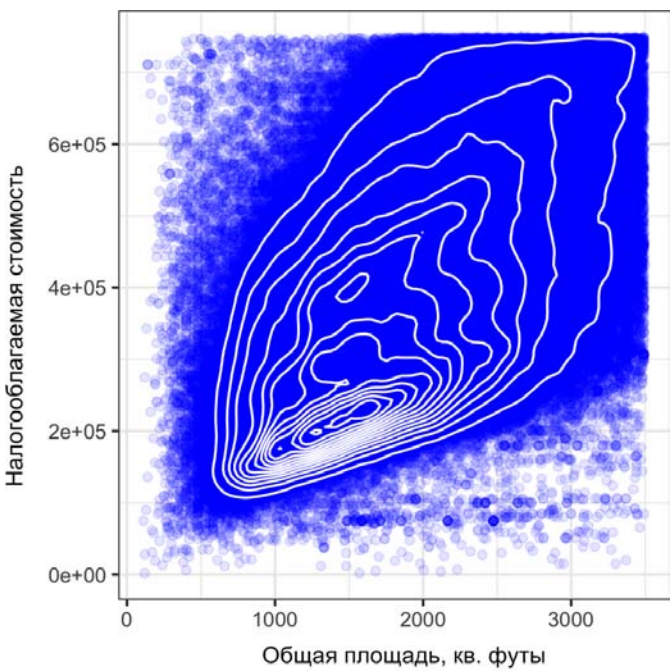


Рис. 1.9. Контурный график для налогооблагаемой стоимости против общей площади в квадратных футах

Две категориальных переменных

Полезным инструментом обобщения двух категориальных переменных является *таблица сопряженности* — таблица количеств по категориям. В табл. 1.8 приведена таблица сопряженности между уровнями персональной ссуды и ее исходом. Таблица построена на основе данных, предоставленных кредитным клубом Lending Club, лидером в инвестиционно-кредитном бизнесе равноправного кредитования. Уровень может быть от A (верхний) до G (низкий). Исходом может быть один из следующих: погашена, активная, просрочена или списана (остаток ссуды не будет взыскан). Эта таблица показывает количества и строчные проценты. Ссуды высокого качества имеют очень низкий процент просрочки/списания по сравнению со ссудами более низкого уровня. Таблицы сопряженности могут содержать лишь количества либо также включать столбцовые и итоговые проценты. Сводные таблицы в Excel являются, возможно, самым общепринятым инструментом, используемым для создания таблиц сопряженности. В R функция `CrossTable` в программном пакете `descr` создает таблицы сопряженности, и следующий далее фрагмент кода использовался для создания табл. 1.8:

```
library(descr)
x_tab <- CrossTable(lc_loans$grade, lc_loans$status,
                    prop.c=FALSE, prop.chisq=FALSE, prop.t=FALSE)
```

Таблица 1.8. *Таблица сопряженности уровня ссуд и их состояния*

Уровень	Полностью погашена	Активна	Просрочена	Списана	Всего
A	20715	52058	494	1588	74855
	0,277	0,695	0,007	0,021	0,161
B	31782	97601	2149	5384	136916
	0,232	0,713	0,016	0,039	0,294
C	23773	92444	2895	6163	125275
	0,190	0,738	0,023	0,049	0,269
D	14036	55287	2421	5131	76875
	0,183	0,719	0,031	0,067	0,165
E	6089	25344	1421	2898	35752
	0,170	0,709	0,040	0,081	0,077
F	2376	8675	621	1556	13228
	0,180	0,656	0,047	0,118	0,028
G	655	2042	206	419	3322
	0,197	0,615	0,062	0,126	0,007
Total	99426	333451	10207	23139	466223

Категориальные и числовые данные

Коробчатые диаграммы *(см. разд. "Процентили и коробчатые диаграммы" ранее в этой главе)* представляют собой простой способ визуального сравнения распределения числовой переменной, чьи значения сгруппированы согласно категориальной переменной. Например, нам нужно сравнить, каким образом процент задержек авиарейсов варьирует среди авиакомпаний. На рис. 1.10 показан процент задержанных авиарейсов за месяц, когда задержка была вызвана перевозчиком.

```
boxplot(pct_delay ~ airline, data=airline_stats, ylim=c(0, 50))
```

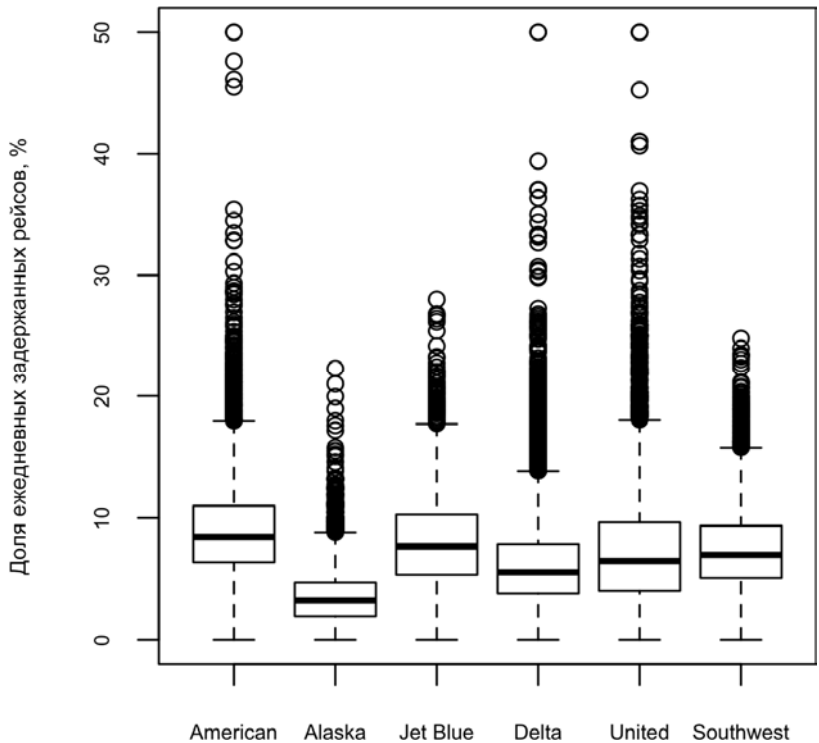


Рис. 1.10. Коробчатая диаграмма ежедневного процента задержек авиарейсов по перевозчикам

Авиакомпания Alaska Airlines выделяется тем, что имеет наименьшее количество задержек авиарейсов, в то время как у American Airlines подавляющее число задержек: нижний квартиль для American Airlines выше верхнего квартиля для Alaska Airlines.

Скрипичный график, введенный в употребление [Hintze-Nelson-1998], представляет собой дополнение к коробчатой диаграмме и графически изображает оценки плотности, где плотности расположены на оси *y*. Плотность зеркально отражена и перевернута, и получившаяся фигура заполнена, создавая изображение, напоминающее скрипку. Преимущество скрипичного графика состоит в том, что он может показывать нюансы в распределении, которые не заметны на коробчатой диаграмме. С другой стороны, коробчатая диаграмма более ясно показывает выбросы в дан-

ных. В программном пакете `ggplot2` функция `geom_violin` используется для создания скрипичного графика следующим образом:

```
ggplot(data=airline_stats, aes(airline, pct_carrier_delay)) +
  ylim(0, 50) + geom_violin() +
  labs(x="", y="Доля ежедневных задержанных рейсов, %")
```

Соответствующий график показан на рис. 1.11. Скрипичный график показывает концентрацию в распределении около нуля для Alaska Airlines и в меньшей степени для Delta Airlines. Этот феномен не так очевиден на коробчатой диаграмме. Скрипичный график можно объединить с коробчатой диаграммой путем добавления к графику `geom_boxplot` (правда, в этом случае лучше всего использовать цвета).

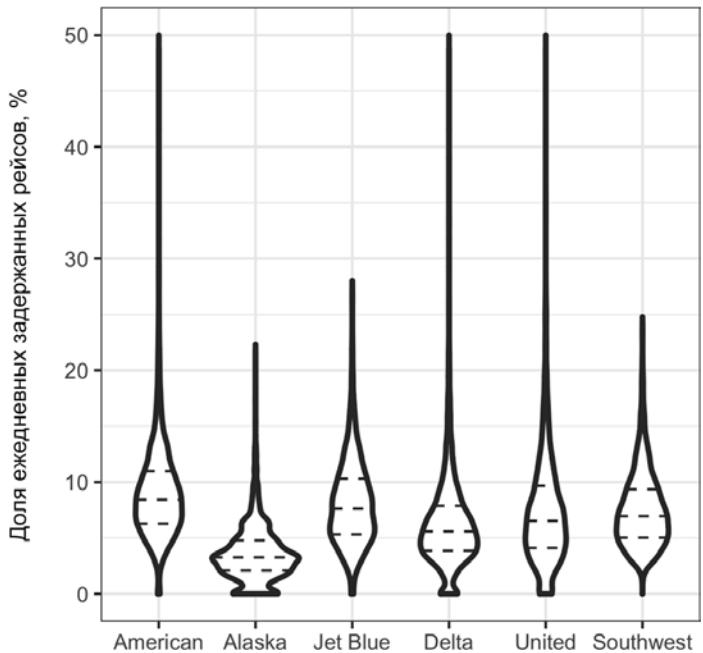


Рис. 1.11. Сочетание коробчатой диаграммы и скрипичного графика, показывающее процент задержек авиарейсов по перевозчикам

Визуализация многочисленных переменных

Все виды диаграмм, которые использовались ранее для сравнения двух переменных — диаграммы рассеяния, графики с шестиугольной сеткой и коробчатые диаграммы — легко расширяемы на большее количество переменных через понятие *кондиционности*. В качестве примера вернемся назад к рис. 1.8, в котором была показана связь между общей площадью домов и налогооблагаемой стоимостью. Мы отметили, что, по всей видимости, существует кластер домов, которые имеют более высокую налогооблагаемую стоимость в расчете на квадратный фут. Копнем поглубже и покажем рис. 1.12, который объясняет эффект местоположения домов, отобразив данные набора почтовых индексов. Теперь картина становится намного яснее: в некоторых почтовых индексах (98112, 98105) налогооблагаемая стоимость

намного выше, чем в других (98108, 98057). Именно эта несоизмеримость дает начало кластерам, наблюдаемым на рис. 1.8.

Мы создали рис. 1.12, используя пакет `ggplot2` и идею *аспектов* (facets), или кондиционных переменных[4] (в данном случае, почтовые индексы):

```
ggplot(subset(kc_tax0, ZipCode %in% c(98188, 98105, 98108, 98126)),
         aes(x=SqFtTotLiving, y=TaxAssessedValue)) +
  stat_binhex(colour="white") +
  theme_bw() +
  scale_fill_gradient( low="white", high="blue") +
  labs(x="Общая площадь, кв. футы", y="Налогооблагаемая стоимость") +
  facet_wrap("ZipCode")
```

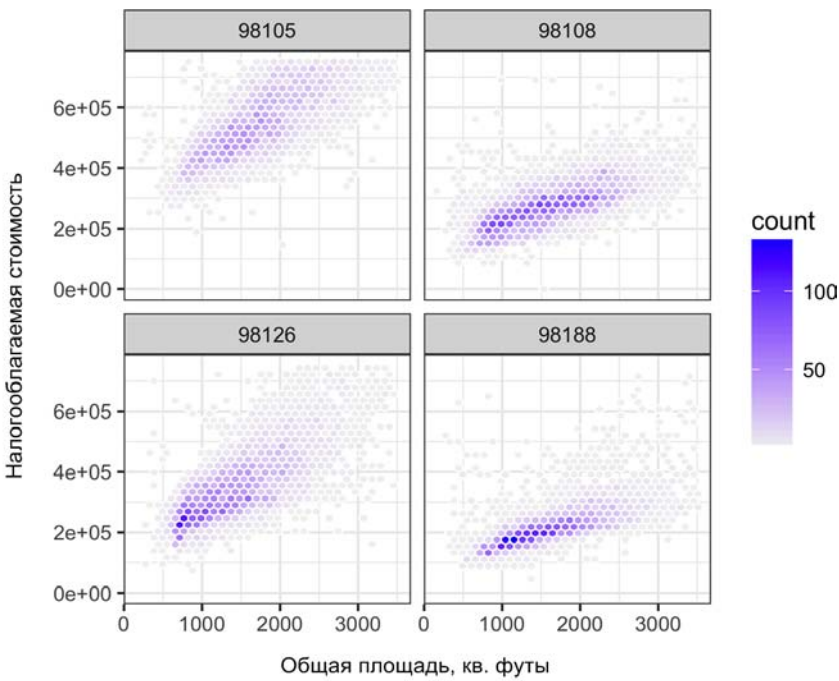


Рис. 1.12. Налогооблагаемая стоимость против общей площади в квадратных футах по почтовому индексу

Понятие кондиционных переменных в графической системе было впервые применено в Trellis graphics — методологии для отображения наборов сложных многомерных данных, которая получила развитие благодаря Рику Беккеру (Rick Becker), Билу Кливленду (Bill Cleveland) и другим в Bell Labs [Trellis-Graphics]. Эта идея распространилась на различные современные графические системы, такие как пакеты `lattice` ([lattice]) и `ggplot2` в R и модули `Seaborn` ([seaborne]) и `Bokeh` ([bokeh])

[4] Кондиционные переменные (conditioning variables) — фоновые переменные, характеризующие объект исследования, которые используются для построения правдоподобных значений путем привнесения недостающих данных. — *Прим. пер.*

в Python. Кондиционные переменные также являются неотъемлемой частью бизнес-аналитических платформ, таких как Tableau и Spotfire. С появлением обширных вычислительных возможностей современные платформы визуализации переместились далеко за пределы скромных начинаний разведочного анализа данных. Однако ключевые понятия и инструменты, разработанные за эти годы, по-прежнему формируют основу для этих систем.

Ключевые идеи для исследования двух или более переменных

- Графики с шестиугольной сеткой и контурные графики — это полезные инструменты, которые позволяют выполнять графический анализ сразу двух числовых переменных, несмотря на огромные объемы данных.
- Таблицы сопряженности — это стандартный инструмент для изучения количеств двух категориальных переменных.
- Коробчатые диаграммы и скрипичные графики позволяют отображать числовую переменную в сопоставлении с категориальной.

Дополнительные материалы для чтения

- В книге Baumer B., Kaplan D., Horton. Modern Data Science with R. — CRC Press, 2017 ("Современная наука о данных вместе с R") представлено превосходное введение в "грамматику для графиков" ("gg" в названии пакета ggplot).
- Ggplot2: Elegant Graphics for Data Analysis. — Hadley Wickham, Springer, 2009 (Ggplot2: изящная графика для анализа данных) — это превосходный ресурс от создателя пакета ggplot2.
- Джозеф Фрювальд (Josef Fruehwald) предлагает онлайновое учебное руководство по пакету ggplot2 (**http://www.ling.upenn.edu/~joseff/avml2012/**).

Резюме

С разработкой разведочного анализа данных (EDA), впервые внедренного Джоном Тьюки, статистика заложила фундамент, который стал предпосылкой для науки о данных. Ключевая идея EDA заключается в том, что первый и самый важный шаг в любом проекте, основанном на данных, состоит в том, чтобы *посмотреть на данные*. Благодаря обобщению и визуализации данных можно получить ценное интуитивное понимание проекта.

Данная глава была посвящена обзору понятий, начиная с простых метрических показателей, таких как оценки центрального положения и вариабельности, и заканчивая насыщенным визуальным отображением с целью анализа связей между многочисленными переменными (см. рис. 1.12). Разнообразный набор инструментов и специальных приемов, создаваемых сообществом разработчиков ПО с открытым исходным кодом, в сочетании с выразительностью R и Python создали огромное количество способов исследования и анализа данных. И разведочный анализ должен быть краеугольным камнем любого проекта науки о данных.

Распределения данных и выборок

Популярное мнение ошибочно утверждает, что эра больших данных сводит на нет потребность в выборке. На самом деле быстрое распространение данных переменного качества и релевантности укрепляет потребность в выборке как инструменте эффективной работы с разнообразными данными и минимизации смещения. Даже в проекте на основе больших данных предсказательные модели, как правило, разрабатываются и апробируются при помощи выборок. Выборки также используются в самых разнообразных тестах (например, в ценообразовании, веб-обработке).

На рис. 2.1 показана схема, которая подкрепляет понятия из данной главы. Слева представлена популяция, которая в статистике предположительно подчиняется базовому, но *неизвестному* распределению. Единственное, что имеется, — это *выборка* данных и ее эмпирическое распределение, показанное справа. Для того чтобы "попасть" из левой стороны в правую, используется процедура *отбора* (представленная стрелкой). Традиционная статистика сосредоточена главным образом на левой стороне, используя теорию, основанную на серьезных предположениях о характере популяции. Современная статистика переместилась в правую сторону, где такие предположения не требуются.

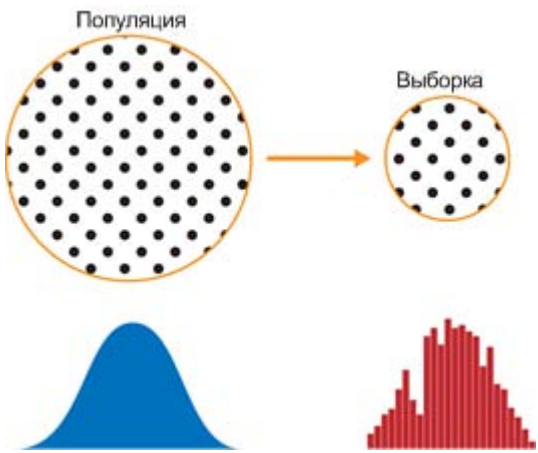


Рис. 2.1. Сравнение популяции с выборкой

В целом аналитикам данных не нужно беспокоиться о теоретической природе левой стороны; вместо этого им следует уделять основное внимание процедурам отбора и располагаемым данным. Правда, существуют некоторые существенные

исключения. Иногда данные генерируются из физического процесса, который может быть смоделирован. Самый простой пример — бросание монеты: оно подчиняется биномиальному распределению. Любая реальная биномиальная ситуация (купить или не купить, мошенничество или не мошенничество, нажать или не нажать) может быть эффективным образом смоделирована монетой (разумеется, с подстроенной вероятностью приземления орла). В этих случаях можно получить дополнительную информацию посредством использования нашего понимания популяции.

Случайный отбор и смещенная выборка

Выборка — это подмножество данных из более крупного набора данных, который в статистике называется *популяцией*, или *генеральной совокупностью*. Популяция в статистике — это не то же самое, что популяция в биологии — это большой, заданный, но нередко теоретический или воображаемый, набор данных.

Ключевые термины

Выборка (sample)

Подмножество большего по размеру набора данных.

Популяция (population)

Более крупный набор данных или идея набора данных.

Синоним: генеральная совокупность.

***N* (*n*)**

Размер популяции (выборки).

Случайный отбор (random sampling)

Выемка элементов в выборку в произвольном порядке.

Стратифицированный отбор (stratified sampling)

Разделение популяции на страты и случайный отбор элементов из каждой страты.

Простая случайная выборка (simple random sample)

Выборка, получаемая в результате случайного отбора без разбиения популяции на страты.

Смещенная выборка (sample bias)

Выборка, которая представляет популяцию в искаженном виде.

Случайный отбор — это процесс, в котором каждый доступный член популяции, подвергаемой отбору, имеет равную возможность попасть в выборку при каждой выемке. Результирующая выборка называется *простой случайной выборкой*. Отбор может быть выполнен с *возвратом*, когда после каждой выемки наблюдения кладутся назад в популяцию для возможно повторного отбора в будущем. Как альтер-

натива, отбор может быть выполнен *без возврата*, и в этом случае однажды выбранные наблюдения недоступны для будущих выемок.

Качество данных часто имеет большее значение, чем их количество, когда выполняется оценка либо создается модель на основе выборки. Качество данных в науке о данных сопряжено с полнотой, последовательностью формата, чистотой и точностью отдельных точек данных. Наука статистика добавляет сюда понятие *репрезентативности*.

Классический пример — опрос, выполненный в 1936 г. журналом "Литературный обзор" (Literary Digest), который предсказал победу Альфреда Лэндона (Al Landon) над Франклином Рузвельтом (Franklin Roosevelt). Периодическое ежедневное издание "Литературный обзор" опросило своих подписчиков, зарегистрированных в базе издания, плюс дополнительно еще людей (в общей сложности более 10 млн человек) и предсказало сокрушительную победу Лэндона. Джордж Гэллап (George Gallup), основатель института опроса общественного мнения, проводил опросы каждые две недели всего у 2 тыс. респондентов и точно предсказал победу Рузвельта. Разница заключалась в том, как выбирались респонденты.

Журнал "Литературный обзор" сделал ставку на количество, мало обращая внимания на метод отбора. В итоге оказалось, что были опрошены люди с относительно высоким социально-экономическим статусом (собственные подписчики издания плюс те, кто входили в списки маркетологов на основании владения предметами роскоши, такими как телефоны и автомобили). Результатом стала *смещенная выборка*, т. е. она отличалась некоторым содержательным неслучайным характером от остальной, более многочисленной популяции, которую эта выборка должна была представлять. Термин *"неслучайный"* очень важен — едва ли любая выборка, включая случайные выборки, будет для популяции строго репрезентативной. Смещенная выборка возникает, когда разница становится содержательной, и ожидается, что она продолжится в отношении других выборок, вынимаемых таким же образом, что и первая.

Систематическая ошибка самоотбора

Отзывы о ресторанах, гостиницах, кафе и других объектах досуга и развлечений, которые вы читаете в социальных сетях, таких как Yelp, подвержены смещению, потому что предлагающие их люди не отбираются в произвольном порядке, а наоборот, они сами взяли на себя инициативу высказаться. Это приводит к так называемой систематической ошибке самоотбора — люди, мотивированные написать отзыв, могут иметь неудачный опыт, быть материально-заинтересованными или же просто отличаться по складу от тех людей, которые отзывы не пишут. Отметим, что, хотя выборки на основе самоотбора могут быть ненадежными индикаторами истинного положения дел, они могут быть более надежными в простом сравнении одного предприятия с аналогичным ему; та же самая систематическая ошибка самоотбора может касаться каждого из них.

Смещение

Статистическое смещение относится к ошибкам измерения и отбора, которые систематичны и порождаются процессом измерения или отбора данных. Важно проводить различие между ошибками, возникающими в силу случайной возможности и ошибками вследствие смещения. Рассмотрим физический процесс стрельбы из ружья по мишени. Поражение абсолютного центра мишени не будет происходить каждый раз или даже не будет происходить вообще. Несмещенный процесс произведет ошибку, но она будет случайной и не проявит тенденцию к любому из направлений (рис. 2.2). Приведенные на рис. 2.3 результаты показывают смещенный

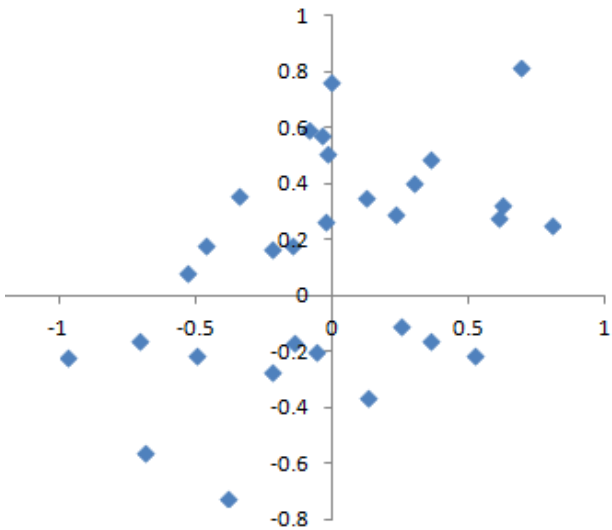


Рис. 2.2. Диаграмма рассеяния выстрелов из ружья с настроенным прицелом

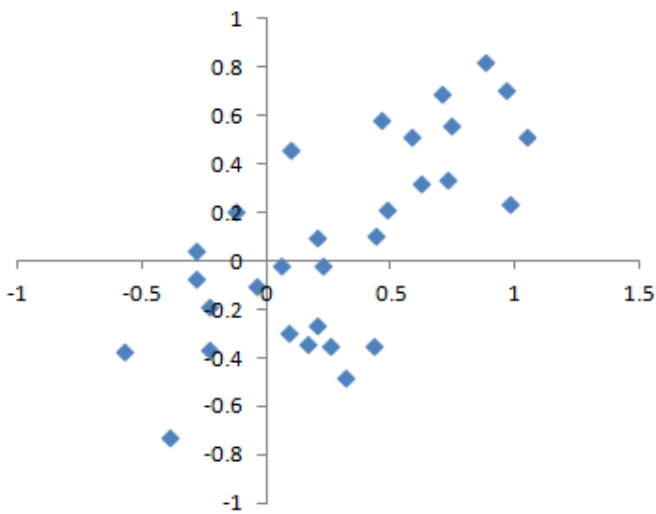


Рис. 2.3. Диаграмма рассеяния выстрелов из ружья со смещенным прицелом

процесс — по-прежнему имеется случайная ошибка как в направлении *x*, так и в направлении *y*, но помимо этого имеется смещение. Выстрелы демонстрируют тенденцию падать в верхний правый квадрант.

Смещение возникает в разных формах и может быть заметным или невидимым. Когда результат действительно наводит на мысль о смещении (например, опираясь на эталон или фактические значения), это сигнализирует о том, что статистическая или машинно-обучаемая модель была задана неправильно, либо не была учтена важная переменная.

Произвольный выбор

Для того чтобы предотвратить проблему смещенной выборки, которая привела журнал "Литературный обзор" к предсказанию победы Лэндона над Рузвельтом, Джордж Гэллап (чья фотография приведена на рис. 2.4) с научной точки зрения сделал ставку на более надежные методы достижения выборки, которая была репрезентативной для избирателей США. Сегодня существуют разнообразные методы, которые позволяют достигать репрезентативности, но в основе их всех лежит *случайный отбор*.

Рис. 2.4. Джордж Гэллап, который стал знаменитым благодаря провалу журнала "Литературный обзор" при работе с большими данными

Случайный отбор не всегда прост, и надлежащее определение доступной популяции является ключом. Предположим, мы хотим сгенерировать репрезентативный профиль покупателей, и нам нужно провести их пилотный статистический обзор. Обзор нужен репрезентативный, но он трудоемок.

Сначала мы должны определить, кто является покупателем. Мы можем выбрать все записи покупателей с суммой покупки более 0. Стоит ли включать всех прошлых покупателей? Стоит ли включать компенсации? Внутренние покупки? Перекупщиков? Биллингового агента и покупателя?

Далее мы должны определить процедуру отбора. Она может заключаться в том, чтобы "выбрать 100 покупателей наугад". Там, где задействован отбор из потока (например, транзакции покупателей в реальном времени или посетители веб-сайта), особую важность принимают соображения, касающиеся времени (например, посетитель веб-сайта в 10:00 в будний день может отличаться от посетителя веб-сайта в 22:00 в выходные).

В *стратифицированном отборе* популяция разделяется на *страты*, и случайные выборки берутся из каждой страты. Политические социологи могут попытаться выяснить электоральные предпочтения белых, афроамериканцев и латиноамериканцев. Простая случайная выборка, взятая из населения США, приведет к очень небольшому числу афроамериканцев и латиноамериканцев, и поэтому в стратифицированном отборе этим стратам может быть придан больший вес, чтобы получить эквивалентные размеры выборок.

Размер против качества: когда размер имеет значение?

В эру больших данных вызывает удивление, что иногда оказывается, что чем меньше, тем лучше. Время и усилие, потраченное на случайный отбор, не только уменьшают смещение, но и позволяют уделять большее внимание разведке данных и их качеству. Например, пропущенные данные и выбросы могут содержать полезную информацию. Розыск отсутствующих значений или вычисление выбросов в миллионах записей могут оказаться непозволительно дорогостоящими, но эта работа в выборке, состоящей из нескольких тысяч записей, вполне выполнима. Отображение данных на графиках и ручное исследование практически бессмысленны, если данных слишком много.

Когда же нужны массивные объемы данных?

В классическом сценарии важности больших данных эти данные не только большие, но и редкие. Возьмем, к примеру, поисковые запросы, получаемые компанией Google, где столбцы — это условия, строки — отдельные поисковые запросы, а значения ячеек равны 0 или 1 в зависимости от того, содержит ли запрос термин или нет. Задача состоит в том, чтобы наилучшим образом определить предсказанное назначение поиска для заданного запроса. В английском языке свыше 150 тыс. слов, и Google обрабатывает более 1 трлн запросов в год. В результате получится огромная матрица, подавляющее большинство записей в которой будут равны "0".

Это настоящая задача для больших данных — по большинству запросов могут быть возвращены эффективные результаты поиска, только когда накоплены такие огромные количества данных. И чем больше данных накапливается, тем лучше результаты. Для популярных критериев поиска это не такая проблема — эффективные данные могут быть найдены довольно быстро для небольшого количества чрезвычайно популярных тем, находящихся в тренде в то или иное время. Действительная важность современной поисковой технологии заключается в способности возвращать подробные и полезные результаты для огромного разнообразия поис-

ковых запросов, включая те, которые возникают с частотой, скажем, всего один на миллион.

Возьмем поисковую фразу "Рики Рикардо и Красная Шапочка" (Ricky Ricardo and Little Red Riding Hood). В первые годы существования Интернета этот запрос, вероятно, возвратил бы результаты по лидеру группы *Рики Рикардо*, телешоу *"Я люблю Люси"* (I love Lucy), в котором он снимался в главной роли, и детскую сказку *"Красная Шапочка"*. Теперь же, когда накоплены триллионы поисковых запросов, этот поисковый запрос возвращает в точности эпизод из телешоу "Я люблю Люси", в котором Рики театрально рассказывает сказку о Красной Шапочке своему новорожденному сыну, пользуясь комичным смешением английского и испанского языков.

Следует иметь в виду: чтобы поисковый запрос был эффективным, число фактических искомых записей, в которых появляется этот точный поисковый запрос или что-то очень похожее (вместе с информацией о том, на какой ссылке люди в конечном итоге нажали), возможно, должно измеряться тысячами. Однако для того чтобы эти релевантные записи получить, необходимо иметь триллионы точек данных (и случайный отбор тут, конечно, не поможет). *См. также разд. "Длиннохвостые распределения" далее в этой главе*.

Выборочное среднее против популяционного среднего

Символ $\bar{x}$ (произносится "x с чертой") используется для представления среднего значения выборки из популяции, тогда как μ (мю) применяется для представления среднего значения по всей популяции. В чем важность этого различия? Информация о выборках наблюдаема, а информация о крупных популяциях часто выводится из меньших по размеру выборок. В статистике предпочитают разделять эти две вещи в плане символического обозначения.

Ключевые идеи для случайного отбора и смещенной выборки

- Даже в эру больших данных случайный отбор остается важным инструментом в арсенале аналитика данных.
- Смещение возникает, когда данные измерений, или наблюдения, систематически ошибочны, потому что они не репрезентативны, т. е. не представляют всю популяцию в целом.
- Качество данных часто важнее их количества, и случайный отбор может уменьшить смещение и оказать содействие повышению качества, достижение которого было бы непозволительно дорогим.

Дополнительные материалы для чтения

- Полезный обзор процедур отбора можно найти в главе Рональда Фрикера (Ronald Fricker) "Методы отбора для статистических опросов в Веб и по электронной почте" (Sampling Methods for Web and E-mail Surveys), в книге "Руководство Sage по онлайновым методам статистического исследования" (Sage Handbook of Online Research Methods). В данной главе представлен обзор модификаций метода случайного отбора, которые часто используются по практическим причинам стоимости или выполнимости.
- Статью о провале проведенного журналом "Литературный обзор" статистического опроса можно найти на веб-сайте Capital Century (**http://www.capitalcentury.com/1935.html**).

Систематическая ошибка отбора

Перефразируя бейсболиста Йоги Берра (Yogi Berra), "если вы не знаете, что ищете, присмотритесь повнимательнее, и вы это найдете".

Систематическая ошибка отбора (или предвзятость при отборе, selection bias) относится к практике избирательного подбора данных — осознанно или неосознанно. Таким образом, что она приводит к обманчивому или недолговечному выводу.

Ключевые термины

Смещение (bias)

Систематическая ошибка.

Прочесывание данных (data snooping)

Подробная ревизия данных с целью найти что-то интересное.

Эффект бескрайнего поиска (vast search effect)

Смещение или невоспроизводимость, вытекающие из многократного моделирования данных либо моделирования данных с большими количествами предикторных переменных.

Если определить гипотезу и провести хорошо проработанный эксперимент с целью ее проверки, то можно быть твердо уверенным в полученном выводе. Однако зачастую это не так. Вместо этого часто смотрят на имеющиеся данные в попытке разглядеть закономерности. Но является ли закономерность реальной, или же она всего лишь продукт *прочесывания данных*, т. е. подробной ревизии данных, пока не появится нечто интересное? Среди статистиков популярна поговорка: "Если мучать данные слишком долго, то рано или поздно они дадут признательные показания".

Разницу между явлением, в котором вы удостоверяетесь, когда проверяете гипотезу при помощи эксперимента, и явлением, которое вы обнаруживаете, преследуя имеющиеся данные, можно разъяснить следующим мысленным экспериментом.

Предположим, что кто-то говорит вам, что он может заставить приземлиться подбрасываемую им монету орлом 10 бросков подряд. Вы принимаете вызов (эквивалент эксперимента), испытатель приступает к 10-кратному подбрасыванию монеты, и всякий раз она приземляется орлом вверх. Совершенно очевидно, что вы припишите этому человеку какой-то особый талант — вероятность, что в результате 10 бросков монеты она просто по чистой случайности повернется орлом, составляет 1 из 1000.

Теперь предположим, что диктор на стадионе просит, чтобы все присутствующие 20 тыс. человек подбросили монету 10 раз и сообщили работнику стадиона, в случае если они получат выпадение 10 орлов подряд. Шанс, что *кто-то* на стадионе доберется до 10 орлов, чрезвычайно высокий (более 99% — это 1 минус вероятность, что никто не получит 10 орлов). Безусловно, выбор задним числом человека (или людей), который получит 10 орлов на стадионе, не говорит о том, что он имеет какой-то особый талант — скорее всего, это просто удача.

Поскольку неоднократная ревизия больших наборов данных является в науке о данных ключевым ценностным предложением, систематической ошибке отбора следует уделять пристальное внимание. Форму систематической ошибки отбора, имеющую особое значение для аналитиков данных, Джон Элдер (John Elder), основатель компании Elder Research, уважаемой консалтинговой компании в области глубинного анализа данных, называет *эффектом бескрайнего поиска*. Если вы неоднократно выполняете разные модели и задаете разные вопросы в условиях больших наборов данных, то вы непременно найдете нечто интересное. Является ли найденный результат по-настоящему чем-то заслуживающим внимания, или же это случайный выброс?

Против этого можно принять защитные меры, задействовав контрольный набор с отложенными данными (holdout), а иногда более одного контрольного набора, на основе которых можно подтвердить результативность. Помимо этого, Элдер также выступает за использование того, что он называет *целевой перетасовкой* (target shuffling — по сути дела, это перестановочный тест) для проверки достоверности предсказательных ассоциаций, которые предлагает модель глубинного анализа данных.

В статистике типичные формы систематической ошибки отбора, в дополнение к эффекту бескрайнего поиска, включают неслучайный отбор *(смещение при отборе образцов)*, данные, полученные в результате отбора по принципу снятия сливок, выбор временны́х интервалов, которые подчеркивают тот или иной статистический эффект, и остановку эксперимента, когда результаты выглядят "интересными".

Регрессия к среднему

Регрессия к среднему значению относится к явлению, связанному с последовательными измерениями заданной переменной: предельные наблюдения имеют тенденцию сопровождаться более центральными. Придание особого внимания и смысла предельному значению может привести к одной из форм систематической ошибки отбора.

Любители спорта знакомы с явлением "новичка года и кризиса прежнего лидера". Среди спортсменов, которые начинают свою карьеру в конкретный сезон (класс новичков), всегда присутствует тот, кто оказывается результативнее, чем все остальные. Обычно этот "новичок года" не достигает таких же результатов в следующем году. Почему?

Почти во всех главных видах спорта, по крайней мере в командных состязаниях с мячом или шайбой, существует два элемента, которые играют важную роль в общей результативности:

- навык;
- удача.

Регрессия к среднему значению является следствием определенной формы систематической ошибки отбора. Когда мы выбираем новобранца с лучшей результативностью, навык и удача, вероятно, этому содействуют. В свой следующий сезон навык по-прежнему будет на месте, но в большинстве случаев удача будет отсутствовать, и поэтому его результативность упадет — она будет регрессировать. Это явление было впервые идентифицировано Фрэнсисом Гальтоном в 1886 г. [Galton-1886], который описал его в связи с генетическими тенденциями; например, дети чрезвычайно высоких мужчин склонны не быть столь же высокими, что и их отцы (рис. 2.5).

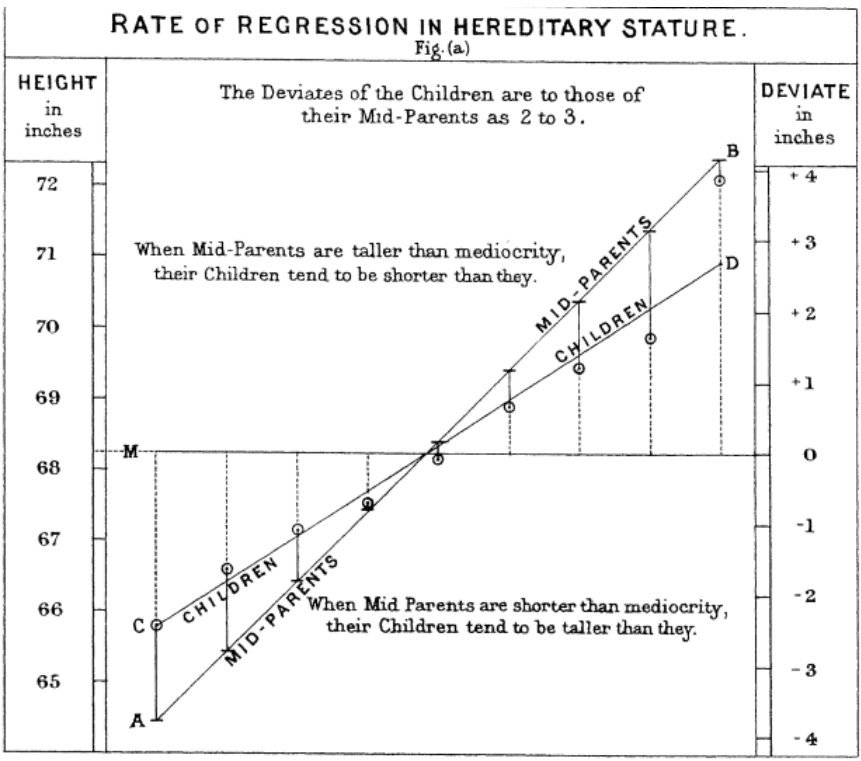


Рис. 2.5. Исследовательская работа Гальтона, в которой он идентифицировал феномен регрессии к среднему

Регрессия, т. е. "возвращение назад", к среднему отличается от метода статистического моделирования линейной регрессии, в котором линейная связь оценивается между предикторными переменными и выходной переменной.

Ключевые идеи для систематической ошибки отбора

- Определение гипотезы, и далее сбор данных, согласно принципам рандомизации и случайного отбора обеспечивают защиту от смещения.
- Все другие формы анализа данных подвергаются риску появления смещения, вытекающего из процесса сбора/анализа данных (многократное выполнение моделей в глубинном анализе данных, прочесывание данных во время статистического исследования и отбор интересующих событий задним числом).

Дополнительные материалы для чтения

- Статья "Идентификация и предотвращение смещения в статистических исследованиях" в сборнике (что удивительно) "Пластмассовая и восстановительная хирургия" (Pannucci C. J., и Wilkins E. G. Identifying and Avoiding Bias in Research // Plastic and Reconstructive Surgery. — 2010. — August) содержит превосходный анализ различных типов смещения, которые могут попасть в статистическое исследование, включая систематическую ошибку отбора.
- Статья "Одурачен произвольностью из-за систематической ошибки отбора" (Fooled by Randomness Through Selection Bias) Майкла Хэрриса (Michael Harris) предоставляет интересный обзор соображений в отношении систематической ошибки отбора в стратегиях торговли на фондовом рынке с точки зрения трейдеров (**http://systemtradersuccess.com/fooled-by-randomness-through-selection-bias/**).

Выборочное распределение статистики

Термин *"выборочное распределение"* статистики обозначает распределение некоторой выборочной статистики, т. е. выборочной статистической величины, на большом числе выборок, вынимаемых из одной и той же популяции. Значительная часть классической науки статистики занимается получением статистических выводов из (малых) выборок и (очень крупных) популяций.

Ключевые термины

Выборочная статистика (sample statistic)

Метрический показатель, который вычисляется для выборки данных, вынимаемой из более крупной популяции.

Синоним: выборочная статистическая величина.

Распределение данных (data distribution)

Частотное распределение индивидуальных значений в наборе данных.

Выборочное распределение (sampling distribution)

Частотное распределение *выборочной статистики* на многочисленных выборках или повторных выборках.

Центральная предельная теорема (central limit theorem)

Тенденция выборочного распределения принимать нормальную форму по мере увеличения размера выборок.

Синоним: ЦПТ.

Стандартная ошибка (standard error)

Вариабельность (стандартное отклонение) выборочной статистики на многочисленных выборках (не путать со *стандартным отклонением*, которое как таковое относится к вариабельности индивидуальных значений данных).

Как правило, выборка вынимается с целью измерения чего-нибудь (при помощи *выборочной статистики*) либо моделирования чего-нибудь (при помощи статистической или машинно-обучаемой модели). Учитывая, что наша оценка или модель основывается на выборке, она может иметь отклонения, т. е. статистические ошибки; она может отличаться, если мы решим вынуть другую выборку. Мы, следовательно, заинтересованы знать, насколько она может отличаться — ключевой проблемой является *выборочная вариабельность, т. е. насколько оценка варьирует между выборками*. Если бы у нас было много данных, то мы могли бы вынимать дополнительные выборки и наблюдать распределение выборочной статистики непосредственно. Как правило, мы будем вычислять нашу оценку или модель, используя столько данных, сколько есть в наличии, так что возможность выемки дополнительных выборок из популяции имеется далеко не всегда.

Выборочное распределение против распределения данных

Важно различать распределение индивидуальных точек данных, именуемое *распределением данных*, и распределение выборочной статистики, известное как *выборочное распределение*.

Распределение выборочной статистики, такой как среднее, вероятно, будет более регулярным и колоколообразным, чем распределение самих данных. Чем больше выборка, на которой основывается статистика, тем более правдоподобной она является. Кроме того, чем больше выборка, тем у́же распределение выборочной статистики.

Это иллюстрируется примером с использованием годового дохода для ссудозаявителей в кредитный клуб Lending Club (*см. разд. "Небольшой пример: предсказание невозврата ссуды" главы 6*, где приводится описание данных). Возьмем из этих данных три выборки: выборку с 1000 значениями, выборку с 1000 средними из 5 значений и выборку с 1000 средними из 20 значений. Затем построим гистограмму каждой выборки, создав рис. 2.6.

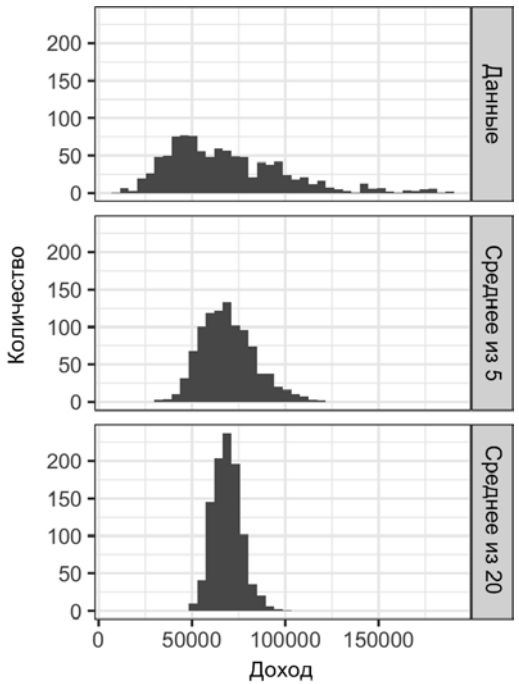


Рис. 2.6. Гистограмма годовых доходов 1000 ссудозаявителей (вверху), затем 1000 средних с числом заявителей n = 5 (в центре) и n = 20 (внизу)

Гистограмма индивидуальных значений данных широко развернута и скошена к более высоким значениям, как и должно ожидаться с данными о доходах. Обе гистограммы средних из 5 и 20 значений более компактны и более колоколообразные. Ниже приведен фрагмент кода на R, который генерирует эти гистограммы при помощи пакета визуализации `ggplot2`.

```
library(ggplot2)
# взять простую случайную выборку
samp_data <- data.frame(income=sample(loans_income, 1000),
                        type='data_dist')
# взять выборку средних из 5 значений
samp_mean_05 <- data.frame(
  income = tapply(sample(loans_income, 1000*5),
                  rep(1:1000, rep(5, 1000)), FUN=mean),
                  type = 'mean_of_5')
```

```
# взять выборку средних из 20 значений
samp_mean_20 <- data.frame(
  income = tapply(sample(loans_income, 1000*20),
                  rep(1:1000, rep(20, 1000)), FUN=mean),
                  type = 'mean_of_20')
# связать кадры данных data.frames и конвертировать тип в фактор
income <- rbind(samp_data, samp_mean_05, samp_mean_20)
income$type = factor(income$type,
                     levels=c('data_dist', 'mean_of_5', 'mean_of_20'),
                     labels=c('Данные', 'Среднее из 5', Среднее 20'))
# построить гистограммы
ggplot(income, aes(x=income)) +
       geom_histogram(bins=40) + facet_grid(type ~ .)
```

Центральная предельная теорема

Феномен, называемый *центральной предельной теоремой*, гласит, что средние значения, вынутые из многочисленных выборок, будут напоминать знакомую колоколообразную нормальную кривую *(см. разд. "Нормальное распределение" далее в этой главе)*, даже если исходная популяция не является нормально распределенной, при условии, что размер выборок достаточно крупный и отклонение данных от нормальности не является слишком высоким. Центральная предельная теорема позволяет использовать такие формулы аппроксимации нормальным распределением, как *t*-распределение, применяемое в вычислении распределений выборок для статистического вывода, а именно доверительные интервалы и проверки статистических гипотез.

В традиционных статистических проверках центральной предельной теореме уделяется большое внимание, потому что она лежит в основе механизма доверительных интервалов и проверок статистических гипотез, которые сами занимают половину содержимого таких текстов. Аналитики данных должны знать об этой ее роли, но поскольку роль формальных проверок гипотез и доверительных интервалов в науке о данных небольшая, и так или иначе всегда есть бутстрапирование, центральная предельная теорема не играет какую-то особо важную роль в практике науки о данных.

Стандартная ошибка

Стандартная ошибка — это одиночный метрический показатель, который обобщает вариабельность в выборочном распределении для статистики. Стандартную ошибку можно оценить с использованием статистики, опираясь на стандартное отклонение *s* значений выборки и размер выборки *n*:

$$\text{Стандартная ошибка} = \frac{s}{\sqrt{n}}.$$

По мере увеличения размера выборки стандартная ошибка уменьшается, соответствуя тому, что наблюдалось на рис. 2.6. Связь между стандартной ошибкой и размером выборки иногда носит название *правила квадратного корня из n*: для сокращения стандартной ошибки в 2 раза, размер выборки должен быть увеличен в 4 раза.

Достоверность формулы стандартной ошибки вытекает из центральной предельной теоремы *(см. предыдущий раздел)*. На деле для понимания стандартной ошибки вам не нужно полагаться на центральную предельную теорему. Рассмотрим следующий подход к измерению стандартной ошибки:

1. Получить несколько совершенно новых выборок из популяции.
2. Для каждой новой выборки вычислить статистику (например, среднее).
3. Рассчитать стандартное отклонение статистики, вычисленной на шаге 2; использовать ее в качестве оценки стандартной ошибки.

На практике этот подход получения новых выборок для оценки стандартной ошибки обычно не выполним (и статистически очень расточителен). К счастью, оказывается, что нет необходимости извлекать совершенно новые выборки; вместо этого можно использовать повторные *бутстраповские выборки (см. разд. "Бутстрап" далее в этой главе)*. В современной статистике бутстрап стал типичным способом оценки стандартной ошибки. Этот метод может использоваться фактически для любой статистики и не опирается на центральную предельную теорему или другие допущения о характере распределения.

Разница между стандартным отклонением и стандартной ошибкой

Не путайте стандартное отклонение (которое показывает вариабельность отдельных точек данных) со стандартной ошибкой (которая показывает вариабельность выборочного метрического показателя).

Ключевые идеи для выборочного распределения статистики

- Частотное распределение выборочной статистики говорит о том, на сколько этот метрический показатель будет отличаться от выборки к выборке.
- Это выборочное распределение можно оценить посредством бутстрапа либо формул, которые опираются на центральную предельную теорему.
- Ключевым метрическим показателем, который обобщает вариабельность выборочной статистики, является его стандартная ошибка.

Дополнительные материалы для чтения

Онлайновый мультимедийный ресурс по статистике Дэвида Лэйна располагает полезной симуляцией, которая позволяет задать выборочную статистику, размер выборки и число итераций, а также визуализировать гистограмму получившегося частотного распределения (**http://onlinestatbook.com/stat_sim/sampling_dist/**).

Бутстрап

Один из простых и эффективных способов оценки выборочного распределения статистики или модельных параметров состоит в том, чтобы вынимать дополнительные выборки с возвратом из самой выборки и повторно вычислять статистику или модель для каждой повторной выборки. Данная процедура называется *бутстрапом* (от англ. *bootstrap* — раскрутка, самонастройка), и она не сопряжена с какими-либо предположениями о нормальном распределении данных или выборочной статистики.

Ключевые термины

Бутстраповская выборка (bootstrap sample)

Выборка, взятая с возвратом из набора наблюдаемых данных.

Синоним: бутстрап-выборка.

Повторный отбор (resampling)

Процесс многократного взятия выборок из наблюдаемых данных; включает процедуры бутстрапа и перестановки (перетасовки).

Синонимы: перевыборка, ресемплинг.

Процесс бутстрапирования можно концептуально представить, как повторение исходной выборки тысячи или миллионы раз с тем, чтобы получить гипотетическую популяцию, которая воплощает все знание, исходя из оригинальной выборки (она просто больше). Затем из этой гипотетической популяции можно извлекать выборки в целях оценки выборочного распределения (рис. 2.7).

Рис. 2.7. Идея бутстрапирования

На практике нет необходимости фактически повторять выборку огромное число раз. Просто после каждой выемки мы возвращаем каждое наблюдение назад; т. е. мы выполняем *отбор с возвратом*. Тем самым мы эффективным образом создаем

бесконечную популяцию, в которой вероятность вынимаемого элемента остается неизменной от выемки к выемке. Алгоритм повторного бутстраповского отбора среднего значения для выборки размера *n* будет следующим:

1. Вынуть выборочное значение, записать его и вернуть назад.
2. Повторить *n* раз.
3. Записать среднее для *n* повторно опробованных значений.
4. Повторить шаги 1–3 *R* раз.
5. Использовать *R* результатов, чтобы:
   - вычислить их стандартное отклонение (оно оценивает стандартную ошибку выборочного среднего);
   - построить гистограмму или коробчатую диаграмму;
   - найти доверительный интервал.

Число итераций *R* процесса бутстрапирования устанавливается несколько произвольно. Чем больше выполняется итераций, тем точнее оценка стандартной ошибки или доверительного интервала. Результатом данной процедуры является бутстраповский набор выборочных статистик или оценочных модельных параметров, которые далее можно обследовать, чтобы увидеть, насколько они переменчивы.

Программный пакет R `boot` совмещает эти шаги в одной функции. Например, в следующем ниже примере бутстрап применяется к доходам людей, которые берут ссуды:

```
library(boot)
stat_fun <- function(x, idx) median(x[idx])
boot_obj <- boot(loans_income, R = 1000, statistic=stat_fun)
```

Функция `stat_fun` вычисляет медиану для заданной выборки, определенной индексом `idx`. Результат будет следующим:

```
Bootstrap Statistics :
    original    bias      std. error
t1*    62000  -70.5595      209.1515
```

Первоначальная оценка медианы составляет 62 тыс. долларов. Бутстраповское распределение говорит о том, что оценка имеет *смещение* порядка –70 долларов и стандартную ошибку 209 долларов.

Бутстрап может использоваться с многомерными данными, где строки отбираются как единое целое (рис. 2.8). Затем на бутстрапированных данных можно выполнить модель, например, чтобы оценить стабильность (вариабельность) модельных параметров или улучшить предсказательную силу. Что касается классификационных и регрессионных деревьев (так называемых *деревьев решений*), то выполнение множественных деревьев на бутстраповских выборках и далее усреднение их предсказаний (или, в случае классификации, принятие решения мажоритарным голосованием, т. е. большинством голосов) обычно оказывается результативнее, чем

использование одиночного дерева. Этот процесс называется бутстрап-агрегированием, или бэггингом (сокращение от bootstrap aggregating: *см. разд. "Бэггинг и случайный лес" главы 6*).

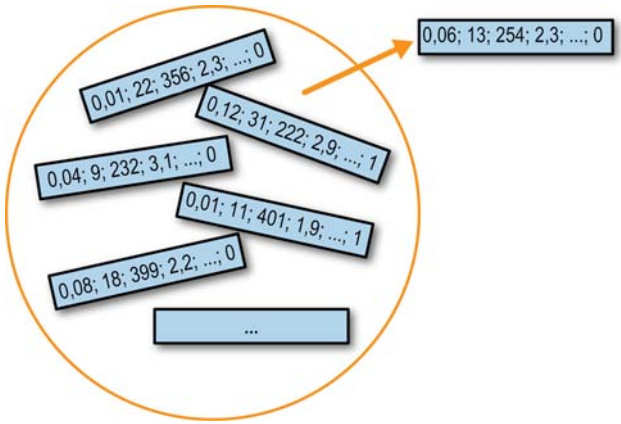


Рис. 2.8. Многомерный отбор бутстраповских выборок

Многократный отбор бутстраповских выборок концептуально не представляет труда, и экономист и демограф Джулиан Саймон (Julian Simon) в своей работе 1969 г. "Методы фундаментального исследования в социологии" (Basic Research Methods in Social Science, Random House) опубликовал резюме примеров повторного отбора, включая бутстрап. Однако этот метод также вычислительно емок и до широкого распространения вычислительных мощностей оставался физически неосуществимой возможностью. Он получил свое название и приобрел популярность после выхода книги стэндфордского статистика Брэдли Эфрона (Bradley Efron) и публикации нескольких статей в журналах в конце 1970-х — начале 1980-х гг. Данный прием в особенности был популярен среди исследователей, которые применяли статистику, но не являлись специалистами в области статистики, и предназначался для использования с метрическими показателями или моделями, где математические приближения не были легко доступны. Выборочное распределение среднего было хорошо проработано, начиная с 1908 г., что нельзя было сказать в отношении выборочного распределения многих других метрических показателей. Бутстрап может использоваться для определения размера выборок, экспериментов с разными значениями *n*, чтобы понять, как они влияют на выборочное распределение.

Когда метод повторения отбора бутстраповских выборок был представлен впервые, он был встречен со значительным скептицизмом; для многих он был связан с ореолом превращения соломы в золото. Этот скептицизм проистекал из непонимания цели бутстрапа.

Бутстрап не компенсирует малый размер выборки; оно не создает новые данные и при этом не заполняет дыры в существующем наборе данных. Оно просто сообщает о том, как поведут себя многочисленные дополнительные выборки, когда они будут выниматься из популяции, такой как наша исходная выборка.

Повторный отбор против бутстрапирования

Иногда термин *"повторный отбор"* используется в качестве синонима термина *"бутстрапирование"*, который был только что представлен в общем виде. Чаще всего термин *"повторный отбор"* также предполагает процедуры перестановки *(см. разд. "Перестановочный тест" главы 3)*, где многочисленные выборки объединяются, и отбор может быть сделан без возврата. В любом случае, термин *"бутстрап"* всегда подразумевает выборку из наблюдаемого набора данных с возвратом.

Ключевые идеи для бутстрапирования

- Бутстрап (отбор из набора данных с возвратом) является мощным инструментом для определения вариабельности выборочной статистики.
- Бутстрап может применяться одинаковым образом в самых различных обстоятельствах без обширного анализа математических приближений выборочных распределений.
- Этот метод также позволяет выполнять оценку выборочных распределений для статистик, где математическое приближение не разработано.
- Когда этот метод применяется к предсказательным моделям, агрегирование многочисленных предсказаний на основе бутстраповских выборок (бэггинг) превосходит по результативности одиночную модель.

Дополнительные материалы для чтения

- "Введение в бутстрап" (Efron B., Tibshirani R. An Introduction to the Bootstrap. — Chapman Hall, 1993) — первая исследовательская работа книжного формата, где в центре внимания был метод бутстрапирования. Данная книга по-прежнему пользуется популярностью.
- Ретроспектива, посвященная бутстрапу, в журнале "Статистическая наука" (Hall P. Prehistory // Statistical Science. — 2003. — Vol. 18. — № 2). В ней обсуждается (среди других предыдущих работ) в разделе "Предыстория" Питера Халла первая публикация о бутстрапе Джулиана Саймона 1969 г.
- В книге "Введение в статистическое обучение" обратитесь к разделам по бутстрапу и, в частности, бэггингу (Gareth James et al. Introduction to Statistical Learning. — Springer, 2013).

Доверительные интервалы

Таблицы частот, гистограммы, коробчатые диаграммы и стандартные ошибки — все они являются способами понять потенциальную ошибку в оценке выборки. Доверительные интервалы — это еще один такой способ.

Ключевые термины

Уровень доверия (confidence level)

Процент доверительных интервалов, созданных одинаковым образом из одной и той же популяции, который ожидаемо будет содержать целевую статистику.

Конечные точки интервала (interval endpoints)

Верх и низ доверительного интервала.

Человеку естественным образом свойственно избегать неопределенности; люди (в особенности эксперты) произносят фразу "Я не знаю" крайне редко. Аналитики и менеджеры, признавая наличие неопределенности, тем не менее неоправданно доверяются оценке, когда она представлена единственным числом *(точечная оценка)*. Метод представить оценку не единственным числом, а диапазоном, противодействует этой тенденции. Доверительные интервалы делают это способом, который берет свое начало в статистических принципах отбора.

Доверительные интервалы всегда сопровождаются уровнем покрытия, выраженным (высоким) процентом, скажем, 90 или 95%. 90-процентный доверительный интервал можно представить следующим образом: это интервал, который окружает центральные 90% бутстраповского выборочного распределения выборочной статистики *(см. разд. "Бутстрап" ранее в этой главе)*. Более широко, *x*-процентный доверительный интервал вокруг выборочной оценки должен в среднем содержать аналогичные выборочные оценки *x*% случаев (когда выполнена аналогичная процедура отбора).

С учетом выборки размера *n* и целевой выборочной статистики алгоритм для бутстраповского доверительного интервала будет следующим:

1. Извлечь случайную выборку размера *n* с возвратом из данных (повторная выборка).
2. Записать целевую статистику для повторной выборки.
3. Повторить шаги 1–2 много (*R*) раз.
4. Для *x*-процентного доверительного интервала отсечь $\left[\left(1-[x/100]\right)/2\right]\%$ от *R* результатов повторного отбора с обоих концов распределения.
5. Точками отсечения являются конечные точки *x*-процентного бутстраповского доверительного интервала.

На рис. 2.9 показан 90-процентный доверительный интервал для среднегодового дохода ссудозаявителей на основе выборки из 20 значений, для которой средним было 57 573 долларов.

Бутстрап широко применяется с целью генерации доверительных интервалов для большинства статистик или модельных параметров. Статистические учебники и программные системы с корнями в более полувековом бескомпьютерном статистическом анализе будут также опираться на доверительные интервалы, генерируемые

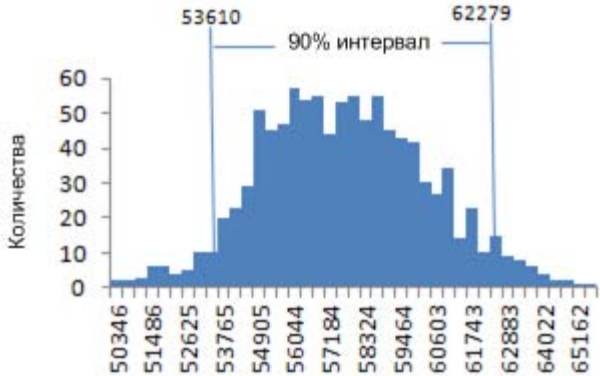


Рис. 2.9. Бутстраповский доверительный интервал для годового дохода ссудозаявителей на основе выборки из 20 значений

формулами, в особенности на *t*-распределение *(см. "t-Распределение Стьюдента" далее в этой главе)*.

Разумеется, когда у нас есть результат в виде выборки, нас больше всего интересует, "какова вероятность, что истинное значение лежит в пределах определенного интервала?" На самом деле это не тот вопрос, на который отвечает доверительный интервал, но в конечном итоге он сводится к тому, как большинство людей интерпретирует ответ.

Вопрос о вероятности, связанной с доверительным интервалом, начинается с фразы "какова вероятность, что с учетом процедуры отбора и популяции..." Ответ на противоположный вопрос — "какова вероятность, что (что-то является истинным в отношении популяции) с учетом результата выборки", сопряжен с более сложными расчетами и более глубокими неизвестными факторами.

Процент, связанный с доверительным интервалом, называется *уровнем доверия*. Чем выше уровень доверия, тем шире интервал. Кроме того, чем меньше выборка, тем шире интервал (т. е. больше неопределенности). Оба свойства достаточно логичны: чем больше вы хотите быть уверенным, и чем меньше данных у вас есть, тем шире следует сделать доверительный интервал, чтобы быть достаточно уверенным в получении истинного значения.

Для аналитика данных доверительный интервал является инструментом для получения представления о том, насколько переменным может быть результат выборки. Аналитики данных используют эту информацию не для публикации академической работы или представления результата контролирующему органу (что обычно и делает научный исследователь), а, скорее всего, чтобы сообщить о потенциальной ошибке в оценке и, возможно, узнать, необходима ли более крупная выборка.

Ключевые идеи для доверительных интервалов

- Доверительные интервалы — это типичный способ представить оценки в виде интервального диапазона.
- Чем больше имеется данных, тем менее переменной будет оценка выборки.
- Чем ниже уровень доверия, который можно допустить, тем у́же будет доверительный интервал.
- Бутстрап — это эффективный способ создания доверительных интервалов.

Дополнительные материалы для чтения

- В изданиях "Вводная статистика и аналитика: под углом повторного отбора" (Bruce P. Introductory statistics and analytics: a resampling perspective. — John Wiley & Sons, 2014) и "Статистика" (Lock R. et al. Statistics. — Wiley, 2012) описывается подход к доверительным интервалам на основе бутстрапа.
- Инженеры, которым необходимо знать прецизионность своих данных измерений, используют доверительные интервалы, возможно, больше, чем в других областях, и как раз книга "Современная инженерная статистика" (Ryan T. Modern Engineering Statistics. — Wiley, 2007) содержит материал о доверительных интервалах. В этой книге дается обзор инструмента, который также полезен, но привлекает меньше внимания: предсказательные интервалы (интервалы вокруг одиночного значения, в противоположность среднему значению или другой сводной статистике).

Нормальное распределение

В традиционной статистике колоколообразное нормальное распределение является каноническим[1]. Тот факт, что распределения выборочных статистик часто имеют нормальную форму, сделал его мощным инструментом в разработке математических формул, которые аппроксимируют эти распределения.

[1] Кривая нормального распределения является канонической, но, возможно, она переоценена. Джордж У. Кобб (George W. Cobb), статистик из Mount Holyoke, известный своим вкладом в концепцию обучения вводного курса статистики, в редакционной статье в журнале "Американский статистик" (American Statistician) за ноябрь 2015 г. утверждает, что "стандартный вводный курс, который отводит нормальному распределению центральное место, пережил полезность своей центральности".

Ключевые термины

Ошибка (error)

Разница между точкой данных и предсказанным либо средним значением.

Стандартизировать (standardize)

Вычесть среднее значение и разделить на стандартное отклонение.

***z*-Оценка (*z*-score)**

Результат стандартизации отдельной точки данных.

Синоним: стандартная оценка.

Стандартное нормальное распределение (standard normal)

Нормальное распределение со средним, равным 0, и стандартным отклонением, равным 1.

Квантиль-квантильный график (*QQ*-plot)

График, позволяющий визуализировать, насколько близким является выборочное распределение к нормальному распределению.

Синонимы: *QQ*-график, график квантиль-квантиль.

В нормальном распределении (рис. 2.10) 68% данных находятся в пределах одного стандартного отклонения от среднего и 95% — в двух стандартных отклонениях.

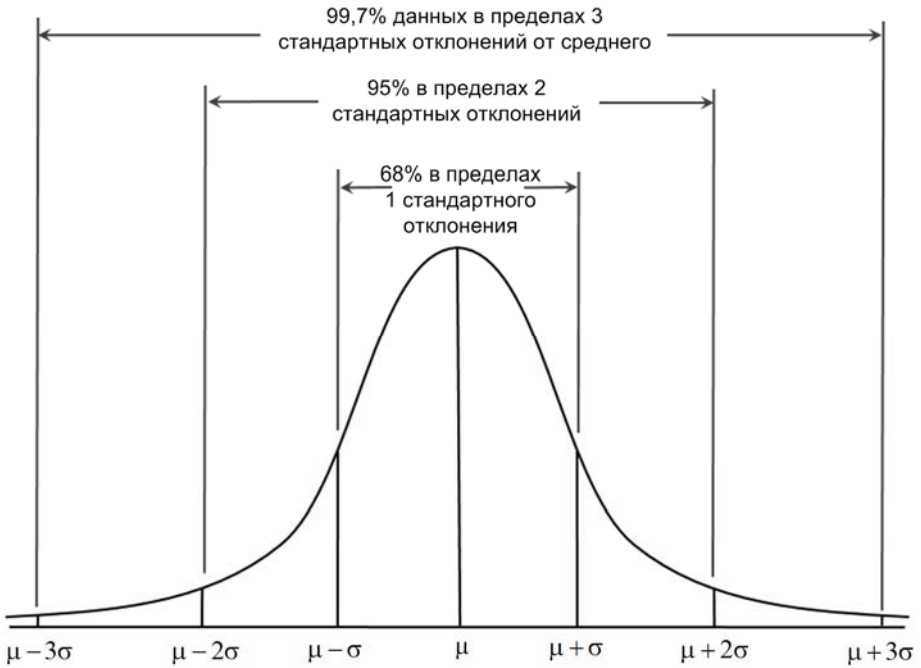


Рис. 2.10. Нормальная кривая

Существует общепринятое заблуждение, что нормальное распределение так вызывается потому, что большинство данных подчиняется нормальному распределению, т. е. что они нормальные. Большинство переменных, используемых в типичном проекте науки о данных, — по сути дела большинство сырых данных в совокупности — *не являются* нормально распределенными *(см. разд. "Длиннохвостые распределения" далее в этой главе)*. Полезность нормального распределения вытекает из того факта, что большое число статистик обычно *действительно* нормально распределено в своих выборочных распределениях. Несмотря на это, предположения о нормальности являются, как правило, крайней мерой, используемой, когда эмпирические вероятностные распределения или бутстраповские распределения отсутствуют.

Нормальное распределение также называется *гауссовым* распределением в честь Карла Фридриха Гаусса (Carl Friedrich Gauss), потрясающего немецкого математика конца XVIII — начала XIX в. Для нормального распределения есть еще одно название, которое использовалось ранее — распределение "ошибок". С точки зрения статистики, *ошибка* — это разница между фактическим значением и статистической оценкой, к примеру, средним по выборке. Например, стандартное отклонение *(см. разд. "Оценки вариативности" главы 1)* основывается на ошибках, или погрешностях, от среднего значения данных. Гаусс разработал нормальное распределение в результате своего исследования ошибок астрономических измерений, которые, как было обнаружено, нормально распределены.

Стандартное нормальное распределение и квантиль-квантильные графики

Стандартное нормальное распределение — это такое распределение, в котором единицы на оси *x* выражены в стандартных отклонениях от среднего. Для того чтобы сравнить данные со стандартным нормальным распределением, нужно вычесть среднее и далее разделить на стандартное отклонение; эта процедура также называется *нормализацией* или *стандартизацией (см. "Стандартизация (нормализация, z-оценки)" главы 6)*. Отметим, что "стандартизация" в данном смысле не связана со стандартизацией записей базы данных (т. е. приведением к общему формату). Преобразованное значение называется *z-оценкой*, или *стандартной оценкой*, а нормальное распределение иногда называют *z-распределением*.

Квантиль-квантильный график используется, чтобы визуально определить, насколько близко выборка находится от нормального распределения. Квантиль-квантильный график упорядочивает *z*-оценки снизу вверх и графически отображает *z*-оценки каждого значения на оси *y*; ось *x* — это соответствующий квантиль нормального распределения для ранга конкретного значения. Поскольку данные нормализованы, единицы соответствуют числу стандартных отклонений данных от среднего. Если точки примерно ложатся на диагональную линию, то распределение выборки можно рассматривать близким к нормальному. На рис. 2.11 показан квантиль-квантильный график для выборки из 100 значений, в произвольном порядке сгенерированных из нормального распределения; как и ожидалось, точки следуют близко к линии. В R это изображение можно получить при помощи функции `qqnorm`:

```
norm_samp <- rnorm(100)
qqnorm(norm_samp)
abline(a=0, b=1, col='grey')
```

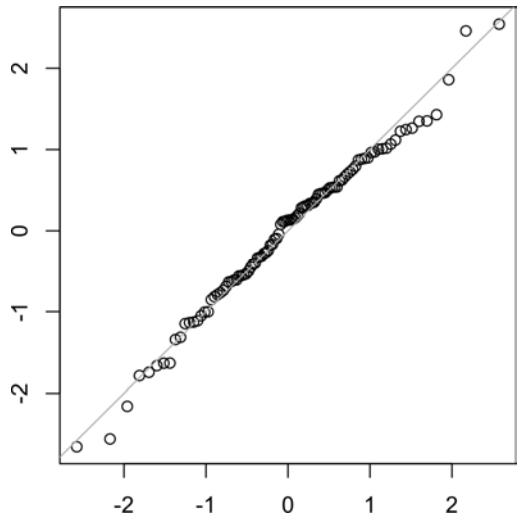


Рис. 2.11. Квантиль-квантильный график выборки из 100 значений, извлеченных из нормального распределения

Преобразование данных в *z*-оценки (т. е. стандартизация или нормализация данных) *не* делают данные нормально распределенными. Эта процедура, часто в целях сравнения, просто помещает данные на ту же шкалу измерения, что и стандартное нормальное распределение.

Ключевые идеи для нормального распределения

- Нормальное распределение имело решающее значение в историческом развитии статистики, поскольку оно позволило развить математическое приближение неопределенности и вариативности.
- Хотя сырые данные, как правило, не являются нормально распределенными, ошибки часто таковыми являются, так же как и средние и общие количества в крупных выборках.
- Для того чтобы преобразовать данные в *z*-оценки, нужно вычесть из данных среднее и разделить на стандартное отклонение; затем данные можно сравнить с нормальным распределением.

Длиннохвостые распределения

Несмотря на важность нормального распределения, в статистике в историческом плане, и в отличие от того, о чем говорит его название, данные обычно нормально распределенными не являются.

Ключевые термины

Хвост (tail)

Длинная узкая часть частотного распределения, где относительно предельные значения встречаются с низкой частотой.

Асимметрия (skew)

Состояние, когда один хвост распределения длиннее другого.

Синоним: скошенность.

В то время как нормальное распределение зачастую является обоснованным и полезным относительно распределения ошибок и выборочных статистик, оно обычно не характеризует распределение сырых данных. Иногда распределение сильно *скошено* (асимметрично), как данные о доходах, или же распределение может быть дискретным, как биномиальные данные. И симметричные, и асимметричные распределения могут иметь *длинные хвосты*. Хвосты распределения соответствуют (малым и большим) предельным значениям. Длинные хвосты и их предотвращение широко признаны на практике. Нассим Талеб (Nassim Taleb) предложил теорию *черного лебедя*, которая предсказывает, что аномальные события, такие как обвал фондового рынка, будут возникать с намного большей вероятностью, чем их предсказание нормальным распределением.

Хорошим примером, который иллюстрирует длиннохвостую природу данных, является доходность акций. На рис. 2.12 показан квантиль-квантильный график ежедневной доходности акций Netflix (NFLX). В R он генерируется следующим образом:

```
nflx <- sp500_px[,'NFLX']
nflx <- diff(log(nflx[nflx>0]))
qqnorm(nflx)
abline(a=0, b=1, col='grey')
```

В отличие от рис. 2.11, точки расположены намного ниже линии для низких значений и намного выше линии для высоких значений. Это означает, что мы наблюдаем предельные значения с намного большей вероятностью, чем можно ожидать, если бы данные имели нормальное распределение. На рис. 2.12 отражен еще один распространенный феномен: точки лежат близко к линии для данных в пределах одного стандартного отклонения от среднего. Тьюки именует это явление как данные, которые "нормальные в середине", но они имеют более длинные хвосты (см. [Tukey-1987]).

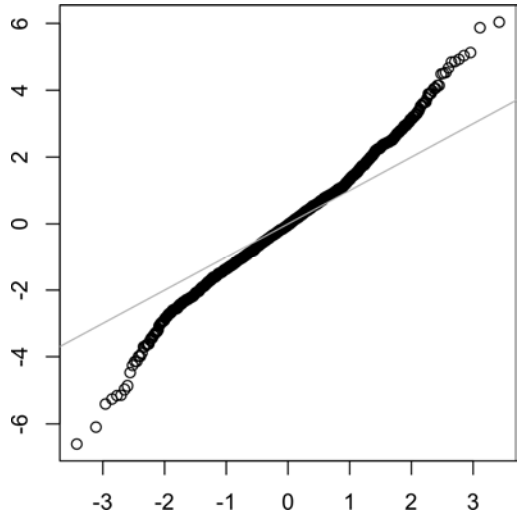


Рис. 2.12. Квантиль-квантильный график доходности NFLX

Существует много литературных источников по статистике, которые посвящены задаче подгонки статистических распределений к наблюдаемым данным. Остерегайтесь чрезмерно информационно-центричного подхода к этой задаче, который в такой же мере является искусством, как и наукой. Данные переменчивы, и часто последовательны, на первый взгляд, характеризуясь более чем одной формой и типом распределения. Как правило, в такой ситуации должно быть пущено в ход предметное и статистическое знание для определения, какое распределение является обоснованным, чтобы смоделировать данную ситуацию. Например, мы можем располагать данными об уровне интернет-трафика на сервере за большое количество 5-секундных периодов подряд. Полезно знать, что наилучшее распределение для моделирования "событий в расчете на период времени" будет пуассоновским *(см. разд. "Распределения Пуассона" далее в этой главе)*.

Ключевые идеи для длиннохвостых распределений

- Большинство данных не являются нормально распределенными.
- Принятие нормального распределения может привести к недооценке предельных событий ("черных лебедей").

Дополнительные материалы для чтения

- Талеб Н. Черный лебедь. Под знаком непредсказуемости (Taleb N. The Black Swan. — 2nd ed. — Random House, 2010).
- "Справочник по статистическим распределениям с применениями" (Krishna K. Handbook of Statistical Distributions with Applications. — 2nd ed. — CRC Press, 2016).

t-Распределение Стьюдента

Распределение Стьюдента, или *t-распределение* — это распределение нормальной формы, но немного толще и длиннее в хвостах. Оно широко используется для изображения распределений выборочных статистик. Распределения выборочных средних, как правило, имеют форму как у *t*-распределения, и существует семейство *t*-распределений, которые отличаются в зависимости от того, насколько большой является выборка. Чем больше выборка, тем более нормальную форму принимает *t*-распределение.

Ключевые термины

n

Размер выборки.

Степени свободы (degrees of freedom)

Параметр, который позволяет *t*-распределению адаптироваться к разным размерам выборок, статистикам и числу групп.

t-Распределение часто называют *t*-распределением Стьюдента, потому что оно было опубликовано в 1908 г. в журнале "Биометрика" (Biometrika) У. С. Госсетом (W. S. Gossett) под псевдонимом "Студент". Работодатели Госсета, руководство пивоваренного завода Guinness, не хотели, чтобы конкуренты знали, что он использует статистические методы, поэтому настояли, чтобы Госсет не использовал свое имя в своей статье.

Госсет хотел ответить на вопрос "Каково выборочное распределение среднего по выборке, вынутой из более крупной популяции?" Он начал работу экспериментом с повторным отбором — извлекая случайные выборки по 4 элемента из набора данных, состоящего из 3000 замеров роста и длины лево-среднего пальца преступников. (Это была эра евгеники, и в центре внимания были данные о преступниках и обнаружение корреляций между склонностями к преступлениям и физическими или психологическими особенностями.) Он нанес стандартизированные результаты (*z*-оценки) на ось *x* и частоты — на ось *y*. Параллельно с этим он получил функцию, ныне известную как функция *t* Стьюдента, и выполнил подгонку этой функции на результатах выборок, графически отобразив сравнение (рис. 2.13).

Пусть целый $\bar{x}$ — это выборочное стандартное отклонение. Тогда 90-процентный доверительный интервал вокруг выборочного среднего задается следующей формулой:

$$\bar{x} \pm t_{n-1}(0,05) \cdot \frac{s}{t},$$

где $t_{n-1}(0,05)$ — это значение *t*-статистики с $(n-1)$ степенями свободы *(см. разд. "Степени свободы" главы 3)*, которое "отсекает" 5% *t*-распределения с обоих концов. *t*-Распределение используется в качестве эталона для распределения выбо-

рочного среднего, разницы между двумя выборочными средними, параметрами регрессии и другими статистиками.

Если бы вычислительные мощности были широко доступны в 1908 г., то вне всякого сомнения статистика с самого начала намного в большей степени опиралась бы на вычислительно емкие методы повторного отбора. Лишенные компьютеров, специалисты в области статистики обратились к математике и функциям, таким как *t*-распределение, чтобы аппроксимировать выборочное распределение. В 1980-х гг. вычислительные мощности компьютеров позволили активизировать практические эксперименты с повторным отбором, но к тому времени использование *t*-распределения и подобных распределений уже глубоко укоренилось в учебниках и программных системах.

Рис. 2.13. Результаты эксперимента Госсета с повторным отбором и подогнанная *t*-кривая (из его работы в журнале "Биометрика", 1908 г.)

Точность *t*-распределения в описании поведения выборочной статистики требует, чтобы распределение этой статистики для этой выборки имело форму как у нормального распределения. Оказывается, что выборочные статистики зачастую *действительно* нормально распределены, даже когда данные базовой популяции такими не являются (факт, который привел к широко распространенному применению *t*-распределения). Это явление называется *центральной предельной теоремой (см. разд. "Центральная предельная теорема" ранее в этой главе)*.

Центральная предельная теорема и *t*-распределение в науке о данных

Что аналитики данных должны знать о *t*-распределении и центральной предельной теореме? Не очень много. Эти распределения используются в классическом статистическом выводе, но не имеют столь важное значение для целей науки о данных. Для аналитиков данных важны понимание и квантификация (т. е. количественная оценка) неопределенности и вариации, но эмпирическое взятие бутстраповских выборок может ответить на большинство вопросов об ошибке в выборке. Однако аналитики данных будут регулярно сталкиваться с *t*-статистикой в данных на выходе из статистических программных систем и статистических процедур в R, например, при *A/B*-тестировании и регрессиях, поэтому знакомство с его назначением будет очень полезным.

Ключевые идеи для распределения Стьюдента

- *t*-Распределение — это на самом деле семейство распределений, напоминающих нормальное распределение, но с более толстыми хвостами.
- Оно широко используется в качестве эталонного базиса для распределения выборочных средних, разниц между двумя выборочными средними, параметров регрессии и т. д.

Дополнительные материалы для чтения

- Исходная работа Госсета в журнале "Биометрика" за 1908 г. доступна в формате PDF (**http://seismo.berkeley.edu/~kirchner/eps_120/Odds_n_ends/Students_original_paper.pdf**).
- Стандартную трактовку *t*-распределения можно найти в онлайновом ресурсе Дэвида Лэйна (**http://onlinestatbook.com/2/estimation/t_distribution.html**).

Биномиальное распределение

Ключевые термины

Испытание (trial)

Событие с дискретным результатом (например, подбрасывание монеты).

Успех (success)

Целевой результат испытания.

Синоним: "1" (в противоположность "0").

Биномиальный (binomial)

Имеющий два результата.

Синонимы: да/нет, 0/1, двоичный, бинарный, двучленный.

Биномиальное испытание (binomial trial)

Испытание с двумя результатами.

Синоним: бернуллиево испытание.

Биномиальное распределение (binomial distribution)

Распределение набора успехов в X испытаниях.

Синоним: бернуллиево распределение.

Биномиальные результаты, т. е. в формате да/нет, лежат в основе аналитики, поскольку они часто являются кульминацией решения или другого процесса; купить/не купить, нажать/не нажать, выжить/погибнуть и т. д. Центральное значе-

ние для понимания биномиального распределения представляет идея множества испытаний, где каждое испытание имеет два возможных исхода с определенными вероятностями.

Например, 10-кратное подбрасывание монеты является биномиальным экспериментом с 10 испытаниями, где каждое испытание имеет два возможных исхода (орел или решка); см. рис. 2.14. Такие исходы в формате да/нет или 0/1 называются *бинарными*, и они не обязательно имеют вероятности 50/50. Возможны любые вероятности, которые в сумме составляют 1,0. В статистике принято называть исход "1" *успехом*; общепринятой практикой также является присвоение "1" более редкому исходу. Использование термина "успех" не говорит о том, что исход желателен или выгоден; на самом деле он скорее говорит о целевом исходе. Например, невозвраты ссуд или мошеннические транзакции являются относительно редкими событиями, в предсказании которых мы можем быть заинтересованы, поэтому им дают название "1" или "успех".

Рис. 2.14. Сторона с решкой бизоньего пятицентовика

Биномиальное распределение — это частотное распределение числа успехов (x) в заданном числе испытаний (n) с указанной вероятностью (p) успеха в каждом испытании. Имеется семейство биномиальных распределений, которые подразделяются в зависимости от значений x, n и p. Биномиальное распределение отвечает на такой вопрос:

> Если вероятность нажатия, которое конвертируется в продажу, составляет 0,02, какова вероятность наблюдать 0 продаж при 200 нажатиях?

Функция R `dbinom` вычисляет биномиальные вероятности. Например:

```
dbinom(x=2, n=5, p=0.1)
```

вернет 0,0729 — вероятность наблюдать ровно $x = 2$ успехов при $n = 5$ испытаниях, где вероятность успеха для каждого испытания равна $p = 0{,}1$.

Часто мы заинтересованы в определении вероятности x или меньшего количества успехов при n испытаниях. В этом случае мы используем функцию `pbinom`:

```
pbinom(2, 5, 0.1)
```

Она вернет 0,9914 — вероятность наблюдать два или меньшее число успехов в пяти испытаниях, где вероятность успеха для каждого испытания равна 0,1.

Среднее биномиального распределения равно $n \times p$; его можно также представить, как ожидаемое число успехов при n испытаниях для вероятности успеха, равного p.

Дисперсия равна $n \times p(1-p)$. При достаточно большом числе испытаний (в особенности, когда p близко к 0,50) биномиальное распределение фактически неотличимо от нормального распределения. На самом деле, вычисление биномиальных вероятностей с выборками больших размеров требует больших вычислительных ресурсов, и в большинстве статистических процедур используется нормальное распределение, где среднее и дисперсия — это приближения.

Ключевые идеи для биномиального распределения

- Биномиальные исходы важны для моделирования, поскольку они представляют, среди всего прочего, фундаментальные решения (купить или не купить, нажать или не нажать, выжить или погибнуть и т. д.).
- Биномиальное испытание — это эксперимент с двумя возможными исходами: один с вероятностью p и другой с вероятностью $1-p$.
- При большом n и при условии, что вероятность p не слишком близка к 0 или 1, биномиальное распределение может быть аппроксимировано нормальным распределением.

Дополнительные материалы для чтения

- Почитайте о quincunx — симуляционном устройстве подобном игре в пинбол, предназначенном для демонстрации биномиального распределения (**https://www.mathsisfun.com/data/binomial-distribution.html**).
- Биномиальное распределение составляет важнейшую часть вводного курса статистики, и все введения в статистику обязательно будут содержать одну-две главы по этой теме.

Распределение Пуассона и другие с ним связанные распределения

Многие процессы порождают события в произвольном порядке при заданной общей интенсивности — посетители, прибывающие на веб-сайт, автомобили, прибывающие в пункт сбора дорожной пошлины (события, распространяющиеся во времени), изъяны в квадратном метре ткани или опечатки в расчете на 100 строк программного кода (события, распространяющиеся в пространстве).

Ключевые термины

Лямбда (lambda)

Интенсивность (в расчете на единицу времени или пространства), с которой события происходят.

Распределение Пуассона (Poisson distribution)

Частотное распределение числа событий в отобранных единицах времени или пространства.

Экспоненциальное распределение (exponential distribution)

Частотное распределение времени или расстояния от одного события до следующего события.

Распределение Вейбулла (Weibull distribution)

Обобщенная версия экспоненциального распределения, в котором допускается смещение интенсивности события во времени.

Распределения Пуассона

Исходя из предшествующих данных, мы можем оценить среднестатистическое число событий в расчете на единицу времени или пространства, но мы также, возможно, захотим узнать, насколько оно может отличаться от одной единицы времени/пространства к другой. Распределение Пуассона говорит нам о распределении событий в расчете на единицу времени или пространства, когда мы выбираем много таких единиц. Оно полезно, когда отвечают на вопросы о массовом обслуживании, к примеру, такой: "Какие мощности нам потребуются, чтобы на 95% быть уверенными в полной обработке интернет-трафика, который прибывает на сервер в любой 5-секундный период?"

Ключевым параметром в распределении Пуассона является λ, или лямбда. Это среднее число событий, которое происходит в указанный интервал времени или пространства. Дисперсия пуассоновского распределения тоже равна λ.

Общепринятый прием состоит в генерировании случайных чисел из распределения Пуассона в качестве составной части симуляции массового обслуживания. Функция `rpois` в R делает это, принимая всего два аргумента — количество искомых случайных чисел и лямбда:

```
rpois(100, lambda=2)
```

Этот фрагмент кода генерирует 100 случайных чисел из распределения Пуассона, где $\lambda = 2$. Например, если среднее число входящих звонков в службу поддержки клиентов равно 2 в минуту, то этот фрагмент кода сымитирует 100 минут, возвращая число вызовов в каждую из этих 100 минут.

Экспоненциальное распределение

Используя тот же параметр λ, который мы использовали в распределении Пуассона, мы также можем смоделировать распределение времени между событиями: время между посещениями веб-сайта или между прибытиями автомобилей в пункт сбора дорожной пошлины. Оно также используется в техническом проектировании для моделирования времени безотказной работы, и в управлении процессами для моделирования, например, времени, требующегося в расчете на сервисный вызов. Фрагмент кода на R для генерации случайных чисел из экспоненциального распределения принимает два аргумента: `n` (количество чисел, которые будут сгенерированы) и `rate` (число событий в расчете на период времени). Например:

```
rexp(n = 100, rate = .2)
```

Этот фрагмент кода сгенерирует 100 случайных чисел из экспоненциального распределения, где среднее число событий в расчете на период времени равняется 2. Таким образом, его можно использовать для моделирования 100 интервалов в минутах между сервисными вызовами, где средняя интенсивность входящих вызовов равна 0,2 в минуту.

Ключевое предположение в любом исследовании симуляций, как для пуассоновского, так и для экспоненциального распределений, состоит в том, что интенсивность λ остается постоянной в течение рассматриваемого периода. Это редко бывает обоснованным в глобальном смысле; например, движение на дорогах или трафик в сетях передачи данных варьирует по времени суток и по дню недели. Однако периоды времени либо области пространства обычно могут быть поделены на сегменты, которые достаточно гомогенны, в результате чего допустимы анализ или симуляция в течение этих периодов.

Оценка интенсивности отказов

Во многих приложениях интенсивность события λ известна или может быть оценена из предшествующих данных. Однако в редких случаях это не обязательно так. Отказ авиационного двигателя, например, случается достаточно редко (к счастью), так что для данного типа двигателя может иметься мало данных, на которых можно было бы базировать оценку времени между отказами. При полном отсутствии данных почти нет никакой базы, на которой можно оценивать интенсивность события. Однако можно высказать какие-то предположения: если никакие события не были замечены по прошествии 20 ч, то можно быть вполне уверенным, что интенсивность не равна 1 в расчете на час. Посредством симуляции, либо прямого вычисления вероятностей, можно определить разные гипотетические интенсивности события и оценить пороговые значения, ниже которых интенсивность вряд ли когда-либо упадет. Если данных немного, но которых недостаточно, чтобы обеспечить точную, надежную оценку интенсивности, то к различным интенсивностям может быть применена проверка оптимальности подгонки *(см. разд. "Проверка на основе статистики хи-квадрат" главы 3)*, чтобы определить, насколько хорошо они соответствуют наблюдаемым данным.

Распределение Вейбулла

Во многих случаях интенсивность события не остается постоянной во времени. Если период, за который она изменяется, намного длиннее, чем типичный интервал между событиями, то никаких проблем; вы просто подразделяете анализ на сегменты, где интенсивности являются относительно постоянными, как упомянуто ранее. Если же интенсивность события изменяется внутри временного интервала, то экспоненциальное либо пуассоновское распределения бесполезны. Это, скорее всего, будет касаться случаев механического отказа — риск отказа увеличивается с течением времени. *Распределение Вейбулла* является расширением экспоненциального распределения, в котором допускается изменение интенсивности события в соответствии с *параметром формы*, β. Если $\beta > 1$, то вероятность события увеличивается во времени, если $\beta < 1$, то уменьшается. Поскольку распределение Вейбулла используется вместе с анализом времени безотказной работы (наработки на отказ) вместо интенсивности события, второй параметр выражен с точки зрения характерного времени жизни (ресурсной характеристики), а не с точки зрения интенсивности событий в расчете на интервал. Здесь используется символ η, греческая буква *эта*, который также называется параметром *масштаба*, или шкалы[2].

С распределением Вейбулла задача оценивания теперь предусматривает оценивание двух параметров: β и η. Программная система используется, чтобы смоделировать данные и произвести оценку оптимально подогнанного распределения Вейбулла.

Фрагмент кода на R для генерации случайных чисел из распределения Вейбулла принимает три аргумента: `n` (количество чисел, которые будут сгенерированы), форму `shape` и масштаб `scale`. Например, следующий фрагмент кода сгенерирует 100 случайных чисел (времена жизни) из распределения Вейбулла с формой 1,5 и характерным временем жизни 5000:

```
rweibull(100,1.5,5000)
```

Ключевые идеи для распределения Пуассона и других с ним связанных распределений

- Для событий, которые происходят с постоянной интенсивностью, число событий в расчете на единицу времени или пространства может быть смоделировано как распределение Пуассона.
- В этом сценарии можно также смоделировать время или расстояние между одним событием и следующим событием как экспоненциальное распределение.
- Изменяющаяся во времени интенсивность события (например, увеличивающаяся вероятность отказа устройства) может быть смоделирована распределением Вейбулла.

[2] Параметр масштаба (scale) — это параметр вероятностного распределения, чье физическое конкретное значение связано с выбором шкалы измерения. — *Прим. пер.*

Дополнительные материалы для чтения

- "Современная инженерная статистика" (Ryan T. Modern Engineering Statistics. — Wiley, 2007) содержит главу, посвященную распределениям вероятностей, используемым в инженерных приложениях.
- Про распределение Вейбулла с точки зрения инженерного дела (главным образом с точки зрения технического проектирования) можно почитать здесь: **http://bit.ly/2qGs8de** и **http://bit.ly/2qAMBkw**.

Резюме

В эру больших данных принципы отбора случайных подвыборок по-прежнему имеют особое значение, когда необходимы точные оценки. Случайный выбор данных может уменьшить смещение и привести к набору данных более высокого качества, чем тот, который можно получить в результате простого использования легко доступных данных. Знание различных выборочных и порождающих данные распределений позволяет нам квантифицировать потенциальные ошибки в оценке, которые могут произойти из-за случайной вариации. В то же время бутстрап (отбор из наблюдаемого набора данных с возвратом) является привлекательным универсальным ("единым для всех") методом для определения возможной ошибки в оценках выборок.

Статистические эксперименты и проверка значимости

Планирование экспериментов является краеугольным камнем практической статистики с приложениями фактически во всех областях исследования. Цель такого планирования — разработка эксперимента, который подтвердит или отклонит гипотезу. Аналитики данных сталкиваются с потребностью проводить непрерывные эксперименты, в особенности относительно пользовательского интерфейса и товарного маркетинга. В этой главе представлен обзор традиционного планирования экспериментов и обсуждается несколько распространенных задач в науке о данных. В ней также будут рассмотрены некоторые часто цитируемые в статистическом выводе понятия и дано объяснение их значения и актуальности (или отсутствие таковой) для науки о данных.

Каждый раз, когда упоминается статистическая значимость, проверка на основе *t*-статистики или *p*-значения, это происходит, как правило, в контексте классического "конвейера" статистического вывода (рис. 3.1). Этот процесс начинается с гипотезы ("препарат *A* — лучше существующего стандартного препарата", "цена *A* — прибыльнее существующей цены *B*"). Эксперимент (это может быть *A/B*-тест) предназначен для проверки гипотезы, построенной таким образом, чтобы обеспечить убедительные результаты. Данные собираются и анализируются, и далее делается вывод. Термин *"вывод"* отражает намерение применить экспериментальные результаты, которые сопряжены с предельным набором данных, к более крупному процессу или популяции.

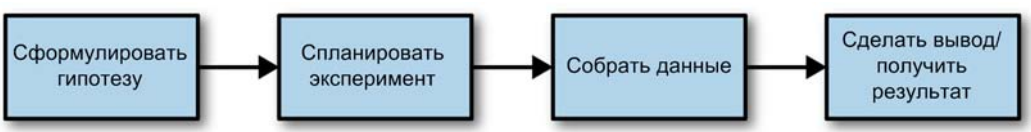


Рис. 3.1. Классический конвейер статистического вывода

A/B-тестирование

A/B-тест — это эксперимент с двумя группами для определения того, какой из двух вариантов продуктов, процедур, лечения и других сравниваемых объектов лучше. Часто один вариант из двух является стандартным существующим вариантом или отсутствует вообще. Если используется стандартный вариант (или же он отсутству-

ет), то он называется *контрольным*. Типичная гипотеза состоит в том, что предлагаемый вариант лучше контрольного.

Ключевые термины

Вариант эксперимента (treatment)

Лекарство, цена, веб-заголовок, т. е. нечто, что испытуемому предлагается в целях тестирования.

Тестовая группа (treatment group)

Группа испытуемых, которой предлагается конкретный вариант.

Контрольная группа (control group)

Группа испытуемых, которой предлагается нестандартный вариант или не предлагается никакой.

Рандомизация (randomization)

Процесс произвольного отнесения испытуемых к вариантам.

Испытуемые (subjects)

Субъекты (посетители веб-сайта, пациенты и т. д.), которым предлагаются варианты.

Проверочная статистика (test statistic)

Метрический показатель, который используется для измерения эффекта условий варианта.

A/B-тесты общеприняты в веб-дизайне и маркетинге, поскольку их результаты очень легко измеряются. Некоторые примеры *A/B*-тестирования включают:

- тестирование двух методов обработки почвы, позволяющее определить, какая из них приводит к наилучшему прорастанию саженцев;
- тестирование двух методов лечения с целью выяснить, какое из них эффективнее подавляет рак;
- тестирование двух цен, позволяющее определить, какая из них приносит больше чистой прибыли;
- тестирование двух веб-заголовков для определения, какой из них порождает больше нажатий (рис. 3.2);
- тестирование двух веб-объявлений, позволяющее выяснить, какое из них генерирует больше конверсий.

Надлежащий *A/B*-тест имеет *испытуемых*, которые могут быть отнесены к тому или иному варианту. Испытуемым может быть человек, саженец, посетитель веб-сайта; главное, что испытуемому предлагается вариант эксперимента. В идеальном случае испытуемые *рандомизируются* (назначаются в произвольном порядке) по двум предлагаемым вариантам. Таким образом, вы знаете, что любая разница между тестовыми группами наблюдается вследствие одного из двух событий:

- эффекта условий разных вариантов;
- чистой случайности, с которой испытуемые назначаются вариантам (т. е. случайное назначение, возможно, привело к тому, что более результативные испытуемые были естественным образом сконцентрированы в группе *A* или группе *B*).

Рис. 3.2. Маркетологи все время тестируют веб-презентации, сравнивая одну с другой

Необходимо также обратить внимание на *проверочную статистику* — метрический показатель, который используется для сравнения группы *A* с группой *B*. Возможно, самым общепринятым метрическим показателем в науке о данных является двоичная переменная: нажатие или отсутствие нажатия, купить или воздержаться от покупки, мошенничество или не мошенничество и т. д. Эти результаты будут обобщены в таблице размера 2×2. В табл. 3.1 приведена таблица 2×2 с результатами тестирования фактической цены.

Таблица 3.1. *Таблица 2 × 2 с результатами тестирования электронной коммерции*

Исход	Цена A	Цена B
Конверсия	200	182
Нет конверсии	23 539	22 406

Если метрический показатель представлен непрерывной величиной (сумма покупки, прибыль и т. д.) или количеством (например, дни в стационаре, посещенные

страницы), то результат может быть отображен по-разному. Если интересует не конверсия, а доход в расчете на один просмотр страницы, то результаты ценового теста в табл. 3.1 могут выглядеть как в стандартном выводе программной системы по умолчанию:

```
Доход/просмотр с ценой A: среднее = 3.87, CO = 51.10
Дохода/просмотра с ценой B: среднее = 4.11, CO = 62.98
```

"CO" обозначает стандартное отклонение значений внутри каждой группы.

Тот факт, что статистические программные системы — включая R — генерируют вывод по умолчанию, не означает, что вся эта информация полезна или релевантна. Можно увидеть, что приведенные выше стандартные отклонения не очень полезны; судя по всему, они предполагают, что многочисленные значения могут быть отрицательными в то время, как отрицательный доход невозможен. Эти данные состоят из небольшого набора относительно высоких значений (просмотры с конверсиями) и огромного числа нулевых значений (просмотры без конверсий). Очень трудно обобщить вариабельность таких данных в одном числе, хотя среднее абсолютное отклонение от среднего (7,68 для *A* и 8,15 для *B*) представляется более разумным, чем стандартное отклонение.

Зачем нужна контрольная группа?

Почему нельзя проигнорировать контрольную группу и просто выполнить эксперимент, применив целевой вариант только для одной группы и сравнив результат с предшествующим опытом?

Без контрольной группы нет никакой гарантии, что "прочие условия будут равными" и что любая разница действительно возникает в силу варианта эксперимента (либо случайным образом). Когда есть контрольная группа, она подчиняется тем же условиям (за исключением целевого варианта), что и тестовая группа. Если просто сравнивать с "базовым" или предшествующим опытом, то помимо варианта могут разниться и другие факторы.

Слепота в статистическом исследовании

Слепое исследование — это такое исследование, в котором испытуемые не осведомлены о том, что им предлагается вариант *A* или вариант *B*. Осведомленность о том или ином варианте может повлиять на отклик. В *двойном слепом исследовании* исследователи и помощники (например, врачи и медсестры в медицинском исследовании) не осведомлены о том, какие испытуемые участвуют и какой вариант им предлагается. Слепое исследование невозможно, когда природа варианта прозрачна, пример — когнитивная психотерапия с использованием компьютера, а не с помощью психолога.

Использование *A/B*-тестирования в науке о данных, как правило, находится в веб-контексте. В качестве вариантов эксперимента могут выступать дизайн веб-страницы, цена товара, формулировка заголовка объявления или какой-либо другой объект. Для того чтобы сохранить принципы рандомизации, необходимо серьезно

продумать эксперимент. Обычно испытуемым в эксперименте является посетитель веб-сайта, а измеряемыми нами целевыми исходами — нажатия, покупки, продолжительность посещения, число посещаемых страниц, просмотрена ли определенная страница и т. п. В стандартном *A/B*-эксперименте нужно выбрать один метрический показатель заранее. Могут быть собраны и представлять интерес многочисленные метрические показатели поведения, но если эксперимент ожидаемо ведет к выбору между вариантом *A* и вариантом *B*, то одиночный метрический показатель (или *проверочная статистика*) должен быть установлен заранее. Выбор проверочной статистики после того, как эксперимент проведен, создает условие для смещения вследствие предвзятости исследователя.

Почему только *A/B*? Почему не *C*, *D*?

A/B-тесты популярны в мире маркетинга и электронной коммерции, но это далеко не единственный тип статистического эксперимента. Могут быть задействованы дополнительные варианты. Испытуемые могут быть подвергнуты повторным измерительным исследованиям. Фармацевтические испытания, где субъекты дефицитны, дорогостоящи и участвуют в течение долгого времени, иногда разрабатываются с многочисленными возможностями остановки эксперимента и получения окончательного заключения.

В традиционном планировании статистического эксперимента центральное внимание уделяется ответу на статический вопрос об эффективности указанных вариантов. Аналитики данных меньше интересуются вопросом:

"Является ли разница между ценой *A* и ценой *B* статистически значимой?",

чем вопросом:

"Какая из многочисленных возможных цен является лучшей?"

Для этого используется относительно новый тип экспериментального плана: *многорукий бандит (см. разд. "Алгоритм многорукого бандита" далее в этой главе)*.

Получение разрешения

В научном и медицинском исследованиях, в которых принимают участие люди, обычно требуется их согласие на проведение эксперимента, а также одобрение институционального ревизионного совета по вопросам этики. Для экспериментов в бизнесе, которые проводятся как составная часть непрерывных операций, разрешения почти никогда не получают. В большинстве случаев (например, в ценовых экспериментах или экспериментах, связанных с выбором, какой заголовок показать либо какое предложение следует сделать) такая практика является общепринятой. Компания Facebook, однако, столкнулась с этим общепринятыми правилами в 2014 г., когда проводила эксперименты с эмоциональным тоном в лентах новостей пользователей. Компания использовала анализ мнений для классификации постов ленты новостей на положительные или отрицательные, затем она поменяла соотношение положительных/отрицательных постов, которые показывались пользователям. Несколько наугад отобранных пользователей испытывали на себе более положительные посты, тогда как другие — более отрицательные. Было обнаружено, что поль-

зователи, которые читали более положительную ленту новостей, с большей вероятностью сами отправляли положительные посты, и наоборот. Однако эффект был незначительным, и компания Facebook столкнулась с большим количеством критики за то, что эксперимент проводился без ведома пользователей. Некоторые пользователи полагали, что компания Facebook вполне могла подтолкнуть чрезвычайно подавленных пользователей к крайним поступкам, когда те получали отрицательную версию своего канала.

Ключевые идеи для *A/B*-тестирования

- Испытуемые распределяются на две (или более) групп, с которыми обращаются строго одинаково, за исключением того, что варианты отличаются один от другого.
- В идеальном случае испытуемые распределяются по группам в произвольном порядке.

Дополнительные материалы для чтения

- Двухгрупповые сравнения (*A/B*-тесты) являются элементом традиционной статистики, и почти любой вводный курс статистики будет содержать разносторонний материал по принципам проектирования экспериментов и процедурам статистического вывода. Тематический материал, в котором *A/B*-тесты помещены в контекст, более соответствующий науке о данных, с использованием повторного отбора, можно найти в издании "Вводная статистика и аналитика: под углом повторного отбора" (Bruce P. Introductory statistics and analytics: a resampling perspective. — John Wiley & Sons, 2014).
- По поводу веб-тестирования можно сказать: логистические аспекты тестирования могут быть столь же сложными, что и статистические. Хорошей отправной точкой является раздел справки, посвященный экспериментам, в Google Analytics (**http://bit.ly/2p6zPsb**).
- Будьте осторожны с советами из широко распространенных руководств по *A/B*-тестированию, которые можно увидеть в сети, такими, как следующие слова из одного такого руководства: "Дождитесь порядка 1000 посетителей и постарайтесь, чтобы тест проводился в течение одной недели". Такие эмпирические правила не являются статистически содержательными; *см. разд. "Мощность и размер выборки" далее в этой главе* для получения подробностей.

Проверка статистических гипотез

Проверки статистических гипотез, так называемые *проверки значимости*, получили широкое распространение в традиционном статистическом анализе, который встречается в публикуемых исследованиях. Цель таких проверок состоит в том, чтобы выяснить, может ли наблюдаемый эффект вызываться случайной возможностью.

Ключевые термины

Нулевая гипотеза (null hypothesis)

Гипотеза о том, что виной всему является случайность.

Альтернативная гипотеза (alternative hypothesis)

Обратная нулевой (то, что вы надеетесь доказать).

Односторонняя проверка (one-way test)

Вариант проверки гипотезы, при котором случайные результаты подсчитываются только в одном направлении.

Двусторонняя проверка (two-way test)

Вариант проверки гипотезы, при котором случайные результаты подсчитываются в двух направлениях.

A/B-тест *(см. разд. "A/B-тестирование" ранее в этой главе)*, как правило, конструируется с учетом гипотезы. Например, гипотеза может заключаться в том, что цена *B* приносит более высокую прибыль. Зачем нужна гипотеза? Почему нельзя просто взглянуть на результат эксперимента и остановиться на любом варианте, который работает лучше?

Ответ заключается в склонности человеческого разума недооценивать размах естественного случайного поведения. Одно из проявлений этой склонности состоит в неумении предвидеть предельные события или так называемых "черных лебедей" *(см. разд. "Длиннохвостые распределения" в главе 2)*. Еще одним ее проявлением является тенденция неправильно истолковывать случайные события, как имеющие признаки какой-либо значимости. Статистическая проверка гипотез была изобретена как способ защитить исследователей от того, чтобы оказаться обманутым случайной возможностью.

Неправильная интерпретация случайность

Склонность людей недооценивать случайность можно наблюдать в следующем эксперименте. Попросите нескольких своих друзей вообразить 50-кратное подбрасывание монеты: пусть они запишут серию случайных О (орел) и Р (решка). Затем попросите их, чтобы они на самом деле подбросили монету 50 раз и записали результаты. Пусть они поместят реальные результаты подбрасывания монеты в один список и выдуманные результаты — в другой. Легко различить, какие результаты являются настоящими: настоящие результаты будут иметь более длинную вереницу, состоящую из О или Р. Увидеть в наборе из 50 *настоящих* подбрасываний пять или шесть О или Р подряд не является чем-то необычным. Однако, когда большинство из нас воображает случайные подбрасывания монеты, и у нас получается три или четыре О подряд, мы говорим себе: чтобы серия выглядела случайной, нам лучше всего переключиться на Р.

Другая сторона этой монеты, если можно так выразиться, состоит в том, что когда мы *на деле* видим реальный эквивалент, состоящий из шести О подряд (например, когда один заголовок объявления превосходит по результативности другой на 10%), мы склонны приписать это чему-то реальному, не просто случайности.

В надлежащем образом разработанном *A/B*-тесте вы собираете данные о вариантах *A* и *B* таким образом, что любая наблюдаемая разница между *A* и *B* должна произойти вследствие одного из двух:

- случайной возможности в назначении испытуемых в группы;
- истинной разницы между *A* и *B*.

Статистическая проверка гипотез является дальнейшим анализом *A/B*-теста либо любого рандомизированного эксперимента с целью установить, является ли случайная возможность разумным объяснением наблюдаемой разницы между группами *A* и *B*.

Нулевая гипотеза

В проверках гипотез используется следующая логика: "С учетом склонности человека реагировать на необычное, но случайное поведение и интерпретировать его как нечто содержательное и реальное, в наших экспериментах нам потребуются доказательства, что разница между группами является более предельной, чем та, которую могла бы обоснованно породить случайность". Эта логика сопряжена с базовым предположением, что варианты эквивалентны, и любая разница между группами является случайной. Это базовое предположение называется *нулевой гипотезой*. И наша надежда тогда состоит в том, что мы сможем на деле доказать *неправильность* нулевой гипотезы и показать, что исходы для групп *A* и *B* отличаются больше, чем то, что может породить случайность.

Один из путей это сделать лежит через процедуру перетасовки на основе повторного отбора, в которой мы перемешиваем результаты групп *A* и *B* и далее неоднократно раздаем данные в группы аналогичных размеров, затем наблюдаем, как часто мы получаем такую же предельную разницу, что и наблюдаемая разница. Для получения дополнительных подробностей *см. разд. "Повторный отбор" далее в этой главе*.

Альтернативная гипотеза

Проверки гипотез по их природе сопряжены не только с нулевой гипотезой, но и с компенсирующей ее альтернативной гипотезой. Вот несколько примеров:

- нулевая гипотеза — "разницы между средними в группе *A* и группе *B* нет", альтернативная гипотеза — "*A* отличается от *B*" (может быть больше или меньше);
- нулевая гипотеза — "$A \leq B$", альтернативная — "$B > A$";

- нулевая гипотеза — "*B* не больше *A* на X%", альтернативная — "*B* больше *A* на *X*%".

Взятые вместе нулевая и альтернативная гипотезы охватывают абсолютно все имеющиеся возможности. Природа нулевой гипотезы определяет структуру проверки гипотезы.

Односторонняя и двухсторонняя проверки гипотез

Нередко в *A*/*B*-тесте вы проверяете новую возможность (скажем, *B*) против заданной по умолчанию возможности (*A*) и презюмируете, что будете придерживаться возможности по умолчанию, если только новая возможность не окажется определенно лучше. В таком случае при проверке гипотезы вам потребуется защититься от того, чтобы не обмануться случайностью в пользу *B*. Вы не заботитесь о том, чтобы быть обманутым случайностью в другом направлении, потому что останетесь на стороне *A*, если *B* не окажется определенно лучше. Поэтому вам нужна *направленная* альтернативная гипотеза (*B* лучше *A*). В таком случае вы используете *одностороннюю* (т. е. с одним хвостом) проверку гипотезы. Это означает, что предельный шанс приводит только к однонаправленному подсчету в сторону *p*-значения.

Если вы хотите, чтобы проверка гипотезы защитила вас от обмана случайностью в любом направлении, то альтернативная гипотеза является *двунаправленной* (*A* отличается от *B* и может быть больше или меньше). В таком случае вы используете *двустороннюю* (или с двумя хвостами) гипотезу. Это означает, что предельный шанс приводит к двунаправленному подсчету в сторону *p*-значения.

Проверка гипотезы с одним хвостом часто соответствует природе принятия решения в *A*/*B*-тестировании, в котором принятие решения обязательно, и одной возможности обычно присваивается состояние "по умолчанию", если другая не оказывается лучше. Однако программные системы, включая R, как правило, предоставляют на выходе двухстороннюю проверку (с двумя хвостами) по умолчанию, и многие специалисты в области статистики отдают предпочтение более консервативной двухсторонней проверке, только чтобы предотвратить полемику. Тема различий между проверками с одним хвостом или с двумя хвостами является довольно запутанной и не имеет прямого отношения к науке о данных, где точность расчетов *p*-значения не слишком важна.

Ключевые идеи для проверки статистических гипотез

- *Нулевая гипотеза* — это логическая конструкция, воплощающая утверждение, что ничего особенного не произошло, и любой эффект, который вы наблюдаете, происходит в силу случайной возможности.
- Проверка *статистической гипотезы* предполагает, что нулевая гипотеза является истинной, создает "нулевую модель" (вероятностную модель) и проверяет, является ли эффект, который вы наблюдаете, разумным результатом этой модели.

Дополнительные материалы для чтения

- "Походка алкаша" (Mlodinow L. The Drunkard's Walk. — Vintage Books, 2008) — это легко читаемый обзор ситуаций, когда "произвольность управляет нашими жизнями".
- "Статистика" (Freedman D., Pisani R., Purves R. Statistics. — 4th Edition. — W.W. Norton & Company, 2007) — классическое статистическое издание, в котором содержатся превосходные нематематические выкладки большинства статистических тем, включая проверку гипотез.
- "Вводная статистика и аналитика: под углом повторного отбора" (Bruce P. Introductory statistics and analytics: a resampling perspective. — John Wiley & Sons, 2014) развивает тему проверки гипотез с использованием повторного отбора.

Повторный отбор

Повторный отбор в статистике означает многократное извлечение выборочных значений из наблюдаемых данных с общей целью определения случайной вариабельности в статистической величине. Этот метод также может использоваться для диагностики и улучшения точности некоторых машинно-обучаемых моделей (например, предсказания в моделях на основе деревьев решений, построенных на многократно бутстрапированных данных, которые могут быть усреднены в результате процесса, известного как *бэггинг*: *см. разд. "Бэггинг и случайный лес" главы 6*).

Существует два главных типа процедур повторного отбора: *бутстрап* и *перестановка*. Бутстрап используется для определения надежности оценки; этот метод обсуждался в предыдущей главе *(см. разд. "Бутстрап" главы 2)*. Перестановочные тесты применяются для проверки гипотез и, как правило, сопряжены с двумя или более группами, и мы обсудим их в данном разделе.

Ключевые термины

Перестановочный тест (permutation test)

Процедура объединения двух или более выборок, и произвольное (или исчерпывающее) перераспределение наблюдений в повторные выборки.

Синонимы: рандомизационный тест, произвольный перестановочный тест, точный тест.

С возвратом или без возврата (with or without replacement)

Вариант отбора элементов, когда элемент возвращается или не возвращается в выборку перед следующей выемкой.

Перестановочный тест

В процедуре *перестановки* задействуются две или более выборок, как правило, группы в *A/B*-тесте или другой проверке гипотезы. *Перестановка* означает изменение порядка следования значений, или их пермутацию. Первый шаг в *перестановочной проверке* гипотезы состоит в объединении результатов из групп *A* и *B* (и групп *C*, *D*, ..., если они используются). В этом заключается логическое воплощение нулевой гипотезы — варианты, которые были предложены группам, не различаются. Затем мы тестируем эту гипотезу путем произвольной выемки групп из этого объединенного набора и смотрим, насколько они отличаются друг от друга. Перестановочная процедура следующая:

1. Объединить результаты из разных групп в один набор данных.
2. Перетасовать объединенные данные, затем в произвольном порядке вынуть (без возврата) повторную выборку того же размера, что и группа *A*.
3. Из оставшихся данных в произвольном порядке вынуть (без возврата) повторную выборку того же размера, что и группа *B*.
4. Сделать то же для групп *C*, *D* и т. д.
5. В зависимости от вычисленной для исходных выборок статистики или оценки (например, разница в долях групп) теперь рассчитать ее для повторных выборок и записать; это будет одной итерацией перестановки.
6. Повторить предыдущие шаги *R* раз для получения перестановочного распределения проверочной статистики.

Теперь вернемся к наблюдаемой разнице между группами и сравним ее с набором перестановочных разниц. Если наблюдаемая разница убедительно лежит в пределах набора перестановочных разниц, то мы ничего не доказали — наблюдаемая разница находится внутри диапазона того, что может породить случайность. Однако если наблюдаемая разница лежит вне большей части перестановочного распределения, то мы приходим к заключению, что случайность не несет ответственности. Говоря техническим языком, разница является *статистически значимой (см. разд. "Статистическая значимость и p-значения" далее в этой главе)*.

Пример: прилипчивость веб-страниц

Компания, продающая относительно дорогостоящую услугу, хочет протестировать, какая из двух веб-презентаций справляется с продажей лучше. Ввиду дороговизны продаваемой услуги, продажи случаются нечасто, и цикл продаж продолжительный; аккумулирование достаточного количества продаж с целью узнать, какая презентация превосходит, заняло бы слишком много времени. Поэтому компания решает измерить результаты при помощи эрзац-переменной, используя подробную внутреннюю страницу веб-сайта, в которой описывается услуга.

Эрзац-переменная, или *прокси-переменная* — это переменная, которая подменяет истинную целевую переменную, возможно, отсутствующую либо слишком дорогостоящую, либо чтобы измерить которую требуется слишком много времени. В климатических исследованиях, например, содержание кислорода в кернах древнего льда используется в качестве эрзаца для температуры. При этом полезно иметь, по крайней мере, *немного* данных об истинной целевой переменной, чтобы можно было определить силу их связи с эрзацем.

Одна из потенциальных эрзац-переменных для нашей компании — это число нажатий на подробной посадочной странице. Еще лучше — сколько времени люди проводят на странице. Разумно полагать, что веб-презентация (страница), которая задерживает внимание людей дольше, приведет к большему количеству продаж. Следовательно, при сравнении страницы *A* со страницей *B* нашим метрическим показателем будет среднее время сеанса.

Речь идет о внутренней странице специального назначения, поэтому она не получает огромного числа посетителей. Также стоит отметить, что служба Google Analytics (GA), при помощи которой мы измеряем время сеанса, не может измерить время сеанса последнего посещения страницы клиентом. Вместо того чтобы удалить этот сеанс из данных, GA записывает его как 0, поэтому данные требуют доработки, чтобы удалить эти сеансы. В результате получается в общей сложности 36 сеансов для двух разных презентаций: 21 сеанс для страницы *A* и 15 сеансов для страницы *B*. Используя программный пакет `ggplot`, можно визуально сравнить времена сеансов при помощи парных коробчатых диаграмм:

```
ggplot(session_times, aes(x=Page, y=Time)) +
  geom_boxplot()
```

Представленная на рис. 3.3 коробчатая диаграмма говорит о том, что страница *B* приводит к более продолжительным сеансам, чем страница *A*. Средние для каждой группы могут быть вычислены следующим образом:

```
mean_a <- mean(session_times[session_times['Page']=='Page A', 'Time'])
mean_b <- mean(session_times[session_times['Page']=='Page B', 'Time'])
mean_b - mean_a
[1] 21.4
```

Времена сеансов страницы *B* продолжительнее в среднем на 21,4 секунды в отличие от страницы *A*. Вопрос состоит в том, находится ли эта разница внутри диапазона того, что может породить случайная возможность, или же, напротив, является статистически значимой. Один из способов ответить на этот вопрос состоит в том, чтобы применить перестановочный тест — объединить все времена сеансов, затем многократно перетасовать и поделить их на группы, состоящие из 21 элемента (вспомним, что $n = 21$ для страницы *A*) и из 15 элементов ($n = 15$ для *B*).

Для того чтобы применить перестановочный тест, нам нужна функция, которая будет произвольно назначать 36 времен сеансов группе из 21 элемента (страница *A*) и группе из 15 элементов (страница *B*):

```
perm_fun <- function(x, n1, n2)
{
  n <- n1 + n2
  idx_b <- sample(1:n, n1)
  idx_a <- setdiff(1:n, idx_b)
  mean_diff <- mean(x[idx_b]) - mean(x[idx_a])
  return(mean_diff)
}
```

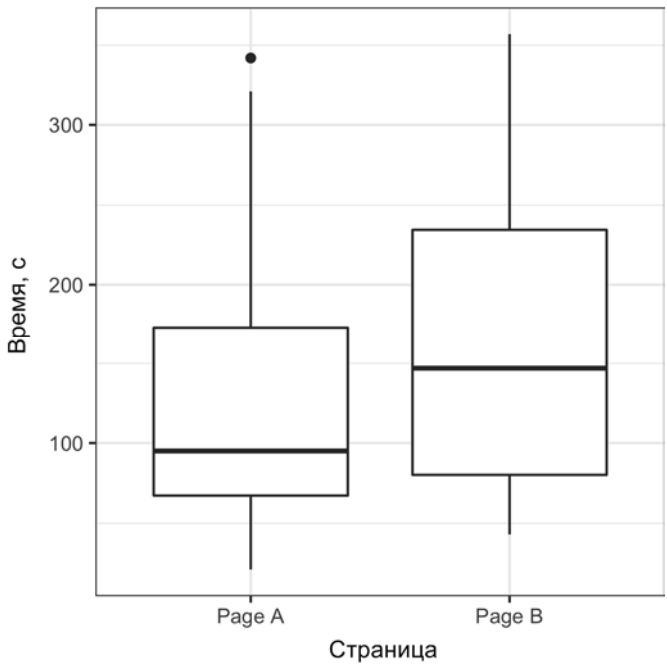


Рис. 3.3. Времена сеансов для веб-страниц *A* и *B*

Данная функция работает путем отбора индексов `n2` без возврата и отнесения их к группе *B*; оставшиеся индексы `n1` назначаются группе *A*. Функция возвращает разницу между двумя средними. Вызов этой функции в количестве $R = 1000$ раз и установка `n2=15` и `n1=21` приводит к перераспределению разниц во временах сеансов, которые могут быть отображены в виде гистограммы.

```
perm_diffs <- rep(0, 1000)
for(i in 1:1000)
  perm_diffs[i] = perm_fun(session_times[,'Time'], 21, 15)
hist(perm_diffs, xlab='Разницы во времени сессий, с')
abline(v = mean_b - mean_a)
```

Гистограмма, приведенная на рис. 3.4, показывает, что средняя разница произвольных перестановок превышает наблюдаемую разницу во временах сеансов (вертикальная линия). Это свидетельствует о том, что наблюдаемая разница во временах

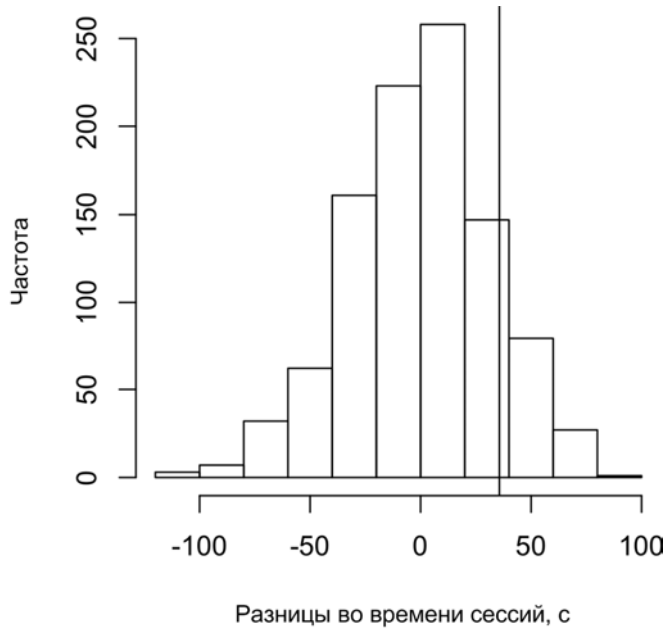


Рис. 3.4. Частотное распределение для разниц во временах сеансов между страницами *A* и *B*

сеансов между страницами *A* и *B* находится далеко внутри диапазона случайной вариации и, следовательно, не является статистически значимой.

Исчерпывающий и бутстраповский перестановочные тесты

В дополнение к предыдущей процедуре произвольной перетасовки, также именуемой *произвольным перестановочным тестом*, или *рандомизационным тестом*, существуют два варианта перестановочного теста:

- исчерпывающий перестановочный тест;
- бутстраповский перестановочный тест.

В исчерпывающем перестановочном тесте вместо простой произвольной перетасовки и разделения данных мы фактически выясняем все возможные способы, которыми они могут быть разделены. Это практически осуществимо только для относительно небольших размеров выборок. При большом количестве повторных перетасовок результаты случайного перестановочного теста аппроксимируют результаты исчерпывающего перестановочного теста и приближаются к ним в пределе. Исчерпывающие перестановочные тесты также иногда именуются *точными тестами* из-за их статистического свойства, которое гарантирует, что нулевая модель не протестируется как "значимая" больше уровня значимости α в тесте *(см. разд. "Статистическая значимость и p-значения" далее в этой главе)*.

В бутстраповском перестановочном тесте выемки, кратко описанные в шагах 2 и 3 случайного перестановочного теста, делаются *с возвратом*, а не без возврата. Тем

самым процедура повторного отбора моделирует не только случайный элемент при отнесении испытуемого к варианту, но и случайный отбор испытуемых из популяции. Обе процедуры встречаются в статистике, и разница между ними носит довольно замысловатый характер и не имеет особой важности в практике науки о данных.

Перестановочные тесты: сухой остаток для науки о данных

Перестановочные тесты являются полезными эвристическими процедурами для исследования роли случайной вариации. Они относительно легко программируются, интерпретируются и объясняются. Они также предлагают полезный обходной путь вместо формализма и "ложного детерминизма" статистики, основанной на формулах.

Одно из достоинств повторного отбора, в отличие от подходов на основе формул, состоит в том, что он почти вплотную сближается с универсальным ("единым для всех") подходом к статистическому выводу. Данные могут быть десятичными или двоичными. Размеры выборок могут быть одинаковыми или разными. Предположения о нормально распределенных данных не требуются.

Ключевые идеи для повторного отбора

- В перестановочном тесте многочисленные выборки объединяются и далее перетасовываются.
- Перетасованные значения далее делятся на изымаемые повторно выборки, и затем вычисляется целевая статистика.
- Этот процесс повторяется, и повторно опробованная статистика сводится в таблицу.
- Сравнение наблюдаемого значения статистики с повторно опробованным распределением позволяет судить, могла ли наблюдаемая разница между выборками произойти случайно.

Дополнительные материалы для чтения

- Прочитайте "Перестановочные тесты" (Randomization Tests, 4th ed., Eugene Edgington, Patrick Onghena, Chapman Hall, 2007), но не слишком лезьте в дебри неслучайного отбора.
- "Вводная статистика и аналитика: под углом повторного отбора" (Bruce P. Introductory statistics and analytics: a resampling perspective. — John Wiley & Sons, 2014).

Статистическая значимость и *p*-значения

Статистическая значимость описывает, как специалисты в области статистики измеряют статистический эксперимент (или вообще любое статистическое исследование существующих данных), отвечая на вопрос: дает ли эксперимент более предельный результат, чем тот, который может породить случайность?

Если результат лежит за рамками случайной вариации, то говорят, что он является *статистически значимым*.

Ключевые термины

***P*-значение (*P*-value)**

С учетом случайной модели, которая воплощает нулевую гипотезу, *p*-значение является вероятностью получения столь же необычных или предельных результатов, что и наблюдаемые результаты.

Альфа (α, alpha)

Вероятностный порог "необычности", который случайные результаты должны превзойти, чтобы фактические исходы считались статистически значимыми.

Синонимы: уровень значимости.

Ошибка 1-го рода (type 1 error)

Ошибочный вывод о том, что эффект — реальный (в то время, как он случайный).

Ошибка 2-го рода (type 2 error)

Ошибочный вывод о том, что эффект — случайный (в то время, как он реальный).

Рассмотрим в табл. 3.2 результаты веб-теста, показанного ранее.

Таблица 3.2. *Таблица 2 × 2 для результатов эксперимента с электронной коммерцией*

Исход	Цена *A*	Цена *B*
Конверсия	200	182
Нет конверсии	23,539	22,406

Цена *A* конвертирует (т. е. превращает посетителей в покупателей) почти на 5% лучше, чем цена *B* (0,8425% против 0,8057% — разность 0,0368 процентных пунктов); это достаточно весомый результат, чтобы компания была значимой в крупномасштабном бизнесе. Здесь имеется более 45 000 точек данных, и возникает соблазн рассматривать их как "большие данные", не требуя проверок статистической значимости (необходимой главным образом, чтобы учитывать выборочную вариабельность в небольших выборках). Однако уровень конверсии настолько низкий (менее 1%), что фактические содержательные значения — конверсии — находятся лишь в сотых долях, при этом необходимый размер выборки действительно опре-

деляется этими конверсиями. Мы можем проверить, лежит ли разность в конверсиях между ценами *A* и *B* внутри диапазона случайной вариации, воспользовавшись процедурой повторного отбора. Под "случайной вариацией" мы подразумеваем случайную вариацию, порожденную вероятностной моделью, которая воплощает нулевую гипотезу, что нет никакой разницы между уровнями конверсии *(см. разд. "Нулевая гипотеза" ранее в этой главе)*. Следующая перестановочная процедура ставит вопрос: если этим двум ценам присущ один и тот же уровень конверсии, то сможет ли случайная вариация привести к разнице на целых 5%?

1. Создать коробку со всеми выборочными результатами: она будет представлять воображаемый совместный уровень конверсии из 382 единиц и 45,945 нулей — $0,008246 = 0,8246\%$.
2. Перетасовать и вынуть повторную выборку размера 23,739 (число *n* такое же, что и у цены *A*) и записать количество единиц.
3. Записать число единиц в оставшихся 22,588 (число *n* такое же, что и у цены *B*).
4. Записать разницу в виде доли единиц.
5. Повторить шаги 2–4.

Как часто наблюдалась разница ≥ 0,0368?

Снова воспользовавшись функцией `perm_fun`, определенной в *разд. "Пример: прилипчивость веб-страниц" ранее в этой главе*, можно создать гистограмму произвольно перестановленных разниц в уровне конверсии:

```
obs_pct_diff <- 100*(200/23739 - 182/22588)
conversion <- c(rep(0, 45945), rep(1, 382))
perm_diffs <- rep(0, 1000)
for(i in 1:1000)
  perm_diffs[i] = 100*perm_fun(conversion, 23739, 22588)
hist(perm_diffs, xlab=Разницы во времени сессий, c')
abline(v = obs_pct_diff)
```

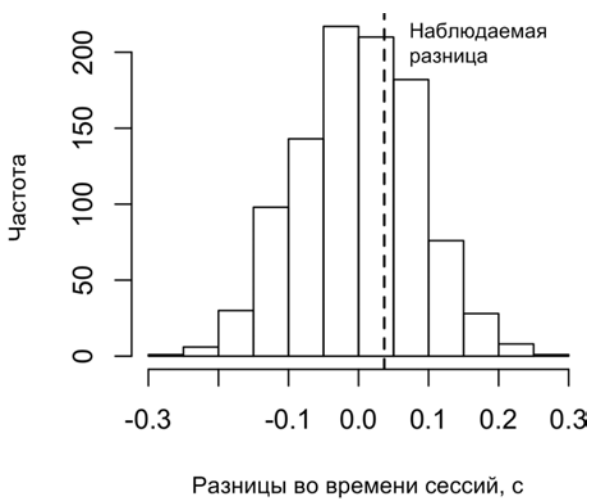


Рис. 3.5. Частотное распределение для разницы в уровнях конверсии между страницами *A* и *B*

Посмотрим на гистограмму из 1000 повторно опробованных результатов на рис. 3.5: как оказалось, в данном случае наблюдаемая разница 0,0368% лежит далеко внутри диапазона случайной вариации.

p-Значение

Простой осмотр графика — не очень точный способ измерить статистическую значимость, поэтому бóльший интерес вызывает *p*-значение. Это частота, с которой случайная модель приводит к результату, более предельному, чем наблюдаемый результат. Мы можем оценить *p*-значение из нашего перестановочного теста путем взятия доли количества раз, когда перестановочный тест порождает разницу, равную или бóльшую, чем наблюдаемое различие:

```
mean(perm_diffs > obs_pct_diff)
[1] 0.308
```

p-Значение составляет 0,308, что означает, что мы более 30% времени ожидаемо будем достигать такого же предельного результата, как и этот, или более предельного в силу случайной возможности.

В данном случае, чтобы получить *p*-значение, нам не нужно было использовать перестановочный тест. Поскольку имеется биномиальное распределение, мы можем аппроксимировать *p*-значение при помощи нормального распределения. Во фрагменте кода на R мы это делаем при помощи функции `prop.test`:

```
> prop.test(x=c(200,182), n=c(23739,22588), alternative="greater")

  2-sample test for equality of proportions with continuity correction

data: c(200, 182) out of c(23739, 22588)
X-squared = 0.14893, df = 1, p-value = 0.3498
alternative hypothesis: greater
95 percent confidence interval:
 -0.001057439 1.000000000
sample estimates:
     prop 1      prop 2
0.008424955 0.008057376
```

Аргумент `x` — это число успехов для каждой группы, аргумент `n` — число испытаний. Нормальное приближение дает *p*-значение 0,3498, близкое к *p*-значению, полученному в результате перестановочного теста.

Альфа

Специалисты в области статистики осуждают практику, когда на усмотрение исследователя оставляется решение, не является ли результат "слишком необычным", чтобы произойти случайно. Вместо этого заранее устанавливается порог, как, например, "более предельный, чем 5% случайных (нулевая гипотеза) результатов".

Этот порог называется *уровнем значимости*, или альфа-уровнем (α). Типичными α-уровнями являются 5 и 1%. При этом выбор любого из двух является произвольным решением — в данной процедуре нет ничего, что будет гарантировать правильные решения в *x*% случаях. Это вызвано тем, что получаемый ответ на вопрос о вероятности состоит *не* в том, "какова вероятность, что он произошел случайно?", а в том, "какова вероятность, что результат будет таким предельным, с учетом случайной модели?" Затем мы делаем обратный вывод о правомерности случайной модели, но такое суждение не предполагает вероятность. Этот момент всегда был источником часто возникающей путаницы.

Чему равно *p*-значение?

В последние годы использование *p*-значения было окружено активной полемикой. Один из журналов по психологии пошел настолько далеко, что "запретил" использование *p*-значений в предоставляемых ему статьях на том основании, что публикационные решения, основанные исключительно на *p*-значении, приводили к публикациям исследовательских работ плохого качества. Слишком много исследователей, лишь смутно представляющих, что в действительности представляет *p*-значение, копаются в данных и среди разных возможных гипотез, подлежащих проверке, пока не находят комбинацию, которая приводит к подходящему *p*-значению и, следовательно, к работе, подходящей для публикации.

Настоящая проблема состоит в том, что люди хотят получать от *p*-значения больший смысл, чем оно имеет. Вот то, что мы *хотели* бы, чтобы *p*-значение содержало:

> Вероятность, что результат является случайным.

Мы выражаем надежду на низкое значение и поэтому можем заключить, что что-то доказали. Именно так многие редакторы журналов интерпретируют *p*-значение. Но вот, что *p*-значение представляет *на самом деле*:

> Вероятность, что *с учетом случайной модели* могут получиться такие же предельные результаты, что и наблюдаемые.

Разница является тонкой, но реальной. Значимое *p*-значение не ведет вас далеко по дороге к "доказательству", как оно, похоже, обещает. Логический фундамент для вывода о "статистической значимости" несколько ослабевает, когда понят настоящий смысл *p*-значения.

В марте 2016 г. Американская статистическая ассоциация (American Statistical Association, ASA) после долгих внутренних дискуссий показала степень недоразумения по поводу *p*-значений, когда выпустила предостережение относительно их использования.

В заявлении ASA было подчеркнуто шесть принципов для исследователей и редакторов журналов.

1. *p*-Значения могут свидетельствовать о том, насколько несовместимы данные с заданной статистической моделью.

2. *p*-Значения не измеряют вероятность, что изучаемая гипотеза является истинной, либо вероятность, что данные были порождены исключительно в силу случайной возможности.
3. Научные выводы и бизнес или стратегические решения не должны базироваться только на том, переходит ли *p*-значение определенный порог.
4. Надлежащий вывод требует полной отчетности и прозрачности.
5. *p*-Значение, или статистическая значимость, не измеряет размер эффекта или важность результата.
6. Как таковое, *p*-значение не обеспечивает хорошую меру доказательства относительно модели или гипотезы.

Ошибки 1-го и 2-го рода

В определении статистической значимости возможны два рода ошибок:

- ошибка 1-го рода, когда вы ошибочно заключаете, что эффект является реальным, в то время как в действительности он чисто случайный;
- ошибка 2-го рода, когда вы ошибочно заключаете, что эффект не является реальным (т. е. является случайным), в то время как он в действительности реальный.

Ошибка 2-го рода в сущности является не столько ошибкой, сколько суждением, что размер выборки слишком маленький для того, чтобы вы смогли обнаружить эффект. Когда *p*-значение лежит далеко от статистической значимости (например, оно превышает 5%), мы фактически хотим сказать, что "эффект не доказан". Может оказаться, что более крупная выборка приведет к меньшему *p*-значению.

Основная функция проверок значимости (или так называемых *проверок гипотез*) — защитить от того, чтобы экспериментатор был обманут случайной возможностью; следовательно, они обычно строятся таким образом, чтобы минимизировать ошибки 1-го рода.

Наука о данных и *p*-значения

Проводимая аналитиками данных работа, как правило, не предназначена для публикации в научных журналах, поэтому дебаты о смысле *p*-значения являются несколько академичными. Для аналитика данных *p*-значение — это полезный метрический показатель в ситуациях, когда вы хотите знать, лежит ли модельный результат, который кажется интересным и полезным, в диапазоне нормальной случайной вариабельности. Как инструмент для принятия решения в эксперименте, *p*-значение следует считать не решающим показателем, а просто еще одной точкой информации, которая оказывает влияние на решение. Например, *p*-значения иногда используются в качестве промежуточных входов в некие статистические или машинно-обучаемые модели — признак может быть включен в состав или исключен из модели в зависимости от его *p*-значения.

Ключевые идеи для статистической значимости и *p*-значения

- Проверки значимости используются для определения, лежит ли наблюдаемый эффект внутри диапазона случайной вариации для модели нулевой гипотезы.
- *p*-Значение — это вероятность, которая приводит к результатам, таким же предельным, какими могут оказаться наблюдаемые результаты с учетом модели нулевой гипотезы.
- Значение α, или уровень значимости, — это порог "необычности" в случайной модели нулевой гипотезы.
- Проверка значимости остается намного более релевантной для формализованной отчетности об исследовании, чем для науки о данных (но чья важность в последнее время начала угасать даже для первой).

Дополнительные материалы для чтения

- Статья "Фишер и 5%-й уровень" (Stigler S. Fisher and the 5% Level // Chance. — 2008. — Vol. 21. — № 4. — P. 12) представляет собой короткий комментарий относительно книги "Статистические методы для научных работников" (Fisher R. Statistical Methods for Research Workers. — 1925), где делается акцент на 5%-м уровне значимости.
- *См. также разд. "Проверка статистических гипотез" ранее в этой главе* и упомянутые там дополнительные материалы для чтения.

Проверка на основе *t*-статистики

Существуют многочисленные типы проверок значимости, которые зависят от того, являются ли данные количественными или измерительными, сколько имеется выборок и что измеряется. Самой общепринятой является *проверка на основе t-статистики* (или *t*-тест), названной в честь *t*-распределения Стьюдента, первоначально разработанного У. С. Госсетом для аппроксимации распределения одновыборочного среднего *(см. разд. "t-Распределение Стьюдента" главы 2)*.

Ключевые термины

Проверочная статистика (test statistic)

Метрический показатель целевой разницы или эффекта.

Синоним: статистика критерия.

***t*-Статистика (*t*-statistic)**

Стандартизированная версия проверочной статистики.

***t*-Распределение (*t*-distribution)**

Эталонное распределение (в данном случае полученное из нулевой гипотезы), с которым может быть сопоставлена наблюдаемая *t*-статистика.

Все проверки значимости требуют, чтобы вы определили *проверочную статистику* (в отечественной литературе — статистику критерия), которая измеряет интересующий вас эффект и помогает установить, находится ли этот наблюдаемый эффект внутри диапазона нормальной случайной вариации. В проверке с повторным отбором (обсуждение перестановки *см. в разд. "Перестановочный тест" ранее в этой главе*) шкала данных не имеет значения. Вы создаете эталонное (нулевая гипотеза) распределение непосредственно из самих данных и используете проверочную статистику как есть.

В 1920–30-х гг., когда проверка статистических гипотез была в стадии разработки, задача произвольной перетасовки данных с несколькими тысячами итераций с целью выполнить проверку на основе повторного отбора образцов была невыполнима. Специалисты в области статистики обнаружили, что хорошим приближением перестановочного (перетасованного) распределения является основанная на *t*-статистике проверка, опирающаяся на *t*-распределение Госсета. Она используется для очень распространенного двухвыборочного сравнения — *A*/*B*-теста, в котором данные являются числовыми. Но для того чтобы *t*-распределение можно было использовать без привязки к какой-либо конкретной шкале данных, должна использоваться стандартизированная форма проверочной статистики.

В классическом статистическом тексте на данном этапе были бы показаны различные формулы, которые имеют в своем составе распределение Госсета и демонстрируют, каким образом выполняется стандартизация ваших данных с целью их сравнения со стандартным *t*-распределением. Эти формулы здесь не приводятся, потому что все статистические программные системы, а также R и Python, содержат команды, которые эти формулы реализуют. В R это функция `t.test`:

```
> t.test(Time ~ Page, data=session_times, alternative='less' )

        Welch Two Sample t-test

data: Time by Page
t = -1.0983, df = 27.693, p-value = 0.1408
alternative hypothesis: true difference in means is less than 0
95 percent confidence interval:
     -Inf 19.59674
sample estimates:
mean in group Page A mean in group Page B
            126.3333             162.0000
```

Альтернативная гипотеза состоит в том, что среднее время сеанса для страницы *A* меньше, чем для страницы *B*. Это достаточно близко к *p*-значению 0,124 перестановочного теста *(см. "Пример: прилипчивость веб-страниц" ранее в этой главе)*.

В режиме повторного отбора мы структурируем решение так, чтобы отразить наблюдаемые данные и подлежащую проверке гипотезу, не заботясь о том, являются ли данные десятичными или двоичными, сбалансированы ли размеры выборок или нет, имеются ли дисперсии выборок либо множество других факторов. В мире

формул многие вариации представляют самих себя, и они могут быть изумительными. Специалистам в области статистики нужно путешествовать по этому миру и изучать его карту, но аналитикам данных в этом нет необходимости — они, как правило, не занимаются выжимкой подробностей из проверок гипотез и доверительных интервалов, как это делает исследователь, который готовит научную работу к публикации.

Ключевые идеи для проверки на основе *t*-статистики

- До появления компьютеров проверки на основе повторного отбора образцов не были практичны, и специалисты в области статистики использовали стандартные эталонные распределения.
- Проверочная статистика далее может быть стандартизирована и сопоставлена с эталонным распределением.
- Одной из таких широко используемых стандартизированных статистик является *t*-статистика.

Дополнительные материалы для чтения

- Любой вводный курс статистики содержит иллюстрации *t*-статистики и ее использования; вот два хороших из них: "Статистика" (Freedman D., Pisani R., Purves R. Statistics. — 4th Edition. — W.W. Norton & Company, 2007) и "Основополагающие принципы статистики" (Moore D. S. The Basic Practice of Statistics. — Palgrave Macmillan, 2010).
- По поводу параллельной трактовки процедур проверки на основе *t*-статистики и повторного отбора образцов обратитесь к книгам "Вводная статистика и аналитика: под углом повторного отбора" (Bruce P. Introductory statistics and analytics: a resampling perspective. — John Wiley & Sons, 2014) или "Статистика" (Lock R. et al. Statistics. — Wiley, 2012).

Множественное тестирование

Как мы упомянули ранее, в статистике существует высказывание: "Если мучить данные слишком долго, то рано или поздно они дадут признательные показания". Это означает, что если смотреть на данные с очень большого числа разных точек зрения и задавать слишком много вопросов, то обязательно можно найти статистически значимый эффект.

Ключевые термины

Ошибка 1-го рода (type 1 error)

Ошибочный вывод, что эффект является статистически значимым.

Коэффициент ложных открытий (false discovery rate)

Доля совершения ошибки 1-го рода в результате множественного тестирования.

Корректировка *p*-значений (adjustment of p-values)

Уточнение значения при выполнении множественного тестирования на одинаковых данных.

Переподгонка (overfitting)

Подгонка к шуму.

Например, если имеется 20 предикторных переменных и одна результирующая переменная, и все они сгенерированы *случайным образом*, то существуют достаточно хорошие шансы, что по крайней мере один предиктор (ложным образом) окажется статистически значимым, если выполнить серию из 20 проверок значимости при α-уровне, равном 0,05. Как уже обсуждалось ранее, эта ситуация называется *ошибкой 1-го рода*. Данную вероятность можно рассчитать, сперва найдя вероятность, что все переменные пройдут проверку, *правильно* показав незначимость на уровне 0,05. Вероятность, что *одна* из переменных пройдет проверку, правильно показав незначимость, равна 0,95, поэтому вероятность, что все 20 предикторов пройдут проверку, правильно показав незначимость, будет равна $0,95 \times 0,95 \times 0,95...$ или $0,95^{20} = 0,36$ [1]. Вероятность, что по крайней мере один предиктор (ложным образом) покажет значимость, обратна этой вероятности, или 1 – (вероятность, что все будут незначимыми) = 0,64.

Этот вопрос связан с проблемой переподгонки в глубинном анализе данных, или "подгонки модели к шуму". Чем больше переменных вы добавляете или больше моделей вы выполняете, тем больше вероятность, что нечто проявится как "значимое" просто по чистой случайности.

В задачах обучения с учителем контрольный набор с отложенными данными, где модели диагностируются на данных, которые модель не видела раньше, снижает этот риск. В задачах статистического и машинного обучения, не сопряженных с помеченным контрольным набором, сохраняется риск прихода к заключениям, основанным на статистическом шуме.

[1] Правило умножения вероятностей утверждает, что вероятность одновременного наступления *n* независимых событий является произведением индивидуальных вероятностей. Например, если Вы и я каждый подбросим монету один раз, то вероятность, что Ваша и моя монеты обе повернутся орлом, равна $0,5 \cdot 0,5 = 0,25$.

В статистике существует несколько процедур, предназначенных для преодоления этой проблемы при очень определенных обстоятельствах. Например, при сравнении результатов по многочисленным контрольным группам можно задавать многочисленные вопросы. Так, для вариантов *A–C* можно спросить:

- *A* отличается от *B*?
- *B* отличается от *C*?
- *A* отличается от *C*?

Или же в клиническом испытании вы можете захотеть посмотреть на результаты терапии на нескольких этапах. В каждом случае вы задаете многочисленные вопросы, и с каждым вопросом вы увеличиваете шанс, что будете обмануты случайностью. Но этот обман можно скомпенсировать за счет корректировочных процедур в статистике, установив более строгую планку для статистической значимости, чем та, которая устанавливается для одиночной проверки гипотезы. Такие корректировочные процедуры, как правило, сопряжены с "делением α-уровня" согласно числу проверок. Данный метод приводит к меньшему α-уровню (т. е. более строгой планке для статистической значимости) для каждой проверки. Одна такая процедура, корректировка Бонферрони, просто делит α на число наблюдений *n*.

Однако проблема множественных сравнений выходит за пределы этих высоко структурированных случаев и связана с феноменом многократного "прочесывания" данных, которое породило высказывание об издевательстве над данными. Говоря иначе, если вы располагаете достаточно сложными данными и не нашли в них ничего интересного, значит, вы просто не всматривались в них долго и внимательно. Сегодня, как никогда ранее, доступно все больше и больше данных. Так, число опубликованных между 2002 и 2010 гг. журнальных статей почти удвоилось. И мы получаем массу возможностей найти что-то интересное в данных, включая вопросы множественности, а именно:

- сравнение многочисленных попарных разниц по всем группам;
- рассмотрение результатов многочисленных подгрупп ("в целом мы не нашли значимого эффекта условий варианта, но зато обнаружили эффект для незамужних женщин моложе 30");
- испытание массы статистических моделей;
- привлечение массы переменных в моделях;
- постановка большого числа разных вопросов (т. е. с получением разных возможных исходов).

Коэффициент ложных открытий

Термин *"коэффициент ложных открытий"* первоначально использовался для описания уровня, на котором данный набор проверок гипотез будет ложно идентифицировать значимый эффект. Он стал в особенности полезным с появлением геномных исследований, в которых могут проводиться массивные объемы статистических проверок в качестве составной части проекта генетического секвенирования. В этих случаях данный термин применяется к протоколу тестирования, и одиночное ложное "открытие" связано с результатом проверки гипотезы (например, между двумя выборками). Исследователи стре-

мились установить параметры процесса тестирования так, чтобы удерживать коэффициент ложных открытий на заданном уровне. Термин "коэффициент ложных открытий" также использовался в сообществе глубинного анализа данных в контексте классификации данных, в котором ложное открытие является неправильной идентификацией метки одиночной записи — в частности, неправильная идентификация нулей как единиц *(см. главу 5 и разд. "Проблема редкого класса" главы 5)*.

По ряду причин, в особенности включая общий вопрос "множественности", дальнейшее исследование не обязательно означает более хорошее исследование. Например, фармацевтическая компания Bayer в 2011 г. обнаружила, что когда она попыталась повторить 67 научных исследований, то смогла полностью повторить только 14 из них. Почти 2/3 невозможно было повторить вообще.

В любом случае корректировочные процедуры для подробно определенных и структурированных статистических проверок являются слишком специфичными и негибкими, чтобы найти широкое применение среди аналитиков данных. Для аналитиков данных в отношении множественности в сухом остатке будет следующее:

- в случае предсказательного моделирования риск получения иллюзорной модели, очевидная эффективность которой является в основном продуктом случайной возможности, уменьшается за счет перекрестной проверки *(см. разд. "Перекрестная проверка" главы 4)* и использования контрольной выборки с отложенными данными;
- что касается других процедур без помеченного контрольного набора для обследования модели, необходимо полагаться на:
  - осознание, что чем больше вы опрашиваете данные и ими манипулируете, тем больше возможность того, что в игру вступит случай;
  - эвристики, связанные с повторным отбором и симуляциями, для обеспечения случайных эталонов (бенчмарков), с которыми могут быть сопоставлены наблюдаемые результаты.

Ключевые идеи для множественного тестирования

- Множественность в исследовательской работе или проекте глубинного анализа данных (многократные сравнения, много переменных, много моделей и т. д.) увеличивает риск сделать вывод о том, что нечто является значимым просто в силу случайной возможности.
- Для ситуаций, сопряженных с множественными статистическими сравнениями (т. е. множественными проверками значимости), существуют статистические корректировочные процедуры.
- В ситуации с глубинным анализом данных использование контрольной выборки с помеченными переменными исходов способно предотвратить недостоверные результаты.

Дополнительные материалы для чтения

- По поводу краткого изложения одной из процедур корректировки (Даннета) под множественные сравнения см. онлайновые статистические материалы Дэвида Лэйна (**http://davidmlane.com/hyperstat/B112114.html**).
- Меган Голдмэн (Megan Goldman) предлагает немного более длинную трактовку процедуры корректировки Бонферрони (**http://www.stat.berkeley.edu/~mgoldman/Section0402.pdf**).
- Книга "Множественная проверка на основе повторного отбора" (Westfall P. H., Young S. S. Resampling-based multiple testing: examples and methods for p-value adjustment. — Wiley, 1993) содержит всестороннюю трактовку более гибких статистических процедур корректировки *p*-значений.
- "Глубинный анализ данных для бизнес-аналитики" (Shmueli G., Bruce P., Patel N. Data mining for business analytics. — 3rd ed. — John Wiley & Sons, 2016. — Chapter 2) содержит материал по разделению данных и использованию контрольных выборок с отложенными данными в предсказательном моделировании.

Степени свободы

В документации и настройках ко многим статистическим проверкам гипотез можно увидеть отсылки на "степени свободы". Данное понятие применяется к статистикам, вычисляемым из выборочных данных, и относится к числу значений, которые могут свободно варьироваться. Например, если известно среднее выборки из 10 значений, а также известно 9 значений, то известно и 10-е значение. И только 9 значений могут свободно варьироваться.

Ключевые термины

***n* или размер выборки (sample size)**

Число наблюдений (строк или записей) данных.

d.f.

Степени свободы.

Число степеней свободы является входом во многие статистические проверки. Например, степени свободы — это название, которое дано знаменателю $n-1$, встречавшемуся при расчетах дисперсии и стандартного отклонения. В чем важность этого понятия? Когда вы используете выборку для оценки дисперсии в отношении популяции, вы заканчиваете оценкой, которая, если использовать в знаменателе *n*, немного смещена вниз. Если же вы используете в знаменателе $n-1$, то оценка будет свободна от этого смещения.

Значительная доля традиционного курса статистики или соответствующего методического материала расходуется на различные стандартные проверки гипотез (проверки на основе *t*-статистики, *F*-статистики и т. д.). Когда выборочные статистики стандартизированы для использования в традиционных статистических формулах, степени свободы являются частью стандартизационных расчетов, призванных гарантировать, что ваши стандартизированные данные совпадают с соответствующим опорным распределением (*t*-распределением, *F*-распределением и т. д.).

Действительно ли степени свободы так важны для науки о данных? Не совсем. По крайней мере в контексте проверки значимости. С одной стороны, формальные статистические проверки используются в науке о данных довольно экономно. С другой, размер данных обычно большой настолько, что для аналитика данных редко играет реально важную роль, имеет ли, например, знаменатель n или $n-1$.

Тем не менее существует один контекст, в котором степени свободы релевантны: применение факторизованных переменных в регрессии (включая логистическую регрессию). Регрессионные алгоритмы глохнут, если присутствуют строго избыточные предикторные переменные. Это чаще всего происходит при факторизации категориальных переменных в двоичные индикаторы (фиктивные переменные). Возьмем день недели. Несмотря на то что в неделе 7 дней, при указании дня недели будет всего 6 степеней свободы. Например, раз вы знаете, что день недели не понедельник по субботу, то вы уверены, что этот день должен быть воскресеньем. Включение индикаторов пн–сб поэтому подразумевает, что включение *еще и* воскресенья станет для регрессии причиной неудачи из-за ошибки *мультиколлинеарности*.

Ключевые идеи для степеней свободы

- Число степеней свободы (*d.f.*) входит в состав вычислений с целью стандартизации проверочных статистик, в результате которой они могут быть сопоставлены с эталонными распределениями (*t*-распределением, *F*-распределением и т. д.).
- Понятие степеней свободы лежит в основе факторизации категориальных переменных в $n-1$ индикаторных или фиктивных переменных при выполнении регрессии (для предотвращения мультиколлинеарности).

Дополнительные материалы для чтения

По теме степеней свободы имеется несколько онлайновых пособий (**http://blog.minitab.com/blog/statistics-and-quality-data-analysis/what-are-degrees-of-freedom-in-statistics**).

ANOVA

Предположим, что вместо *A/B*-теста мы сравнивали многочисленные группы, скажем, *A-B-C-D*, где каждая содержит числовые данные. Статистическая процедура, которая выполняет проверку на статистически значимую разницу среди групп, называется *дисперсионным анализом*, или ANOVA (от англ. *analysis of variance*).

Ключевые термины

Попарное сравнение (pairwise comparison)

Проверка гипотезы (к примеру, о средних значениях) между двумя группами из множества групп.

Универсальный тест (omnibus test)

Одиночная проверка гипотезы об общей дисперсии среди средних значений множества групп.

Синонимы: омнибус-тест, омнибус-критерий.

Разложение дисперсии (decomposition of variance)

Выделение компонентов, которые вносят вклад в индивидуальное значение (к примеру, из общего среднего, из среднего значения для варианта и из остаточной ошибки).

***F*-статистика (*F*-statistic)**

Стандартизированная статистика, измеряющая степень, с которой разницы в групповых средних превышают то, что можно ожидать в случайной модели.

SS

"Сумма квадратов" обозначает отклонения от какого-либо среднего значения.

В табл. 3.3 показана прилипчивость (stickiness, т. е. степень удержания внимания посетителей) четырех веб-страниц, в количестве секунд, проведенных на странице. Четыре страницы в произвольном порядке отключаются, чтобы каждый посетитель веб-сайта получал одну из них вразброс. В общей сложности имеется 5 посетителей на каждую страницу, и в табл. 3.3 каждый столбец представляет независимый набор данных. Первый посетитель страницы 1 никак не связан с первым посетителем страницы 2. Отметим, что в веб-тесте такого рода мы не можем полностью реализовать классический рандомизированный план отбора, в котором каждый посетитель выбирается в произвольном порядке из некоторой многочисленной популяции. Мы должны брать посетителей по мере их прибытия. Посетители могут систематически отличаться в зависимости от времени суток, времени недели, времени года, состояния Интернета, характера используемого ими устройства и т. д. Эти факторы следует рассматривать, как потенциальное смещение, когда изучаются результаты эксперимента.

Таблица 3.3. *Прилипчивость (в секундах) для четырех веб-страниц*

	Страница 1	**Страница 2**	**Страница 3**	**Страница 4**
	164	178	175	155
	172	191	193	166
	177	182	171	164
	156	185	163	170
	195	177	176	168
Среднее	172	185	176	162
Общее среднее				173,75

Теперь возникает сложная задача (рис. 3.6). Когда мы сравнивали всего две группы, все было просто: мы всего лишь смотрели на разницу между средними каждой группы. В условиях четырех средних между группами имеется шесть возможных сравнений:

- страница 1 по сравнению со страницей 2;
- страница 1 по сравнению со страницей 3;
- страница 1 по сравнению со страницей 4;

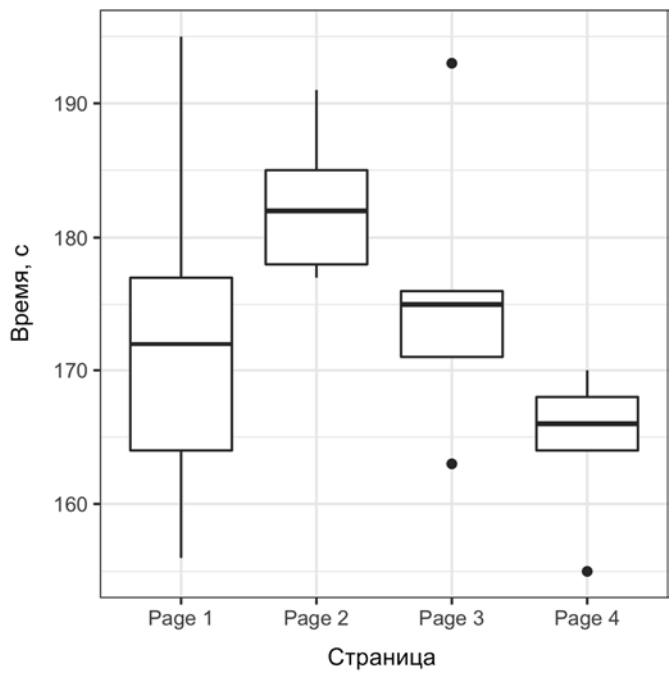


Рис. 3.6. Коробчатые диаграммы для четырех групп демонстрируют существенные расхождения среди страниц

- страница 2 по сравнению со страницей 3;
- страница 2 по сравнению со страницей 4;
- страница 3 по сравнению со страницей 4.

Чем больше мы делаем таких *попарных* сравнений, тем больше потенциал для того, чтобы оказаться обманутым случайной возможностью *(см. разд. "Множественное тестирование" ранее в этой главе)*. Вместо того чтобы беспокоиться обо всех возможных сравнениях между индивидуальными страницами, которые мы могли бы сделать, можно выполнить всего один всеобщий *универсальный* тест, который дает ответ на вопрос: "Могут ли все страницы иметь одинаковую прилипчивость, и могут ли разницы между этими страницами вызываться случайным характером распределения между ними общего набора времен сеансов?"

Для ответа на этот вопрос используется процедура ANOVA. Основополагающие принципы ее применения можно увидеть в следующей процедуре повторного отбора (заданной здесь для *A-B-C-D*-теста прилипчивости веб-страниц):

1. Объединить все данные в одной коробке.
2. Перетасовать и вынуть четыре повторных выборки с пятью значениями для каждой.
3. Записать среднее значение каждой из четырех групп.
4. Записать дисперсию среди средних значений четырех групп.
5. Повторить шаги 2–4 множество раз (скажем, 1000).

Какую долю случаев повторно опробованная дисперсия превышала наблюдаемую дисперсию? Это и есть *p*-значение.

Этот тип перестановочного теста немного сложнее, чем тест из *разд. "Перестановочный тест" ранее в этой главе*. К счастью, для этого случая в программном пакете `lmPerm` имеется функция `aovp`, которая вычисляет перестановочный тест:

```
> library(lmPerm)
> summary(aovp(Time ~ Page, data=four_sessions))
[1] "Settings: unique SS "
Component 1 :
            Df R Sum Sq R Mean Sq Iter Pr(Prob)
Page         3    831.4    277.13 3104  0.09278 .
Residuals   16   1618.4    101.15
---
Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

p-Значение, заданное в `Pr(Prob)`, равно 0,09278. Столбец `Iter` выводит число итераций, потребовавшихся в перестановочном тесте. Другие столбцы соответствуют традиционной таблице ANOVA и описаны далее.

F-статистика

Аналогично проверке гипотезы на основе *t*-статистики, которая может использоваться вместо перестановочного теста для сравнения средних из двух групп, для ANOVA существует статистическая проверка на основе *F-статистики*. Данная *F*-статистика опирается на отношение дисперсии по всем групповым средним (т. е. эффекте условий варианта) к дисперсии из-за остаточной ошибки. Чем выше это отношение, тем более статистически значим результат. Если данные подчиняются нормальному распределению, то статистическая теория диктует, что статистика должна иметь определенное распределение. На этом основании есть возможность вычислить *p*-значение.

В R можно вычислить *таблицу* ANOVA, используя функцию `aov`:

```
> summary(aov(Time ~ Page, data=four_sessions))
            Df Sum Sq Mean Sq F value Pr(>F)
Page         3  831.4   277.1    2.74 0.0776 .
Residuals   16 1618.4   101.2
---
Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

`Df` — это "степени свободы", `Sum Sq` — "сумма квадратов", `Mean Sq` — "средние квадраты" (аббревиатура для среднеквадратических отклонений), `F value` — *F*-статистика. Для общего среднего, т. е. среднего всех средних значений, сумма квадратов — это отклонение общего среднего от 0, возведенное в квадрат и умноженное на 20 (по числу наблюдений). Степени свободы для общего среднего по определению равны 1. Для средних значений варианта степени свободы равны 3 (после того как определены три значения и далее определено общее среднее, среднее значение для варианта варьироваться не может). Сумма квадратов для средних значений варианта — это сумма квадратических отклонений между средними значениями варианта и общим средним. Для остатков степени свободы равны 20 (все наблюдения могут варьироваться), и *SS* — это сумма квадратических разниц между индивидуальными наблюдениями и средними значениями варианта. Средние квадраты (*MS*) — это сумма квадратов, поделенная на степени свободы. *F*-статистика равна *MS*(вариант)/*MS*(ошибка). Значение *F*, таким образом, зависит исключительно от данного отношения и может быть сопоставлено со стандартным *F*-распределением для того, чтобы можно было установить, являются ли разницы в средних значениях вариантов больше, чем ожидается в вариации, возникающей в силу случайной возможности.

Разложение дисперсии

Наблюдаемые значения в наборе данных можно рассматривать как суммы, состоящие из разных компонентов. Любое наблюдаемое значение в наборе данных можно разбить на общее среднее, эффект условий варианта и остаточную ошибку. Данная процедура называется *разложением дисперсии*.

1. Начать с общего среднего (173,75 для данных о прилипчивости веб-страниц).
2. Добавить эффект условий варианта, который может быть отрицательным (независимая переменная = веб-страница).
3. Добавить остаточную ошибку, которая может отрицательной.

Таким образом, разложение дисперсии для верхнего левого значения в таблице *A-B-C-D*-теста следующее:

1. Начать с общего среднего: 173,75.
2. Добавить эффект условий варианта (группа): $-1,75$ $(172-173,75)$.
3. Добавить остаток: -8 $(164-172)$.
4. Итого: 164.

Двухсторонняя процедура ANOVA

Только что описанный *A-B-C-D*-тест является "односторонней" процедурой ANOVA, в которой имеется один варьирующийся фактор (группа). Но может быть вовлечен второй фактор — скажем, "выходной по сравнению с будним днем", где данные собираются по каждой комбинации (группа *A*-выходной, группа *A*-будний, группа *B*-выходной и т. д.). Это и есть "двухсторонняя процедура ANOVA", и мы будем работать с ней точно так же, как и с односторонней процедурой ANOVA путем идентификации "эффекта взаимодействия". После идентификации эффекта общего среднего и эффекта условий варианта мы разделяем наблюдения, проводившиеся в выходные и будние дни для каждой группы и находим разницу между средними для этих подмножеств и средним варианта.

Можно видеть, что процедура ANOVA и двухсторонняя процедура ANOVA являются первыми шагами по дороге к полной статистической модели, такой как регрессия и логистическая регрессия, в которых могут быть смоделированы многочисленные факторы и их эффекты *(см. главу 4)*.

Ключевые идеи для процедуры ANOVA

- ANOVA — это статистическая процедура для анализа результатов эксперимента с многочисленными группами.
- Данная процедура является расширением аналогичных процедур для *A/B*-теста и используется для оценки, находится ли общая вариация среди групп внутри диапазона случайной вариации.
- Полезным результатом ANOVA является идентификация компонентов дисперсии, связанных с групповыми вариантами, эффектами взаимодействия и ошибками.

Дополнительные материалы для чтения

- "Вводная статистика и аналитика: под углом повторного отбора" (Bruce P. Introductory statistics and analytics: a resampling perspective. — John Wiley & Sons, 2014) содержит главу по ANOVA.
- "Введение в планирование и анализ экспериментов" (Cobb G. Introduction to design and analysis of experiments. — John Wiley & Sons, 2008) предлагает всестороннюю и легко читаемую трактовку заявленной в заголовке темы.

Проверка на основе статистики хи-квадрат

Веб-тестирование часто выходит за рамки *A*/*B*-тестирования и проверяет сразу несколько вариантов. Проверка на основе статистики χ-квадрат (χ^2, хи-квадрат) используется с количественными данными, чтобы выяснить, насколько хорошо они соответствуют ожидаемому распределению. В статистической практике наиболее общепринято использовать статистику χ^2 вместе с таблицами сопряженности $r \times c$, чтобы установить, имеются ли основания для нулевой гипотезы о независимости среди переменных.

Проверка на основе статистики χ^2 первоначально была разработана Карлом Пирсоном в 1900 г. Термин "хи" происходит от греческой буквы χ, которую Пирсон использовал в своей статье.

Ключевые термины

Статистика χ^2 (chi-square statistic)

Метрический показатель, который измеряет степень, с которой наблюдаемые данные отступают от ожидания.

Ожидание или ожидаемое (expectation or expected)

Поведение данных, по нашим ожиданиям, в соответствии с каким-либо предположением, как правило, нулевой гипотезой.

d.f.

Степени свободы.

$r \times c$ означает "строки на столбцы" — таблица размера 2×3 (два на три) имеет две строки и три столбца.

Проверка χ^2: подход на основе повторного отбора

Предположим, что вы проверяете три разных заголовка объявления — *A*, *B* и *C* — и выполняете проверку каждого из них на 1000 посетителей. Результаты представлены в табл. 3.4.

Таблица 3.4. *Результаты веб-тестирования трех разных заголовков*

	Заголовок *A*	**Заголовок *B***	**Заголовок *C***
Нажатия	14	8	12
Нет нажатий	986	992	988

Заголовки без всякого сомнения явно разнятся. Заголовок *A* дает почти в 2 раза больше нажатий, чем заголовок *B*. Правда, фактические числа небольшие. Процедурой повторного отбора можно проверить, отличается ли процент нажатий в еще большей степени, чем может вызвать случайность. Для такой проверки мы должны располагать "ожидаемым" распределением нажатий, и в этом случае проверка будет осуществляться, исходя из предположения нулевой гипотезы, что всем трем заголовкам присущ один и тот же процент нажатий при общем проценте нажатий, равном 34/3000. Исходя из этого предположения наша таблица сопряженности будет выглядеть так, как показано в табл. 3.5.

Таблица 3.5. *Ожидаемое значение, если все три заголовка имеют одинаковый процент нажатий (нулевая гипотеза)*

	Заголовок *A*	**Заголовок *B***	**Заголовок *C***
Нажатия	11,33	11,33	11,33
Нет нажатий	988,67	988,67	988,67

Остаток Пирсона задается следующей формулой:

$$R = \frac{\text{наблюдаемое} - \text{ожидаемое}}{\sqrt{\text{ожидаемое}}}$$

и показывает, насколько фактические количества отличаются от их ожидаемых количеств (табл. 3.6).

Таблица 3.6. *Остатки Пирсона*

	Заголовок *A*	**Заголовок *B***	**Заголовок *C***
Нажатия	0,792	–0,990	0,198
Нет нажатий	–0,085	0,106	–0,021

Статистика χ^2 задается как сумма квадратических остатков Пирсона:

$$\chi^2 = \sum_{i}^{r} \sum_{j}^{c} R^2,$$

где r и c — соответственно число строк и столбцов. Статистика χ^2 для данного примера составит 1,666. Действительно ли это больше, чем может обоснованно произойти в случайной модели?

Мы можем проверить это при помощи следующего алгоритма повторного отбора:

1. Положить в коробку 34 единицы (нажатия) и 2966 нулей (нажатия отсутствуют).
2. Перетасовать, вынуть три отдельных выборки по 1000 элементов и подсчитать нажатия в каждом.

3. Найти квадратичные разницы между перетасованными количествами и ожидаемыми количествами и просуммировать их.
4. Повторить шаги 2 и 3, скажем, 1000 раз.

Как часто повторно опробованная сумма квадратических отклонений превышает наблюдаемые? Это и есть *p*-значение.

Функция `chisq.test` может использоваться для вычисления повторно отобранной статистики χ^2. Для данных с нажатиями проверка χ^2 будет следующей:

```
> chisq.test(clicks, simulate.p.value=TRUE)

  Pearson's Chi-squared test with simulated p-value (based on 2000 replicates)

data: clicks
X-squared = 1.6659, df = NA, p-value = 0.4853
```

Проверка показывает, что этот результат мог легко быть получен в силу произвольности.

Проверка χ^2: статистическая теория

Асимптотическая статистическая теория показывает, что распределение статистики χ^2 может быть приближено *распределением* χ^2. Соответствующее стандартное распределение χ^2 определяется *степенями свободы (см. разд. "Степени свободы" ранее в этой главе)*. Для таблицы сопряженности степени свободы связаны с числом строк (*r*) и столбцов (*c*) следующим образом:

$$\text{степени свободы} = (r-1)(c-1).$$

Распределение χ^2 обычно имеет асимметричный вид, длинный хвост которого скошен вправо (см. рис. 3.7 относительно распределения с 1, 2, 5 и 10 степенями свободы). Чем дальше на распределении χ^2 находится наблюдаемая статистика, тем ниже *p*-значение.

Функция `chisq.test` может применяться для вычисления *p*-значения с использованием распределения χ^2 в качестве эталона:

```
> chisq.test(cticks, simulate.p.value=FALSE)
Pearson's Chi-squared test test

data:  clicks
X-squared = 1.6659, df = 2, p-value = 0.4348
```

Данное *p*-значение немного меньше, чем *p*-значение повторного отбора: это вызвано тем, что распределение хи-квадрат является всего лишь приближением фактического распределения статистики.

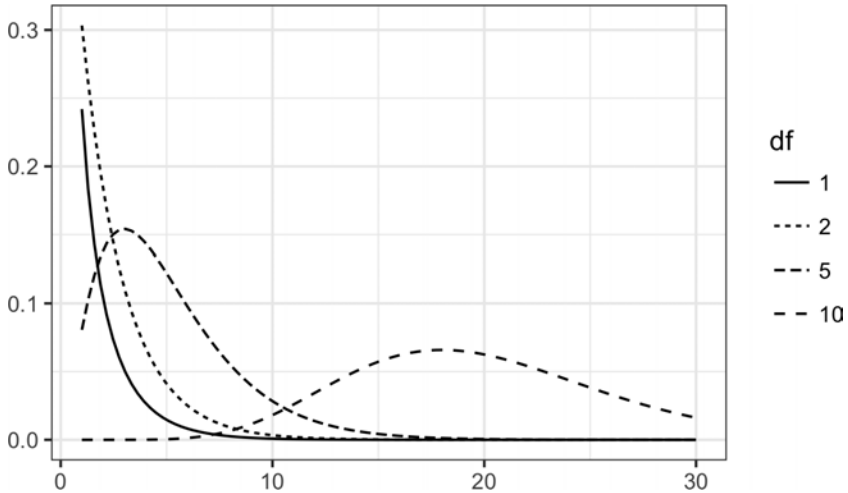


Рис. 3.7. Распределение χ^2 с разными степенями свободы (вероятность — на оси *y*, значение статистики χ^2 — на оси *x*)

Точная проверка Фишера

Распределение χ^2 является хорошим приближением только что описанной перетасованной проверки с повторным отбором, кроме тех случаев, когда количества чрезвычайно низкие (одиночные разряды, в особенности пять или меньше). В таких случаях процедура с повторным отбором будет давать более точные *p*-значения. На деле в большинстве статистических программных систем имеется процедура для фактического перечисления *всех* возможных перестановок, которые могут произойти, она сводит в таблицу их частоты и точно определяет, насколько предельным является наблюдаемый результат. Эта процедура называется *точной статистической проверкой Фишера* в честь великого статистика Р. А. Фишера. Программный код на R для точной проверки Фишера в своей канонической форме очень прост:

```
> fisher.test(cticks)
Fisher's Exact Test for Count Data

data:  clicks
p-value = 0.4824
alternative hypothesis: two.sided
```

p-Значение очень близко к *p*-значению 0,4853, полученному с использованием метода повторного отбора.

Там, где одни количества очень низкие, а другие довольно высокие (например, знаменатель в уровне конверсии), может возникнуть необходимость выполнить перетасованный перестановочный тест вместо полной точной проверки из-за трудности вычисления всех возможных перестановок. Ранее упомянутая R-функция имеет несколько аргументов, которые управляют тем, как использовать это при-

ближение (`simulate.p.value=TRUE` или `FALSE`), сколько следует использовать итераций (`B=...`), и задают вычислительное ограничение (`workspace=...`) на то, насколько далеко должны зайти расчеты для получения *точного* результата.

Обнаружение мошенничества в науке

Интересный пример представлен исследователем Университета Тафтса Терезой Иманиси-Кари (Thereza Imanishi-Kari), которая в 1991 г. обвинялась в фабрикации данных ее научной работы. В деле оказался замешанным конгрессмен Джон Динджелл (John Dingell), и этот случай в конечном счете привел к отставке ее коллеги Дэвида Балтимора (David Baltimore) от президентства в Рокфеллеровском университете.

После долгого разбирательства Иманиси-Кари была реабилитирована. Тем не менее один элемент в данном деле покоился на статистических уликах относительно ожидаемого распределения цифр в ее лабораторных данных, где каждое наблюдение имело много цифр. Следователи сосредоточились на внутренних цифрах, которые ожидаемо должны были подчиняться *равномерному* распределению *случайной величины*. Иными словами, они должны появляться в произвольном порядке, где каждая цифра имеет равную вероятность появления (ведущая цифра может иметь преимущественно одно значение, а заключительные цифры могут быть затронуты округлением). В табл. 3.7 перечислены частоты внутренних цифр из фактических данных этого нашумевшего дела.

Таблица 3.7. *Центральная цифра в лабораторных данных*

Цифра	Частота	Цифра	Частота
0	14	5	19
1	71	6	12
2	7	7	45
3	65	8	53
4	23	9	6

Распределение 315 цифр, показанное на рис. 3.8, конечно же выглядит неслучайным.

Следователи рассчитали отклонение от ожидания (31,5, т. е. как часто каждая цифра появлялась в строго равномерном распределении), и использовали проверку χ^2 (в равной степени могла быть использована процедура повторного отбора), чтобы показать, что фактическое распределение было далеко за пределами диапазона нормальной случайной вариации.

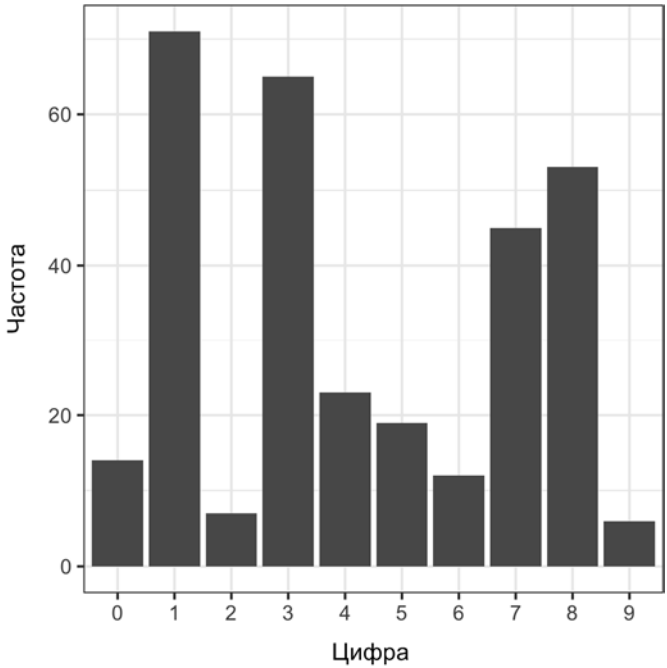


Рис. 3.8. Частотная гистограмма для лабораторных данных Иманиси-Кари

Актуальность проверок для науки о данных

Большинство стандартных применений проверки χ^2 или точной проверки Фишера не особо актуальны для науки о данных. В большинстве экспериментов, будь то *A-B* или *ABC*..., цель состоит не в том, чтобы просто установить статистическую значимость, а в том, чтобы выяснить лучший вариант. С этой целью многорукие бандиты *(см. разд. "Алгоритм многорукого бандита" далее в этой главе)* предлагают более полное решение.

Одно из применений проверки χ^2, и в особенности ее точной версии Фишера, в науке о данных заключается в установлении надлежащих размеров выборок для веб-экспериментов. Такие эксперименты часто имеют очень низкие показатели нажатий и, несмотря на тысячи показов, проценты количеств могут быть слишком малыми, чтобы получить категорические заключения в эксперименте. В таких случаях точная проверка Фишера, проверка χ^2 и другие проверки могут быть полезными как составной компонент вычислений мощности и размеров выборок *(см. разд. "Мощность и размер выборки" далее в этой главе)*.

Проверки χ^2 широко используются в сборе информации экспертами в поисках неуловимого статистически значимого *p*-значения, которое дает возможность опубликовать работу. Проверки χ^2 или подобные симуляции с повторным отбором используются в приложениях науки о данных больше как фильтр для определения того, заслуживает ли эффект или признак дальнейшего рассмотрения, чем фор-

мальная проверка значимости. Например, они используются в геопространственной статистике и картографии для определения, соответствуют ли пространственные данные указанному нулевому распределению. (Например, действительно преступления сконцентрированы в определенной области в большей степени, чем это позволяет случайная возможность?) Их также можно использовать в автоматическом отборе признаков в машинном обучении, чтобы определить распространенность класса во всех признаках и идентифицировать признаки, в которых распространенность определенного класса является необычно высокой или низкой, что не совместимо со случайной вариацией.

Ключевые идеи для проверки на основе статистики χ^2

- Общепринятая процедура в статистике состоит в проверке, соответствуют ли наблюдаемые количества данных предположению о независимости (например, склонность покупать ту или иную вещь не зависит от пола).
- Распределение χ^2 — это эталонное распределение (оно воплощает предположение о независимости), с которым должна быть сопоставлена наблюдаемая вычисленная статистика χ^2.

Дополнительные материалы для чтения

- Знаменитый пример Р. А. Фишера "Lady Tasting Tea" (процедура *"леди дегустирует чай"* обозначает рандомизированный эксперимент) с начала XX в. остается простой и эффективной иллюстрацией его точной статистической проверки. Погуглите "Lady Tasting Tea", и вы найдете много хороших рецензий.
- Сайт Stat Trek предлагает хорошее учебное пособие по статистической проверке χ^2 (**http://stattrek.com/chi-square-test/independence.aspx?Tutorial=AP**).

Алгоритм многорукого бандита

Многорукие бандиты предлагают подход к тестированию, в особенности вебтестированию, который позволяет выполнять явную оптимизацию и более быстрое принятие решения, чем традиционный статистический подход к разработке экспериментов.

Ключевые термины

Многорукий бандит (multi-arm bandit)

Воображаемый игровой автомат с несколькими пусковыми рычагами, или руками, на которые игрок может нажимать по выбору, при этом каждая рука имеет разный выигрыш; здесь термин приведен в качестве аналогии эксперимента с несколькими вариантами.

Рука (arm)

Вариант в эксперименте (например, "заголовок *A* в веб-тесте").

Выигрыш (win)

Экспериментальный аналог выигрыша в автомате (например, "клиент щелкает на ссылке").

Традиционный *A/B*-тест сопряжен с данными, которые собраны в эксперименте согласно определенному плану с целью ответа на конкретный такой вопрос, вроде "Что лучше: вариант *A* или вариант *B*?" Презюмируется, что как только мы получаем ответ на этот вопрос, эксперимент заканчивается, и мы переходим к действиям, руководствуясь полученными результатами.

Вы, вероятно, усмотрите несколько трудностей в таком подходе. Во-первых, наш ответ может быть неокончательным: "эффект не доказан". Другими словами, результаты эксперимента могут свидетельствовать об эффекте, но если есть эффект, может не оказаться достаточно большой выборки для его подтверждения (к всеобщему удовлетворению традиционных статистических стандартов). Какие шаги нам предпринять? Во-вторых, мы, возможно, захотим начать пользоваться результатами, которые поступают до окончания эксперимента. В-третьих, мы имеем право передумать либо попробовать нечто другое, основываясь на дополнительных данных, которые поступят после того, как эксперимент закончен. Традиционный подход к экспериментам и проверке гипотез восходит к 1920-м гг. и весьма негибок. Вычислительные мощности компьютеров и программное обеспечение активировали более эффективные и гибкие подходы. Кроме того, наука о данных и бизнес в целом не особо беспокоятся по поводу статистической значимости и более заинтересованы в оптимизации общих усилий и результатов.

"Бандитские" алгоритмы, которые очень популярны в веб-тестировании, позволяют сразу протестировать несколько вариантов и сделать выводы быстрее, чем традиционные статистические проекты. Они берут свое имя от автоматов, используемых в азартных играх, так называемых одноруких бандитов (поскольку они настроены таким образом, чтобы извлекать из игрока деньги непрерывным потоком). Если вообразить автомат, у которого больше одного пускового рычага, или руки, и при этом каждая возвращает выигрыши по разным ставкам, то у вас получится многорукий бандит, который дает полное название этому алгоритму.

Ваша цель состоит в том, чтобы выиграть как можно больше денег, а именно, чтобы идентифицировать выигрышный рычаг и остановиться на нем как можно скорее. Проблема состоит в том, что вы не знаете, по какой ставке они выплачивают выигрыши — вы можете узнать результаты, только если нажмете на рычаг. Предположим, что каждый "выигрыш" приносит одинаковую сумму независимо от того, на какой рычаг вы нажали. Отличается только вероятность выигрыша. Далее допустим, что сначала вы пробуете каждый рычаг по 50 раз и получаете следующие результаты:

- рука *A*: 10 выигрышей из 50;
- рука *B*: 2 выигрыша из 50;
- рука *C*: 4 выигрыша из 50.

Один из предельных подходов состоит в том, чтобы сказать: "Похоже, рычаг *A* выигрышный — надо прекратить пробовать другие и остановиться на *A*". Этот шаг в полной мере пользуется информацией начального испытания. Если *A* действительно превосходит, то мы извлекаем из этого пользу с самого начала. С другой стороны, если *B* или *C* реально лучше, то мы теряем любую возможность это обнаружить. Другой предельный подход состоит в том, чтобы сказать: "Сдается, что все это во власти случая — попробую подергать их одинаково". Это дает максимальную возможность проявиться другим альтернативам помимо *A*. Однако по ходу мы задействуем варианты, которые выглядят менее удачными. Сколько времени мы позволим этому продолжаться? Бандитские алгоритмы принимают гибридный подход: мы чаще начинаем дергать за рычаг *A*, пользуясь его очевидным превосходством, но мы не отказываемся от *B* и *C*. Мы просто обращаемся к ним не так часто. Если *A* продолжает превосходить по результативности, то мы продолжаем смещать ресурсы (нажимаем рычаги) в сторону от *B* и *C* и чаще нажимаем на *A*. Если с другой стороны *C* начинает работать лучше, а *A* — хуже, то мы можем сместиться от *A* назад к *C*. Если окажется, что один из них превосходит *A*, и это было скрыто в начальном испытании в силу случайности, то теперь он имеет возможность проявить себя при дальнейшем тестировании.

Теперь подумаем, как это применить к веб-тестированию. Вместо нескольких пусковых рычагов автомата у вас могут быть многочисленные варианты, заголовки объявлений, цвета и другие объекты и характеристики, которые тестируются на веб-сайте. Клиенты либо нажимают ("выигрыш" для коммерсанта), либо не нажимают. Первоначально, варианты предлагаются произвольно и одинаково. Если, однако, один вариант начинает превосходить по результативности другие, то он может предлагаться ("нажиматься") чаще. Но какими должны быть параметры алгоритма, который изменяет интенсивность нажатий рычага? На какую "интенсивность нажатий рычага" мы должны перейти, и когда мы должны это сделать?

Вот один из простых алгоритмов, эпсилон-жадный (ε) алгоритм для *A/B*-теста:

1. Сгенерировать случайное число между 0 и 1.
2. Если число находится между 0 и ε (при этом ε — это число между 0 и 1, обычно довольно малое), подбросьте "справедливую" монету (с вероятностью 50/50), и:
   - если монета повернется орлом, предложите вариант *A*;
   - если монета повернется решкой, предложите вариант *B*.
3. Если число больше или равно ε, предложите любой вариант, который до настоящего времени имел самую высокую интенсивность откликов.

Эпсилон (ε) — это единственный параметр, который управляет этим алгоритмом. Если $\varepsilon = 1$, то мы заканчиваем стандартным простым *A/B*-экспериментом (случайное распределение между *A* и *B* для каждого испытуемого). Если $\varepsilon = 0$, то мы за-

канчиваем чисто *жадным* алгоритмом — он не стремится продолжать экспериментировать, просто относя испытуемых (посетителей веб-сайта) к наиболее результативному варианту.

Более сложный алгоритм использует "отбор образцов по методу Томпсона". Данная процедура "вынимает выборки" (нажимает пусковой рычаг бандита) на каждом этапе для максимизации вероятности выбора наилучшего рычага. Разумеется, вы не знаете, какой рычаг лучший — в этом-то и проблема! — но по мере того как вы наблюдаете за выплатой при каждой последующей выемке, вы получаете больше информации. В отборе образцов по методу Томпсона используется байесовский подход: первоначально принимается какое-то априорное распределение вознаграждений на основе так называемого *бета-распределения* (это общепринятый механизм для указания априорной информации в байесовской задаче). По мере накопления информация после каждой выемки может обновляться, давая возможность лучше оптимизировать следующую выемку с точки зрения выбора нужного рычага.

"Бандитские" алгоритмы способны эффективно обрабатывать три варианта и более и двигаться к оптимальному выбору "лучшего". Что касается традиционных статистических процедур тестирования, то сложность процесса выбора в бандитских алгоритмах относительно трех и более вариантов далеко превосходит традиционный *A/B*-тест, и тем самым преимущество "бандитских" алгоритмов гораздо выше.

Ключевые идеи для алгоритма многорукого бандита

- Традиционные *A/B*-тесты предусматривают случайный процесс отбора, который может привести к чрезмерному показу худшего по качеству варианта.
- Многорукие бандиты, напротив, изменяют процесс отбора для включения информации, извлеченной во время эксперимента, и уменьшают частоту показа менее качественного варианта.
- Они также упрощают эффективную обработку более двух вариантов.
- Существуют разные алгоритмы, которые смещают вероятность отбора от менее качественного варианта к (предполагаемому) более качественному.

Дополнительные материалы для чтения

- Книга "Бандитские алгоритмы для веб-оптимизации" (White J. M. Bandit algorithms for website optimization: developing, deploying, and debugging. — O'Reilly, 2012) предлагает превосходное краткое изложение алгоритмов многоруких бандитов. Автор приводит примеры на Python, а также результаты имитационного моделирования для анализа результативности бандитов.
- По поводу дополнительных сведений об отборе образцов по методу Томпсона, см. работу "Анализ отбора образцов по методу Томпсона для задачи многоруко-

го бандита" (Agrawal S., Goyal N. Analysis of Thompson Sampling for the Multi-armed Bandit Problem, **http://jmlr.org/proceedings/papers/v23/agrawal12/agrawal12.pdf**).

Мощность и размер выборки

Если вы запускаете веб-тест, то каким образом решаете, сколько времени он должен выполняться (т. е. сколько необходимо показов в расчете на вариант)? Несмотря на общие советы, которые можно прочесть во многих руководствах по веб-тестированию в сети, хорошие рекомендации отсутствуют — главным образом все зависит от частоты, с которой достигается желаемая цель.

Ключевые термины

Размер эффекта (effect size)

Минимальная величина эффекта, которую вы надеетесь обнаружить в результате статистической проверки, в частности, "20-процентный рост количества нажатий".

Синоним: величина эффекта.

Мощность (power)

Вероятность обнаружить заданный размер эффекта при заданным размере выборки.

Уровень значимости (significance level)

Уровень статистической значимости, при котором будет проводиться проверка.

Синонимы: альфа, α.

Один из шагов в статистических расчетах в отношении размера выборки состоит в том, чтобы задать вопрос: "Покажет ли статистическая проверка гипотезы на самом деле разницу между вариантами *A* и *B*?" Исход проверки гипотезы — *p*-значение — зависит от того, какова реальная разница между вариантами *A* и *B*. Он также зависит от чистой случайности — того, кто отобран в группы для эксперимента. Но вполне логично предположить, что чем больше фактическая разница между вариантами *A* и *B*, тем больше вероятность, что наш эксперимент ее покажет; и чем меньше разница, тем больше данных будет необходимо для ее обнаружения. Для того чтобы отличить 0,350-х бьющих игроков в бейсболе от 0,200-х бьющих[2], потребуется не так уж много подходов к бите. А вот чтобы различить 0,300-х бьющих и 0,280-х бьющих, подходов потребуется гораздо больше.

[2] Бьющий игрок с соотношением количества подходов или попыток подходов к бите к количеству хитов, т. е. любым точным ударам по мячу, которые позволяют бьющему добежать до базы. Например, если в игре бьющий имеет четыре похода или попытки ударить и записывает два хита, то его коэффициент попаданий составит 2/4, или 0,500. — *Прим. пер.*

Мощность — это вероятность обнаружения указанного размера эффекта с указанными характеристиками выборки (размером и вариабельностью). Например, (гипотетически) можно сказать, что вероятность различения 0,330-х бьющих и 0,200-х бьющих при 25 подходах к бите составляет 0,75. Размер эффекта здесь — это разница 0,130. И "обнаружение" означает, что проверка гипотезы отклонит нулевую гипотезу "нет разницы" и придет здесь к заключению, что существует реальный эффект. Так, эксперимент с 25 подходами к бите ($n = 25$) для двух бьющих с размером эффекта 0,130 имеет (гипотетическую) мощность 0,75 или 75%.

Вы можете видеть, что здесь есть несколько подвижных элементов, и легко запутаться в многочисленных статистических предположениях и формулах, которые будут необходимы (чтобы указать выборочную вариабельность, размер эффекта, размер выборки, α-уровень для проверки гипотезы и другое и рассчитать мощность). В действительности существуют статистические программные системы специального назначения для расчета мощности. Большинству аналитиков данных не придется проходить через все формальные этапы, необходимые для того, чтобы сообщить мощность, например, в опубликованной работе. Однако они могут столкнуться со случаями, когда для *A/B*-теста им понадобится собрать немного данных, а сбор или обработка данных влечет за собой какие-то затраты. В этом случае приблизительные сведения о том, сколько данных необходимо собрать, способны помочь предотвратить ситуацию, когда вы собираете данные, затрачивая на это определенные усилия, и в итоге результат оказывается неокончательным. Вот весьма интуитивно понятный альтернативный подход:

1. Начать с каких-либо гипотетических данных, представляющих вашу наилучшую догадку о выходных данных, которые получатся в итоге (возможно, основываясь на априорных данных) — например, коробка с 20 единицами и 80 нулями, которая представляет 0,200-х бьющих, или коробка с несколькими наблюдениями о "времени, проведенном на веб-сайте".
2. Создать вторую выборку, добавив к первой выборке желаемый размер эффекта — например, вторая коробка с 33 единицами и 67 нулями или вторая коробка с 25 секундами, добавленными к каждому исходному "времени, проведенному на веб-сайте".
3. Вынуть бутстраповскую выборку размера *n* из каждой коробки.
4. Выполнить перестановочную (либо основанную на формуле) проверку гипотезы на двух бутстраповских выборках и записать, является ли разница между ними статистически значимой.
5. Повторить предыдущие два шага много раз и определить, как часто разница была значимой — это и есть расчетная мощность.

Размер выборки

Самое общепринятое использование расчетов мощности состоит в оценке того, насколько крупная выборка вам нужна.

Например, предположим, что вы смотрите на соотношение числа нажатий к числу показов (нажатия как процент показов) и тестируете новое объявление против существующего объявления. Сколько нажатий необходимо накопить в исследовании? Если вы интересуетесь только теми результатами, которые показывают огромную разницу (скажем, 50-процентную разницу), то может сработать относительно небольшая выборка. Если же, с другой стороны, интерес представляет даже незначительная разница, то необходима намного более крупная выборка. Стандартный подход состоит в принятии политики, что новое объявление должно непременно работать лучше, чем существующее объявление, на несколько процентов, скажем 10%; в противном случае существующее объявление останется на месте. Эта цель, "размер эффекта", в дальнейшем управляет размером выборки.

Например, предположим, что текущие соотношения числа нажатий к числу показов составляют порядка 1,1%, и вы стремитесь к 10%-му росту до 1,21%. Поэтому у нас будет две коробки: коробка *A* с 1,1% единиц (скажем, 110 единиц и 9890 нулей) и коробка *B* с 1,21% единиц (скажем, 121 единиц и 9879 нулей). Для начала попробуем из каждой коробки вынуть по 300 (это будет соответствовать 300 "показам" каждого объявления). Допустим, что наша первая выемка приводит к следующему:

- коробка *A*: 3;
- коробка *B*: 5.

Сразу же видим, что любая проверка гипотезы показала бы, что эта разница (5 против 3) далеко внутри диапазона случайной вариации. Данное сочетание размера выборки ($n = 300$ в каждой группе) и размера эффекта (10%-я разница) слишком небольшое для любой проверки гипотезы, чтобы та могла надежно показать разницу.

Поэтому мы можем попытаться увеличить размер выборки (попробуем 2000 показов) и предусмотрим больший рост (30% вместо 10%).

Например, предположим, что текущие соотношения числа нажатий к числу показов по-прежнему составляют 1,1%, но теперь мы стремимся к 50%-му росту до 1,65%. Таким образом, у нас есть две коробки: коробка *A* по-прежнему с 1,1% единицами (скажем, 110 единиц и 9890 нулей) и коробка *B* с 1,65% единицами (скажем, 165 единиц и 9868 нулей). Теперь мы попробуем вынуть по 2000 из каждой коробки. Допустим, что наша первая выемка приводит к следующему:

- коробка *A*: 19;
- коробка *B*: 34.

Проверка значимости на этой разнице ($34 - 19$) показывает, что она по-прежнему регистрируется, как "незначимая" (хотя намного ближе к значимой, чем более ранняя разница $5 - 3$). Для того чтобы рассчитать мощность, нам потребуется повторить предыдущую процедуру много раз либо использовать статистическую программу, которая может вычислять мощность, но наша исходная выемка свидетель-

ствует о том, что даже обнаружение 50%-го роста потребует нескольких тысяч показов.

Таким образом, для вычисления мощности или требуемого размера выборки существует четыре подвижных элемента:

- размер выборки;
- размер эффекта, который вы хотите обнаружить;
- уровень значимости (α), при котором будет проводиться тест;
- мощность.

Определите любые три из них, и четвертое может быть вычислено. Чаще всего вы захотите вычислить размер выборки, поэтому необходимо задать три других элемента. Далее приведен фрагмент кода на R для теста с привлечением двух пропорций, где обе выборки имеют одинаковый размер (в примере используется программный пакет `pwr`):

```
pwr.2p.test(h = ..., n = ..., sig.level = ..., power = )
```

где `h` — размер эффекта (в виде доли); `n` — размер выборки; `sig.level` — уровень значимости (α), при котором будет проводиться проверка; `power` — мощность (вероятность обнаружить размер эффекта).

Ключевые идеи для мощности и размера выборки

- Выяснение того, насколько большой размер выборки вам нужен, обязывает задуматься о перспективе статистической проверки, которую вы планируете провести.
- Необходимо определить минимальный размер эффекта, который вы хотите обнаружить.
- Необходимо также определить требуемую вероятность обнаружить этот размер эффекта (мощность).
- Наконец, необходимо определить уровень значимости (α), при котором будет проводиться проверка.

Дополнительные материалы для чтения

- Книга "Определение размера выборки и мощность" (Ryan T. P. Sample Size Determination and Power. — John Wiley & Sons, 2013) предлагает всесторонний и читаемый анализ данной темы.
- Стив Саймон (Steve Simon), статистический консультант, написал очень привлекательный пост в стиле рассказа по данной теме (**http://www.pmean.com/09/AppropriateSampleSize.html**).

Резюме

Принципы планирования эксперимента — рандомизация испытуемых на две группы или более, которым предлагаются разные варианты — позволяет делать допустимые выводы о том, насколько хорошо эти варианты работают. Лучше всего при этом задействовать контрольный вариант, "не вносящий никакого изменения". Предмет формального статистического вывода — проверка гипотез, *p*-значения, проверки на основе *t*-статистики и многое другое далее по списку — занимает много времени и пространства в традиционном курсе или учебнике по статистике, и формализованность является главным образом не нужной с точки зрения науки о данных. Однако по-прежнему остается важным признать тот вклад, который случайная вариация может внести в обман человеческого мозга. Интуитивно понятные процедуры повторного отбора (перестановка и бутстрап) позволяют аналитикам данных измерять степень, с которой случайная вариация может сыграть свою роль в анализе их данных.

Регрессия и предсказание

Возможно, наиболее распространенная цель в статистике состоит в том, чтобы ответить на вопросы: связана ли переменная X (или, вероятнее всего, $X_1, ..., X_p$) с переменной Y, и если да, то в чем заключается эта связь, и можем ли мы ее использовать для предсказания Y?

Нигде единение статистики и науки о данных не является более тесным, чем в области предсказания, в частности предсказания исхода (целевой) переменной на основе значений других "предикторных" переменных. Еще одна важная взаимосвязь лежит в области *обнаружения аномалий*, где диагностика регрессии, первоначально предназначенной для анализа данных и совершенствования регрессионной модели, может использоваться для обнаружения необычных записей в данных. Предпосылки корреляции и линейной регрессии насчитывают более одного столетия.

Простая линейная регрессия

Простая линейная регрессия, или линейная парная регрессия, моделирует связь между величиной одной переменной и величиной второй — например, по мере увеличения X, увеличивается и Y. Или же по мере увеличения X, уменьшается Y[1]. Корреляция — это еще один способ измерить, каким образом связаны две переменные *(см. разд. "Корреляция" главы 1)*. Разница между ними состоит в том, что корреляция измеряет связи между двумя переменными, тогда как регрессия квантифицирует природу этой связи.

Ключевые термины

Отклик (response)

Переменная, которую мы пытаемся предсказать.

Синонимы: зависимая переменная, Y-переменная, цель, исход.

Независимая переменная (independent variable)

Переменная, которая используется для предсказания отклика.

Синонимы: независимая переменная, X-переменная, предиктор, признак, атрибут.

[1] Этот и последующие разделы в настоящей главе используются с разрешения © 2017 Datastats, LLC, Питер Брюс и Эндрю Брюс.

Запись (record)

Вектор, который состоит из значений предикторов и значения исхода для индивидуального элемента данных или случая.

Синонимы: строка, случай, прецедент, образец, экземпляр, пример.

Пересечение (intercept)

Пересечение регрессионной прямой, т. е. предсказанное значение, когда $X = 0$.

Синонимы: b_0, β_0, точка пересечения.

Коэффициент регрессии (regression coefficient)

Наклон регрессионной прямой.

Синонимы: наклон, b_1, β_1, оценки параметров, веса.

Подогнанные значения (fitted values)

Оценки $\hat{Y}_i$, полученные из регрессионной прямой.

Синоним: предсказанные значения.

Остатки (residuals)

Разница между наблюдаемыми значениями и подогнанными значениями.

Синоним: ошибки.

Наименьшие квадраты (least squares)

Метод подгонки регрессии путем минимизации суммы квадратов остатков.

Синонимы: обычный метод наименьших квадратов, обычный МНК.

Уравнение регрессии

Простая линейная регрессия оценивает, насколько именно изменится *Y*, когда *X* изменится на определенную величину. Для коэффициента корреляции переменные *X* и *Y* взаимозаменяемы. В случае же с регрессией мы пытаемся предсказать переменную *Y* из переменной *X*, используя линейное соотношение (т. е. прямую):

$$Y = b_0 + b_1 X.$$

Эта формула читается, как "*Y* равняется b_1 умноженное на *X* плюс константа b_0". Свободный член b_0 называется *пересечением* (или константой), и коэффициент b_1 — *наклоном* для *X*. Оба члена отображаются на выходе в R как *коэффициенты*, хотя в повсеместном использовании термин "коэффициент" часто зарезервирован за b_1. Переменная *Y* называется *откликом*, или *зависимой* переменной, поскольку она зависит от *X*. Переменная *X* называется *предиктором* (от англ. *predictor* — предсказатель), или *независимой* переменной. Сообщество машинного обучения демонстрирует тенденцию использовать другие термины, называя *Y* целью и *X* — вектором признаков.

Рассмотрим диаграмму рассеяния на рис. 4.1, показывающую число лет, в течение которых рабочий был подвержен воздействию хлопчатобумажной пыли (`Exposure`) против показателя объема легких (`PEFR` — пиковая объемная скорость выдоха). Каким образом переменная `PEFR` связана с `Exposure`? Трудно сказать что-то конкретное на основе простого изображения.

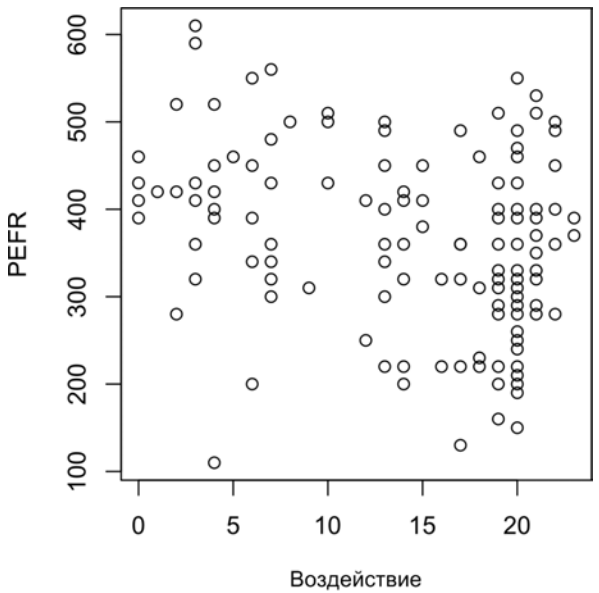


Рис. 4.1. Воздействие хлопка против объема легких

Простая линейная регрессия пытается подобрать "оптимальную" прямую, чтобы предсказать отклик `PEFR`, как функцию от предикторной переменной `Exposure`.

$$\texttt{PEFR} = b_0 + b_1\texttt{Exposure}.$$

Функция `lm` в R может использоваться для подгонки линейной регрессии.

```
model <- lm(PEFR ~ Exposure, data=lung)
```

`lm` обозначает линейную модель (linear model), и символ `~` обозначает, что переменная `PEFR` предсказывается переменной `Exposure`.

Распечатка объекта `model` произведет на выходе следующие данные:

```
Call:
lm(formula = PEFR ~ Exposure, data = lung)

Coefficients:
(Intercept)     Exposure
    424.583       -4.185
```

Пересечение, или b_0, равно 424,583 и может быть интерпретировано, как предсказанное для рабочего значение `PEFR` с нулевым воздействием, т. е. в размере 0 лет.

Коэффициент регрессии, или b_1, может быть интерпретирован следующим образом: для каждого дополнительного года, в течение которого рабочий подвергается воздействию хлопчатобумажной пыли, результат измерения PEFR рабочего уменьшается на $-4{,}185$.

Прямая регрессии данной модели изображена на рис. 4.2.

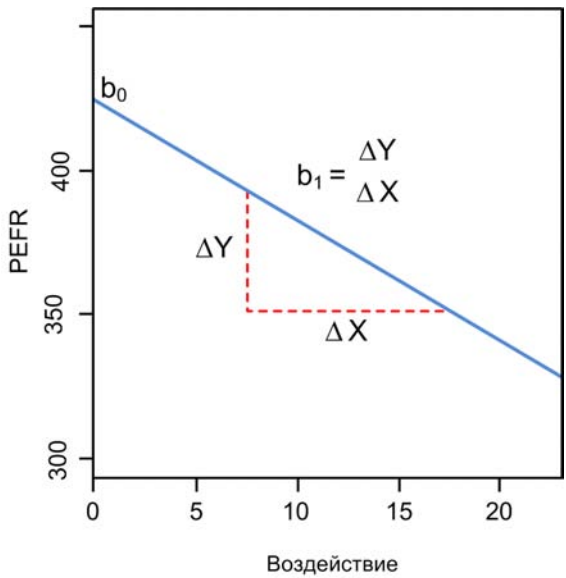


Рис. 4.2. Наклон и пересечение для регрессии, подогнанной к данным о легких

Подогнанные значения и остатки

В регрессионном анализе важными понятиями являются *подогнанные* значения и *остатки*. Обычно прямая не проходит точно через имеющиеся данные, поэтому уравнение регрессии должно включать заданный в явной форме остаточный член e_i:

$$Y = b_0 + b_1 X + e_i.$$

Подогнанные значения, или *предсказанные* значения, обычно обозначаются как $\hat{Y}_i$ (Y-hat, Y с шляпой). Они задаются следующей формулой:

$$\hat{Y}_i = \hat{b}_0 + \hat{b}_1 X_i.$$

Форма записи $\hat{b}_0$ и $\hat{b}_1$ говорит о том, что эти коэффициенты оценочные (расчетные) в отличие от известных (фактических).

Форма записи с шляпой: оценки против известных значений

Форма записи с "шляпой" используется для различения между оценками и известными значениями. Так, символ $\hat{b}$ ("b с шляпой") — это оценка неизвестного параметра b. Почему в статистике дифференцируют оценку от истинного значения? Оценка имеет неопределенность, тогда как истинное значение фиксировано[2].

Мы вычисляем остатки $\hat{e}_i$ путем вычитания *предсказанных* значений из исходных данных:

$$\hat{e}_i = Y_i - \hat{Y}_i.$$

В R можно получить подогнанные значения и остатки при помощи функций `predict` и `residuals`:

```
fitted <- predict(model)
resid <- residuals(model)
```

На рис. 4.3 иллюстрируются остатки, т. е. отклонения от прямой регрессии, подогнанной к данным о легких. Остатки — это длина вертикальных пунктирных линий от данных до прямой.

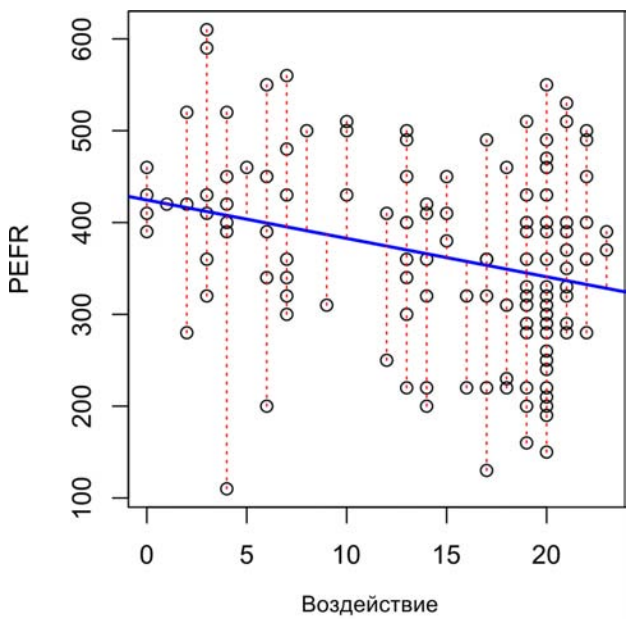


Рис. 4.3. Остатки, т. е. отклонения от прямой регрессии (обратите внимание на другую шкалу оси *y* на рис. 4.2, отсюда и очевидным образом другой наклон прямой)

[2] В байесовской статистике предполагается, что истинное значение является случайной величиной с заданным распределением. В байесовском контексте вместо оценок неизвестных параметров существуют априорные и апостериорные распределения.

Наименьшие квадраты

Каким образом выполняется подгонка модели к данным? Когда существует четкая связь, вы можете мысленно представить подгонку прямой вручную. На практике прямая регрессии является оценкой, которая минимизирует сумму квадратических значений остатков, также именуемых остаточной суммой квадратов, или RSS (residual sum of squares):

$$\text{RSS} = \sum_{i=1}^{n}\left(Y_i - \hat{Y}_i\right)^2 = \sum_{i=1}^{n}\left(Y_i - \hat{b}_0 - \hat{b}_1 X_i\right)^2.$$

Оценки $\hat{b}_0$ и $\hat{b}_1$ — это значения, которые минимизирует RSS.

Метод минимизации суммы квадратических остатков называется *регрессией наименьших квадратов*, или обычным методом наименьших квадратов (обычным МНК). Данный метод часто приписывается Карлу Фридриху Гауссу, немецкому математику, он был в 1805 г. впервые опубликован французским математиком Андре-Мари Лежандром (Adrien-Marie Legendre). Регрессия наименьших квадратов приводит к простой формуле вычисления коэффициентов:

$$\hat{b}_1 = \frac{\sum_{i=1}^{n}\left(Y_i - \bar{Y}\right)\left(X_i - \bar{X}\right)}{\sum_{i=1}^{n}\left(X_i - \bar{X}\right)^2};$$

$$\hat{b}_0 = \bar{Y} - \hat{b}_1 X.$$

Исторически, вычислительное удобство является одной из причин широкого применения метода наименьших квадратов в регрессии. С появлением больших данных его вычислительная скорость по-прежнему остается важным фактором. Метод наименьших квадратов, как и среднее значение *(см. разд. "Медиана и робастные оценки" главы 1)*, чувствителен к выбросам, хотя этот факт имеет тенденцию быть значимой проблемой только в небольших или умеренных по размеру задачах. *См. разд. "Выбросы" далее в этой главе*, где рассматриваются выбросы в регрессии.

Терминология регрессионного анализа

Когда аналитики и исследователи используют термин *"регрессия"* отдельно, они обычно ссылаются на линейную регрессию; в центре внимания, как правило, находится разработка линейной модели для объяснения связи между предикторными переменными и числовой переменной исхода (результирующей переменной). В своем формальном статистическом смысле регрессия также охватывает нелинейные модели, которые выдают функциональную связь между предикторными переменными и переменной исхода. В сообществе машинного обучения данный термин также временами используется в свободном толковании для ссылки на использование любой предсказательной модели, которая производит предсказанный числовой исход (в отличие от методов классификации, которые предсказывают бинарный или категориальный исход).

Предсказание против объяснения (профилирование)

Исторически, первостепенное использование регрессии состояло в выявлении предполагаемой линейной связи между предикторными переменными и переменной исхода. Цель заключалась в том, чтобы понять связь и объяснить ее при помощи данных, к которым выполнялась подгонка регрессии. В этом случае основное внимание находится на оценочном значении наклона уравнения регрессии $\hat{b}$. Экономисты хотят знать связь между потребительскими расходами и ростом ВВП. Чиновники здравоохранения могут захотеть понять, является ли кампания по информированию общественности эффективной при продвижении методов безопасного секса. В таких случаях в центре внимания находится не предсказание отдельных случаев, а, наоборот, понимание общей связи.

С появлением больших данных регрессия широко используется для формирования модели с целью предсказания индивидуальных исходов для новых данных вместо статистического объяснения имеющихся под рукой данных (т. е. предсказательной модели). В этом случае главными целевыми составляющими являются подогнанные значения $\hat{Y}$. В маркетинге регрессия может использоваться для предсказания изменения в доходе в ответ на размер рекламной кампании. Университеты используют регрессию для предсказания среднего академического балла GPA студентов на основании их отметок за экзамен на определение академических способностей SAT[3].

Модель регрессии, которая подогнана к данным хорошо, настраивается таким образом, что изменения в X приводят к изменениям в Y. Однако, как таковое уравнение регрессии не доказывает направление причинной обусловленности. Выводы о причинной обусловленности должны делаться на основании более широкого контекста понимания связи. Например, уравнение регрессии может показать определенную связь между числом нажатий на веб-рекламе и числом конверсий. Как раз, наше знание маркетингового процесса, а не уравнения регрессии, приводит нас к заключению, что именно нажатия на объявлении генерируют продажи, а не наоборот.

Ключевые идеи для простой линейной регрессии

- Уравнение регрессии моделирует связь между переменной отклика Y и предикторной переменной X в виде прямой.
- Регрессионная модель выдает подогнанные значения и остатки — предсказания отклика и ошибки предсказаний.
- Подгонка регрессионных моделей, как правило, выполняется методом наименьших квадратов.
- Регрессия используется как для предсказания, так и для статистического объяснения.

[3] GPA (grade point average) — средний академический балл студентов; SAT (scholastic assessment test) — отборочный экзамен для выпускников школ, поступающих в вузы. — *Прим. пер.*

Дополнительные материалы для чтения

Подробности о всесторонней трактовке темы сравнения предсказания со статистическим объяснением см. в статье Галита Шмуели (Galit Shmueli) "Объяснить или предсказать" (To Explain or to Predict?, **https://projecteuclid.org/euclid.ss/1294167961**).

Множественная линейная регрессия

Когда предикторов несколько, данное уравнение просто расширяется для их размещения:

$$Y = b_0 + b_1 X_1 + b_2 X_2 + ... + b_p X_p + e.$$

Вместо прямой теперь у нас линейная модель — связь между каждым коэффициентом и его переменной (признаком) линейная.

Ключевые термины

Среднеквадратическая ошибка (root mean squared error)

Квадратный корень из среднеквадратической ошибки регрессии (наиболее широко используемый метрический показатель для сравнения регрессионных моделей).

Синоним: RMSE.

Стандартная ошибка остатков (residual standard error)

То же самое, что и среднеквадратическая ошибка, но скорректированная для степеней свободы.

Синоним: RSE.

***R*-квадрат (R-squared)**

Доля дисперсии, объясненная моделью, со значениями в диапазоне от 0 до 1.

Синонимы: коэффициент детерминации, R2.

***t*-статистика (*t*-statistic)**

Метрический показатель для сравнения важности переменных в модели, получаемый в результате деления регрессионного коэффициента для какого-либо предиктора на стандартную ошибку коэффициента.

Взвешенная регрессия (weighted regression)

Регрессия, в которой записям поставлены в соответствие разные веса.

Все другие понятия из простой линейной регрессии, такие как подгонка наименьшими квадратами и определение подогнанных значений и остатков, расширяются на множественную линейную регрессию. Например, подогнанные значения задаются следующей формулой:

$$\hat{Y}_i = \hat{b}_0 + \hat{b}_1 X_{1,i} + \hat{b}_2 X_{2,i} + ... + \hat{b}_p X_{p,i} + e.$$

Пример: данные о жилом фонде округа Кинг

Примером использования регрессии является оценивание стоимости домов. Оценщики округа должны оценивать стоимость домов в целях обложения налогами. Потребители недвижимости и профессионалы в этой области консультируются на популярных веб-сайтах, таких как Zillow, чтобы удостовериться в справедливости цены. Ниже приведено несколько строк данных о жилом фонде округа Кинг (Сиэтл, шт. Вашингтон) из кадра данных `data.frame house`:

```
head(house[, c("AdjSalePrice", "SqFtTotLiving", "SqFtLot", "Bathrooms",
               "Bedrooms", "BldgGrade")])
Source: local data frame [6 x 6]
 AdjSalePrice SqFtTotLiving SqFtLot Bathrooms Bedrooms BldgGrade
        (dbl)         (int)   (int)     (dbl)    (int)     (int)
1      300805          2400    9373      3.00        6         7
2     1076162          3764   20156      3.75        4        10
3      761805          2060   26036      1.75        4         8
4      442065          3200    8618      3.75        5         7
5      297065          1720    8620      1.75        4         7
6      411781           930    1012      1.50        2         8
```

Цель состоит в том, чтобы предсказать продажную цену на основе остальных переменных. Переменная `lm` обрабатывает случай множественной регрессии просто путем включения большего количества членов на правой стороне уравнения; аргумент `na.action=na.omit` заставляет модель отбрасывать записи, в которых имеются отсутствующие значения:

```
house_lm <- lm(AdjSalePrice ~ SqFtTotLiving + SqFtLot + Bathrooms +
                 Bedrooms + BldgGrade,
               data=house, na.action=na.omit)
```

Распечатка объекта `house_lm` произведет на выходе следующие данные:

```
house_lm

Call:
lm(formula = AdjSalePrice ~ SqFtTotLiving + SqFtLot + Bathrooms +
    Bedrooms + BldgGrade, data = house, na.action = na.omit)

Coefficients:
(Intercept)  SqFtTotLiving       SqFtLot     Bathrooms
 -5.219e+05      2.288e+02    -6.051e-02    -1.944e+04
   Bedrooms      BldgGrade
 -4.778e+04      1.061e+05
```

Интерпретация коэффициентов такая же, как и в простой линейной регрессии: предсказанное значение $\hat{Y}$ изменяется коэффициентом b_j для каждого единичного изменения в X_j с учетом всех остальных переменных, X_k для $k \neq j$ остается прежним. Например, добавление дополнительного квадратного фута общей площа-

ди к дому увеличивает оценочную стоимость примерно на 229 долларов; добавление 1000 кв. футов предполагает, что цена возрастет на 228 800 долларов.

Диагностика модели

Самым важным метрическим показателем результативности с точки зрения науки о данных является *среднеквадратическая ошибка*, или RMSE (root mean squared error). RMSE — это квадратный корень из средней квадратической ошибки в предсказанных значениях $\hat{y}_i$:

$$\text{RMSE} = \sqrt{\frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{n}}.$$

Она измеряет общую точность модели и является основанием для ее сравнения с другими моделями (включая модели, подогнанные с использованием специальных приемов машинного обучения). Аналогичной RMSE является *стандартная ошибка остатков*, или RSE (residual standard error). В этом случае мы имеем p предикторов, и RSE задается следующей формулой:

$$\text{RSE} = \sqrt{\frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{n-p-1}}.$$

Единственная разница состоит в том, что знаменатель является степенями свободы, в противоположность числу записей *(см. разд. "Степени свободы" главы 3)*. На практике для линейной регрессии разница между RMSE и RSE очень мала, в особенности для приложений больших данных.

Функция `summary` в R вычисляет RSE, а также другие метрические показатели регрессионной модели:

```
summary(house_lm)

Call:
lm(formula = AdjSalePrice ~ SqFtTotLiving + SqFtLot + Bathrooms +
    Bedrooms + BldgGrade, data = house, na.action = na.omit)

Residuals:
     Min       1Q   Median       3Q      Max
-1199508  -118879   -20982    87414   9472982

Coefficients:
               Estimate Std. Error t value Pr(>|t|)
(Intercept)   -5.219e+05  1.565e+04 -33.349  < 2e-16 ***
SqFtTotLiving  2.288e+02  3.898e+00  58.699  < 2e-16 ***
SqFtLot       -6.051e-02  6.118e-02  -0.989    0.323
Bathrooms     -1.944e+04  3.625e+03  -5.362 8.32e-08 ***
```

```
Bedrooms       -4.778e+04  2.489e+03 -19.194  < 2e-16 ***
BldgGrade       1.061e+05  2.396e+03  44.287  < 2e-16 ***
---
Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 261200 on 22683 degrees of freedom
Multiple R-squared: 0.5407,     Adjusted R-squared: 0.5406
F-statistic: 5340 on 5 and 22683 DF, p-value: < 2.2e-16
```

Еще один полезный метрический показатель, который вы увидите в данных на выходе из программных систем, — это *коэффициент детерминации*, также именуемый статистикой *R-квадрат*, или R^2. R-квадрат колеблется в пределах от 0 до 1 и измеряет объясняемую в модели долю вариации в данных. Он полезен главным образом в использовании регрессии в объяснительных целях, где вы хотите проанализировать, как хорошо модель подогнана к данным. Формула для R^2 следующая:

$$R^2 = 1 - \frac{\sum_{i=1}^{n}\left(y_i - \hat{y}_i\right)^2}{\sum_{i=1}^{n}\left(y_i - \overline{y}_i\right)^2}.$$

Знаменатель пропорционален дисперсии *Y*. Выходные данные в R также сообщают о *скорректированном коэффициенте детерминации* (скорректированном R-квадрате), который корректируется с учетом степеней свободы; в множественной регрессии он очень редко имеет значимую разницу.

Вместе с оценочными коэффициентами R сообщает о стандартной ошибке коэффициентов (SE) и *t-статистике*:

$$t_b = \frac{\hat{b}}{\mathrm{SE}\left(\hat{b}\right)}.$$

t-Статистика — и ее зеркальное отображение, *p*-значение — измеряет степень, с которой коэффициент является "статистически значимым", т. е. за пределами диапазона того, что может породить случайная расстановка предикторной и целевой переменной. Чем выше *t*-статистика (и ниже *p*-значение), тем более значим предиктор. Поскольку экономность модели является ее ценной особенностью, полезно иметь такого рода инструмент, который позволяет ориентировать в выборе переменных для включения в качестве предикторов *(см. разд. "Отбор модели и шаговая регрессия" далее в этой главе)*.

В дополнение к *t*-статистике R и другие программные пакеты часто будут сообщать о *p-значении* (в R это `Pr(>|t|)`) среди выводимых данных) и *F-статистике*. Аналитики данных обычно не слишком увлекаются интерпретацией этих статистических данных, равно как и вопросом статистической значимости. Аналитики данных преимущественно сосредотачиваются на *t*-статистике как полезном ориентире, подсказывающем, включать предиктор в модель или нет. Высокие *t*-статистики (которые сопровождаются *p*-значениями, находя-

щимися около 0) говорят о том, что предиктор должен быть сохранен в модели, в то время как очень низкие *t*-статистики указывают на то, что предиктор может быть отброшен. *См. разд. "p-Значение" главы 3* для получения более подробной информации.

Перекрестная проверка

Классические статистические метрические показатели регрессии (R^2, *F*-статистики и *p*-значения) являются "внутривыборочными" показателями — они применяются к тем же данным, которые использовались для подгонки модели. На интуитивном уровне вы видите, что имеет большой смысл отложить немного исходных данных, не используя их для подгонки модели, и далее применить модель к зарезервированным (отложенным в сторону) данным, чтобы увидеть, как хорошо она справляется со своей работой. Обычно значительную часть данных вы будете использовать, чтобы выполнить подгонку модели, а оставшуюся часть — чтобы ее проверить.

Такая идея "вневыборочной" проверки не является новой, но она не утвердилась до тех пор, пока большие наборы данных не получили широкое распространение; имея в распоряжении небольшой набор данных, аналитики, как правило, хотят использовать все имеющиеся данные и на их основе выполнять подгонку лучшей модели.

Использование контрольной выборки с отложенными данными, тем не менее, ставит вас в зависимость от некоторой неопределенности, которая возникает просто из-за вариабельности в малой контрольной выборке. Насколько будут отличаться результаты диагностики модели, если взять другую контрольную выборку с отложенными данными?

Перекрестная проверка расширяет идею контрольной выборки с отложенными данными до множественных последовательных контрольных выборок. Алгоритм базовой *k-блочной перекрестной проверки* выглядит следующим образом:

1. Отложить $1/k$ данных в качестве контрольной выборки.
2. Натренировать модель на оставшихся данных.
3. Применить модель к контрольной выборке $1/k$ (оценить результаты) и записать необходимые метрические показатели диагностики модели.
4. Восстановить первые $1/k$ данных и отложить следующее $1/k$ (исключая любые записи, которые были выбраны в первый раз).
5. Повторить шаги 2 и 3.
6. Повторять, пока каждая запись не будет использована в процентной доле, предназначенной для контрольной выборки.
7. Усреднить или же скомбинировать метрические показатели диагностики модели.

Подразделение данных на тренировочную выборку и контрольную выборку также называется *разделением на блоки* (fold).

Отбор модели и шаговая регрессия

В некоторых задачах многие переменные могут применяться в качестве предикторов в регрессии. Например, для предсказания стоимости дома могут использоваться дополнительные переменные, такие как размер подвала или год постройки. В R их легко добавить к уравнению регрессии:

```
house_full <- lm(AdjSalePrice ~ SqFtTotLiving + SqFtLot + Bathrooms +
                  Bedrooms + BldgGrade + PropertyType + NbrLivingUnits +
                  SqFtFinBasement + YrBuilt + YrRenovated +
                  NewConstruction,
                data=house, na.action=na.omit)
```

Добавление большего количества переменных, однако, не обязательно означает, что у нас получится более хорошая модель. В статистике в качестве ориентира пользуются принципом *бритвы Оккама*, который позволяет выбрать модель: при прочих равных условиях предпочтение отдается использованию более простой модели над более сложной.

Включение в состав дополнительных переменных всегда уменьшает RMSE и увеличивает R^2. Следовательно, они не подходят в качестве ориентиров, которые помогают выбрать модель. В 1970-х гг. Хиротугу Акаике (Hirotugu Akaike), выдающийся японский статистик, разработал метрический показатель, названный информационным критерием Акаике — AIC (Akaike's Information Criteria), который штрафует добавление членов в модель. В случае регрессии AIC имеет следующую форму:

$$\text{AIC} = 2P + n\log\left(\text{RSS}/n\right),$$

где p — это число переменных; n — число записей. Цель состоит в том, чтобы найти модель, которая минимизирует AIC; модели с k дополнительными переменными штрафуются на $2k$.

AIC, BIC и CP Мэллоуза

Формула AIC может показаться немного таинственной, но фактически она основывается на асимптотических результатах в теории информации. Существует несколько вариантов AIC:

- AICc — версия AIC, скорректированная для размеров небольших выборок;
- BIC, или байесовский информационный критерий, — аналогичный AIC, но с более сильным штрафом за включение в состав модели дополнительных переменных;
- CP Мэллоуза — вариант AIC, разработанный Колином Мэллоузом (Colin Mallows).

Аналитикам данных обычно не приходится беспокоиться по поводу различий среди этих внутривыборочных метрических показателей или теории, лежащей в их основе.

Как отыскать модель, которая минимизирует AIC? Один из подходов состоит в переборе всех возможных моделей, который называется *регрессией всех подмножеств* (all subset regression). Данный подход вычислительно затратен и не выполним для задач с большими данными и многими переменными. Привлекательной альтернативой является использование *шаговой регрессии*, которая последовательно добавляет и отбрасывает предикторы для нахождения модели, которая понижает AIC. Программный пакет `MASS`, созданный Венеблз (Venebles) и Рипли (Ripley), предлагает функцию шаговой регрессии, которая называется `stepAIC`:

```
library(MASS)
step <- stepAIC(house_full, direction="both")
step

Call:
lm(formula = AdjSalePrice ~ SqFtTotLiving + Bathrooms + Bedrooms +
    BldgGrade + PropertyType + SqFtFinBasement + YrBuilt, data = house0,
    na.action = na.omit)

Coefficients:
              (Intercept)              SqFtTotLiving
                6227632.22                    186.50
                 Bathrooms                  Bedrooms
                   44721.72                -49807.18
                 BldgGrade  PropertyTypeSingle Family
                  139179.23                  23328.69
    PropertyTypeTownhouse            SqFtFinBasement
                   92216.25                      9.04
                   YrBuilt
                   -3592.47
```

Функция выбрала модель, в которой несколько переменных были отброшены из `house_full`: `SqFtLot`, `NbrLivingUnits`, `YrRenovated` и `NewConstruction`.

Более простыми являются *прямой отбор* и *обратный отбор*. В прямом отборе вы начинаете без предикторов и на каждом шаге добавляете по одному предиктору, который имеет самый большой вклад в R^2, останавливаясь, когда вклад перестает быть статистически значимым. В обратном отборе, или *обратном исключении* (backward elimination), вы начинаете с полной модели и устраняете предикторы, которые не являются статистически значимыми, пока у вас не останется модель, в которой все предикторы являются статистически значимыми.

Штрафная регрессия аналогична по духу критерию AIC. Вместо того чтобы явным образом перебирать дискретный набор моделей, уравнение подгонки моделей содержит ограничение, которое штрафует модель за слишком большое число переменных (параметров). Взамен полного устранения предикторных переменных — как в шаговой регрессии, прямом и обратном отборе — штрафная регрессия применяет штраф путем понижения коэффициентов в некоторых случаях почти до

нуля. Общепринятыми штрафными методами регрессии являются *гребневая регрессия* и *лассо-регрессия*.

Шаговая регрессия и регрессия всех подмножеств — это *внутривыборочные* методы диагностики и настройки моделей. Это означает, что отбор модели, возможно, подвержен переподгонке и не сможет показывать такую же хорошую результативность применительно к новым данным. Один из общепринятых подходов для ее предотвращения состоит в использовании перекрестной проверки с целью подтверждения результативности моделей. В линейной регрессии переподгонка, как правило, не является главной проблемой, вследствие простой (линейной) глобальной структуры, накладываемой на данные. Для более сложных типов моделей, в особенности итеративных процедур, которые откликаются на локальную структуру данных, перекрестная проверка является очень важным инструментом; по поводу дальнейших подробностей *см. разд. "Перекрестная проверка" ранее в этой главе*.

Взвешенная регрессия

Взвешенная регрессия в статистике используется для самых различных целей; в частности, она имеет большое значение для анализа сложных опросов. Аналитики данных могут посчитать взвешенную регрессию полезной в двух случаях:

- инверсно-дисперсионное взвешивание, когда разные наблюдения были измерены с разной точностью;
- анализ данных в агрегированной форме, т. е. такой, когда в весовой переменной кодируется, сколько исходных наблюдений каждая строка представляет в агрегированных данных.

Например, в данных о жилой недвижимости более старые продажи менее надежны, чем более свежие. Используя `DocumentDate` для определения года продажи, можно вычислить вес `Weight` как число лет начиная с 2005 (начала данных).

```
library(lubridate)
house$Year = year(house$DocumentDate)
house$Weight = house$Year - 2005
```

Мы можем вычислить взвешенную регрессию при помощи функции `lm`, используя аргумент `weight`.

```
house_wt <- lm(AdjSalePrice ~ SqFtTotLiving + SqFtLot + Bathrooms +
                 Bedrooms + BldgGrade,
               data=house, weight=Weight)
round(cbind(house_lm=house_lm$coefficients,
            house_wt=house_wt$coefficients), digits=3)
```

```
                  house_lm     house_wt
(Intercept)     -521924.722  -584265.244
SqFtTotLiving       228.832      245.017
SqFtLot              -0.061       -0.292
```

```
Bathrooms       -19438.099   -26079.171
Bedrooms        -47781.153   -53625.404
BldgGrade       106117.210   115259.026
```

Коэффициенты во взвешенной регрессии немного отличаются от исходной регрессии.

Ключевые идеи для множественной линейной регрессии

- Множественная линейная регрессия моделирует связь между переменной отклика Y и многочисленными предикторными переменными X_1, ..., X_p.
- Самыми важными метрическими показателями, используемыми для оценки модели, являются среднеквадратическая ошибка (RMSE) и коэффициент детерминации (R^2).
- Стандартная ошибка коэффициентов может использоваться для измерения надежности вклада переменной в модель.
- Шаговая регрессия — это метод автоматического определения того, какие переменные должны быть включены в состав в модели.
- Взвешенная регрессия используется для того, чтобы дать определенным записям больший или меньший вес при подгонке уравнения.

Предсказание на основе регрессии

Основная цель регрессии в науке о данных заключается в предсказании. Это полезно учитывать, поскольку регрессия, будучи проверенным временем и признанным статистическим методом, сопровождается багажом, который более соответствует своей традиционной роли объяснительного моделирования, чем предсказанию.

Ключевые термины

Предсказательный интервал (prediction interval)

Интервал неопределенности вокруг индивидуального предсказанного значения.

Экстраполяция (extrapolation)

Расширение модели за пределы диапазона данных, которые используются для ее подгонки.

Опасности экстраполяции

Регрессионные модели не следует использовать для экстраполирования за пределы диапазона данных. Модель допустима только для тех значений предикторов, для которых имеется достаточно значений в данных (и даже в случае достаточности данных могут иметься другие проблемы: *см. разд. "Проверка допущений: диагностика регрессии" далее в этой главе*). В качестве крайнего случая предположим, что `model_lm` используется для предсказания значения незанятого земельного участка площадью 5000 кв. футов. В данном случае все предикторы, связанные со зданием, будут иметь значение 0, и уравнение регрессии выдаст абсурдное предсказание $-521\,900 + 5000 \cdot (-0,0605) = -522\,202$ долларов. Почему это произошло? Данные содержат только участки со зданиями, а записи, соответствующие свободным участкам, отсутствуют. Следовательно, модель не имеет никакой информации, говорящей ей, каким образом предсказывать продажную цену на свободный земельный участок.

Доверительный и предсказательный интервалы

Значительная часть статистических данных предусматривает понимание и измерение вариабельности (неопределенности). *t*-Статистики и *p*-значения, о которых сообщается на выходе из регрессии, решают это формальным образом, что иногда полезно для отбора переменной *(см. разд. "Диагностика модели" ранее в этой главе)*. Более полезными метрическими показателями являются доверительные интервалы, которые суть интервалы неопределенности, помещенные вокруг коэффициентов регрессии и предсказаний. Простой способ разобраться в них — применить бутстрап (*см. разд. "Бутстрап" главы 2* с дополнительной информацией об общей процедуре бутстрапирования). Самые распространенные доверительные интервалы регрессии, которые встречаются на выходе из программных систем, — это доверительные интервалы для параметров (коэффициентов) регрессии. Далее приведен алгоритм бутстрапа, который генерирует доверительные интервалы для параметров (коэффициентов) регрессии с использованием набора данных с P предикторами и n записями (строками):

1. Рассматривать каждую строку (включая переменную исхода) как одиночный "пакет" и поместить все n пакеты в коробку.
2. Вынуть пакет наугад, записать его значения и вернуть его в коробку.
3. Повторить шаг 2 n раз; теперь у вас есть одна повторно отобранная бутстраповская выборка.
4. Выполнить подгонку регрессии к бутстраповской выборке и записать оценочные коэффициенты.
5. Повторить шаги 2–4, скажем, 1000 раз.
6. Теперь у вас есть 1000 бутстраповских значений по каждому коэффициенту; найти соответствующие процентили для каждого из них (например, 5-й и 95-й для 90%-го доверительного интервала).

В R вы можете использовать функцию `Boot` с целью сгенерировать фактические бутстраповские доверительные интервалы для коэффициентов или просто использовать интервалы на основе формул, которые в R являются рутинными выходными данными. Концептуальное значение и интерпретация остаются теми же и не являются первоочередной необходимостью для аналитиков данных, потому что эти данные касаются коэффициентов регрессии. Больший интерес для аналитиков данных представляют интервалы вокруг предсказанных значений y ($\hat{Y}_i$). Неопределенность вокруг $\hat{Y}_i$ вытекает из двух источников:

- неопределенность в том, каковы релевантные предикторные переменные и их коэффициенты (см. приведенный выше алгоритм бутстрапа);
- дополнительная ошибка, присущая индивидуальным точкам данных.

Ошибку индивидуальной точки данных можно представить следующим образом: даже если известно, каким было уравнение регрессии (например, если имелось огромное число записей для выполнения подгонки), *фактические* значения исхода для заданного набора значений предикторов будут варьироваться. Например, несколько домов — каждый с 8 комнатами, общей площадью 6500 кв. футов, 3 ванными комнатами и подвалом — могут иметь разные стоимости. Эту индивидуальную ошибку можно смоделировать остатками от подогнанных значений. Алгоритм бутстрапа для моделирования ошибки регрессионной модели и ошибки индивидуальной точки данных будет выглядеть следующим образом:

1. Взять бутстраповскую выборку из данных (в деталях разъясненную ранее).
2. Выполнить подгонку регрессии и предсказать новое значение.
3. Взять наугад одиночный остаток из первоначальной подгонки регрессии, добавить его к предсказанному значению и записать результат.
4. Повторить шаги 1–3, скажем, 1000 раз.
5. Найти 2,5-й и 97,5-й процентили результатов.

Предсказательный интервал или доверительный интервал?

Предсказательный интервал касается неопределенности вокруг одиночного значения, в то время как доверительный интервал связан со средним или другой статистикой, рассчитанными из многочисленных значений. Таким образом, для одного и того же значения предсказательный интервал, как правило, будет намного шире, чем доверительный интервал. Мы моделируем ошибку индивидуального значения в бутстраповской модели путем отбора индивидуального остатка с тем, чтобы его прикрепить к предсказанному значению. Который из двух интервалов использовать? Все зависит от контекста и цели анализа, но в целом аналитики данных интересуются специфическими индивидуальными предсказаниями, поэтому предсказательный интервал будет более подходящим. Выбор доверительного интервала тогда, когда необходимо использовать предсказательный интервал, значительно недооценивает неопределенность в конкретном предсказанном значении.

Ключевые идеи для предсказания с использованием регрессии

- Экстраполяция за пределы диапазона данных может привести к ошибке.
- Доверительные интервалы квантифицируют неопределенность вокруг коэффициентов регрессии.
- Предсказательные интервалы квантифицируют неопределенность в индивидуальных предсказаниях.
- Большинство программных систем, включая R, на выходе будут производить стандартные предсказательные и доверительные интервалы с использованием формул.
- Также может использоваться бутстрап; его интерпретация и идея остаются прежними.

Факторные переменные в регрессии

Факторные переменные, именуемые также *категориальными* переменными, принимают предельное число дискретных значений. Например, целью ссуды могут быть "консолидация задолженности", "свадьба", "автомобиль" и т. д. Двоичная (да/нет) переменная, именуемая также *индикаторной* переменной, является особым случаем факторной переменной. Регрессия требует на входе числовые данные, поэтому факторные переменные нужно перекодировать, чтобы их можно было использовать в модели. Общепринятый подход состоит в конвертировании переменной в набор двоичных *фиктивных* переменных.

Ключевые термины

Фиктивные переменные (dummy variables)

Двоичные переменные в формате 0-1, полученные путем перекодирования факторных данных, для использования в регрессии и других моделях.

Опорное кодирование (reference coding)

Общепринятый тип кодирования, используемый в статистике, при котором один уровень фактора используется в качестве опорного, а другие факторы сопоставляются с этим уровнем.

Синоним: комбинированное кодирование.

Кодировщик с одним активным состоянием (one hot encoder)

Тип кодирования, общепринятый в сообществе машинного обучения, при котором сохраняются все уровни факторов. Широко используется в определенных алгоритмах машинного обучения; вместе с тем этот прием не подходит для множественной линейной регрессии.

Кодирование отклонений (deviation coding)

Тип кодирования, при котором каждый уровень сравнивается не с опорным уровнем, а с общим средним.

Синонимы: контрасты сумм, кодирование эффектов, маргинальное кодирование.

Представление фиктивные переменных

В данных о жилом фонде округа Кинг имеется факторная переменная, соответствующая типу собственности; ниже показано небольшое подмножество из шести записей.

```
head(house[, 'PropertyType'])
Source: local data frame [6 x 1]

   PropertyType
         (fctr)
1     Multiplex
2 Single Family
3 Single Family
4 Single Family
5 Single Family
6     Townhouse
```

Имеется три возможных значения: `Multiplex` (мультиплекс), `Single Family` (односемейный) и `Townhouse` (таунхаус). Для того чтобы воспользоваться данной факторной переменной, мы должны ее конвертировать в набор двоичных переменных. Это делается путем создания двоичной переменной для каждого возможного значения факторной переменной. Для того чтобы сделать это в R, мы используем функцию `model.matrix`[4]:

```
prop_type_dummies <- model.matrix(~PropertyType -1, data=house)
head(prop_type_dummies)
  PropertyTypeMultiplex PropertyTypeSingle Family PropertyTypeTownhouse
1                     1                         0                     0
2                     0                         1                     0
3                     0                         1                     0
4                     0                         1                     0
5                     0                         1                     0
6                     0                         0                     1
```

Функция `model.matrix` конвертирует кадр данных в матрицу, подходящую для линейной модели. Факторная переменная `PropertyType`, которая имеет три отличаю-

[4] Аргумент –1 в `model.matrix` генерирует представление, используя кодирование с одним активным состоянием (путем удаления константы пересечения, отсюда и "–"). В остальном, в R по умолчанию порождается матрица с $P - 1$ столбцами, где первый факторный уровень является опорным.

щихся уровня, представлена матрицей с тремя столбцами. В сообществе машинного обучения такое представление имеет название *кодирование с одним активным состоянием (см. разд. "Кодировщик с одним активным состоянием" главы 6)*. В определенных машинно-обучаемых алгоритмах, таких как ближайшие соседи и древовидные модели, кодирование с одним активным состоянием является стандартным способом представления факторных переменных (например, *см. разд. "Древовидные модели" главы 6*).

В настройках регрессии факторная переменная с P отличающимися уровнями обычно представлена матрицей только с $P-1$ столбцами. Это вызвано тем, что регрессионная модель, как правило, включает константный член пересечения. Что касается пересечения, то, как только вы определили двоичные значения для $P-1$, значение P-го известно и может считаться избыточным. Добавление P-го столбца вызовет ошибку мультиколлинеарности *(см. разд. "Мультиколлинеарность" далее в этой главе)*.

В R принятое по умолчанию представление состоит в использовании первого уровня фактора в качестве *опоры* и интерпретации оставшихся уровней относительно этого фактора.

```
lm(AdjSalePrice ~ SqFtTotLiving + SqFtLot + Bathrooms +
+      Bedrooms + BldgGrade + PropertyType, data=house)

Call:
lm(formula = AdjSalePrice ~ SqFtTotLiving + SqFtLot + Bathrooms +
    Bedrooms + BldgGrade + PropertyType, data = house)

Coefficients:
              (Intercept)             SqFtTotLiving
                -4.469e+05                 2.234e+02
                   SqFtLot                Bathrooms
                -7.041e-02                -1.597e+04
                  Bedrooms                BldgGrade
                -5.090e+04                 1.094e+05
PropertyTypeSingle Family      PropertyTypeTownhouse
                -8.469e+04                -1.151e+05
```

Данные на выходе из регрессии в R отражаются двумя коэффициентами, соответствующими типу собственности `Property Type`: `PropertyTypeSingle Family` и `PropertyTypeTownhouse`. Коэффициент `Multiplex` отсутствует, поскольку он неявно определяется, когда `PropertyTypeSingle Family == 0` и `PropertyTypeTownhouse == 0`. Эти коэффициенты интерпретируются как относительные для `Multiplex`, таким образом, стоимость дома с типом `Single Family` меньше почти на 85 тыс. долларов, а стоимость дома с типом `Townhouse` меньше более чем на 150 тыс. долларов[5].

[5] На первый взгляд это выглядит нелогичным, но может быть объяснено влиянием местоположения, которое рассматривается как искажающая переменная *(см. разд. "Искажающие переменные" далее в этой главе)*.

Другие кодировки факторов

Существует несколько других способов кодировки факторных переменных, известных как системы *контрастного кодирования*. Например, *кодирование отклонений*, именуемое также *контрастами сумм*, сравнивает каждый уровень с общим средним. Другой вариант — *полиномиальное кодирование*, которое подходит для порядковых факторов *(см. разд. "Порядковые факторные переменные" далее в этой главе)*. За исключением порядковых факторов, аналитики данных обычно не будут сталкиваться с другими типами кодирования помимо опорного кодирования либо кодировщика с одним активным состоянием.

Многоуровневые факторные переменные

Некоторые факторные переменные могут производить огромное число двоичных фиктивных переменных — почтовые индексы — это образец факторной переменной, и в США имеется 43 тыс. почтовых индексов. В таких случаях полезно обследовать данные, а также связи между предикторными переменными и исходом, чтобы определить, содержится ли полезная информация в категориях. Если да, то далее необходимо решить, имеет ли смысл сохранить все факторы, или же уровни должны быть консолидированы.

В округе Кинг имеется 82 почтовых индекса, связанных с продажей домов:

```
table(house$ZipCode)

 9800 89118 98001 98002 98003 98004 98005 98006 98007 98008 98010 98011
    1     1   358   180   241   293   133   460   112   291    56   163
98014 98019 98022 98023 98024 98027 98028 98029 98030 98031 98032 98033
   85   242   188   455    31   366   252   475   263   308   121   517
98034 98038 98039 98040 98042 98043 98045 98047 98050 98051 98052 98053
  575   788    47   244   641     1   222    48     7    32   614   499
98055 98056 98057 98058 98059 98065 98068 98070 98072 98074 98075 98077
  332   402     4   420   513   430     1    89   245   502   388   204

98092 98102 98103 98105 98106 98107 98108 98109 98112 98113 98115 98116
  289   106   671   313   361   296   155   149   357     1   620   364
98117 98118 98119 98122 98125 98126 98133 98136 98144 98146 98148 98155
  619   492   260   380   409   473   465   310   332   287    40   358
98166 98168 98177 98178 98188 98198 98199 98224 98288 98354
  193   332   216   266   101   225   393     3     4     9
```

Переменная `ZipCode` имеет особое значение, поскольку является эрзацем для эффекта местоположения, влияющего на стоимость дома. Включение в состав всех уровней требует 81 коэффициента, что соответствует 81 степени свободы. Исходная модель `house_lm` имеет всего 5 степеней свободы *(см. разд. "Диагностика модели" ранее в этой главе)*. Кроме того, несколько почтовых индексов имеют всего одну продажу. В некоторых задачах почтовый индекс можно консолидировать при по-

мощи первых двух или трех цифр, что соответствует субметропольной географической области. Для округа Кинг почти все продажи происходят в индексах 980xx или 981xx, так что это не помогает.

Альтернативный подход состоит в группировке почтовых индексов согласно еще одной переменной, такой как продажная цена. Еще лучше будет, если формировать группы с почтовыми индексами, используя остатки от начальной модели. В следующем ниже фрагменте кода `dplyr` консолидируют эти 82 почтовых индекса в пять групп, основываясь на медиане остатка от регрессии `house_lm`:

```
zip_groups <- house %>%
  mutate(resid = residuals(house_lm)) %>%
  group_by(ZipCode) %>%
  summarize(med_resid = median(resid),
            cnt = n()) %>%
  arrange(med_resid) %>%
  mutate(cum_cnt = cumsum(cnt),
         ZipGroup = ntile(cum_cnt, 5))
house <- house %>%
  left_join(select(zip_groups, ZipCode, ZipGroup), by='ZipCode')
```

Медианные остатки вычислены для каждого почтового индекса, и функция `ntile` используется для разбиения почтовых индексов, сортированных по медиане, в пять групп. *См. разд. "Искажающие переменные" далее в этой главе* по поводу примера того, как это используется в качестве члена в уравнении регрессии, который улучшает первоначальную подгонку.

Принцип использования остатков с целью помочь сориентировать подгонку регрессии является фундаментальным шагом в процессе моделирования *(см. разд. "Проверка допущений: диагностика регрессии" далее в этой главе)*.

Порядковые факторные переменные

Некоторые факторные переменные отражают уровни фактора; они называются *порядковыми факторными переменными* или *порядковыми категориальными переменными*. Например, уровень ссуды может быть *A*, *B*, *C* и т. д. — каждый уровень несет бóльший риск, чем предыдущий. Порядковые факторные переменные, как правило, могут быть конвертированы в числовые значения и использоваться как есть. Например, переменная `BldgGrade` — это порядковая факторная переменная. Несколько типов ее уровней приведены в табл. 4.1. Хотя эти уровни имеют определенное значение, ее числовые значения упорядочены снизу вверх, соответствуя домам более высокого уровня. В модели регрессии `house_lm`, подгонка которой была выполнена в *разд. "Множественная линейная регрессия" ранее в этой главе*, `BldgGrade` рассматривалась как числовая переменная.

Рассмотрение порядковых факторов в качестве числовой переменной сохраняет информацию, содержащуюся в упорядоченности, которая будет потеряна при конвертировании в фактор.

Таблица 4.1. *Типичный формат данных*

Значение	Описание
1	Низкобюджетное
2	Ниже среднего
5	Порядочное
10	Очень хорошее
12	Роскошное
13	Особняк

Ключевые идеи для факторных переменных в регрессии

- Факторные переменные нужно конвертировать в числовые переменные для их использования в регрессии.
- Общепринятый метод кодирования факторной переменной с P отличающимися значениями состоит в их представлении с использованием $P-1$ фиктивных переменных.
- Многоуровневая факторная переменная даже в очень больших наборах данных может потребовать консолидации в переменную с меньшим количеством уровней.
- Некоторые факторы имеют упорядоченные уровни и могут быть представлены как одна числовая переменная.

Интерпретация уравнения регрессии

В науке о данных самое важное применение регрессии состоит в предсказании зависимой переменной (исхода). В некоторых случаях, однако, большую роль может сыграть ознакомление непосредственно с самим уравнением для понимания природы связи между предикторами и исходом. В данном разделе приведены ориентиры, касающиеся исследования уравнения регрессии и его интерпретации.

Ключевые термины

Коррелированные предикторные переменные (correlated predictor variables)

Когда предикторные переменные высоко коррелированы, сложно интерпретировать индивидуальные коэффициенты.

Мультиколлинеарность (multicollinearity)

Когда предикторные переменные имеют идеальную или почти идеальную корреляцию, регрессия может быть нестабильной, либо ее невозможно вычислить.

Синоним: коллинеарность.

Искажающие переменные (confounding variables)

Важный предиктор, который при его пропуске приводит к мнимым связям в уравнении регрессии.

Синоним: спутывающие переменные.

Главные эффекты (main effects)

Связь между предикторной переменной и переменной исхода, которая не зависит от других переменных.

Взаимодействия (interactions)

Взаимозависимая связь между двумя или несколькими предикторами и откликом.

Коррелированные предикторы

В множественной регрессии предикторные переменные часто коррелируют друг с другом. В качестве примера предлагаем обследовать коэффициенты регрессии для модели `step_lm`, подогнанной в *разд. "Отбор модели и шаговая регрессия" ранее в этой главе*:

```
step_lm$coefficients
              (Intercept)              SqFtTotLiving
             6.227632e+06               1.865012e+02
                Bathrooms                  Bedrooms
             4.472172e+04              -4.980718e+04
                BldgGrade  PropertyTypeSingle Family
             1.391792e+05               2.332869e+04
    PropertyTypeTownhouse            SqFtFinBasement
             9.221625e+04               9.039911e+00
                  YrBuilt
            -3.592468e+03
```

Коэффициент для `Bedrooms` отрицательный! Это подразумевает, что добавление спальни в дом уменьшит его стоимость. Как это может быть? Это вызвано тем, что предикторные переменные коррелированы: в более крупных домах демонстрируется тенденция наличия больших спален, и именно размер дома управляет его стоимостью, а не число спален. Рассмотрим два дома одного и того же размера: разумно ожидать, что дом с большим числом, но меньшими по площади спальнями будет считаться менее желательным.

Наличие коррелированных предикторов может усложнить интерпретацию знака и значения коэффициентов регрессии (и может раздуть стандартную ошибку оценок). Переменные для спален, размера дома и количества ванных — все они коррелируются. Это иллюстрируется следующим далее примером, в котором выполняется подгонка еще одной регрессии путем удаления переменных `SqFtTotLiving`, `SqFtFinBasement` и `Bathrooms` из уравнения:

```
update(step_tm, . ~ . -SqFtTotLiving - SqFtFinBasement - Bedrooms)
Call:
lm(formuia = AdjSaiePrice ~ Bedrooms + BidgGrade + PropertyType +
    YrBuilt, data = house0, na.action = na.omit)
Coefficients:
              (Intercept)                    Bedrooms
                  4834680                       27657
                BidgGrade PropertyTypeSingie Family
                   245709                      -17604
    PropertyTypeTownhouse                   YrBuilt
                  -47477                       -3161
```

Функция обновления `update` может использоваться для добавления или удаления переменных из модели. Теперь коэффициент для спален положительный — в соответствии с тем, что мы ожидали (хотя он действительно действует как эрзац для размера дома теперь, когда эти переменные были убраны).

Коррелированные переменные являются лишь одной проблемой, связанной с интерпретацией коэффициентов регрессии. В модели `house_lm` нет переменной, которая отчитывается за местоположения дома, и модель смешивает совсем разные типы жилых районов. Местоположение может быть *искажающей* переменной (*см. разд. "Искажающие переменные" далее в этой главе* для последующего обсуждения).

Мультиколлинеарность

Предельный случай коррелированных переменных приводит к мультиколлинеарности — условию, в котором существует избыток среди предикторных переменных. Идеальная мультиколлинеарность случается, когда одна предикторная переменная может быть выражена как линейная комбинация других. Мультиколлинеарность происходит, когда:

- переменная включается в состав модели многократно по ошибке;
- из факторной переменной создаются P фиктивных переменных, вместо $P-1$ *(см. разд. "Факторные переменные в регрессии" ранее в этой главе)*;
- две переменные почти идеально коррелируются друг с другом.

Мультиколлинеарность в регрессии необходимо устранять — переменные следует убирать до тех пор, пока мультиколлинеарность не исчезнет. Регрессия не имеет четко определенного решения в присутствии идеальной мультиколлинеарности. Многие программные пакеты, в том числе R, обрабатывают определенные типы мультиколлинеарности автоматически. Например, если включить переменную `SqFtTotLiving` в регрессию данных `house` дважды, то результаты будут такими же, что и для модели `house_lm`. В случае неидеальной мультиколлинеарности программная система может получить решение, но результаты могут быть нестабильными.

Мультиколлинеарность не представляет какую-то особую проблему для нерегрессионных методов, таких как деревья, кластеризация и ближайшие соседи, и в таких методах может быть желательным сохранять P фиктивных переменных (вместо $P-1$). Тем не менее, даже в этих методах неизбыточность в предикторных переменных по-прежнему остается достоинством.

Искажающие переменные

Что касается коррелированных переменных, то проблема состоит во включении переменных в состав: включение разных переменных, которые имеют аналогичную предсказательную связь с откликом. А вот относительно искажающих переменных проблемой является исключение переменных из состава: важная переменная не включается в уравнение регрессии. Наивная интерпретация коэффициентов уравнения может привести к необоснованным выводам.

Возьмем, например, уравнение регрессии `house_lm` для округа Кинг из *разд. "Пример: данные о жилом фонде округа Кинг" ранее в этой главе*. Коэффициенты регрессии `SqFtLot`, `Bathrooms` и `Bedrooms` — все отрицательные. Исходная модель регрессии не содержит переменную, которая представляла бы местоположение — очень важный предиктор цены на недвижимость. Для того чтобы смоделировать местоположение, включите переменную `ZipGroup`, которая отнесет почтовый индекс в одну из пяти групп от наименее дорогого (1) до самого дорогого (5).[6]

```
lm(AdjSalePrice ~ SqFtTotLiving + SqFtLot +
     Bathrooms + Bedrooms +
     BldgGrade + PropertyType + ZipGroup,
   data=house, na.action=na.omit)

Coefficients:
              (Intercept)             SqFtTotLiving
               -6.709e+05                 2.112e+02
                  SqFtLot                Bathrooms
                 4.692e-01                 5.537e+03
                  Bedrooms                BldgGrade
                -4.139e+04                 9.893e+04
PropertyTypeSingle Family     PropertyTypeTownhouse
                 2.113e+04                -7.741e+04
                 ZipGroup2                 ZipGroup3
                 5.169e+04                 1.142e+05
                 ZipGroup4                 ZipGroup5
                 1.783e+05                 3.391e+05
```

[6] В округе Кинг имеется 82 почтовых индекса, по некоторым из которых зарегистрировано всего несколько продаж. Как альтернатива прямому использованию почтового индекса в качестве факторной переменной, `ZipGroup` кластеризует подобные почтовые индексы в одну группу. Подробности *см. в разд. "Многоуровневые факторные переменные ранее в этой главе*.

Несомненно, переменная ZipGroup очень важна: дом в самой дорогой группе почтовых индексов оценивается как имеющий продажную цену выше почти на 340 тыс. долларов. Коэффициенты SqFtLot и Bathrooms теперь положительны, и добавление ванной увеличивает продажную цену на 7500 долларов.

Коэффициент для Bedrooms по-прежнему отрицательный. Хотя это выглядит нелогичным, данный феномен известен, и связан он с недвижимостью. Наличие у домов одинаковой общей жилой площади и большего количества и, следовательно, меньших по размеру спален ассоциируется с менее ценными домами.

Взаимодействия и главные эффекты

Специалисты в области статистики предпочитают различать главные эффекты, или независимые переменные, от взаимодействий между главными эффектами. Главные эффекты — это то, что называется предикторными переменными в уравнении регрессии.

Неявное предположение, когда в модели используются только главные эффекты, состоит в том, что связь между предикторной переменной и откликом не зависит от других предикторных переменных. Зачастую это не так.

Например, модель, подогнанная к данным о жилом фонде округа Кинг в *разд. "Искажающие переменные" ранее в этой главе* включает несколько переменных в качестве главных эффектов, в том числе ZipCode. Местоположение в торговле недвижимостью — краеугольный камень, и естественно предположить, что связь, скажем, между размером дома и продажной ценой зависит от местоположения. Большой дом, построенный в районе с низкой арендной платой, не будет сохранять одинаковую стоимость, что и большой дом, построенный в дорогом районе. В R взаимодействия между переменными можно включить в расчеты, используя оператор *. Для данных округа Кинг представленный далее пример выполняет подгонку взаимодействия между SqFtTotLiving и ZipGroup:

```
lm(AdjSalePrice ~ SqFtTotLiving*ZipGroup + SqFtLot +
     Bathrooms + Bedrooms + BldgGrade + PropertyType,
   data=house, na.action=na.omit)

Coefficients:
              (Intercept)              SqFtTotLiving
                -4.919e+05                  1.176e+02
                 ZipGroup2                  ZipGroup3
                -1.342e+04                  2.254e+04
                 ZipGroup4                  ZipGroup5
                 1.776e+04                 -1.555e+05
                   SqFtLot                  Bathrooms
                 7.176e-01                 -5.130e+03
                  Bedrooms                  BldgGrade
                -4.181e+04                  1.053e+05
PropertyTypeSingle Family      PropertyTypeTownhouse
                 1.603e+04                 -5.629e+04
```

```
SqFtTotLiving:ZipGroup2    SqFtTotLiving:ZipGroup3
              3.165e+01                  3.893e+01
SqFtTotLiving:ZipGroup4    SqFtTotLiving:ZipGroup5
              7.051e+01                  2.298e+02
```

У результирующей модели четыре новых члена: `SqFtTotLiving:ZipGroup2`, `SqFtTotLivlng:ZlpGroup3` и т. д.

Расположение и размер дома, похоже, имеют сильное взаимодействие. Для дома в самой низкой группе `ZipGroup` наклон прямой такой же, что и наклон для главного эффекта `SqFtTotLiving`, который составляет 177 долларов за кв. фут (это вызвано тем, что R использует *опорное* кодирование для факторных переменных; *см. разд. "Факторные переменные в регрессии" ранее в этой главе*). Для дома в самой высокой группе `ZipGroup` наклон является суммой главного эффекта плюс `SqFtTotLiving:ZipGroup5`, или $177 + 230 = 447$ долларов за кв. фут. Другими словами, добавление квадратного фута в группе наиболее дорогих почтовых индексов повышает предсказанную продажную цену почти в 2,7 раза по сравнению с повышением в группе наименее дорогих почтовых индексов.

Отбор модели при помощи членов взаимодействия

В задачах, сопряженных со многими переменными, может оказаться сложным принять решение, какие члены уравнения, характеризующие взаимодействия между переменными, должны быть включены в модель. Чаще всего принимается несколько разных подходов.

- В некоторых задачах предварительные знания и интуиция могут служить ориентиром для выбора того, какие члены взаимодействия включать в модель.
- Пошаговый отбор *(см. разд. "Отбор модели и шаговая регрессия" ранее в этой главе)* может использоваться для отсеивания различных моделей.
- Штрафная регрессия может автоматически выполнять подгонку к большому набору возможных членов взаимодействия.

По-видимому, самым общепринятым подходом является использование *древовидных моделей*, а также их потомков, *случайного леса* и градиентно-бутстированных деревьев. Данный класс моделей автоматически выполняет поиск оптимальных членов взаимодействия; *см. разд. "Древовидные модели" главы 6*.

Ключевые идеи для интерпретации уравнения регрессии

- Вследствие корреляции между предикторами необходимо проявлять осторожность в интерпретации коэффициентов в множественной линейной регрессии.
- Мультиколлинеарность может вызвать числовую нестабильность в подгонке уравнения регрессии.
- Искажающая переменная является важным предиктором, который не учтен в модели и может привести к уравнению регрессии с мнимыми связями.

- Член уравнения, характеризующий взаимодействия между двумя переменными, необходим, если связь между переменными и откликом является взаимозависимой.

Проверка допущений: диагностика регрессии

В объяснительном моделировании (т. е. в контексте исследования) принимаются различные шаги в дополнение к упомянутым ранее метрическим показателям *(см. разд. "Диагностика модели" ранее в этой главе)*, чтобы диагностировать, насколько хорошо модель подогнана к данным. Большинство основывается на анализе остатков, который может выполнить проверку допущений, лежащих в основе модели. Эти шаги непосредственно не решают вопрос предсказательной точности, но они могут обеспечить полезные сведения о настройке предсказания.

Ключевые термины

Стандартизированные остатки (standardized residuals)

Остатки, деленные на стандартную ошибку остатков.

Выбросы (Outliers)

Записи (или значения исхода), которые находятся на удалении от остальной части данных (или предсказанного исхода).

Влиятельное значение (influential value)

Значение, или запись, присутствие или отсутствие которого имеет большое значение в уравнении регрессии.

Плечо (leverage)

Степень влиятельности, которую одиночная запись имеет на уравнение регрессии.

Синонимы: hat-значение, диагональ в проекционной матрице.

Ненормальные остатки (non-normal residuals)

Ненормально распределенные остатки могут аннулировать некоторые технические условия регрессии; вместе с тем они обычно не являются предметом озабоченности в науке о данных.

Гетероскедастичность (heteroskedasticity)

Ситуация, когда некоторые диапазоны исхода показывают остатки с более высокой дисперсией (что может говорить о предикторе, который в уравнении отсутствует).

Графики частных остатков (partial residual plots)

Диагностический график для выявления связи между переменной исхода и одиночным предиктором.

Синоним: график с добавленными переменными.

Выбросы

Вообще говоря, предельное значение, так называемый *выброс*, — это такое значение, которое находится далеко от большинства других наблюдений. Подобно тому, как с выбросами приходится управляться для получения оценок центрального положения и вариабельности *(см. разд. "Оценки центрального положения" и "Оценки вариабельности" главы 1)*, выбросы могут вызывать проблемы с моделями регрессии. В регрессии выброс — это запись, фактическое значение *y* которой находится далеко от предсказанного значения. Выбросы можно обнаружить путем обследования *стандартизированного остатка*, т. е. остатка, деленного на стандартную ошибку остатков.

Нет статистической теории, которая отделяет выбросы от невыбросов. Вместо нее существуют (произвольные) эмпирические правила в отношении того, насколько далеко от основной части данных должно находиться наблюдение, чтобы его можно было назвать выбросом. Например, в коробчатой диаграмме выбросами являются те точки данных, которые находятся слишком высоко или слишком низко от границ коробки *(см. разд. "Процентили и коробчатые диаграммы" главы 1)*, где "слишком" означает величину, превышающую "1,5 умножить на межквартильный размах". В регрессии метрический показатель стандартизированного остатка, как правило, используется для определения, не относится ли запись к категории выбросов. Стандартизированные остатки можно интерпретировать, как "число стандартных ошибок от прямой регрессии".

Давайте выполним подгонку регрессии к данным о продажах домов в округе Кинг для всех продаж в почтовом индексе 98105:

```
house_98105 <- house[house$ZipCode == 98105,]
lm_98105 <- lm(AdjSalePrice ~ SqFtTotLiving + SqFtLot + Bathrooms +
                Bedrooms + BldgGrade, data=house_98105)
```

Мы извлекаем стандартизированные остатки при помощи функции `rstandard` и получаем индекс наименьшего остатка при помощи функции `order`:

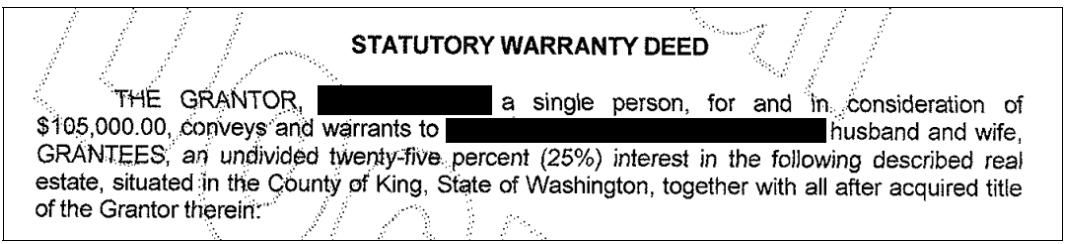
```
sresid <- rstandard(lm_98105)
idx <- order(sresid)
sresid[idx[1]]
    20431
-4.326732
```

Самая большая завышенная оценка из модели составляет более четырех стандартных ошибок выше прямой регрессии, что соответствует переоценке в 757 753 долларов. Исходная запись данных, которая соответствует этому выбросу, следующая:

```
house_98105[idx[1], c('AdjSalePrice', 'SqFtTotLiving', 'SqFtLot',
                      'Bathrooms', 'Bedrooms', 'BldgGrade')]

AdjSalePrice SqFtTotLiving SqFtLot Bathrooms Bedrooms BldgGrade
       (dbl)         (int)   (int)     (dbl)    (int)     (int)
1     119748          2900    7276         3        6         7
```

В данном случае, по всей видимости, что-то не так с записью: дом такого размера, как правило, продается за намного бóльшую цену, чем 119 748 долларов в этом почтовом индексе. На рис. 4.4 показана выдержка из установочного акта этой продажи: ясно, что продажа была связана всего лишь с долей в собственности. В этом случае выброс соответствует продаже, которая является аномальной и не должна быть включена в регрессию. Кроме того, выбросы могут быть результатом других проблем, таких как ввод данных из-за "толстого пальца" или несоответствие единиц измерения (например, сообщение о продаже в тысячах долларов вместо просто долларов).



STATUTORY WARRANTY DEED

THE GRANTOR, [redacted] a single person, for and in consideration of $105,000.00, conveys and warrants to [redacted] husband and wife, GRANTEES, an undivided twenty-five percent (25%) interest in the following described real estate, situated in the County of King, State of Washington, together with all after acquired title of the Grantor therein:

Рис. 4.4. Установочный акт для самого большого отрицательного остатка

Для задач на основе больших данных выбросы обычно не представляют проблему в подгонке регрессии, которая будет использоваться в предсказании новых данных. Однако выбросы занимают центральное место в обнаружении аномалий, где отыскание выбросов является смыслом всего дела. Кроме того, выброс может соответствовать эпизоду мошенничества или случайного действия. Так или иначе, обнаружение выбросов может быть критической потребностью бизнеса.

Влиятельные значения

Значение, отсутствие которого существенно изменит уравнение регрессии, называется *влиятельным наблюдением*. В регрессии такое значение не обязательно должно быть связано с большим остатком. В качестве примера рассмотрим прямую регрессии на рис. 4.5. Пунктирная прямая соответствует регрессии со всеми данными, тогда как жирная прямая соответствует регрессии с точкой в удаленном правом верхнем углу. Безусловно, это значение данных имеет огромное влияние на регрессию, хотя оно не связано с большим выбросом (удаленном от полной регрессии). Считается, что это значение данных имеет в регрессии сильное *плечо* (leverage).

В дополнение к стандартизированным остаткам *(см. разд. "Выбросы" ранее в этой главе)*, статистики разработали несколько метрических показателей для определения влиятельности одиночной записи на регрессию. Общепринятой мерой плеча является hat-значение; значения выше $2(P+1)/n$ говорят о значении данных с высоким плечом[7].

[7] Термин "hat-значение" происходит от понятия hat-матрицы, т. е. проекционной матрицы с проекцией на пространство регрессоров. Множественная линейная регрессия может быть выражена формулой $Y = \mathbf{H}Y$, где **H** — это hat-матрица. Hat-значения соответствуют диагонали в **H**.

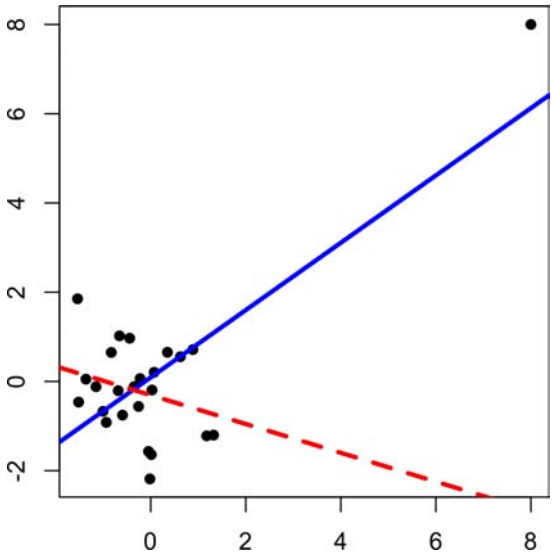


Рис. 4.5. Пример влиятельной точки данных в регрессии

Еще один метрический показатель — *расстояние Кука* (Cook's distance), которое характеризует влияние как комбинацию плеча и размера остатка. Эмпирическое правило состоит в том, что наблюдение имеет высокое влияние, если расстояние Кука превышает $4/(n-P-1)$.

График влиятельности, или *пузырьковый график*, объединяет стандартизированные остатки, hat-значение и расстояние Кука в одном графике. На рис. 4.6 показан

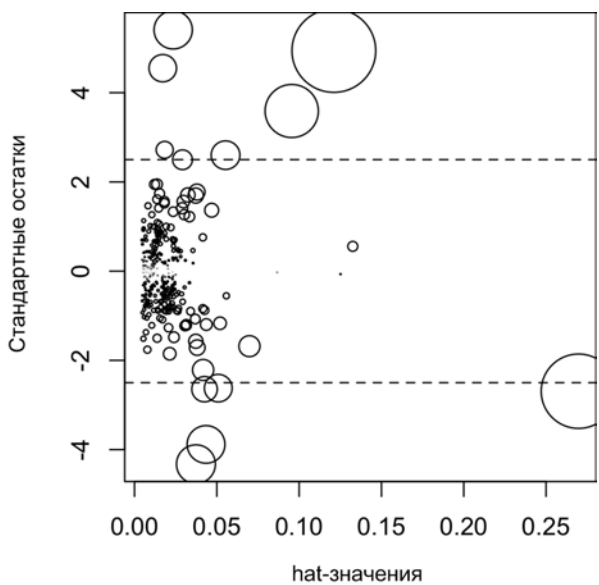


Рис. 4.6. График для установления, какие наблюдения имеют высокое влияние

график влиятельности для данных о жилом фонде округа Кинг. Данный график может быть создан при помощи следующего ниже фрагмента кода на R.

```
std_resid <- rstandard(lm_98105)
cooks_D <- cooks.distance(lm_98105)
hat_values <- hatvalues(lm_98105)
plot(hat_values, std_resid, cex=10*sqrt(cooks_D))
abline(h=c(-2.5, 2.5), lty=2)
```

Судя по всему, имеется несколько точек данных, которые оказывают большое влияние в данной регрессии. Расстояние Кука можно вычислить при помощи функции `cooks.distance`, и вы можете использовать `hatvalues` для вычисления диагностического показателя. Hat-значения отображены на оси *x*, остатки — на оси *y*, и размер точек связан со значением расстояния Кука.

В табл. 4.2 приведено сравнение регрессии с полным набором данных и с исключенными очень влиятельными точками данных. Коэффициент регрессии для `Bathrooms` изменяется вполне заметно[8].

Таблица 4.2. *Сравнение коэффициентов регрессии с полными данными и с убранными влиятельными данными*

	Оригинал	**Влиятельные убраны**
(пересечение)	–772 550	–647 137
`SqFtTotLiving`	210	230
`SqFtLot`	39	33
`Bathrooms`	2282	–16132
`Bedrooms`	–26320	–22 888
`BldgGrade`	130 000	114 871

В целях подгонки регрессии, которая надежно предсказывает будущие данные, идентификация влиятельных наблюдений полезна только в меньших по размеру наборах данных. Для регрессий, сопряженных с большим числом записей, маловероятно, что какое-либо наблюдение будет нести достаточный вес, чтобы вызывать предельное влияние на подогнанное уравнение (хотя регрессия может по-прежнему иметь большие выбросы). В целях обнаружения аномалии, тем не менее, идентификация влиятельных наблюдений может быть очень полезной.

[8] Коэффициент для `Bathrooms` становится отрицательным, что нелогично. Местоположение не было принято во внимание, и почтовый индекс 98105 содержит районы несопоставимых типов домов. *См. разд. "Искажающие переменные" ранее в этой главе* по поводу обсуждения искажающих переменных.

Гетероскедастичность, ненормальность и коррелированные ошибки

В статистике значительное внимание уделяется распределению остатков. Оказывается, что обычные наименьшие квадраты *(см. разд. "Наименьшие квадраты" ранее в этой главе)* являются несмещенными и в некоторых случаях единственными "оптимальными" критериями оценки на основе широкого спектра допущений о характере распределения. Это означает, что в большинстве задач аналитикам данных не приходится слишком беспокоиться о характере распределения остатков.

Распределение остатков релевантно главным образом для подтверждения достоверности формального статистического вывода (проверки гипотез и *p*-значения), который имеет минимальное значение для аналитиков данных, озабоченных главным образом предсказательной точностью. Для того чтобы формальный вывод был полноценным, принимается допущение, что остатки нормально распределены, имеют такую же дисперсию и независимы. Одной из областей, где это может представлять интерес для аналитиков науки о данных, является стандартное вычисление доверительных интервалов для предсказанных значений, которые основаны на допущениях о характере остатков *(см. разд. "Доверительные и предсказательные интервалы" ранее в этой главе)*.

Гетероскедастичность — это отсутствие постоянной остаточной дисперсии по всему диапазону предсказанных значений. Другими словами, для некоторых частей диапазона ошибки больше, чем для других. Программный пакет `ggplot2` располагает несколькими удобными инструментами для анализа остатков.

Следующий ниже фрагмент кода выводит график с абсолютными остатками против предсказанных значений для регрессии `lm_98105`, подогнанной в *разд. "Выбросы" ранее в этой главе*.

```
df <- data.frame(
  resid = residuals(lm_98105),
  pred = predict(lm_98105))
ggplot(df, aes(pred, abs(resid))) +
  geom_point() +
  geom_smooth()
```

На рис. 4.7 представлен результирующий график. При помощи `geom_smooth` очень легко наложить сглаженную из абсолютных остатков кривую. Данная функция вызывает метод `loess` для создания визуальной сглаженной кривой для оценки связи между переменными на осях *x* и *y* в диаграмме рассеяния *(см. врезку "Сглаживатели для диаграмм рассеяния" далее в этой главе)*.

Совершенно очевидно, что дисперсия остатков имеет тенденцию увеличиваться для домов с более высокой стоимостью, но является также большой для домов с более низкой стоимостью. Данный график говорит о том, что регрессия `lm_98105` имеет *гетероскедастичные* ошибки.

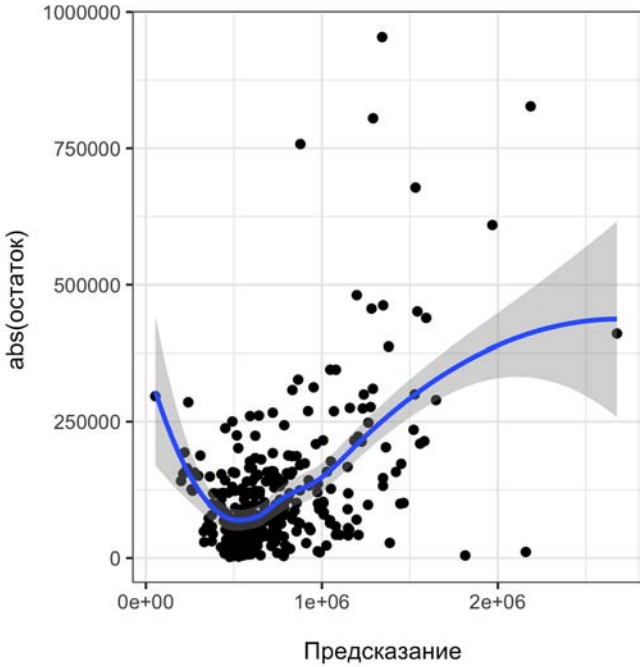


Рис. 4.7. График абсолютного значения остатков против предсказанных значений

Почему аналитик данных должен интересоваться гетероскедастичностью?

Гетероскедастичность говорит о том, что ошибки предсказания отличаются для разных диапазонов предсказанного значения и могут свидетельствовать о неполной модели. Например, гетероскедастичность в `lm_98105` может говорить о том, что регрессия что-то не учла в домах с высоким и низким диапазонами.

На рис. 4.8 приведена гистограмма стандартизированных остатков для регрессии `lm_98105`. Ее распределение однозначно имеет более длинные хвосты, чем у нормального распределения, и показывает умеренную асимметричность в сторону более крупных остатков.

Специалисты в области статистики могут также проверить допущение, что ошибки независимы. Это особенно верно в отношении данных, которые собираются в течение долгого времени. Статистика *Дурбина — Уотсона* (Durbin — Watson) может использоваться для обнаружения того, существует ли значительная автокорреляция в регрессии, сопряженной с данными временно́го ряда.

Хотя регрессия может нарушить одно из допущений о характере распределения, должно ли это нас заботить? В большинстве случаев в науке о данных главным объектом интереса является предсказательная точность, и поэтому какой-то анализ гетероскедастичности не помешает. Вы можете обнаружить, что в данных имеется некий сигнал, который ваша модель не охватила. Однако удовлетворение допущений о характере распределения просто ради подтверждения достоверности фор-

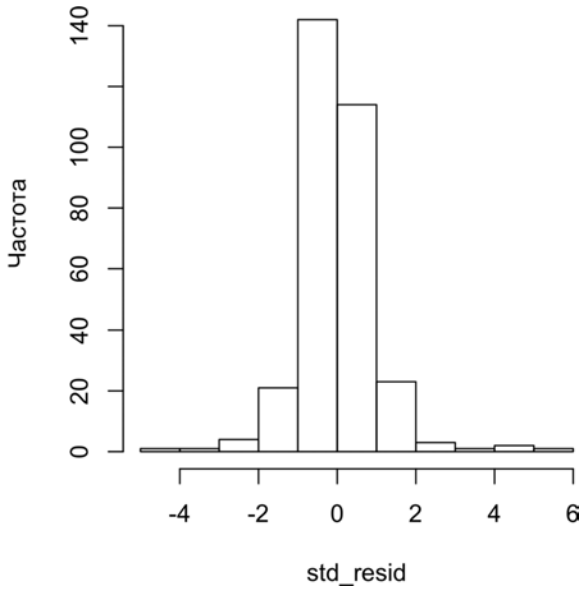


Рис. 4.8. Гистограмма остатков от регрессии данных о жилом фонде

мального статистического вывода (*p*-значения, *F*-статистики и т. д.) не представляет для аналитика данных какую-то особую важность.

Сглаживатели диаграмм рассеяния

Регрессия — это прежде всего моделирование связи между откликом и предикторными переменными. В оценивании регрессионной модели полезно использовать *сглаживатель диаграмм рассеяния* (scatterplot smoother), чтобы визуально высветить связи между двумя переменными.

Например, на рис. 4.7 сглаженная связь между абсолютными остатками и предсказанным значением показывает, что дисперсия остатков зависит от значения остатка. В этом случае использовалась функция `loess`; она работает путем неоднократной подгонки серии локальных регрессий к смежным подмножествам, чтобы выработать сглаженную. Хотя `loess`, вероятно, является общепринятым сглаживателем, в R имеются и другие сглаживатели, такие как суперсглаживатель (`supsmu`) и ядерный сглаживатель (`ksmooth`). Для оценивания регрессионной модели, как правило, нет надобности беспокоиться по поводу деталей сглаживателей диаграмм рассеивания.

Графики частных остатков и нелинейность

Графики частных остатков — это способ визуализации того, насколько хорошо вычисленная подгонка объясняет связь между предиктором и исходом. Вместе с обнаружением выбросов это, вероятно, самый важный диагностический показатель для аналитиков данных. Основная идея графика частных остатков состоит в изолировании связи между предикторной переменной и откликом, *принимая во внимание все другие предикторные переменные*. Частный остаток можно представить, как

значение "синтетического исхода", комбинируя предсказание на основе одиночного предиктора с фактическим остатком от полного уравнения регрессии. Частный остаток для предиктора X_i — это обычный остаток плюс связанный с X_i член регрессии.

$$\text{Частный остаток} = \text{Остаток} + \hat{b}_i X_i,$$

где $\hat{b}_i$ — это оценочный коэффициент регрессии. Функция `predict` в R имеет возможность возвращать индивидуальные члены регрессии $\hat{b}_i X_i$:

```
terms <- predict(lm_98105, type='terms')
partial_resid <- resid(lm_98105) + terms
```

График частных остатков отображает X_i на оси *x* и частные остатки — на оси *y*. Использование программного пакета `ggplot2` упрощает наложение сглаженной из частных остатков.

```
df <- data.frame(SqFtTotLiving = house_98105[, 'SqFtTotLiving'],
                 Terms = terms[, 'SqFtTotLiving'],
                 PartialResid = partial_resid[, 'SqFtTotLiving'])
ggplot(df, aes(SqFtTotLiving, PartialResid)) +
  geom_point(shape=1) + scale_shape(solid = FALSE) +
  geom_smooth(linetype=2) +
  geom_line(aes(SqFtTotLiving, Terms))
```

Результирующий график показан на рис. 4.9. Частный остаток — это оценка вклада, который `SqFtTotLiving` вносит в продажную цену. Легко видно, что связь между

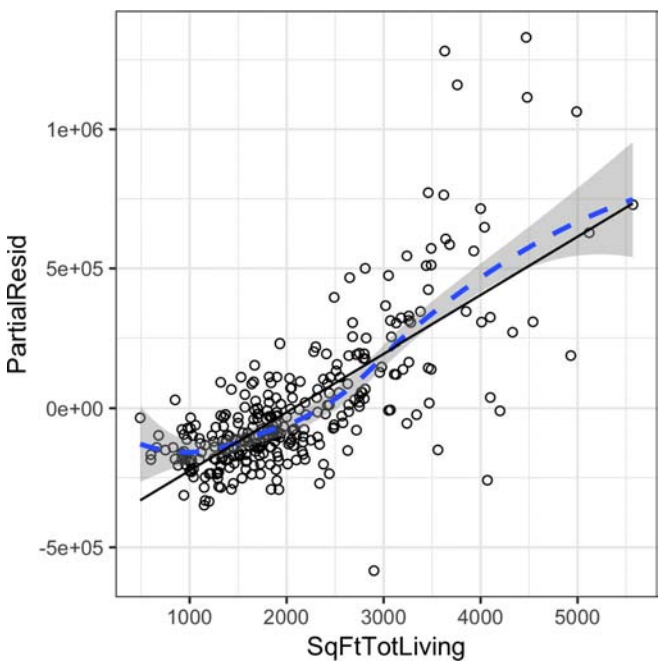


Рис. 4.9. График частных остатков для переменной `SqFtTotLiving`

SqFtTotLiving и продажной ценой не линейна. Прямая регрессии недооценивает продажную цену для домов площадью менее 1000 кв. футов и переоценивает цену для домов с площадью между 2000 и 3000 кв. футов. Выше 4000 кв. футов имеется слишком мало точек данных, чтобы сделать выводы для этих домов.

В данном случае эта нелинейность имеет смысл: добавление 500 футов в малом доме имеет гораздо бóльшую разницу, чем добавление 500 футов в большом доме. Это свидетельствует о том, что для SqFtTotLiving вместо простого линейного члена следует рассмотреть нелинейный член *(см. следующий раздел)*.

Ключевые идеи для проверки допущений

- Хотя выбросы могут вызывать проблемы для небольших наборов данных, первостепенный интерес к выбросам состоит в том, чтобы идентифицировать проблемы с данными или локализовать аномалии.
- Одиночные записи (включая регрессионные выбросы) могут иметь большое влияние на уравнение регрессии с небольшими данными, но этот эффект размывается в больших данных.
- Если регрессионная модель используется для формального вывода (*p*-значения и т. п.), то должна быть выполнена сверка определенных допущений о характере распределения остатков. В целом, однако, распределение остатков не является критически важным в науке о данных.
- График частных остатков может использоваться для качественной диагностики подгонки для каждого члена регрессии, возможно, приводя к спецификации альтернативной модели.

Нелинейная регрессия

Связь между откликом и предикторной переменной не обязательно линейна. Отклик на дозу препарата часто нелинеен: удвоение дозы обычно не приводит к удвоенному отклику. Спрос на продукт не является линейной функцией маркетинга расходования денег, поскольку в какой-то момент спрос, вероятно, будет удовлетворен. Существует несколько способов, которыми регрессия может быть расширена для получения этих нелинейных эффектов.

Ключевые термины

Параболическая регрессия (polynomial regression)

Добавляет в регрессию полиномиальные члены (квадраты, кубы и т. д.).

Синоним: полиномиальная регрессия.

Сплайновая регрессия (spline regression)

Подгонка гладкой кривой с серией полиномиальных сегментов.

Узлы (knots)

Значения, которые отделяют сплайновые сегменты.

Обобщенные аддитивные модели (generalized additive models)

Сплайновые модели с автоматизированным выбором узлов.

Нелинейная регрессия

Когда статистики говорят о *нелинейной регрессии*, они всегда ссылаются на модели, которые не могут быть подогнаны при помощи наименьших квадратов. Какие модели не являются нелинейными? По существу, это все модели, где отклик не может быть выражен как линейная комбинация предикторов или некая трансформация предикторов. Нелинейные регрессионные модели более жесткие и вычислительно более емкие для выполнения подгонки, поскольку они требуют численной оптимизации. По этой причине, если это возможно, предпочтение отдается использованию линейной модели.

Параболическая регрессия

Параболическая, или *полиномиальная*, *регрессия* связана с включением в состав уравнения регрессии полиномиальных членов. Использование параболической регрессии практически началось с разработки непосредственно самой регрессии в статье Жергонна (Gergonne) в 1815 г. Например, квадратичная регрессия между откликом Y и предиктором X примет следующую форму:

$$Y = b_0 + b_1 X + b_2 X^2 + e.$$

Параболическая регрессия может быть подогнана в R посредством функции `poly`. Например, представленный далее фрагмент кода выполняет подгонку квадратичного полинома для `SqFtTotLiving` с данными о жилом фонде округа Кинг:

```
lm(AdjSalePrice ~ poly(SqFtTotLiving, 2) + SqFtLot +
                BldgGrade + Bathrooms + Bedrooms,
                    data=house_98105)

Call:
lm(formula = AdjSalePrice ~ poly(SqFtTotLiving, 2) + SqFtLot +
    BldgGrade + Bathrooms + Bedrooms, data = house_98105)

Coefficients:

            (Intercept)  poly(SqFtTotLiving, 2)1
             -402530.47               3271519.49
poly(SqFtTotLiving, 2)2                  SqFtLot
              776934.02                    32.56
             BldgGrade                Bathrooms
              135717.06                -1435.12
              Bedrooms
               -9191.94
```

Теперь с `SqFtTotLiving` связано два коэффициента: один для линейного члена и другой для квадратичного члена (члена второй степени).

График частных остатков *(см. разд. "Графики частных остатков и нелинейность" ранее в этой главе)* говорит о некоторой кривизне в уравнении регрессии, связанном с `SqFtTotLiving`. Подогнанная линия более точно соответствует сглаженной *(см. следующий раздел)* из частных остатков по сравнению с линейной подгонкой (рис. 4.10).

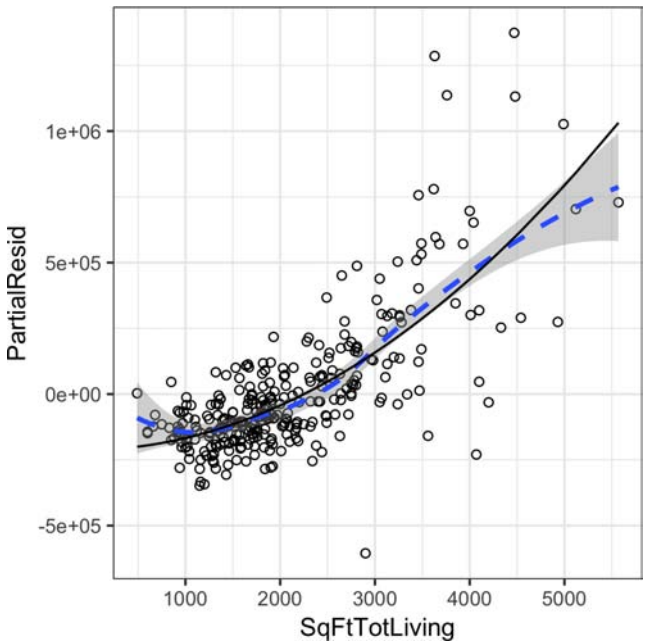


Рис. 4.10. Параболическая регрессия, подогнанная для переменной `SqFtTotLiving` (сплошная линия) против сглаженной (жирная линия; смотрите следующий раздел о сплайнах)

Сплайновая регрессия

Параболическая регрессия захватывает только определенное количество кривизны в нелинейной связи. Добавление членов более высоких степеней, таких как кубический биквадратный полином, часто приводит к нежелательной "волнистости" в уравнении регрессии. Альтернативный, и часто превосходящий, подход к моделированию нелинейных связей состоит в использовании *сплайнов*. Сплайны предоставляют способ гладко интерполировать между фиксированными точками. Сплайны первоначально использовались чертежниками для нанесения плавной кривой, в частности, в судостроении и самолетостроении.

Сплайны создавались путем деформирования тонкого куска дерева при помощи гирь, которые назывались "утками" (рис. 4.11).

Рис. 4.11. Сплайны первоначально создавались на основе деформируемой древесины и "уток" и использовались в качестве инструмента чертежника для подгонки кривых. Фото с разрешения Боба Перри (Bob Perry)

Техническое определение сплайна является серией кусочно-непрерывных полиномов. Впервые они были разработаны во время Второй мировой войны на Абердинском испытательном полигоне в США румынским математиком Дж. Шенбергом (J. Schoenberg). Полиномиальные части гладко соединялись в серии фиксированных точек в предикторной переменной, которые назывались *узлами*. Формулировка для сплайнов намного более сложнее, чем параболическая регрессия; подробностями подгонки сплайнов обычно занимаются статистические программные системы. Пакет `splines` в R содержит функцию `bs` для создания в модели регрессии члена с *B-сплайном*, т. е. совокупностью последовательных полиномиальных кривых. Например, следующий ниже фрагмент кода добавляет член B-сплайна к модели регрессии дома:

```
library(splines)
knots <- quantile(house_98105$SqFtTotLiving, p=c(.25, .5, .75))
lm_spline <- lm(AdjSalePrice ~ bs(SqFtTotLiving, knots=knots, degree=3) +
  SqFtLot + Bathrooms + Bedrooms + BldgGrade, data=house_98105)
```

Необходимо определить два параметра: степень полинома и расположение узлов. В этом случае предиктор `SqFtTotLiving` включается в модель при помощи кубического сплайна (`degree=3`). По умолчанию `bs` помещает узлы на границах; кроме того, узлы также были помещены в нижний, медианный и верхний квартили.

В отличие от линейного члена, для которого коэффициент имеет прямое значение, коэффициенты для сплайнового члена не интерпретируемы. Вместо этого полезнее использовать визуальное отображение для выявления природы сплайновой подгонки. На рис. 4.12 показан график зависимости частных остатков от регрессии. В отличие от полиномиальной модели, сплайновая модель намного ближе соответствует сглаженной, демонстрируя бóльшую гибкость сплайнов. В этом случае линия подогнана к данным намного ближе. Означает ли это, что сплайновая регрессия

является более хорошей моделью? Не обязательно. С экономической точки зрения нет никакого смысла в том, чтобы очень небольшие дома (площадью менее 1000 кв. футов) имели более высокую стоимость, чем дома немного большего размера. Это, возможно, артефакт искажающей переменной *(см. разд. "Искажающие переменные" ранее в этой главе)*.

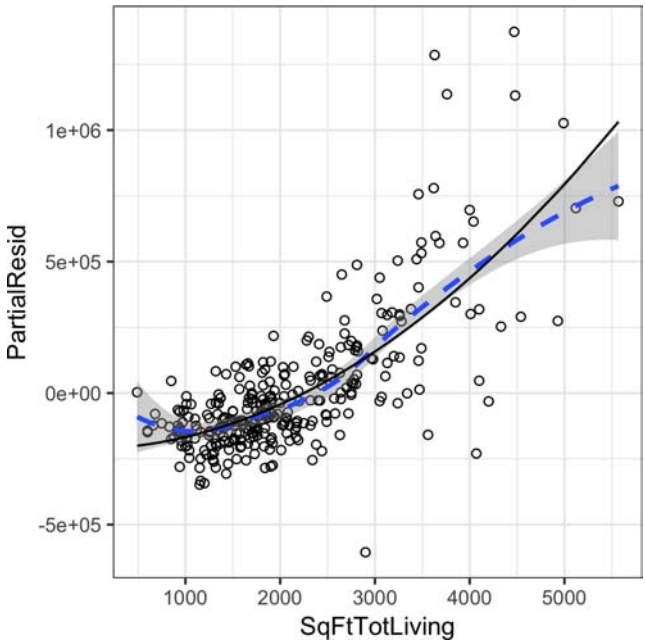


Рис. 4.12. Сплайновая регрессия, подогнанная для переменной `SqFtTotLiving` (сплошная линия) по сравнению со сглаженной (пунктирная линия)

Обобщенные аддитивные модели

Предположим, что вы заподозрили нелинейную связь между откликом и предикторной переменной в силу априорного знания либо вследствие обследования диагностических показателей регрессии. Полиномиальные члены могут быть недостаточно гибкими для захвата связи, а сплайновые члены требуют определения узлов. Обобщенные аддитивные модели (generalized additive models, GAM) — это специальный метод, предназначенный для автоматической подгонки сплайновой регрессии. Программный пакет `gam` в R может использоваться для подгонки модели GAM к данным жилого фонда:

```
library(mgcv)
lm_gam <- gam(AdjSalePrice ~ s(SqFtTotLiving) + SqFtLot +
                      Bathrooms + Bedrooms + BldgGrade,
                      data=house_98105)
```

Член `s(SqFtTotLiving)` говорит функции `gam` найти "наилучшие" узлы для сплайнового члена уравнения (рис. 4.13).

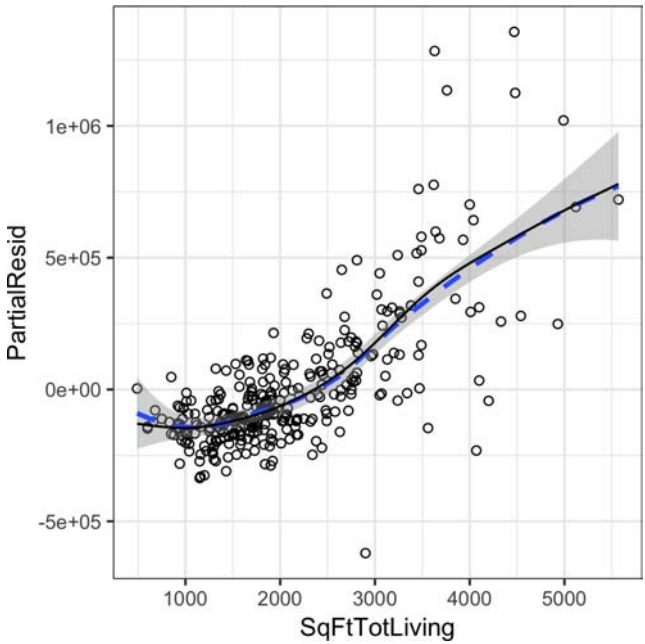


Рис. 4.13. Регрессия GAM, подогнанная для переменной `SqFtTotLiving` (сплошная линия) по сравнению со сглаженной (пунктирная линия)

Ключевые идеи для нелинейной регрессии

- Выбросы в регрессии — это записи с большим остатком.
- Мультиколлинеарность может вызвать числовую нестабильность в подгонке уравнения регрессии.
- Искажающая переменная — это важный предиктор, который упущен из модели и может привести к уравнению регрессии с мнимыми связями.
- Член уравнения, характеризующий взаимодействия между двумя переменными, необходим, если эффект одной переменной зависит от *уровня* другой.
- Параболическая регрессия может подгонять нелинейные связи между предикторами и переменной исхода.
- Сплайны — это серия нанизанных полиномиальных сегментов, присоединенных в узлах.
- Обобщенные аддитивные модели (GAM) автоматизируют процесс определения узлов в сплайнах.

Дополнительные материалы для чтения

Подробнее о сплайновых моделях и GAM см. книгу "Элементы статистического обучения" (Hastie T., Tibshirani R., Friedman J. Elements of Statistical Learning. — 2nd ed. — Springer, 2009) и ее более краткий вариант на основе R "Введение в статистическое обучение" (Gareth J., Witten D., Hastie T., Tibshirani R. An Introduction to Statistical Learning. — Springer, 2013).

Резюме

Возможно, никакой другой статистический метод не видел более широкого применения на протяжении многих лет, чем регрессия — процесс установления связи между многочисленными предикторными переменными и переменной исхода. Ее фундаментальная форма линейна: каждая предикторная переменная имеет коэффициент, который описывает линейную связь между предиктором и исходом. Более усовершенствованные формы регрессии, такие как параболическая и сплайновая регрессия, допускают нелинейность связи. В классической статистике главный упор делается на отыскании хорошей подгонки к наблюдаемым данным, чтобы объяснить или описать какое-либо явление, и сила этой подгонки зависит от того, как используются традиционные ("внутривыборочные") метрические показатели для диагностики модели. В науке о данных, в отличие от этого, как правило, цель состоит в том, чтобы предсказывать значения для новых данных, поэтому используются метрические показатели, основанные на предсказательной точности для вневыборочных данных. Кроме того, применяются методы отбора переменных с целью уменьшить размерность и создать более компактные модели.

Классификация

Аналитики данных часто сталкиваются с проблемой, которая требует автоматизированного решения. Является ли электронное письмо попыткой фишинга? Перейдет ли клиент к другому поставщику? Нажмет ли интернет-пользователь на рекламе? Все эти вопросы относятся к задачам *классификации*. Классификация — это, возможно, самая важная форма предсказания: ее цель состоит в том, чтобы предсказать, является ли запись нулем или единицей (фишинг/не фишинг, нажмет/не нажмет, перейдет/не перейдет) либо в некоторых случаях одной из нескольких категорий (например, фильтрация ваших входящих сообщений в Gmail на "основные", "соцсети", "промоакции" или "форумы").

Зачастую нам требуется больше, чем простая бинарная классификация: мы хотим знать предсказательную вероятность, что конкретный случай принадлежит классу.

Вместо того чтобы иметь модель, которая просто назначает бинарную категорию классифицируемой записи, большинство алгоритмов могут возвращать балльную оценку вероятности (склонность) принадлежать целевому классу. На деле, если вести речь о логистической регрессии, в R стандартные выходные данные находятся в шкале логарифма шансов, и ее необходимо трансформировать в склонность. Затем для преобразования балльной оценки склонности в решение может использоваться скользящий порог отсечения. Общий подход будет следующим:

1. Установить для целевого класса пороговую вероятность, выше которой мы рассматриваем запись как принадлежащую этому классу.
2. Оценить (любой моделью) вероятность, что запись принадлежит целевому классу.
3. Если эта вероятность выше пороговой вероятности, то назначить новую запись целевому классу.

Чем выше порог отсечения, тем меньше записей, предсказанных как 1, т. е. принадлежащих целевому классу. Чем ниже порог отсечения, тем больше записей, предсказанных как 1.

Эта глава посвящена нескольким ключевым приемам классификации и оценке склонностей; дополнительные методы, которые могут использоваться как для классификации, так и для численного предсказания, описаны в следующей главе.

Более двух категорий?

Подавляющее большинство задач сопряжено с бинарным откликом. Некоторые задачи классификации, однако, связаны с откликом из более чем двух возможных исходов. Например, по истечении срока действия клиентского договора подписки может быть три исхода: клиент покидает, или "переходит к другому поставщику" ($Y = 2$), переводится на помесячный договор ($Y = 1$) либо подписывает новый долгосрочный договор ($Y = 0$). Цель состоит в том, чтобы предсказать $Y = j$ для $j = 0,\ 1$ либо 2. Большинство методов классификации в данной главе могут применяться непосредственно либо со скромной адаптацией к откликам, которые имеют более двух исходов. Даже в случае более двух исходов задача часто может быть переработана в серию бинарных задач при помощи условных вероятностей. Например, чтобы предсказать исход договора, можно решить две задачи бинарного предсказания:

- предсказать, является ли $Y = 0$ или $Y > 0$;
- если дано $Y > 0$, предсказать, является ли $Y = 1$ или $Y = 2$.

В последнем случае имеет смысл разбить задачу на два случая: перейдет ли клиент к другому поставщику, и если он не перейдет, какой договор он выберет. С точки зрения подгонки модели часто выгодно трансформировать мультиклассовую задачу в серию бинарных задач. Это в особенности характерно, когда одна категория распространена намного больше, чем другие.

Наивный байесовский алгоритм

Наивный байесовский алгоритм использует вероятность наблюдать значения предикторных переменных при условии исхода с целью оценить вероятность наблюдать исход $Y = i$ при условии набора значений предикторных переменных[1].

Ключевые термины

Условная вероятность (conditional probability)

Вероятность наблюдать какое-то событие (скажем, $X = i$) *при условии, что имеется* какое-то другое событие (скажем, $Y = i$); записывается, как $P(X_i \mid Y_i)$.

Апостериорная вероятность (posterior probability)

Вероятность исхода после того, как предикторная информация была учтена (в отличие от *априорной вероятности* исходов, которая ее не учитывает).

[1] Этот и последующие разделы в настоящей главе используются с разрешения © 2017 Datastats, LLC, Питер Брюс и Эндрю Брюс.

Для понимания байесовской классификации можно начать с того, чтобы представить "не наивную" байесовскую классификацию. Для каждой записи, которая будет классифицирована:

1. Найти все остальные записи с одинаковым предикторным профилем (т. е. в которых значения предикторных переменных одинаковы).
2. Определить, к каким классам эти записи принадлежат и какой класс является преобладающим (т. е. вероятным).
3. Назначить этот класс новой записи.

Приведенный выше подход сводится к отысканию всех записей в выборке, которые выглядят в точности как новая классифицируемая запись в том смысле, что все значения предикторных переменных идентичны.

В стандартном наивном байесовском алгоритме предикторные переменные должны быть категориальными (факторными) переменными. *См. разд. "Числовые предикторные переменные" далее в этой главе* касательно двух обходных решений для использования непрерывных переменных.

Почему точная байесовская классификация непрактична?

Когда число предикторных переменных становится немалым, многие классифицируемые записи будут без точных совпадений. Это можно понять в контексте модели, которая предсказывает итоги голосования на основе демографических переменных. Даже внушительная выборка не будет содержать ни единого совпадения для новой записи с мужчиной латиноамериканцем с высоким доходом, из Среднего Запада США, который голосовал на последних выборах, не голосовал на предшествующих выборах, имеет трех дочерей и одного сына и разведен. И это всего восемь переменных — совсем небольшое число для большинства задач классификации. Добавление всего одной новой переменной с пятью одинаково частыми категориями уменьшает вероятность совпадения в 5 раз.

Несмотря на свое название, наивный байесовский алгоритм не считается методом байесовской статистики. Наивный байесовский алгоритм — это управляемый данными эмпирический метод, требующий относительно небольшой статистической компетенции. Название происходит от похожих на *правило Байеса* расчетов с целью формирования предсказаний — в частности, сначала вычисляются вероятности значений предикторных переменных при заданном исходе и далее выполняется заключительная калькуляция вероятностей исходов.

Наивное решение

В наивном байесовском решении мы больше не ограничиваем вычисление вероятности теми записями, которые совпадают с классифицируемой записью. Вместо этого мы используем весь набор данных. Наивная байесовская модификация имеет следующий вид:

1. Относительно бинарного отклика $Y = i$ ($i = 0$ либо 1) оценить индивидуальные условные вероятности для каждого предиктора $P(X_j \mid Y = i)$; речь идет о вероятностях, что значение предиктора находится в записи, когда мы наблюдаем $Y = i$. Данная вероятность оценивается долей значений X_j среди записей $Y = i$ в тренировочном наборе.
2. Перемножить эти вероятности друг с другом, а затем на долю записей, которые принадлежат $Y = i$.
3. Повторить шаги 1 и 2 для всех классов.
4. Оценить вероятность для исхода *i*, взяв значение, вычисленное на шаге 2 для класса *i*, и поделив его на сумму таких значений для всех классов.
5. Отнести запись к классу с самой высокой вероятностью для этого набора значений предикторов.

Данный наивный байесовский алгоритм можно также записать в виде уравнения вероятности наблюдать исход $Y = i$ при условии, что имеется набор значений предикторов $X_1, ..., X_p$:

$$P\left(X_1, X_2, ..., X_p\right).$$

Значение $P\left(X_1, X_2, ..., X_p\right)$ — это поправочный коэффициент, который гарантирует, что вероятность находится между 0 и 1 и не зависит от *Y*:

$$\begin{aligned} &P\left(X_1, X_2, ..., X_p\right) = \\ &= P(Y=0)\left(P\left(X_1 \mid Y=0\right)P\left(X_2 \mid Y=0\right)\cdots P\left(X_p \mid Y=0\right)\right) + \\ &+ P(Y=1)\left(P\left(X_1 \mid Y=1\right)P\left(X_2 \mid Y=1\right)\cdots P\left(X_p \mid Y=1\right)\right). \end{aligned}$$

Почему эта формула называется "наивной"? Дело в том, что мы приняли упрощающее допущение, что *точная условная вероятность* вектора значений предикторов в условиях наблюдаемого исхода достаточно хорошо оценивается произведением индивидуальных условных вероятностей $P\left(X_j \mid Y = i\right)$. Другими словами, в оценке $P\left(X_j \mid Y = i\right)$ вместо $P\left(X_1, X_2, ..., X_p \mid Y = i\right)$ мы принимаем, что X_j *независима* от всех остальных предикторных переменных X_k для $k \neq j$.

В R для оценки наивной байесовской модели могут использоваться несколько программных пакетов. Представленный далее фрагмент кода выполняет подгонку модели при помощи пакета `klaR`:

```
library(klaR)
naive_model <- NaiveBayes(outcome ~ purpose_ + home_ + emp_len_,
                          data = na.omit(loan_data))
naive_model$table
$purpose_
          var
grouping   credit_card debt_consolidation home_improvement major_purchase
  paid off   0.1857711          0.5523427       0.07153354     0.05541148
  default    0.1517548          0.5777144       0.05956086     0.03708506
          var
grouping      medical      other small_business
  paid off 0.01236169 0.09958506     0.02299447
  default  0.01434993 0.11415111     0.04538382

$home_
          var
grouping    MORTGAGE        OWN      RENT
  paid off 0.4966286 0.08043741 0.4229340
  default  0.4327455 0.08363589 0.4836186

$emp_len_
          var
grouping    > 1 Year   < 1 Year
  paid off 0.9690526 0.03094744
  default  0.9523686 0.04763140
```

На выходе из модели будут условные вероятности $P\left(X_j \mid Y = i\right)$. Модель может использоваться для предсказания исхода новой ссуды:

```
new_loan
        purpose_    home_  emp_len_
1 small_business MORTGAGE  > 1 Year
```

В данном случае модель предсказывает невозврат ссуды:

```
predict(naive_model, new_loan)
$class
[1] default
Levels: paid off default

$posterior
      paid off   default
[1,] 0.3717206 0.6282794
```

Предсказание также возвращает апостериорную оценку `posterior` вероятности невозврата ссуды. Наивный байесовский классификатор известен тем, что производит *смещенные* оценки. Однако там, где целью является *ранжирование* записей согласно вероятности, что $Y = 1$, несмещенные оценки вероятности не нужны, и наивный байесовский классификатор приводит к хорошим результатам.

Числовые предикторные переменные

Из определения мы видим, что байесовский классификатор работает только с категориальными предикторами (например, с классификацией спама, где присутствие или отсутствие слов, фраз, символов и других анализируемых параметров лежат в основе предсказательной задачи). Для применения наивного байесовского классификатора к числовым предикторам должен быть принят один из двух подходов:

- разбить на интервальные группы и конвертировать числовые предикторы в категориальные предикторы и далее применить алгоритм из предыдущего раздела;
- использовать вероятностную модель — например, нормальное распределение *(см. разд. "Нормальное распределение" главы 2)* — для оценки условной вероятности $P\left(X_j \mid Y = i\right)$.

Когда предикторная категория в тренировочных данных отсутствует, алгоритм назначает переменной исхода в новых данных нулевую вероятность вместо того, чтобы просто проигнорировать эту переменную и использовать информацию от других переменных, как это делают иные методы. На это обстоятельство следует обратить внимание при разбиении непрерывных переменных на интервальные группы.

Ключевые идеи для наивного байесовского алгоритма

- Наивный байесовский алгоритм работает с категориальными (факторными) предикторами и исходами.
- Он отвечает на вопрос: какие предикторные категории наиболее вероятны внутри каждой категории исходов?
- Эта информация далее инвертируется для оценки вероятностей категорий исходов с учетом значений предикторов.

Дополнительные материалы для чтения

- "Элементы статистического обучения" (Hastie T., Tibshirani R., Friedman J. Elements of Statistical Learning. — 2nd ed. — Springer, 2009).
- "Глубинный анализ данных для бизнес-аналитики" (Shmueli G., Bruce P., Patel N. Data mining for business analytics. — 3rd ed. — John Wiley & Sons, 2016) содержит полную главу по наивному байесовскому классификатору с вариантами для R, Excel и JMP.

Дискриминантный анализ

Дискриминантный анализ — это самый ранний статистический классификатор; он было представлен Р. А. Фишером в 1936 г. в статье, опубликованной в журнале "Летопись евгеники" (Annals of Eugenics)[2].

Ключевые термины

Ковариация (covariance)

Метрический показатель, отражающий степень, с которой переменная варьируется совместно с другой (т. е. имеет аналогичную величину и направление).

Дискриминантная функция (discriminant function)

Функция, которая при ее применении к предикторным переменным, максимизирует разделение классов.

Дискриминантные веса (discriminant weights)

Оценки, получаемые в результате применения дискриминантной функции, которые используются для исчисления вероятностей принадлежности тому или иному классу.

Хотя дискриминантный анализ охватывает несколько специализированных приемов, самое широкое распространение получил *линейный дискриминантный анализ* (linear discriminant analysis, LDA). Исходный метод, предложенный Фишером, на самом деле немного отличался от LDA, но механизм по существу остался прежним. Сегодня LDA используется менее широко с появлением более изощренных специализированных приемов, таких как древовидные модели и логистическая регрессия.

Однако LDA можно все еще встретить в некоторых приложениях, и он имеет связи с другими более широко используемыми методами (такими как анализ главных компонент; *см. разд. "Анализ главных компонент" главы 7*). Кроме того, дискриминантный анализ может обеспечить меру важности предиктора и используется как вычислительно эффективный метод отбора признаков.

Линейный дискриминантный анализ не следует путать с латентным размещением Дирихле, имеющим такую же аббревиатуру (LDA, Latent Dirichlet Allocation). Метод латентного размещения Дирихле используется в обработке текста и естественного языка и никоим образом не связан с линейным дискриминантным анализом.

[2] Конечно удивительно, что первая статья о статистической классификации была опубликована в журнале, посвященном евгенике. Действительно, существует дезориентирующая связь между ранней разработкой статистики и евгеникой (см. **http://www.statistics.com/blog/1/1459877037-subject-eugenics-journey-to-the-dark-side-at-the-dawn-of-statistics**).

Ковариационная матрица

Для понимания дискриминантного анализа сначала необходимо ввести понятие *ковариации* между двумя или несколькими переменными. Ковариация измеряет связь между двумя переменными *x* и *z*. Обозначим среднее каждой переменной как $\bar{x}$ и $\bar{z}$ *(см. разд. "Среднее" главы 1)*. Ковариация $s_{x,z}$ между *x* и *z* задается следующей формулой:

$$s_{x,z} = \frac{\sum_{i=1}^{n}(x_i - \bar{x})(z_i - \bar{z})}{n-1},$$

где *n* — это число записей (отметим, что мы делим на $n-1$ вместо *n*: *см. врезку "Степени свободы и n или n – 1?" главы 1*).

Как и с коэффициентом корреляции *(см. разд. "Корреляция" главы 1)*, положительные значения говорят о положительной связи, а отрицательные значения — об обратной. Корреляция, однако, ограничена значениями между -1 и 1, тогда как ковариация находится на той же шкале измерения, что и переменные *x* и *z*. *Ковариационная матрица* $\mathbf{\Sigma}$ для *x* и *z* состоит из дисперсий индивидуальных переменных s_x^2 и s_z^2 на диагонали (где строка и столбец — это одинаковая переменная) и ковариаций между парами переменных, стоящих вне диагоналей.

$$\hat{\Sigma} = \begin{bmatrix} s_x^2 & s_{x,z} \\ s_{x,z} & s_z^2 \end{bmatrix}.$$

Вспомним, что стандартное отклонение применяется для нормализации переменной в стандартизированную оценку (*z*-меру); ковариационная матрица используется в многомерном расширении этого процесса стандартизации. Такое расширение известно как расстояние Махаланобиса *(см. примечание "Другие метрические показатели расстояния" главы 6)* и связано с функцией LDA.

Линейный дискриминант Фишера

Для простоты сосредоточимся на задаче классификации, в которой мы хотим предсказать бинарный результат *y* при помощи всего двух непрерывных числовых переменных (*x*, *z*). В техническом плане дискриминантный анализ предполагает, что предикторные переменные представляют собой нормально распределенные непрерывные величины, но на практике метод работает хорошо даже на непредельных отклонениях от нормальности, а также для бинарных предикторов. Линейный дискриминант Фишера различает, с одной стороны, вариацию *между* группами и, с другой, вариацию *внутри* групп. В частности, стремясь разделить записи на две группы, LDA фокусируется на максимизации "между" суммой квадратов $SS_{\text{между}}$ (измеряя вариацию между этими двумя группами) относительно "внутригрупповой" суммы квадратов $SS_{\text{внутри}}$ (измеряющей внутригрупповую вариацию). В этом случае эти две группы соответствуют записям (x_0, z_0), для которых $y = 0$, и запи-

сям (x_1, z_1), для которых $y = 1$. Данный метод позволяет найти линейную комбинацию $w_x x + w_z z$, которая максимизирует это отношение суммы квадратов:

$$\frac{SS_{\text{между}}}{SS_{\text{внутри}}}.$$

Межгрупповая сумма квадратов — это квадратическое расстояние между двумя групповыми средними, а внутригрупповая сумма квадратов — это разброс вокруг средних внутри каждой группы, взвешенный на ковариационную матрицу. Интуитивно можно предположить, что путем максимизации межгрупповой суммы квадратов и минимизации внутригрупповой суммы квадратов этот метод дает самое большое разделение между этими двумя группами.

Простой пример

Программный пакет `MASS`, связанный с книгой "Современная прикладная статистика вместе с *S*" (Venables W. N., Ripley B. D. Modern Applied Statistics with S. — Springer, 1994), предлагает функцию для LDA в R. Представленный далее фрагмент кода демонстрирует применение этой функции к выборке данных о ссудах при помощи двух предикторных переменных — балльной оценки заемщика `borrower_score` и соотношения платежа к доходу `payment_inc_ratio`, и распечатывает оценочные веса линейной дискриминантной функции.

```
library(MASS)
loan_lda <- lda(outcome ~ borrower_score + payment_inc_ratio,
                    data=loan3000)
loan_lda$scaling
                           LD1
borrower_score     -6.2962811
payment_inc_ratio   0.1288243
```

Использование дискриминантного анализа для отбора признаков

Если предикторные переменные нормализуются до выполнения LDA, веса дискриминантной функции становятся мерами важности переменных, следовательно, обеспечивают вычислительно эффективный метод отбора признаков.

Функция `lda` может предсказать вероятность "невозврата" относительно "погашено":

```
pred <- predict(loan_lda)
head(pred$posterior)
       paid off   default
25333 0.5554293 0.4445707
27041 0.6274352 0.3725648
7398  0.4014055 0.5985945
35625 0.3411242 0.6588758
17058 0.6081592 0.3918408
2986  0.6733245 0.3266755
```

График предсказаний помогает проиллюстрировать, каким образом метод LDA работает. Используя выходные данные из функции `predict`, график оценочной вероятности невозврата ссуды строится следующим образом:

```
lda_df <- cbind(loan3000, prob_default=pred$posterior[,'default'])
ggplot(data=lda_df,
       aes(x=borrower_score, y=payment_inc_ratio, color=prob_default)) +
  geom_point(alpha=.6) +
  scale_color_gradient2(low='white', high='blue') +
geom_line(data=lda_df0, col='green', size=2, alpha=.8) +
```

Результирующий график показан на рис. 5.1.

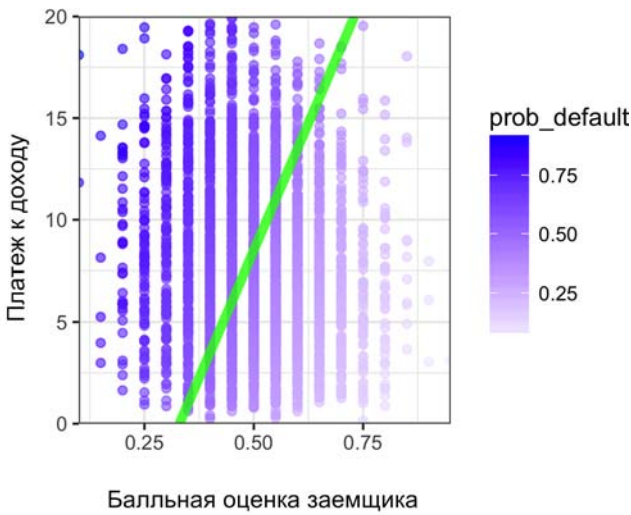


Рис. 5.1. Предсказание на основе LDA невозврата ссуды с использованием двух переменных — балла оценки кредитоспособности заемщика и соотношения платежей к доходу

Используя веса дискриминантной функции, LDA разбивает пространство предиктора на две области, как показано жирной линией. Предсказания, которые расположены от линии намного дальше, имеют более высокий уровень доверия (т. е. вероятность намного больше 0,5).

Расширения дискриминантного анализа

Больше предикторных переменных. Хотя в тексте и примере этого раздела использовались всего две предикторные переменные, метод LDA работает точно так же с тремя и более предикторными переменными. Единственным ограничивающим фактором является число записей (оценивание ковариационной матрицы требует достаточного числа записей в расчете на переменную, что, как правило, не представляет проблему в приложениях науки о данных).

Квадратический дискриминантный анализ. Существуют другие варианты дискриминантного анализа. Самым известным является квадратический дискриминантный анализ (quadratic discriminant analysis, QDA). Несмотря на его название, QDA все же является линейной дискриминантной функцией. Главное

различие состоит в том, что в LDA ковариационная матрица принимается одинаковой для двух групп, соответствующих $Y = 0$ и $Y = 1$. В QDA допускается, что ковариационная матрица может быть разной для двух групп. На практике эта разница в большинстве применений не является критической.

Ключевые идеи для дискриминантного анализа

- Дискриминантный анализ работает с непрерывными или категориальными предикторами, а также с категориальными исходами.
- При использовании ковариационной матрицы он вычисляет линейную дискриминантную функцию, которая используется для различения записей, принадлежащих разным классам.
- Данная функция применяется к записям для получения весов, или балльных оценок, для каждой записи (один вес для каждого возможного класса), которая определяет его оценочный класс.

Дополнительные материалы для чтения

- Книга "Элементы статистического обучения" (Hastie T., Tibshirani R., Friedman J. Elements of Statistical Learning. — 2nd ed. — Springer, 2009) и ее более краткий вариант "Введение в статистическое обучение" (Gareth J., Witten D., Hastie T., Tibshirani R. An Introduction to Statistical Learning. — Springer, 2013) содержат раздел по дискриминантному анализу.
- "Глубинный анализ данных для бизнес-аналитики" (Shmueli G., Bruce P., Patel N. Data mining for business analytics. — 3rd ed. — John Wiley & Sons, 2016) с вариантами для R, Excel и JMP. Данная книга содержит полную главу по дискриминантному анализу.
- Для интереса в историческом плане оригинальную статью Фишера по указанной теме "Использование многократных замеров в таксономических задачах" (The Use of Multiple Measurements in Taxonomic Problems), опубликованную в 1936 г. в журнале "Летопись евгеники" (Annals of Eugenics) (теперь носящем название "Летопись генетики" (Annals of Genetics)), можно найти онлайн: **http://onlinelibrary.wiley.com/doi/10.1111/j.1469-1809.1936.tb02137.x/epdf**.

Логистическая регрессия

Логистическая регрессия аналогична множественной линейной регрессии за одним исключением — исход является бинарным. При этом используются различные преобразования для того, чтобы привести задачу к тому виду, в котором может быть подогнана линейная модель. Как и дискриминантный анализ, и в отличие от метода *K* ближайших соседей и наивного байесовского алгоритма, логистическая регрессия — это подход со структурированной моделью нежели информационно-центричный подход. Данный метод стал популярным благодаря своему высокому

вычислительному быстродействию и выходным данным модели, которые допускают оперативную оценку новых данных.

Ключевые термины

Логит-преобразование (logit)

Функция, которая отображает вероятность принадлежности классу (в диапазоне 0–1) на диапазон $\pm\infty$.

Синонимы: логарифм шансов (см. далее), ло́гит.

Шансы (odds)

Отношение "успеха" (1) к "неуспеху" (0).

Логарифм шансов (log odds)

Отклик в преобразованной модели (теперь линейный), который отображается назад в вероятность.

Как перейти от переменной бинарного исхода к переменной исхода, которую можно смоделировать линейно, а затем возвратить в бинарный исход?

Функция логистического отклика и логит-преобразование

Ключевыми компонентами являются *функция логистического отклика* и *логит-преобразование*, которые позволяют отобразить вероятность (выраженную по шкале 0–1) в более расширенную шкалу, подходящую для линейного моделирования.

Первый шаг состоит в том, чтобы представить переменную исхода не как бинарную метку, а как вероятность p, чья метка равна "1". Мы наивно можем захотеть смоделировать p как линейную функцию предикторных переменных:

$$p = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + ... + \beta_q x_q .$$

Однако подгонка этой модели не гарантирует, что p окажется между 0 и 1, какой и должна быть вероятность.

Вместо этого мы смоделируем p, применив к предикторам функцию *логистического отклика* или *обратного логит-преобразования*:

$$p = \frac{1}{1 + e^{-(\beta_0 + \beta_1 x_1 + \beta_2 x_2 + ... + \beta_q x_q)}}.$$

Данное преобразование гарантирует, что p останется между 0 и 1.

Для того чтобы извлечь экспоненциальное выражение из знаменателя, вместо вероятностей мы рассматриваем *шансы*, или перевесы. Шансы, знакомые всем делающим ставки игрокам, являются соотношением "успехов" (1) к "неуспехам" (0). С точки зрения вероятностей шансы — это вероятность события, деленная на веро-

ятность, что событие не произойдет. Например, если вероятность, что лошадь выиграет скачки, равна 0,5, то вероятность, что "не выиграет", составит $1-0,5=0,5$, и шансы равны 1,0.

$$\text{Шансы}\ (Y=1)=\frac{p}{1-p}.$$

Получить вероятность шансов можно при помощи обратной функции шансов:

$$p=\frac{\text{Шансы}}{1+\text{Шансы}}.$$

Комбинируем эту формулу с функцией логистического отклика, показанной ранее, и получаем:

$$\text{Шансы}\ (Y=1)=e^{\beta_0+\beta_1 x_1+\beta_2 x_2+\ldots+\beta_q x_q}.$$

Наконец, взяв логарифм выражений, стоящих справа и слева от знака равенства, получаем выражение, которое включает линейную функцию предикторов:

$$\log\left(\text{Шансы}\ (Y=1)\right)=\beta_0+\beta_1 x_1+\beta_2 x_2+\ldots+\beta_q x_q.$$

Функция *логарифма шансов*, так называемая *логит*-функция, отображает вероятность *p* из интервала (0, 1) в любое значение из интервала $(-\infty, +\infty)$, рис. 5.2. Цикл преобразования завершен; мы использовали линейную модель для предсказания вероятности, которую в свою очередь можем отобразить на метку класса путем применения правила отсечения — любая запись с вероятностью, большей, чем порог отсечения, классифицируется как 1.

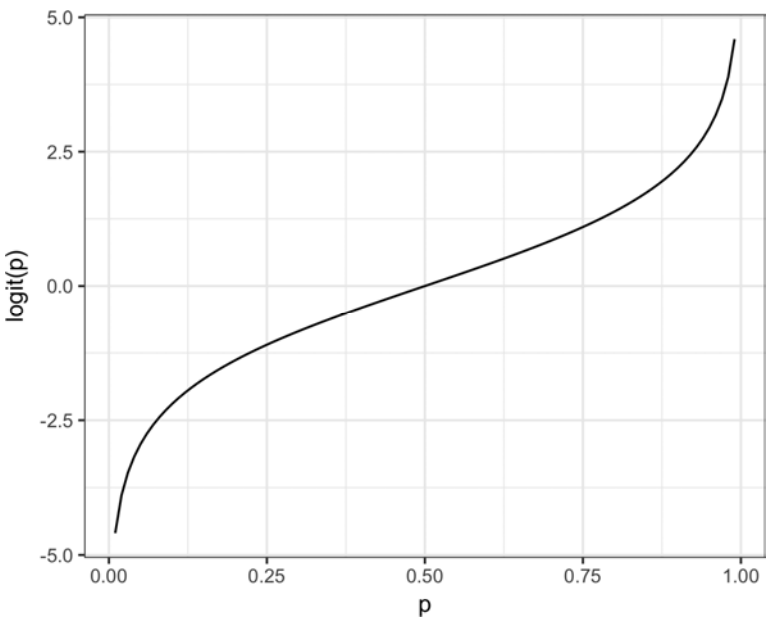


Рис. 5.2. Функция, которая отображает вероятность в шкалу, подходящую для линейной модели (логит)

Логистическая регрессия и обобщенная линейная модель

Отклик в формуле логистической регрессии — это логарифм шансов бинарного исхода 1. Мы наблюдаем только бинарный исход, а не логарифм шансов, и поэтому для подгонки уравнения требуются специальные статистические методы. *Логистическая регрессия* — это особая разновидность обобщенной линейной модели (ОЛМ — GLM, generalized linear model), разработанной для расширения линейной регрессии до других параметрических настроек.

В R для подгонки логистической регрессии используется функция `glm` с аргументом `family`, установленным в значение `binomial`. Приведенный далее фрагмент кода выполняет подгонку логистической регрессии к данным о персональных ссудах, представленных в *разд. "К ближайших соседей" главы 6*.

```
logistic_model

Call: glm(formula = outcome ~ payment_inc_ratio + purpose_ + home_ +
    emp_len_ + borrower_score, family = "binomial", data = loan_data)

Coefficients:
               (Intercept)           payment_inc_ratio
                   1.26982                     0.08244
purpose_debt_consolidation    purpose_home_improvement
                   0.25216                     0.34367
    purpose_major_purchase             purpose_medical
                   0.24373                     0.67536
             purpose_other      purpose_small_business
                   0.59268                     1.21226
                  home_OWN                  home_RENT
                   0.03132                     0.16867
        emp_len_ < 1 Year              borrower_score
                   0.44489                    -4.63890

Degrees of Freedom: 46271 Total (i.e. Null); 46260 Residual
Null Deviance:      64150
Residual Deviance: 58530        AIC: 58550
```

Откликом является `outcome`, который принимает значение 0, если ссуда погашена, и 1, если ссуда не возвращена. Факторные переменные `purpose_` и `home_` представляют цель ссуды и статус домовладельца. Как и в регрессии, факторная переменная с P уровнями представлена $P-1$ столбцами. В R по умолчанию используется *опорное* кодирование и все уровни сравниваются с опорным уровнем *(см. разд. "Факторные переменные в регрессии" главы 4)*. Опорные уровни для этих факторов — это соответственно `credit_card` и `MORTGAGE`. Переменная `borrower_score` — это балльная оценка от 0 до 1, которая обозначает кредитоспособность заемщика (от плохой до превосходной). Данная переменная была создана из несколь-

ких других переменных при помощи *K* ближайших соседей *(см. разд. "Метод KNN как конструктор признаков" главы 6)*.

Обобщенные линейные модели

Обобщенные линейные модели (ОЛМ, GLM) — это второй по важности класс моделей помимо регрессии. ОЛМ характеризуются двумя главными компонентами:

- вероятностным распределением или семейством (биномиальным в случае логистической регрессии);
- связывающей функцией, отображающей отклик на предикторы (логит-преобразование в случае логистической регрессии).

Логистическая регрессия является, безусловно, общепринятой формой ОЛМ. Аналитик данных будет сталкиваться и с другими типами ОЛМ. Иногда вместо логита используется логарифм связывающей функции; на практике использование логарифма связывающей функции вряд ли приведет к сильно отличающимся результатам для большинства применений. Распределение Пуассона чаще всего выбирается для моделирования количественных данных (например, число посещений пользователем веб-страницы на определенное количество времени). Другие семейства включают отрицательное биномиальное и гамма-распределение, часто используемые для моделирования истекшего времени (например, времени безотказной работы). В отличие от логистической регрессии, применение ОЛМ с этими моделями обставлено большим количеством нюансов и сопряжено с мерами предосторожности. Их лучше всего избегать, только если вы с ними не знакомы и не понимаете полезность и ловушки этих методов.

Предсказанные значения в логистической регрессии

Предсказанное значение, полученное из логистической регрессии, рассматривается с точки зрения логарифма шансов: $\hat{Y} = \log\left(\text{Шансы}\,(Y=1)\right)$. Предсказанная вероятность задается функцией логистического отклика:

$$\hat{p} = \frac{1}{1+e^{-\hat{Y}}}.$$

Например, посмотрим на предсказания из модели `logistic_model`:

```
pred <- predict(logistic_model)
summary(pred)
     Min.   1st Qu.    Median      Mean  3rd Qu.     Max.
-2.728000 -0.525100 -0.005235 0.002599 0.513700 3.658000
```

Конвертация этих значений в вероятности является простой операцией трансформации:

```
prob <- 1/(1 + exp(-pred))
> summary(prob)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
0.06132 0.37170 0.49870 0.50000 0.62570 0.97490
```

Значения находятся в пределах диапазона от 0 до 1 и еще не позволяют сделать вывод о том, является ли предсказанное значение "невозврат" или "погашено". Можно объявить любое значение свыше 0,5 как невозврат, аналогично классификатору на основе *K* ближайших соседей. На практике, если цель состоит в том, чтобы идентифицировать членов редкого класса, часто обоснован более низкий порог отсечения *(см. разд. "Проблема редкого класса" далее в этой главе)*.

Интерпретация коэффициентов и отношений шансов

Одно из преимуществ логистической регрессии состоит в том, что она порождает модель, которую можно оперативно применять для оценки новых данных без повторного вычисления. Еще одно преимущество — относительная простота интерпретации модели по сравнению с другими методами классификации. Ключевая концептуальная идея заключается в понимании *отношения шансов*. Отношение шансов легче всего понять на основе бинарной факторной переменной *X*:

$$\text{отношение шансов} = \frac{\text{Шансы}\left(Y = 1 \mid X = 1\right)}{\text{Шансы}\left(Y = 1 \mid X = 0\right)}.$$

Данная формула интерпретируется, как шансы, что $Y = 1$, когда $X = 1$, против шансов, что $Y = 1$, когда $X = 0$. Если отношение шансов равно 2, то шансы, что $Y = 1$ в два раза выше, когда $X = 1$, чем когда $X = 0$.

Зачем возиться с отношением шансов вместо вероятностей? Мы работаем с шансами, потому что коэффициент β_j в логистической регрессии является логарифмом отношения шансов для X_j.

Пример все объяснит. Относительно подгонки модели в *разд. "Логистическая регрессия и обобщенная линейная модель" ранее в этой главе* ее коэффициент регрессии для `purpose_small_business` равен 1,21226. Это означает, что ссуда малому бизнесу в сравнении со ссудой для погашения задолженности по кредитной карте сокращает шансы невозврата против шансов погашения на $\exp(1,21226) \approx 3,4$. Безусловно, кредиты в целях создания или расширения малого бизнеса значительно более рискованны, чем другие типы кредитов.

На рис. 5.3 показана связь между отношением шансов и его логарифмом для отношений шансов больше 1. Поскольку коэффициенты находятся на логарифмической шкале, увеличение на 1 в коэффициентах приводит к увеличению в $\exp(1) \approx 2,72$ раз в отношении шансов.

Отношения шансов для числовых переменных *X* можно проинтерпретировать аналогичным образом: они измеряют изменение в отношении шансов для единичного изменения в *X*. Например, эффект от увеличения соотношения платежей к доходам, скажем, с 5 до 6 увеличивает шансы невозврата ссуды в $\exp(0,08244) \approx 1,09$ раз. Переменная `borrower_score` — это балльная оценка кредитоспособности заемщиков, она колеблется в диапазоне от 0 (низкая) до 1 (высокая). Шансы лучших заемщиков относительно худших, которые не возвращают ссуды, меньше

в $\exp(-4,63890) \approx 0,01$ раз. Другими словами, риск невозврата от заемщиков с самой слабой кредитоспособностью в 100 раз больше, чем риск невозврата у лучших заемщиков!

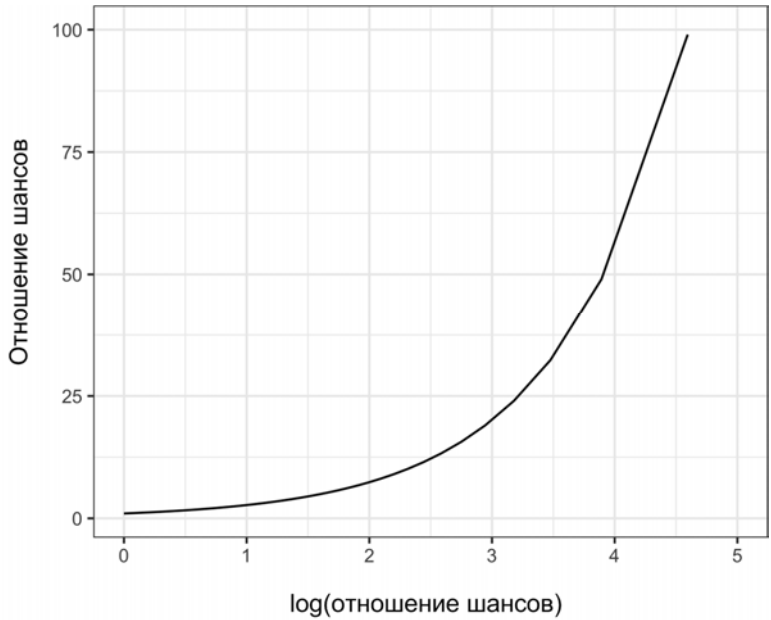


Рис. 5.3. Связь между отношением шансов и его логарифмом

Линейная и логистическая регрессии: сходства и различия

Множественная линейная регрессия и логистическая регрессия имеют массу схожих черт. Обе принимают параметрическую линейную форму, связывающую предикторы с откликом. Обследование и нахождение лучшей модели в обоих случаях очень похожи. Общий характер использования в линейной модели сплайновой трансформации предиктора одинаковым образом применим в параметрических настройках логистической регрессии. Однако логистическая регрессия отличается двумя фундаментальными составляющими:

- характером выполнения подгонки модели (наименьшие квадраты не применимы);
- природой и анализом остатков от модели.

Подгонка модели

Подгонка линейной регрессии выполняется с использованием наименьших квадратов, и качество подгонки оценивается с использованием статистик RMSE и R^2. В логистической регрессии (в отличие от принятого в линейной регрессии) замкнутое решение отсутствует, и подгонка модели должна выполняться с использовани-

ем оценки максимального правдоподобия (MLE, maximum likelihood estimation). Оценка максимального правдоподобия — это процедура, пытающаяся найти модель, которая вероятнее всего породила данные, имеющиеся у нас. В уравнении логистической регрессии откликом является не 0 или 1, а оценка логарифма шансов, что отклик равняется 1. Метод MLE находит такое решение, что оценочный логарифм шансов наилучшим образом описывает наблюдаемый исход. Механизм данного алгоритма сопряжен с квазиньютоновской оптимизацией, которая итеративно выполняется на основе текущих параметров между шагом оценки результативности (оценка в баллах по Фишеру) и обновлением параметров для улучшения подгонки.

Оценка максимального правдоподобия

Рассмотрим этот метод подробнее, если вам нравятся статистические символы: начнем с набора данных $(X_1, X_2, ..., X_n)$ и вероятностной модели $\mathcal{P}_\theta(X_1, X_2, ..., X_n)$, которая зависит от набора параметров $\hat{\theta}$. Цель метода MLE состоит в том, чтобы найти набор параметров $\hat{\theta}$, который максимизирует значение $\mathcal{P}_\theta(X_1, X_2, ..., X_n)$, т. е. максимизирует вероятность наблюдать $(X_1, X_2, ..., X_n)$ в условиях модели $\mathcal{P}(...)$. В процессе подгонки модель оценивается при помощи метрического показателя, который называется *погрешностью*:

$$\text{погрешность} = -2\log\left(\mathcal{P}_{\hat{\theta}}(X_1, X_2, ..., X_n)\right).$$

Более низкая погрешность соответствует более удачной подгонке.

К счастью, большинству пользователей не придется заниматься деталями алгоритма подгонки, поскольку они обрабатываются программным обеспечением. Кроме того, большинству аналитиков данных нет надобности беспокоиться по поводу метода подгонки, кроме понимания того, что он представляет собой способ найти хорошую модель при наличии определенных предположений.

Обработка факторных переменных

В логистической регрессии факторные переменные должны кодироваться как в линейной регрессии *(см. разд. "Факторные переменные в регрессии" главы 4)*. В R и других программных системах это обычно обрабатывается автоматически, и вдобавок, как правило, используется опорное кодирование. Все другие методы классификации, охваченные в этой главе, в основном используют представление в виде кодировщика с одним активным состоянием *(см. разд. "Кодировщик с одним активным состоянием" главы 6)*.

Диагностика модели

Как и другие методы классификации, логистическая регрессия диагностируется тем, насколько точно модель классифицирует новые данные *(см. разд. "Оценивание моделей классификации" далее в этой главе)*. Как и в случае с линейной регрессией, есть несколько дополнительных стандартных статистических инструментов, кото-

рые позволяют выполнить диагностику модели и ее улучшить. Вместе с оценочными коэффициентами R сообщает о стандартной ошибке коэффициентов (SE), *z*-оценке и *p*-значении:

```
summary(logistic_model)

Call:
glm(formula = outcome ~ payment_inc_ratio + purpose_ + home_ +
    emp_len_ + borrower_score, family = "binomial", data = loan_data)

Deviance Residuals:

     Min        1Q    Median        3Q       Max
-2.71430  -1.06806  -0.04482   1.07446   2.11672

Coefficients:
                          Estimate  Std. Error z value Pr(>|z|)
(Intercept)               1.269822    0.051929  24.453  < 2e-16 ***
payment_inc_ratio         0.082443    0.002485  33.177  < 2e-16 ***
purpose_debt_consolidation 0.252164    0.027409   9.200  < 2e-16 ***
purpose_home_improvement   0.343674    0.045951   7.479 7.48e-14 ***
purpose_major_purchase     0.243728    0.053314   4.572 4.84e-06 ***
purpose_medical            0.675362    0.089803   7.520 5.46e-14 ***
purpose_other              0.592678    0.039109  15.154  < 2e-16 ***
purpose_small_business     1.212264    0.062457  19.410  < 2e-16 ***
home_OWN                   0.031320    0.037479   0.836 0.403
home_RENT                  0.168670    0.021041   8.016 1.09e-15 ***
emp_len_ < 1 Year          0.444892    0.053342   8.340  < 2e-16 ***
borrower_score            -4.638902    0.082433 -56.275  < 2e-16 ***
---
Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
(Dispersion parameter for binomial family taken to be 1)
    Null deviance: 64147  on 46271  degrees of freedom
Residual deviance: 58531  on 46260  degrees of freedom
AIC: 58555

Number of Fisher Scoring iterations: 4
```

Интерпретация *p*-значения сопровождается теми же оговорками, что и в регрессии, и должна рассматриваться больше как относительный индикатор важности переменной *(см. разд. "Диагностика модели" главы 4)*, чем как формальная мера статистической значимости. С моделью логистической регрессии, которая имеет бинарный отклик, не связан показатель RMSE либо R^2. Вместо этого модель логистической регрессии обычно оценивается при помощи более общих метрических показателей, предназначенных для классификации *(см. разд. "Оценивание модели классификации" далее в этой главе)*.

Многие другие понятия, относящиеся к линейной регрессии, переносятся на параметрическую настройку логистической регрессии (и других ОЛМ). Например,

можно использовать шаговую регрессию, выполнить подгонку членов уравнения, характеризующих взаимодействие, или включить сплайновые члены. Те же вопросы касаются применения к логистической регрессии искажающих и коррелированных переменных *(см. разд. "Интерпретация уравнения регрессии" главы 4)*. Подгонку обобщенных аддитивных моделей можно выполнить *(см. разд. "Обобщенные аддитивные модели" главы 4)* с помощью пакета `mgcv`:

```
logistic_gam <- gam(outcome ~ s(payment_inc_ratio) + purpose_ +
                      home_ + emp_len_ + s(borrower_score),
                    data=loan_data, family='binomial')
```

Одна из областей, где логистическая регрессия иная, касается анализа остатков. Как и в регрессии (рис. 4.9), вычисление частных остатков выполняется прямолинейно:

```
terms <- predict(logistic_gam, type='terms')
partial_resid <- resid(logistic_model) + terms
df <- data.frame(payment_inc_ratio = loan_data[, 'payment_inc_ratio'],
                 terms = terms[, 's(payment_inc_ratio)'],
                 partial_resid = partial_resid[, 's(payment_inc_ratio)'])
ggplot(df, aes(x=payment_inc_ratio, y=partial_resid, solid = FALSE)) +
  geom_point(shape=46, alpha=.4) +
  geom_line(aes(x=payment_inc_ratio, y=terms),
            color='red', alpha=.5, size=1.5) +
  labs(y='Partial Residual')
```

Результирующий график отображен на рис. 5.4. Оценочная подгонка, показанная линией, проходит между двумя наборами точечных облаков. Верхнее облако соот-

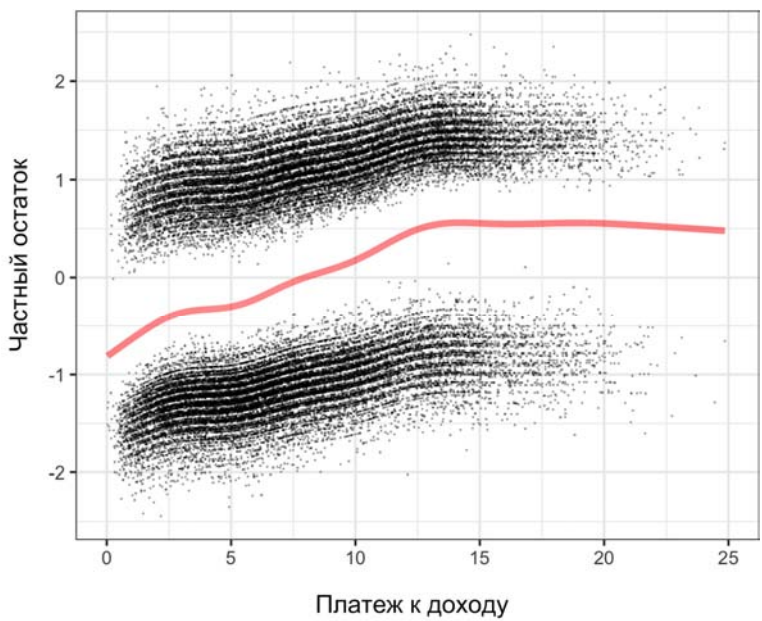


Рис. 5.4. Частные остатки от логистической регрессии

ветствует отклику 1 (невозвращенные ссуды) и нижнее облако — отклику 0 (погашенные ссуды). Такой вид очень типичен для остатков от логистической регрессии, поскольку выходные данные бинарные. Хотя частные остатки в логистической регрессии имеют меньшую ценность, чем в регрессии, они по-прежнему полезны для подтверждения нелинейного поведения и идентификации очень влиятельных записей.

Часть выходных данных функции `summary` можно практически проигнорировать. Параметр дисперсии к логистической регрессии не применяется и существует для других типов ОЛМ. Остаточная погрешность и число итераций оценивания результативности (или вклада) связаны с оценкой максимального правдоподобия *(см. врезку "Оценка максимального правдоподобия" ранее в этой главе)*.

Ключевые идеи для логистической регрессии

- Логистическая регрессия похожа на линейную регрессию, за исключением того, что откликом является бинарная переменная.
- Для получения модели в линейной форме, где в качестве переменной отклика выступает логарифм отношения шансов, необходимо несколько преобразований.
- После подгонки линейной модели (в результате итеративного процесса) логарифмические шансы отображаются назад на вероятности.
- Логистическая регрессия популярна по причине своего вычислительного быстродействия, в том числе потому, что она порождает модель, которую можно оперативно применять для оценки новых данных без повторного вычисления.

Дополнительные материалы для чтения

- Стандартный справочник по логистической регрессии "Прикладная логистическая регрессия" (Hosmer D., Lemeshow S., Sturdivan R. Applied Logistic Regression. — 3rd ed. — Wiley, 2013).
- Также популярные две книги Джозефа Хильбе (Joseph Hilbe): "Модели логистической регрессии" (Logistic Regression Models) в углубленном изложении и "Практическое руководство по логистической регрессии" (Practical Guide to Logistic Regression) в компактном изложении, обе опубликованы в издательстве CRC Press.
- Книга "Элементы статистического обучения" (Hastie T., Tibshirani R., Friedman J. Elements of Statistical Learning. — 2nd ed. — Springer, 2009) и ее более краткий вариант "Введение в статистическое обучение" (Gareth J., Witten D., Hastie T., Tibshirani R. An Introduction to Statistical Learning. — Springer, 2013) содержат разделы по логистической регрессии.
- Книга "Глубинный анализ данных для бизнес-аналитики" (Shmueli G., Bruce P., Patel N. Data mining for business analytics. — 3rd ed. — John Wiley & Sons, 2016)

с вариантами для R, Excel и JMP содержит полную главу по логистической регрессии.

Оценивание моделей классификации

В предсказательном моделировании общепринято испытывать большое число разных моделей, применять каждую к контрольной выборке с отложенными данными (также именуемой *тестовой* или *проверочной* выборкой) и диагностировать их работоспособность. В сущности, подход сводится к наблюдению за тем, какая из них производит самые точные предсказания.

Ключевые термины

Точность (accuracy)

Процент (или доля) случаев, классифицированных правильно.

Матрица несоответствий (confusion matrix)

Отображение в табличной форме (2×2 в бинарном случае) количеств записей по их предсказанному и фактическому состояниям, или результату, классификации.

Синонимы: матрица ошибок, матрица неточностей.

Чувствительность (sensitivity)

Процент (или доля) правильно классифицированных единиц.

Синоним: полнота.

Специфичность (specificity)

Процент (или доля) правильно классифицированных нулей.

Прецизионность (precision)

Процент (или доля) предсказанных единиц, которые фактически являются нулями.

ROC-кривая (ROC curve)

График чувствительности против специфичности.

Лифт (lift)

Метрический показатель, который измеряет степень эффективности модели при идентификации (сравнительно редких) единиц при разных порогах вероятности.

Простой способ измерить результативность модели классификации состоит в том, чтобы подсчитать долю правильных предсказаний.

В большинстве алгоритмов классификации каждому случаю назначена "оценочная вероятность того, чтобы он равен 1"[3]. По умолчанию точка принятия решения, или отсечение, как правило, равна 0,50 или 50%. Если вероятность выше 0,5, то определяется категория "1", в противном случае — "0". Альтернативным отсечением по умолчанию является преобладающая вероятность единиц в данных.

Точность (accuracy) — это просто мера общей ошибки[4]:

$$\text{точность} = \frac{\Sigma_{\text{истинноположительный}} + \Sigma_{\text{истинноотрицательный}}}{\text{размер выборки}}.$$

Матрица несоответствий

В основе системы метрических показателей классификации лежит *матрица несоответствий* — таблица, показывающая число правильных и неправильных предсказаний, сгруппированных в категории по типу отклика. В R имеется несколько программных пакетов, предназначенных для вычисления матрицы несоответствий, но в бинарном случае она легко вычисляется вручную.

Для того чтобы проиллюстрировать матрицу несоответствий, рассмотрим модель `logistic_gam`, которая была натренирована на сбалансированном наборе данных с равным количеством невозвращенных и погашенных ссуд (см. рис. 5.4). Следуя принятым правилам, $Y = 1$ соответствует целевому событию (например, невозврат), а $Y = 0$ соответствует отрицательному (либо обычному) событию (например, погашено). Приведенный далее фрагмент кода вычисляет матрицу несоответствий для модели `logistic_gam`, примененной ко всему (несбалансированному) тренировочному набору:

```
pred <- predict(logistic_gam, newdata=train_set)
pred_y <- as.numeric(pred > 0)
true_y <- as.numeric(train_set$outcome=='default')
true_pos <- (true_y==1) & (pred_y==1)
true_neg <- (true_y==0) & (pred_y==0)
false_pos <- (true_y==0) & (pred_y==1)
false_neg <- (true_y==1) & (pred_y==0)

conf_mat <- matrix(c(sum(true_pos), sum(false_pos),
                     sum(false_neg), sum(true_neg)), 2, 2)
```

[3] Не все методы обеспечивают несмещенные оценки вероятности. В большинстве случаев достаточно, что метод обеспечивает ранжирование, эквивалентное ранжированию, которое было бы результатом несмещенной оценки вероятности; метод отсечения тогда функционально эквивалентен.

[4] В соответствии с ГОСТ Р ИСО 5725-1 — ГОСТ Р ИСО 5725-5 правильность (trueness) — это степень близости результата измерений к истинному или условно истинному (действительному) значению измеряемой величины, а точность (accuracy) и погрешность (error) результатов измерений, как правило, определяются сравнением результата измерений с истинным или действительным (условно истинным) значением измеряемой физической величины (являющимися фактически эталонными значениями измеряемых величин, выраженными в узаконенных единицах). — *Прим. пер.*

```
colnames(conf_mat) <- c('Yhat = 1', 'Yhat = 0')
rownames(conf_mat) <- c('Y = 1', 'Y = 0')
conf_mat
     Yhat = 1 Yhat = 0
Y = 1 14635 8501
Y = 0 8236 14900
```

Предсказанными исходами являются столбцы, а истинными результатами — строки. Диагональные элементы матрицы показывают число правильных предсказаний, внедиагональные элементы — число неправильных предсказаний. Например, 6126 невозвращенных ссуд было предсказано правильно, как невозвращенные, и 17 010 невозвращенных ссуд было предсказано неправильно, как погашенные.

На рис. 5.5 показана связь между матрицей несоответствий для бинарного отклика Y и разных метрических показателей (*см. разд. "Прецизионность, полнота и специфичность" далее в этой главе*, чтобы узнать подробнее о метрических показателях). Как и в примере с данными о ссудах, фактический отклик расположен вдоль строк, а предсказанный отклик — вдоль столбцов. (Можно встретить матрицы несоответствий с инвертированным расположением строк и столбцов.) Диагональные поля (левый верхний, правый нижний) показывают, когда предсказания $\hat{Y}$ правильно предсказывают отклик. Один из важных метрических показателей, который явно не упоминается, — это *доля* ложноположительных исходов (зеркальное отражение прецизионности). Когда единицы встречаются редко, соотношение ложноположительных исходов ко всем предсказанным положительным исходам может быть высоким, приводя к нелогичной ситуации, где предсказанная 1 скорее всего является 0. Эта проблема является бичом для широко применяемых диагностических тестов при медицинском обследовании (например, маммограммы): из-за относительной редкости условия положительные результаты тестов, вероятнее всего, не означают рак молочной железы. Это приводит к дезориентации публики.

Рис. 5.5. Матрица несоответствий для бинарного отклика и различных метрических показателей

Проблема редкого класса

Во многих случаях существует несбалансированность в предсказываемых классах, когда один класс намного более преобладающий, чем остальные — например, законные страховые претензии против мошеннических либо простые посетители против покупателей на веб-сайте. Редкий класс (например, мошеннические претензии) — это обычно класс, который представляет больший интерес и, как правило, обозначается 1, в отличие от более преобладающего, обозначаемого 0. В типичном сценарии единицы — это более важный случай в том смысле, что неправильная их классификация, как нули, стоит дороже, чем неправильная классификация нуля, как единицы. Например, правильная идентификация мошеннической страховой претензии может сэкономить тысячи долларов. С другой стороны, правильная идентификация не мошеннической претензии просто сэкономит вам затраты и усилия на просмотр претензии вручную с более осторожным анализом (то, что вы как раз и сделаете, если бы претензия была помечена как "мошенническая").

В таких случаях, если только классы легко не разделимы, наиболее точной моделью классификации может быть та, которая просто-напросто классифицирует все случаи как 0. Например, если только 0,1% простых посетителей в веб-магазине в конечном итоге делают покупку, то модель, которая предсказывает, что каждый простой покупатель уйдет без покупки, будет точна на 99,9%. И тем не менее она будет бесполезной. Вместо этого мы были бы довольны моделью, которая менее точна в целом, но способна различать покупателей, даже если она по пути неправильно классифицирует каких-либо непокупателей.

Прецизионность, полнота и специфичность

Метрические показатели, иные чем чистая точность — показатели, носящие более нюансированный характер, — широко используются при оценивании моделей классификации. Некоторые из них в статистике применяются давно — в особенности, в биостатистике, где они используются для описания ожидаемой результативности диагностических тестов. Прецизионность измеряет точность предсказанного положительного исхода (рис. 5.5):

$$\text{Прецизионность} = \frac{\sum \text{ИП}}{\sum \text{ИП} + \sum \text{ЛП}},$$

где ИП — истинноположительный; ЛП — ложноположительный.

Полнота, также именуемая *чувствительностью*, измеряет силу модели в предсказании положительного исхода — доля единиц, которые она правильно идентифицирует (рис. 5.5). Термин *"чувствительность"* часто используется в биостатистике и медицинской диагностике, тогда как *полнота* больше применяется в области машинного обучения. Определение полноты имеет следующий вид:

$$\text{Полнота} = \frac{\sum \text{ИП}}{\sum \text{ИП} + \sum \text{ЛО}},$$

где ИП — истинноположительный; ЛО — ложноотрицательный.

Еще один используемый метрический показатель — это *специфичность*, которая измеряет способность модели предсказывать отрицательный исход:

$$\text{Полнота} = \frac{\sum \text{ИО}}{\sum \text{ИО} + \sum \text{ЛО}},$$

где ИО — истинноотрицательный; ЛО — ложноотрицательный.

```
# precision
conf_mat[1,1]/sum(conf_mat[,1])
# recall
conf_mat[1,1]/sum(conf_mat[1,])
# specificity
conf_mat[2,2]/sum(conf_mat[2,])
```

ROC-кривая

Вы можете видеть, что между полнотой и специфичностью имеется компромисс. Охват большего числа единиц обычно означает неправильную идентификацию большего числа нулей, как единиц. Идеальный классификатор превосходно справится с идентификацией единиц без неправильной идентификации большего числа нулей, как единиц.

Метрический показатель, который фиксирует этот компромисс, называется *кривой* "рабочих характеристик получателя", обычно называемой *ROC-кривой* (receiver operating characteristics). Для построения графика ROC-кривой по оси *y* откладывается полнота (чувствительность) в сопоставлении со специфичностью по оси *x*.[5] ROC-кривая показывает компромисс между полнотой и специфичностью по мере изменяемого вами порога отсечения, чтобы вы смогли определить, каким образом классифицировать запись. На графике чувствительность (полнота) отображается на оси *y*, а для оси *x* можно встретить две формы маркировки:

- специфичность наносится на оси *x*, где 1 слева и 0 справа;
- специфичность наносится на оси *x*, где 0 слева и 1 справа.

Кривая выглядит идентичной независимо от формы маркировки. Процесс вычисления ROC-кривой следующий:

1. Отсортировать записи по предсказанной вероятности, указывающей на принадлежность к категории 1, начиная с наиболее вероятной и заканчивая наименее вероятной.
2. Вычислить совокупную специфичность и полноту на основе сортированных записей.

[5] Кривая ROC сначала использовалась во время Второй мировой войны для описания производительности радарных приемных станций, работа которых состояла в том, чтобы правильно идентифицировать (классифицировать) отраженные радарные сигналы и предупреждать силы обороны о приближающемся самолете.

Вычисление ROC-кривой в R выполняется достаточно прямолинейно. Следующий ниже фрагмент кода вычисляет ROC для данных о ссудах:

```
idx <- order(-pred)
recall <- cumsum(true_y[idx]==1)/sum(true_y==1)
specificity <- (sum(true_y==0) - cumsum(true_y[idx]==0))/sum(true_y==0)
roc_df <- data.frame(recall = recall, specificity = specificity)
ggplot(roc_df, aes(x=specificity, y=recall)) +
  geom_line(color='blue') +
  scale_x_reverse(expand=c(0, 0)) +
  scale_y_continuous(expand=c(0, 0)) +
  geom_line(data=data.frame(x=(0:100)/100), aes(x=x, y=1-x),
            linetype='dotted', color='red')
```

Результат показан на рис. 5.6. Пунктирная диагональная линия соответствует классификатору, который не лучше случайной возможности. Чрезвычайно эффективный классификатор (или в медицинских ситуациях чрезвычайно эффективный диагностический тест) будет иметь ROC-кривую, которая прижимается к левому верхнему углу — она правильно идентифицирует много единиц без неправильной классификации многих нулей, как единицы. Если для этой модели нам нужен классификатор со специфичностью по крайней мере 50%, то полнота составит порядка 75%.

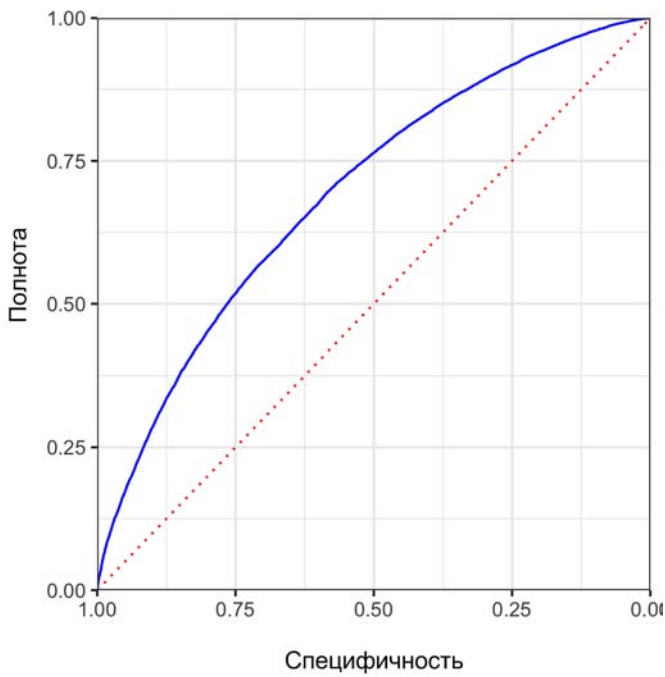


Рис. 5.6. ROC-кривая для данных о ссудах

Кривая прецизионности — полноты

В дополнение к ROC-кривым в информативных целях может быть полезным обследовать *кривую прецизионности — полноты* (precision–recall, PR). PR-кривые вычисляются аналогичным образом за исключением того, что данные упорядочены от наименьшей до наибольшей вероятности, и вычисляются совокупные меры прецизионности и полноты. PR-кривые в особенности полезны в оценивании данных с очень несбалансированными исходами.

Метрический показатель AUC

ROC-кривая является ценным графическим инструментом, но как таковая не представляет единственную меру результативности классификатора. ROC-кривая может использоваться, тем не менее, для создания метрического показателя AUC (area under the ROC curve). Метрический показатель AUC — это просто общая площадь под ROC-кривой. Чем больше значение AUC, тем эффективнее классификатор. AUC, равный 1, говорит об идеальном классификаторе: он правильно идентифицирует все 1 и не идентифицирует неправильно любые 0, как 1.

Абсолютно неэффективный классификатор — диагональная линия — будет иметь AUC, равный 0,5.

На рис. 5.7 показана область под ROC-кривой для модели ссуды. Значение AUC может быть вычислено путем численного интегрирования:

```
sum(roc_df$recall[-1] * diff(1-roc_df$specificity))
[1] 0.5924072
```

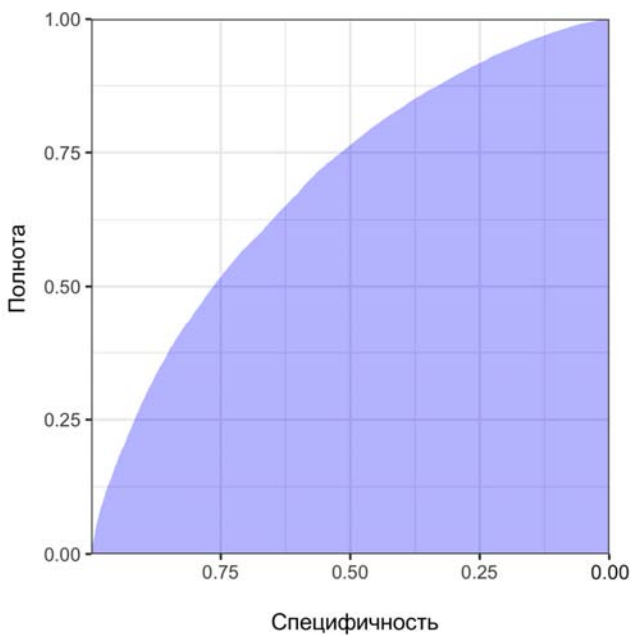


Рис. 5.7. Область под ROC-кривой (AUC) для данных ссуды

Данная модель имеет AUC порядка 0,59, что соответствует относительно слабому классификатору.

Путаница с долей ложноположительных исходов

Доли ложноположительных/ложноотрицательных исходов часто путают или объединяют в одну категорию со специфичностью или чувствительностью (даже в публикациях и программном обеспечении!). Иногда доля ложноположительных исходов определяется, как доля истинноотрицательных, которые дают положительный результат. Во многих случаях (таких как обнаружение вторжения в сеть) данный термин используется для обозначения доли положительных сигналов, которые суть истинноотрицательные.

Лифт

Использование значения AUC как метрического показателя является улучшением по сравнению с простой точностью, поскольку данный показатель может диагностировать, насколько хорошо классификатор обрабатывает компромисс между общей точностью и потребностью идентифицировать более важные единицы. Однако он не полностью решает проблему редкого случая, где необходимо понизить пороговую вероятность модели ниже 0,5, чтобы предотвратить идентификацию всех записей, как 0. В таких случаях, чтобы классифицировать запись как 1, может быть достаточным иметь вероятность 0,4; 0,3 или ниже. В результате мы приходим к тому, что сверхидентифицируем единицы, отражая их бóльшую важность.

Изменение этого порога отсечения повысит шансы охватить единицы (за счет неправильной классификации большего числа нулей, как единиц). Но каким является оптимальный порог отсечения?

Понятие лифта позволяет отложить ответ на этот вопрос. Вместо этого записи рассматриваются в порядке их предсказанной вероятности принадлежать к категории единиц. Скажем, насколько лучше алгоритм сработал в верхних 10%, идентифицированных как единицы, в сравнении с критерием, когда класс просто отбирается вслепую? Если в этом верхнем дециле можно получить 0,3% отклика вместо 0,1% отклика, который вы получаете в целом, подбирая в произвольном порядке, то говорят, что алгоритм имеет *лифт* (или *прирост*, от англ. *gains*), *равный* 3 в верхнем дециле. График лифта (график прироста) квантифицирует это на диапазоне данных. Такой график можно построить подецильно либо непрерывно на диапазоне данных.

Для того чтобы вычислить график лифта, сначала строят график *совокупного прироста*, который показывает полноту на оси *y* и общее число записей — на оси *x*. *Кривая лифта* — это соотношение совокупного прироста к диагональной линии, соответствующей случайному выбору. *Графики подецильного прироста* — это один из самых старых приемов в предсказательном моделировании, датируемый периодом до появления интернет-коммерции. Они были особенно популярны среди профессионалов продажи товаров по почте. Такой вид продаж представляет собой дорогой метод рекламы, если его применять неразборчиво, и рекламодатели ис-

пользовали предсказательные модели (довольно простые в первые годы) для идентификации потенциальных клиентов с наиболее вероятной перспективой оплаты.

Надбавка

Иногда термин *uplift* (надбавка) используется для обозначения того же самого, что и лифт. Альтернативное значение используется в более строгих условиях, когда был проведен *A/B*-тест, и далее в предсказательной модели в качестве предикторной переменной используется вариант (*A* или *B*). Надбавка — это улучшение в отклике, предсказанном для *отдельного случая* с вариантом *A* против варианта *B*. Она определяется путем оценивания результата отдельного случая, сначала, когда предиктор установлен в *A*, и затем снова, когда предиктор переустановлен в *B*. Маркетологи и консультанты политических кампаний используют этот метод для определения, какой из двух вариантов послания должен использоваться в отношении каких клиентов или избирателей.

Кривая лифта позволяет посмотреть на последствия установки разных порогов вероятности для классификации записей как единиц. Он может быть промежуточным шагом в окончательном решении остановиться на подходящем пороге отсечения. Например, налоговый орган может иметь лишь ограниченный объем ресурсов, которые может потратить на налоговые аудиты, и хочет потратить их на наиболее вероятные случаи налогового мошенничества. Учитывая ограниченность ресурсов, руководство воспользуется графиком лифта для оценки, где прочертить линию между налоговыми декларациями, отобранными для аудита и отставленными в сторону.

Ключевые идеи для оценивания моделей классификации

- Точность (процент предсказанных идентификаций, которые являются правильными) следует рассматривать всего лишь как первый шаг в оценке модели.
- Другие метрические показатели (полнота, специфичность, прецизионность) сосредоточены на более специфических характеристиках результативности (например, полнота измеряет, насколько хорошо модель справляется в правильной идентификации единиц).
- AUC (область под ROC-кривой) — это общепринятый метрический показатель способности модели отличать единицы от нулей.
- Аналогичным образом, лифт измеряет, насколько эффективна модель в идентификации единиц, и он часто вычисляется подецильно, начиная с наиболее вероятных единиц.

Дополнительные материалы для чтения

Оценивание и диагностирование, как правило, рассматриваются в контексте той или иной модели (например, *K* ближайших соседей или деревьев решений). Далее приведены три книги, которые излагают эту тему в отдельных главах.

- "Глубинный анализ данных" (Whitten I., Frank E., Hall M. Data Mining. — 3rd ed. — Morgan Kaufmann, 2011).
- "Современная наука о данных с R" (Baumer B., Kaplan D., Horton N. Modern Data Science with R. — CRC Press, 2017).
- "Глубинный анализ данных для бизнес-аналитики" (Shmueli G., Bruce P., Patel N. Data mining for business analytics. — 3rd ed. — John Wiley & Sons, 2016) с вариантами для R, Excel и JMP.

Превосходный материал по перекрестной проверке и повторному отбору можно найти в книге "Введение в статистическое обучение" (Gareth J., Witten D., Hastie T., Tibshirani R. An Introduction to Statistical Learning. — Springer, 2013).

Стратегии в отношении несбалансированных данных

В предыдущем разделе рассматривалась оценка моделей классификации при помощи метрических показателей, которые выходят за пределы простой точности и годятся для несбалансированных данных — данных, в которых целевой исход (покупают на веб-сайте, страховое мошенничество и т. д.) редок. В данном разделе мы обратимся к дополнительным стратегиям, которые могут улучшить результативность предсказательного моделирования с несбалансированными данными.

Ключевые термины

Понижающая выборка (undersample)

В модели классификации использовать меньше записей с преобладающим классом.

Повышающая выборка (oversample)

В модели классификации использовать больше записей с редкими классами, при необходимости прибегая к помощи бутстрапирования.

Повышающая или понижающая перевесовка (up weight or down weight)

Назначать больший (или меньший) вес редкому (или преобладающему) классу в модели.

Генерация данных (data generation)

Процедура, аналогичная бутстрапированию, за исключением того, что каждая новая бутстраповская запись немного отличается от своего источника.

***Z*-оценка (*z*-score)**

Значение, которое получается после стандартизации.

K

Число соседей, учитываемых при вычислении ближайших соседей.

Понижающий отбор

Если данных достаточно, как в случае с данными о ссудах, одно из решений состоит в *понижающем отборе* преобладающего класса, в результате чего моделируемые данные становятся более сбалансированными между нулями и единицами. Основная идея понижающего отбора состоит в том, что данные для доминирующего класса имеют много избыточных записей. Работа с меньшим, более сбалансированным набором данных приводит к преимуществам в результативности модели и упрощает подготовку данных, а также обследование и апробирование экспериментальных моделей.

Какое количество данных будет достаточным? Это зависит от приложения, но в целом, наличие десятков тысяч записей для менее доминирующего класса будет достаточным. Чем легче различимы единицы и нули, тем меньше необходимо данных.

Данные о ссудах, проанализированные в *разд. "Логистическая регрессия" ранее в этой главе* были основаны на сбалансированном тренировочном наборе: половина ссуд была погашена, а другая половина не возвращена. Предсказанные значения были схожими: половина вероятностей была меньше 0,5, а половина — больше 0,5. В полном наборе данных всего лишь порядка 5% ссуд были невозвратными:

```
mean(loan_all_data$outcome == 'default')
[1] 0.05024048
```

Что происходит, если использовать полный набор данных для тренировки модели?

```
full_model <- glm(outcome ~ payment_inc_ratio + purpose_ +
                        home_ + emp_len_+ dti + revol_bal + revol_util,
                      data=train_set, family='binomial')
pred <- predict(full_model)
mean(pred > 0)
[1] 0.00386009
```

Только 0,39% ссуд были предсказаны, что они будут невозвратными, или менее 1/12 ожидаемого числа. Погашенные ссуды подавляют количеством невозвращенные ссуды, потому что модель натренирована с использованием всех данных одинаково. Если подумать об этом на интуитивном уровне, то присутствие такого количества ссуд, не относящихся к невозвратным, в сопряжении с неизбежной вариабельностью в предикторных данных означает, что даже для невозвратной ссуды модель скорее всего найдет несколько не относящихся к невозвратным ссуд, с которыми они будут схожи случайным образом. Когда же использовалась сбалансированная выборка, примерно 50% ссуд были предсказаны как невозвратные.

Повышающий отбор и повышающая/понижающая перевесовка

Одним из критических замечаний по поводу метода понижающего отбора записей является то, что он отбрасывает данные и не использует всю имеющуюся под рукой информацию. Если вы имеете относительно небольшой набор данных, и более ред-

кий класс содержит несколько сотен или несколько тысяч записей, тогда понижающий отбор доминирующего класса имеет риск выбросить полезную информацию. В этом случае вместо понижающего отбора доминирующего случая следует выполнить повышающий отбор более редкого класса путем выемки дополнительных строк с возвратом (бутстрапирование).

Вы можете добиться аналогичного эффекта путем взвешивания данных. Многие алгоритмы классификации принимают весовой аргумент, который позволит вам выполнить повышающую/понижающую перевесовку данных. Например, применить весовой вектор к данным о ссудах при помощи аргумента `weight` к `glm`:

```
wt <- ifelse(loan_all_data$outcome=='default',
             1/mean(loan_all_data$outcome == 'default'), 1)
full_model <- glm(outcome ~ payment_inc_ratio + purpose_ +
                    home_ + emp_len_+ dti + revol_bal + revol_util,
                  data=loan_all_data, weight=wt, family='binomial')
pred <- predict(full_model)
mean(pred > 0)
[1] 0.4344177
```

Веса для невозвратных ссуд установлены в $1/p$, где p — это вероятность невозврата. Ссуды, не относящиеся к невозвратным, имеют вес 1. Суммы весов для невозвратных ссуд и ссуд, не относящихся к невозвратным, примерно эквивалентны. Среднее предсказанных значений теперь составляет 43% вместо 0,39%.

Отметим, что перевесовка обеспечивает альтернативу как для повышающего отбора более редкого класса, так и для понижающего отбора доминирующего класса.

Адаптация функции потерь

Многие алгоритмы классификации и регрессии оптимизируют определенный критерий, или *функцию потерь*. Например, логистическая регрессия пытается минимизировать погрешность. В специализированной литературе некоторые разработчики предлагают модифицировать функцию потерь, чтобы предотвратить проблемы, вызываемые редким классом. На практике это сделать трудно: алгоритмы классификации могут быть многосложными, и их трудно модифицировать. Перевесовка является простым способом внесения изменений в функцию потерь, обесценивая ошибки для записей с низкими весами в пользу записей с более высокими весами.

Генерация данных

Генерация данных — это вариант повышающего отбора записей посредством бутстрапирования *(см. разд. "Понижающий отбор" ранее в этой главе)* с перебором существующих записей для создания новых записей. Лежащая в основе этой идеи логика состоит в том, что поскольку мы наблюдаем только предельный набор случаев, алгоритм не имеет богатого набора информации для создания "правил" классификации. Путем создания новых записей, которые аналогичны, но не идентичны существующим записям, алгоритм имеет возможность научиться более робастному

набору правил. Это понятие по духу подобно ансамблевым статистическим моделям, в частности, бэггингу и бустингу *(см. главу 6)*.

Данная идея набрала обороты с публикацией алгоритма SMOTE, название которого расшифровывается, как методика повышающего отбора синтетического меньшинства (synthetic minority oversampling technique). Алгоритм SMOTE находит запись, которая аналогична записи, подвергаемой повышающему отбору *(см. разд. "К ближайших соседей" главы 6)*, и создает синтетическую запись, которая является произвольным образом взвешенным средним исходной записи и соседней записи, где вес генерируется отдельно для каждого предиктора. Число синтетических записей, взятых повышающим отбором, зависит от коэффициента повышающего отбора, который требуется для приведения набора данных в приблизительное равновесие относительно классов исходов.

В R имеется несколько реализаций SMOTE. Самым исчерпывающим программным пакетом для обработки несбалансированных данных является `unbalanced`. Он предлагает разнообразные специализированные приемы, включая гоночный алгоритм ("Racing") для отбора лучшего метода. Однако алгоритм SMOTE достаточно прост и может быть реализован непосредственно в R с использованием пакета `knn`.

Стоимостно-ориентированная классификация

На практике показатели точности и AUC — это упрощенческие приемы выбора правила классификации. Зачастую назначают оценочную стоимость ложноположительным исходам против ложноотрицательных, и более целесообразно включать эти стоимости для определения лучшего порога отсечения при классификации единиц и нулей. Например, предположим, что оценочная стоимость невозврата новой ссуды равна C, а ожидаемый доход от погашенной ссуды равняется R. Тогда ожидаемый доход для этой ссуды составит:

$$\text{ожидаемый доход} = P(Y = 0) \cdot R + P(Y = 1) \cdot C.$$

Вместо того чтобы просто пометить ссуду как невозвратную или погашенную либо определить вероятность невозврата, разумнее определить, имеет ли ссуда положительный ожидаемый доход. Предсказанная вероятность невозврата является промежуточным шагом и должна быть объединена с итоговой стоимостью ссуды для определения ожидаемой прибыли, которая является окончательным плановым метрическим показателем в бизнесе. Например, ссуду с меньшей величиной можно обойти молчанием в пользу более крупной с немного более высокой вероятностью предсказания невозврата.

Обследование предсказаний

Одиночный метрический показатель, такой как AUC, не может охватить все аспекты адекватности модели для конкретной ситуации. На рис. 5.8 отображены правила решения для четырех разных моделей, подогнанных к данным о ссудах, использующим всего две предикторные переменные: `borrower_score` и `payment_inc_ratio`. Моделями являются: линейный дискриминантный анализ (LDA), логистическая

линейная регрессия, логистическая регрессия, подогнанная с использованием обобщенной аддитивной модели (GAM), и древовидная модель *(см. разд. "Древовидные модели" главы 6)*. Область слева вверху от линий соответствует предсказанному невозврату. LDA и логистическая линейная регрессия в этом случае дают почти идентичные результаты. Древовидная модель производит наименее регулярное правило: на деле, имеются ситуации, в которых увеличение балла оценки заемщика смещает предсказание от "погашено" в сторону "невозврат"! И наконец, подгонка логистической регрессии на основе GAM представляет собой компромисс между древовидными и линейными моделями.

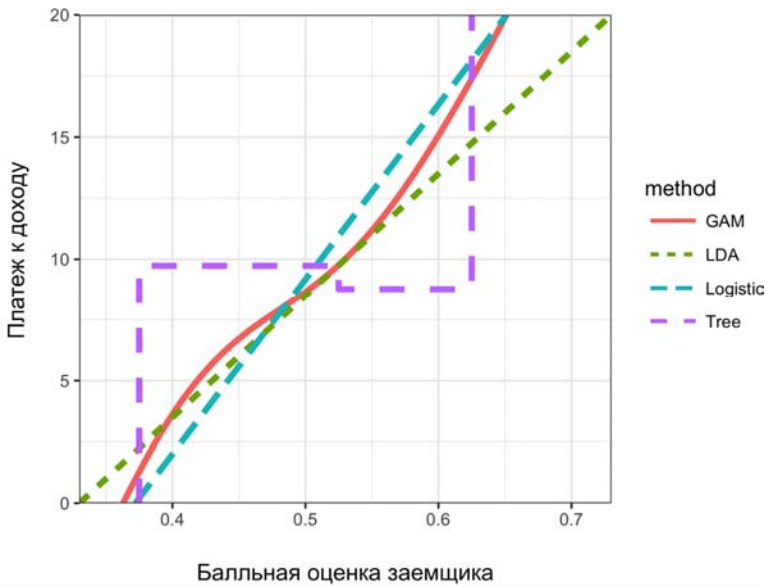


Рис. 5.8. Сравнение правил классификации для четырех разных методов

Визуализировать правила предсказания в более высоких размерностях, или в случае GAM и древовидной модели, даже сгенерировать области для таких правил, очень нелегко.

В любом случае разведочный анализ предсказанных значений всегда оправдан.

Ключевые идеи для стратегий в отношении несбалансированных данных

- Очень несбалансированные данные (т. е. где интересующие исходы, единицы, являются редкими) представляют проблему для алгоритмов классификации.
- Одна из стратегий состоит в том, чтобы сбалансировать тренировочные данные посредством понижающего отбора многочисленного случая (либо повышающего отбора редкого случая).

- Если использование всех единиц по-прежнему дает вам слишком мало единиц, то можно применить бутстрапирование редких случаев либо использовать алгоритм SMOTE для создания синтетических данных, подобных существующим редким случаям.
- Несбалансированные данные обычно говорят о том, что правильная идентификация одного класса (единиц) имеет более высокую стоимость, и что в диагностический метрический показатель следует встроить стоимостной коэффициент.

Дополнительные материалы для чтения

- У Тома Фосетта (Tom Fawcett), автора книги "Наука о данных для бизнеса" (Data Science for Business), есть хорошая статья о несбалансированных классах (**https://svds.com/learning-imbalanced-classes**).
- Подробности об алгоритме SMOTE см. в статье "Метод повышающего отбора синтетического меньшинства" (Chawla N. V., Bowyer K. W, Hall L. O., Kegelmeyer W. P.. SMOTE: Synthetic Minority Over-sampling Technique // Journal of Artificial Intelligence Research. — 2002. — № 16. — P. 321–357).
- См. также "Практическое руководство по решению проблем несбалансированной классификации в R" (Practical Guide to deal with Imbalanced Classification Problems in R // 2016. — March. — 28) от группы Analytics Vidya Content Team.

Резюме

Классификация, т. е. процесс предсказания, к какой из двух категорий (или из небольшого количества категорий) запись принадлежит, — это фундаментальный инструмент предсказательной аналитики. Будет ли ссуда невозвратной (да или нет)? Будет ли она погашена досрочно? Нажмет ли посетитель веб-сайта на ссылке? Купит ли он что-нибудь? Является ли страховая претензия мошеннической? Нередко в задачах классификации один класс представляет главный интерес (например, мошенническая страховая претензия) и, в бинарной классификации этот класс получает значение 1, а другой более преобладающий (многочисленный) класс — значение 0. Нередко ключевая часть процесса состоит в исчислении *балльной оценки склонности*, вероятности принадлежать целевому классу. Общепринятый сценарий — это когда целевой класс относительно редок. Этой главой мы заканчиваем обсуждение разнообразных метрических показателей диагностики модели, которые выходят за пределы простой точности; они важны в ситуации наличия редкого класса, когда идентификация всех записей как нулей может привести к высокой точности.

Статистическое машинное обучение

Недавние достижения в статистике были посвящены разработке более мощных автоматизированных приемов в области предсказательного моделирования — как регрессии, так и классификации. Эти приемы являются составной частью более общей методологии *статистического машинного обучения* и отличаются от классических статистических методов тем, что они управляемы данными и не стремятся описать данные линейной или иной общей функцией. Метод *K* ближайших соседей, например, довольно прост: он классифицирует запись в соответствии с тем, насколько записи схожи. Самые успешные и широко используемые приемы опираются на *ансамблевое обучение* применительно к *деревьям решений*. Основная идея ансамблевого обучения состоит в том, чтобы для формирования предсказания использовать много моделей в отличие от одной-единственной модели. Деревья решений — это гибкий и автоматический прием, предназначенный для того, чтобы обучаться правилам о связях между предикторными переменными и переменными исходов. Оказывается, что комбинация ансамблевого обучения с деревьями решений приводит к высокорезультативным стандартным приемам предсказательного моделирования.

Разработку многих из этих приемов в статистическом машинном обучении можно проследить до статистиков Лео Бреймана (Leo Breiman) (рис. 6.1) в Калифорнийском университете в Беркли и Джерри Фридмана (Jerry Friedman) в Стэнфордском университете. Их совместная с другими исследователями работа в Беркли и Стэнфорде началась в 1984 г. с создания древовидных моделей. Последующая в 1990-х го-

Рис. 6.1. Лео Брейман, работавший преподавателем статистики в Беркли, находился на передовой линии разработки многих приемов, лежащих в основе инструментария аналитика данных

дах разработка ансамблевых методов бэггинга и бустинга установила фундамент статистического машинного обучения.

Машинное обучение против статистики

Какова разница между машинным обучением и статистикой в контексте предсказательного моделирования? Четкой разграничительной линии, которая разделяет эти две дисциплины, нет. Машинное обучение имеет тенденцию уделять больше внимания разработке эффективных алгоритмов, которые масштабируются до больших данных в целях оптимизации предсказательной модели. Статистика обычно больше сосредоточена на теории вероятностей и глубинной структуре модели. Бэггинг и случайный лес *(см. разд. "Бэггинг и случайный лес" далее в этой главе)* выросли, прочно опираясь на статистику. Бустинг *(см. разд. "Бустинг" далее в этой главе)*, с другой стороны, был разработан в обеих дисциплинах, но получает больше внимания на стороне машинного обучения. Независимо от истории, перспективы бустинга гарантируют, что этот метод будет процветать и в статистике, и в машинном обучении.

K ближайших соседей

В основе метода *K* ближайших соседей (KNN, от англ. *k-nearest neighbors*) лежит очень простая идея[1]. Для каждой записи, которая будет классифицирована или предсказана:

1. Найти *K* записей, которые имеют схожие признаки (т. е. схожие значения предикторов).
2. Для классификации: выяснить среди этих схожих записей мажоритарный класс и назначить этот класс новой записи.
3. Для предсказания (также именуемой KNN-регрессией): найти среди этих схожих записей среднее и предсказать это среднее для новой записи.

Ключевые термины

Сосед (neighbor)

Запись, чьи предикторные значения схожи с другой записью.

Метрические показатели расстояния (distance metrics)

Метрические показатели, которые обобщают в одном числе, насколько далеко одна запись находится от другой.

Стандартизация (standardization)

Вычесть среднее и разделить на стандартное отклонение.

Синоним: нормализация.

[1] Этот и последующие разделы в настоящей главе используются с разрешения © 2017 Datastats, LLC, Питер Брюс и Эндрю Брюс.

***Z*-оценка (*z*-score)**

Значение, которое получается после стандартизации.

Синоним: стандартная оценка.

K

Число соседей, учитываемых при вычислении алгоритма ближайших соседей.

KNN — это один из более простых приемов предсказания/классификации: модель, подлежащая подгонке (как в регрессии), отсутствует. Это не означает, что использование KNN является автоматической процедурой. Результаты предсказания зависят от того, каким образом признаки прошкалированы, каким образом измерено сходство и какая задана величина *K*. Кроме того, все предикторы должны быть в числовой форме. Проиллюстрируем работу данного метода примером классификации.

Небольшой пример: предсказание невозврата ссуды

В табл. 6.1 представлены несколько записей данных о персональных ссудах в инвестиционно-кредитной компании Lending Club. Компания Lending Club является лидером в равноправном кредитовании, в котором пулы инвесторов выдают персональные ссуды физическим лицам. Цель анализа будет состоять в том, чтобы предсказать исход новой потенциальной ссуды: погашено против невозврат.

Таблица 6.1. *Несколько записей и столбцов из данных о ссудах инвестиционно-кредитной компании Lending Club*

Исход	Величина ссуды	Доход	Цель	Стаж работы	Домовладение	Штат
Погашено	10 000	79 100	Консолидация долга	11	ИПОТЕКА	NV
Погашено	9600	48 000	Переезд	5	ИПОТЕКА	TN
Погашено	18 800	120 036	Консолидация долга	11	ИПОТЕКА	MD
Невозврат	15 250	232 000	Малый бизнес	9	ИПОТЕКА	CA
Погашено	17 050	35 000	Консолидация долга	4	АРЕНДА	MD
Погашено	5500	43 000	Консолидация долга	4	АРЕНДА	KS

Рассмотрим очень простую модель всего с двумя предикторными переменными: `dti`, т. е. соотношением выплат по задолженности (исключая ипотеку) к доходу, и `payment_inc_ratio`, т. е. соотношением выплат по ссуде к доходу. Оба соотношения умножены на 100. На основе небольшого набора из 200 ссуд `loan200` с известными бинарными исходами (невозврат или его противоположность, заданных в предикторе `outcome200`) и *K*, установленным в 20, оценка `newloan` новой ссуды, подлежащей

предсказанию, с `dti=22.5` и `payment_inc_ratio=9` алгоритмом KNN может быть вычислена в R следующим образом:

```
library(FNN)
knn_pred <- knn(train=loan200, test=newloan, cl=outcome200, k=20)
knn_pred == 'default'
[1] TRUE
```

Предсказание KNN — ссуда не будет возвращена.

В то время как R имеет собственную функцию `knn`, сторонний программный R-пакет `FNN` (fast nearest neighbor — *быстрый ближайший сосед*) масштабирует до больших данных лучше и предоставляет больше гибкости.

На рис. 6.2 дано визуальное отображение этого примера. Новая ссуда, подлежащая предсказанию, представлена квадратом в центре. Круги (невозврат) и крестики (погашено) — это тренировочные данные. Овал показывает границу самых близких 20 точек. В этом случае 14 невозвратных ссуд лежат внутри овала по сравнению всего с 6 погашенными ссудами. Следовательно, предсказанный исход ссуды — невозврат.

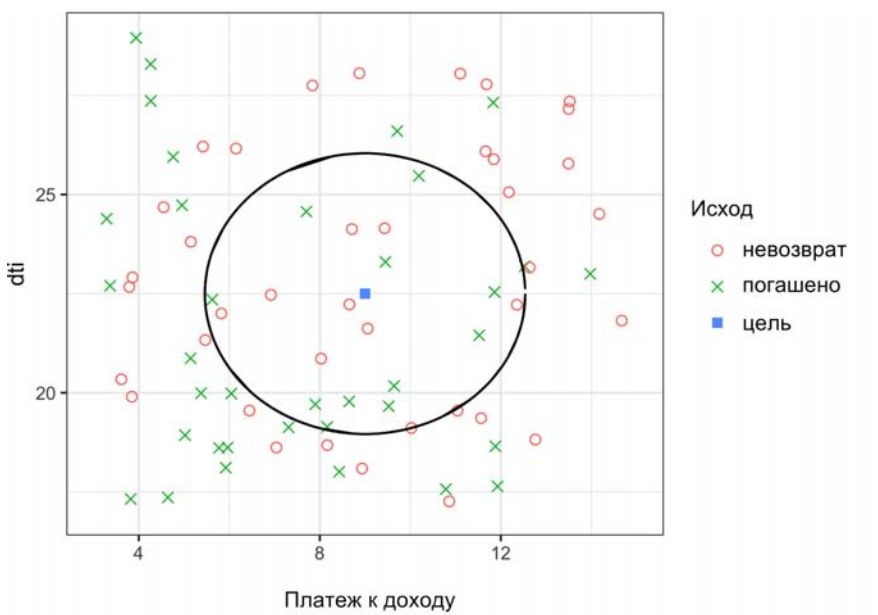


Рис. 6.2. Предсказание алгоритма KNN невозврата ссуды с использованием двух переменных: соотношение задолженности к доходу и соотношение платежей по ссуде к доходу

Хотя на выходе из KNN, предназначенном для классификации, как правило, будет бинарное решение, в частности, невозврат или погашено для данных о ссудах, подпрограммы KNN обычно предлагают возможность показывать на выходе вероятность (склонность) между 0 и 1. Вероятность основывается на доле класса единиц в *K* самых близких соседях. В предыдущем примере эта

вероятность невозврата была бы оценена в 14/20, или 0,7. Использование балльной оценки вероятности позволяет применять иные правила классификации, чем простое мажоритарное голосование (вероятность 0,5). Это имеет особое значение в задачах с несбалансированными классами *(см. разд. "Стратегии в отношении несбалансированных данных" главы 5*. Например, если цель состоит в идентификации членов редкого класса, то порог отсечения, как правило, будет установлен ниже 50%. Один из общепринятых подходов состоит в установке порога на уровне вероятности редкого случая

Метрические показатели расстояния

Сходство (близость) определяется при помощи *метрического показателя расстояния*, т. е. функции, которая измеряет, насколько далеко находятся две записи $(x_1, x_2, ..., x_p)$ и $(u_1, u_2, ..., u_p)$ друг от друга. Самым популярным метрическим показателем расстояния между двумя векторами является *евклидово расстояние*. Для того чтобы измерить евклидово расстояние между двумя векторами, надо вычесть один из другого, разности возвести в квадрат, просуммировать их и взять квадратный корень:

$$\sqrt{(x_1 - u_1)^2 + (x_2 - u_2)^2 + ... + (x_p - u_p)^2}.$$

Евклидово расстояние предлагает особые вычислительные преимущества. Это в особенности важно для крупных наборов данных, поскольку KNN предполагает $K \times n$ попарных сравнений, где n — это число строк.

Еще одним общепринятым метрическим показателем для числовых данных является *манхэттенское расстояние* (или расстояние Минковского):

$$|x_1 - u_1| + |x_2 - u_2| + ... + |x_p - u_p|.$$

Евклидово расстояние соответствует расстоянию по прямой между двумя точками (как говорится, "так, как летит ворон"). Манхэттенское расстояние — это расстояние между двумя точками, пересекаемыми в одном направлении за один раз (например, перемещаясь вдоль прямоугольных городских кварталов). По этой причине манхэттенское расстояние является полезным приближением, в случае если сходство определяется как поточечное время в пути.

В измерении расстояния между двумя векторами переменные (признаки), которые измеряются по сравнительно крупной шкале, будут над этой мерой доминировать. Например, для данных о ссудах расстояние было бы почти исключительно функцией от переменных дохода и суммы кредита, которые измеряются в десятках или сотнях тысяч. Переменные на основе соотношений при сравнении будут практически сведены на нет. Эта проблема решается путем стандартизации данных *(см. разд. "Стандартизация (нормализация, z-оценка)" далее в этой главе)*.

Другие метрические показатели расстояния

Для измерения расстояния между векторами существуют другие многочисленные метрические показатели. Для числовых данных привлекательным является *расстояние Махаланобиса*, поскольку оно объясняет корреляцию между двумя переменными. Это полезно, т. к. если две переменные высоко коррелированы, то с точки зрения расстояния Махаланобиса они в сущности будут восприниматься как одна переменная. Евклидово и манхэттенское расстояния не объясняют корреляцию, практически возлагая больший вес на атрибут, который лежит в основе этих признаков. Оборотной стороной применения расстояния Махаланобиса является увеличение вычислительных усилий и сложности; оно вычисляется при помощи *ковариационной матрицы (см. разд. "Ковариационная матрица" главы 5)*.

Кодировщик с одним активным состоянием

Данные о ссудах в табл. 6.1 включают несколько факторных (строковых) переменных. Большинство моделей статистического и машинного обучения требует, чтобы этот тип переменной был преобразован в серию двоичных фиктивных переменных, несущих одинаковую информацию, как в табл. 6.2. Вместо одной переменной, обозначающей статус домовладельца: "владеет с ипотекой", "владеет без ипотеки", "арендует" или "другой", мы в конечном итоге приходим к четырем двоичным переменным. Первая будет "владеет с ипотекой — Y/N", вторая будет "владеет без ипотеки — Y/N" и т. д. Этот один предиктор, статус домовладельца, таким образом, порождает вектор с одной 1 и тремя 0, который может использоваться в алгоритмах статистического и машинного обучения. Словосочетание *"кодирование с одним активным состоянием"* (one hot encoding) пришло из терминологии цифровых интегральных микросхем, где оно описывает конфигурацию микросхемы, в которой допускается, чтобы только один бит был положительным (активным)[2].

Таблица 6.2. *Представление факторных данных о домовладельце с помощью числовой фиктивной переменной*

ИПОТЕКА	ДРУГОЕ	ВЛАДЕЛЕЦ	АРЕНДА
1	0	0	0
1	0	0	0
1	0	0	0
1	0	0	0
0	0	0	1
0	0	0	1

[2] В отечественной специализированной литературе для данного типа кодировщика нередко используется альтернативный термин *"прямой унитарный кодировщик". — Прим. пер.*

В линейной и логистической регрессиях кодирование с одним активным состоянием вызывает проблемы, связанные с мультиколлинеарностью *(см. разд. "Мультиколлинеарность" главы 4)*. В таких случаях одна фиктивная переменная исключается (а ее значение может быть выведено из других значений). Это не представляет проблемы с KNN и другими методами.

Стандартизация (нормализация, *z*-оценки)

В данных, полученных в результате измерений, нас часто в первую очередь интересует не их величина, а насколько они отличаются от среднего. Процедура стандартизации, или *нормализации*, помещает все переменные на аналогичные шкалы путем вычитания среднего и деления на стандартное отклонение. Таким образом мы гарантируем, что переменная чрезмерно не влияет на модель просто в силу шкалы ее исходного измерения.

$$z = \frac{x - \bar{x}}{s}.$$

Полученные в результате стандартизации величины принято называть *стандартными оценками*, или *z-оценками*. Данные измерений в дальнейшем используются в "стандартных отклонениях от среднего". Таким образом, влияние переменной на модель не затрагивается шкалой ее исходного измерения.

Нормализацию в данном статистическом контексте не следует путать с *нормализацией баз данных*, т. е. удалением избыточных данных и верификацией зависимостей в данных.

Для KNN и нескольких других процедур (например, анализа главных компонент и кластеризации) крайне важно учитывать стандартизацию данных до применения процедуры. Для того чтобы проиллюстрировать эту идею, KNN применяется к данным о ссудах с использованием `dti` и `payment_inc_ratio` *(см. разд. "Небольшой пример: предсказание невозврата ссуды" ранее в этой главе)* плюс двух других переменных: `revol_bal` — общего возобновляемого кредита, доступного для заявителя в долларах, и `revol_util` — процента используемого кредита. Новая предсказываемая запись показана ниже:

```
newloan
  payment_inc_ratio dti revol_bal revol_util
1            2.3932   1      1687        9.4
```

Величина `revol_bal`, которая исчисляется в долларах, намного больше другой переменной. Функция `knn` возвращает индекс самых близких соседей, как атрибут `nn.index`, и он может использоваться для показа верхних пяти самых близких строк в кадре данных `loan_df`:

```
loan_df <- model.matrix(~ -1 + payment_inc_ratio + dti + revol_bal +
                          revol_util, data=loan_data)
```

```
knn_pred <- knn(train=loan_df, test=newloan, cl=outcome, k=5)
loan_df[attr(knn_pred,"nn.index"),]
      payment_inc_ratio  dti revol_bal revol_util
36054           2.22024 0.79      1687        8.4
33233           5.97874 1.03      1692        6.2
28989           5.65339 5.40      1694        7.0
29572           5.00128 1.84      1695        5.1
20962           9.42600 7.14      1683        8.6
```

Значение `revol_bal` в этих соседях очень близко к его значению в новой записи, но другие предикторные переменные идут вразброс и по существу не играют роли в определении соседей.

Сравним это с KNN, примененным к стандартизированным данным при помощи R-функции `scale`, которая вычисляет *z*-оценку для каждой переменной:

```
loan_std <- scale(loan_df)
knn_pred <- knn(train=loan_std, test=newloan_std, cl=outcome, k=5)
loan_df[attr(knn_pred,"nn.index"),]
      payment_inc_ratio  dti revol_bal revol_util
2081            2.61091 1.03      1218        9.7
36054           2.22024 0.79      1687        8.4
23655           2.34286 1.12       523       10.7
41327           2.15987 0.69      2115        8.1
39555           2.76891 0.75      2129        9.5
```

Пять ближайших соседей гораздо больше похожи во всех переменных, обеспечивая более разумный результат. Отметим, что результаты отображены в исходной шкале, но KNN был применен к прошкалированным данным и предсказываемой новой ссуде.

Использование *z*-оценки — это всего лишь один способ перемасштабирования переменных. Вместо среднего может использоваться более робастная оценка центрального положения, такая как медиана. Похожим образом может использоваться и другая оценка шкалы, такая как межквартильный размах, вместо стандартного отклонения. Иногда переменные "впихивают" в диапазон 0–1. Также важно понять, что шкалирование каждой переменной до единичной дисперсии носит несколько произвольный характер. Это подразумевает, что каждая переменная, предположительно, имеет одинаковую важность в предсказательной силе. Если вы располагаете субъективным знанием, что какие-то переменные важнее других, тогда их можно прошкалировать вертикально. Например, если говорить о данных о ссудах, разумно ожидать, что бóльшую важность представляет соотношение платежей к доходу.

Нормализация (стандартизация) не меняет форму распределения данных; она не придает им нормальную форму, если только они ее уже не имеют *(см. разд. "Нормальное распределение" главы 2)*.

Выбор *K*

Выбор *K* имеет чрезвычайно важное значение для результативности KNN. Самый простой выбор состоит в том, чтобы установить $K=1$, что соответствует классификатору 1-го ближайшего соседа. Предсказание интуитивно понятно: оно основывается на нахождении в тренировочном наборе записи, наиболее схожей с новой предсказываемой записью. Принятие за основу $K=1$ редко является лучшим выбором; вы почти всегда будете получать превосходную результативность, используя $K>1$ ближайших соседей.

Вообще говоря, если значение *K* слишком низкое, то мы можем вызвать переподгонку: включив в модель шум в данных. Более высокие значения *K* обеспечивают сглаживание, которое снижает риск переподгонки в тренировочных данных. С другой стороны, если *K* слишком высокое, то мы можем вызвать излишнее сглаживание данных и упустить способность KNN захватывать локальную структуру в данных — одно из его главных преимуществ.

Значение *K*, которое лучше балансирует между переподгонкой и сверхсглаживанием, как правило, определяется точностными метрическими показателями и, в частности, точностью на основе контрольной выборки с отложенными данными или перекрестной проверки. Нет никакого общего правила относительно лучшего значения *K* — все зависит главным образом от природы данных. Для высоко структурированных данных с небольшим шумом меньшие значения *K* работают лучше всего. Заимствуя термин из области обработки сигналов, этот тип данных иногда называют данными с высоким соотношением "сигнал/помеха" (SNR, signal-to-noise ratio). Примерами данных с высоким SNR, как правило, являются данные для распознавания почерка и речи. Для шумных данных с меньшей структурированностью (данных с низким SNR), таких как данные о ссудах, уместными являются более крупные значения *K*. Как правило, значения *K* попадают в диапазон от 1 до 20. Нередко выбирается нечетное число, чтобы избежать равенства голосов при голосовании.

Компромисс между смещением и дисперсией

Разность потенциалов между сверхсглаживанием и переподгонкой является вариантом *компромисса между смещением и дисперсией*, повсеместно распространенной проблемы в подгонке статистических моделей. Дисперсия обозначает ошибку моделирования, которая происходит из-за выбора тренировочных данных; т. е. если бы вы решите выбрать другой набор тренировочных данных, то результирующая модель будет иной. Смещение обозначает ошибку моделирования, которая происходит, потому что вы должным образом не идентифицировали реальный базовый сценарий; эта ошибка не исчезнет, если просто добавить больше тренировочных данных. Когда гибкая модель переподогнана, дисперсия увеличивается. Вы можете уменьшить гибкость при помощи более простой модели, но смещение может увеличиться из-за потери гибкости в моделировании реальной базовой ситуации. Общий подход к решению этого компромисса лежит через *перекрестную проверку*. Для получения дополнительной информации *см. разд. "Перекрестная проверка" главы 4*.

Метод KNN как конструктор признаков

Метод KNN получил свою популярность из-за его простоты и интуитивно понятной природы. С точки зрения результативности, KNN как таковой обычно не конкурентоспособен по сравнению с более изощренными приемами классификации. При подгонке моделей в практических условиях, однако, KNN может использоваться для добавления "локального знания" в многоэтапном процессе с другими приемами классификации.

1. KNN выполняется на данных, и для каждой записи формируется результат классификации (либо квазивероятность класса).
2. Этот результат добавляется в качестве нового признака к записи, и затем на данных выполняется еще один метод классификации. Исходные предикторные переменные таким образом используются дважды.

Поначалу можно засомневаться, не вызывает ли этот процесс проблему, связанную с мультиколлинеарностью ввиду того, что некоторые предикторы используются им дважды *(см. разд. "Мультиколлинеарность" главы 4)*. Это не является проблемой, поскольку информация, включаемая в модель второго этапа, очень локальна, получена только из нескольких соседних записей и является поэтому не избыточной информацией, а дополнительной.

Такое поэтапное применение KNN можно представить как форму ансамблевого обучения, в котором многочисленные предсказывающие методы моделирования используются в сочетании друг с другом. Его также можно рассматривать как форму конструирования признаков, где цель состоит в том, чтобы получить признаки (предикторные переменные), которые имеют предсказательную силу. Нередко это сопряжено с некоторым ручным анализом данных; KNN предоставляет вполне автоматический способ достижения этого.

Например, рассмотрим данные о жилом фонде округа Кинг. При установлении продажной цены на дом агент по продаже недвижимости будет основывать цену на схожих домах, которые были недавно проданы, так называемых "продажах-аналогах". В сущности, агенты по продаже недвижимости выполняют ручную версию KNN: глядя на продажные цены схожих домов, они могут оценить, за что дом будет продан. Мы можем создать новый признак для статистической модели, которая будет имитировать профессионала в области торговли недвижимостью путем применения KNN к недавним продажам. Предсказываемое значение является продажной ценой, и существующие предикторные переменные могут включать местоположение, общую площадь в кв. футах, тип строения, размер земельного участка и количество спален и ванных комнат. Новая предикторная переменная (признак), которую мы добавляем посредством KNN, — это предиктор KNN для каждой записи (аналогичной продажам-аналогам у агентов по продаже недвижимости). Поскольку предсказываемое значение является числовым, вместо мажоритарного голосования используется среднее K ближайших соседей (так называемая KNN-регрессия).

Аналогичным образом, для данных о ссудах мы можем создать признаки, которые представляют разные стороны процесса выдачи ссуд. Например, приведенный

далее фрагмент кода создает признак, который представляет кредитоспособность заемщика:

```
borrow_df <- model.matrix(~ -1 + dti + revol_bal + revol_util + open_acc +
                          delinq_2yrs_zero + pub_rec_zero, data=loan_data)
borrow_knn <- knn(borrow_df, test=borrow_df, cl=loan_data[, 'outcome'],
                  prob=TRUE, k=10)
prob <- attr(borrow_knn, "prob")
borrow_feature <- ifelse(borrow_knn=='default', prob, 1-prob)
summary(borrow_feature)
   Min. 1st Qu.  Median    Mean 3rd Qu.   Max.
0.0000   0.4000  0.5000  0.5012  0.6000 1.0000
```

Результатом является признак, который предсказывает правдоподобие ситуации, что заемщик не вернет ссуду, опираясь на его кредитную историю.

Ключевые идеи для *K* ближайших соседей

- Метод *K* ближайших соседей (KNN) классифицирует запись путем ее отнесения к классу, которому принадлежат схожие записи.
- Сходство (расстояние) определяется евклидовым расстоянием или другими подобными метрическими показателями.
- Число ближайших соседей, с которыми сравнивается запись, *K*, определяется тем, насколько хорошую результативность алгоритм показывает на тренировочных данных с использованием разных значений *K*.
- Как правило, предикторные переменные стандартизуются, в результате чего переменные с большой шкалой не доминируют над метрическим показателем расстояния.
- В предсказательном моделировании KNN часто используется на первом этапе, и предсказанное значение добавляется в данные в качестве *предиктора* для моделирования на втором (не KNN) этапе.

Древовидные модели

Древовидные модели, так называемые *классификационные и регрессионные деревья* (classification and regression trees, CART)[3], *деревья решений*, или *просто деревья* — это эффективный и популярный метод классификации (и регрессии), первоначально разработанный в 1984 г. Лео Брейманом и др. Древовидные модели и их более мощные потомки *случайные леса* и *бустинг (см. разд. "Бэггинг и случайный лес" и "Бустинг" далее в этой главе)* формируют основание для наиболее широко исполь-

[3] Термин CART является зарегистрированной торговой маркой Salford Systems, которая связана с их конкретной реализацией древовидных моделей.

зуемых и мощных предсказательных инструментов моделирования в науке о данных как для регрессии, так и для классификации.

Ключевые термины

Рекурсивное сегментирование (recursive partitioning)

Многократное разбиение данных на разделы и подразделы с целью создания максимально однородных исходов в каждом итоговом подразделе.

Значение в точке разбиения (split value)

Значение предиктора, которое делит записи на те, где этот предиктор меньше и где он больше значения в точке разбиения.

Узел (node)

В дереве решений или в наборе соответствующих правил ветвления узел — это графическое либо в виде правила представление значения в точке разбиения.

Лист (leaf)

Конец набора правил в формате "если-то", или ветвлений дерева, т. е. правила, которые приводят к листу, обеспечивают одно из правил классификации для любой записи в дереве.

Потеря (loss)

Число неправильных результатов классификации на конкретном этапе в процессе разбиения; чем больше потерь, тем больше разнородность.

Разнородность (impurity)

Степень смешанности классов в подразделе данных (чем больше смешанность, тем больше разнородность).

Синонимы: гетерогенность, нечистота.

Антонимы: однородность, чистота, гомогенность.

Подрезание (pruning)

Процесс поступательного подрезания ветвей полностью выращенного дерева с целью снижения переподгонки.

Древовидная модель — это набор правил импликации вида "если-то-иначе", которые просто понять и реализовать. В отличие от регрессии и логистической регрессии, деревья имеют способность обнаруживать скрытые шаблоны (образы, паттерны), соответствующие сложным взаимодействиям в данных. Вместе с тем, в отличие от KNN или наивного байесовского классификатора, простые древовидные модели могут быть выражены с точки зрения связей между предикторами, которые легко поддаются интерпретации.

Деревья решений в исследовании операций

Термин "деревья решений" имеет другой (и более старый) смысл в теории принятия решений и исследовании операций, где он обозначает процесс анализа решений человеком. В этом смысле точки принятия решения, возможные исходы и их оценочные вероятности располагаются на диаграмме ветвления, и аналитик выбирает путь решения с максимальным математическим ожиданием (т. е. ожидаемым значением).

Простой пример

В R существует два главных программных пакета для подгонки древовидных моделей — `rpart` и `tree`. При помощи пакета `rpart` модель подгоняется к выборке из 3000 записей данных о ссудах с использованием переменных `payment_inc_ratio` и `borrower_score` (*см. разд. "К ближайших соседей" ранее в этой главе* относительно описания данных).

```
library(rpart)
loan_tree <- rpart(outcome ~ borrower_score + payment_inc_ratio,
                   data=loan_data, control = rpart.control(cp=.005))
plot(loan_tree, uniform=TRUE, margin=.05)
text(loan_tree)
```

Результирующее дерево показано на рис. 6.3. Эти правила классификации устанавливаются путем обхода иерархического дерева, начиная в корне, пока не будет достигнут лист.

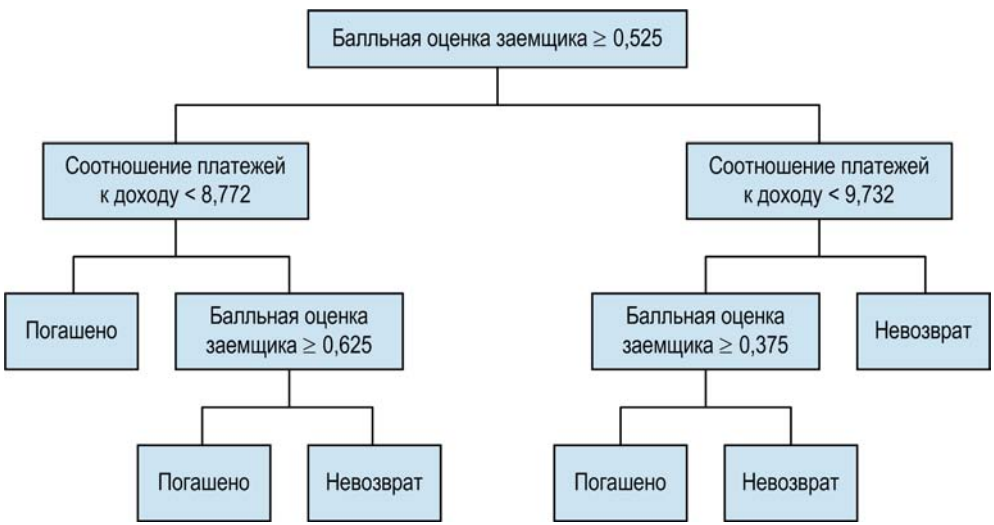


Рис. 6.3. Правила для простой древовидной модели, подогнанной к данным о ссудах

Как правило, дерево отображается в перевернутом виде так, что корень находится вверху, а листья — внизу. Например, если мы получаем ссуду с балльной оценкой заемщика `borrower_score`, равной 0,6, и соотношением платежей к доходу `payment_`

inc_ratio, равным 8,0, то мы приходим к крайнему левому листу и предсказываем, что ссуда будет погашена.

Сгенерировать структурно распечатанную версию дерева тоже не представляет труда:

```
loan_tree
n= 3000

node), split, n, loss, yval, (yprob)
      * denotes terminal node

1) root 3000 1467 paid off (0.5110000 0.4890000)
  2) borrower_score>=0.525 1283 474 paid off (0.6305534 0.3694466)
    4) payment_inc_ratio< 8.772305 845 249 paid off (0.7053254 0.2946746) *
    5) payment_inc_ratio>=8.772305 438 213 default (0.4863014 0.5136986)
      10) borrower_score>=0.625 149 60 paid off (0.5973154 0.4026846) *
      11) borrower_score< 0.625 289 124 default (0.4290657 0.5709343) *
  3) borrower_score< 0.525 1717 724 default (0.4216657 0.5783343)
    6) payment_inc_ratio< 9.73236 1082 517 default (0.4778189 0.5221811)
      12) borrower_score>=0.375 784 384 paid off (0.5102041 0.4897959) *
      13) borrower_score< 0.375 298 117 default (0.3926174 0.6073826) *
    7) payment_inc_ratio>=9.73236 635 207 default (0.3259843 0.6740157) *
```

Глубина дерева показана отступом. Каждый узел соответствует предварительной классификации, определяемой преобладающим исходом в данном сегменте. "Потеря" — это число неправильных результатов классификации, производимое предварительной классификацией в сегменте. Например, в узле 2 было 474 неправильных результатов классификации из общего числа 1467 записей. Значения в круглых скобках представляют собой долю записей соответственно о погашенных и невозвратных ссудах. Например, в узле 13, который предсказывает невозврат, более 60% записей — это невозвратные ссуды.

Алгоритм рекурсивного сегментирования

Алгоритм *рекурсивного сегментирования* для построения дерева решений достаточно прямолинеен и интуитивно понятен. Данные многократно делятся при помощи значений предикторов, которые делают все возможное, чтобы разложить данные на относительно однородные сегменты. На рис. 6.4 представлено изображение сегментов, созданных для дерева на рис. 6.3. Первое правило borrower_score >= 0.525 на графике обозначено под номером 1. Второе правило payment_inc_ratio < 9.732 делит правостороннюю область на две.

Предположим, что у нас есть переменная отклика Y и набор из P предикторных переменных X_j для $j = 1, ..., P$. Для сегмента A с записями алгоритм рекурсивного сегментирования найдет лучший способ разбить A на два подсегмента:

1. Для каждой предикторной переменной X_j:
   - для каждого значения s_j из X_j:
     - отнести записи в A со значениями $X_j < s_j$ в один сегмент и оставшиеся записи, где $X_j \geq s_j$, — в другой сегмент;
     - измерить однородность классов в каждом подсегменте A;
   - выбрать значение s_j, которое порождает максимальную внутрисегментную однородность класса.
2. Выбрать переменную X_j и значение разбиения s_j, которое порождает максимальную внутрисегментную однородность класса.

Теперь наступает очередь рекурсивной части:

1. Инициализировать A всем набором данных.
2. Применить алгоритм сегментирования, чтобы разбить A на два подсегмента, A_1 и A_2.
3. Повторить шаг 2 на подсегментах A_1 и A_2.
4. Алгоритм завершается, когда невозможно создать никакой дальнейший сегмент, который в достаточной мере улучшает однородность сегментов.

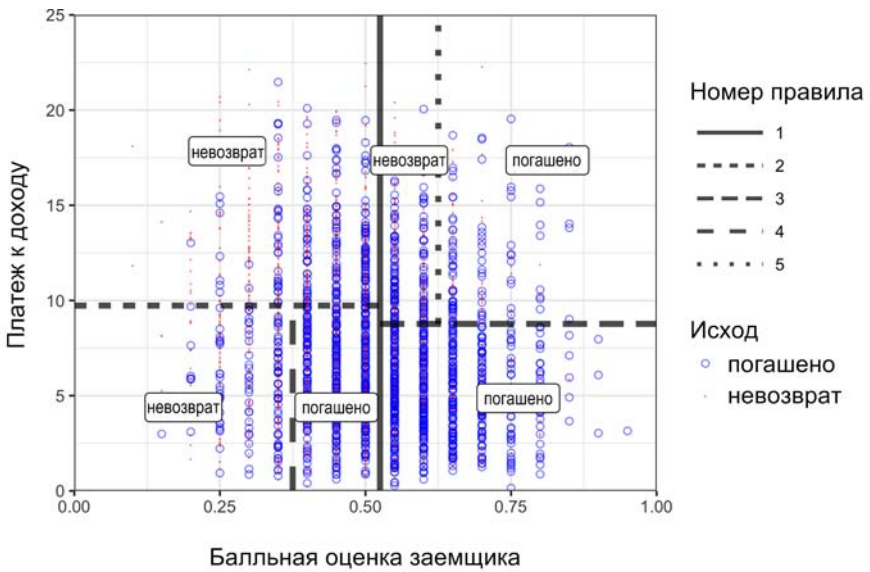


Рис. 6.4. Правила для простой древовидной модели, подогнанной к данным о ссудах

Конечным результатом является сегментирование данных, как на рис. 6.4, за исключением P-размерностей, где каждый сегмент предсказывает результат 0 либо 1 в зависимости от мажоритарного голосования отклика в этом сегменте.

В дополнение к бинарному предсказанию в формате 0/1, древовидные модели могут производить оценку вероятности, основанную на количестве нулей и единиц в сегменте. Оценкой является простая сумма нулей или единиц в сегменте, деленная на число наблюдений в сегменте.

$$\text{Prob}(Y = 1) = \frac{\text{Число единиц в сегменте}}{\text{Размер сегмента}}.$$

Затем оценочная вероятность $\text{Prob}(Y = 1)$ может быть конвертирована в бинарное решение; например, оценка устанавливается в 1, если $\text{Prob}(Y = 1) > 0,5$.

Измерение однородности или разнородности

Древовидные модели рекурсивно создают сегменты (наборы записей) A, которые предсказывают исход $Y = 0$ или $Y = 1$. Из предыдущего алгоритма видно, что нам нужен способ измерить однородность, так называемую *чистоту класса*, в сегменте. Или, что то же самое, нам нужно измерить разнородность сегмента. Точность предсказаний — это доля p неправильно классифицированных записей внутри этого сегмента, которая колеблется от 0 (идеально) до 0,5 (чисто случайное угадывание).

Оказывается, что точность не является хорошей мерой разнородности. Вместо нее общеприняты две другие меры разнородности — *коэффициент разнородности Джини* и *энтропия*, или *информация*. В то время как эти (и другие) меры разнородности применяются к задачам классификации с более чем двумя классами, мы сосредоточимся на бинарном случае. Коэффициент разнородности Джини для набора записей A имеет вид:

$$I(A) = p(1 - p).$$

Энтропийная мера задается следующей формулой:

$$I(A) = -p \log_2(p) - (1 - p) \log_2(1 - p).$$

На рис. 6.5 показано, что мера разнородности Джини (перешкалированная) и мера энтропии схожи, при этом энтропия дает более высокие оценки разнородности для умеренных и высоких уровней точности.

Коэффициент Джини

Разнородность Джини не следует путать с *коэффициентом Джини*. Эти два показателя представляют схожие понятия, но разнородность Джини ограничена задачей бинарной классификации и связана с метрическим показателем AUC *(см. разд. "Метрический показатель AUC" главы 5)*.

Метрический показатель разнородности используется в описанном ранее алгоритме сегментирования. По каждому предлагаемому разбиению данных разнородность вычисляется для каждого сегмента, который получается в результате разбиения. Затем вычисляется взвешенное среднее, и (на каждом этапе) выбирается любой сегмент, который выдает самое низкое взвешенное среднее.

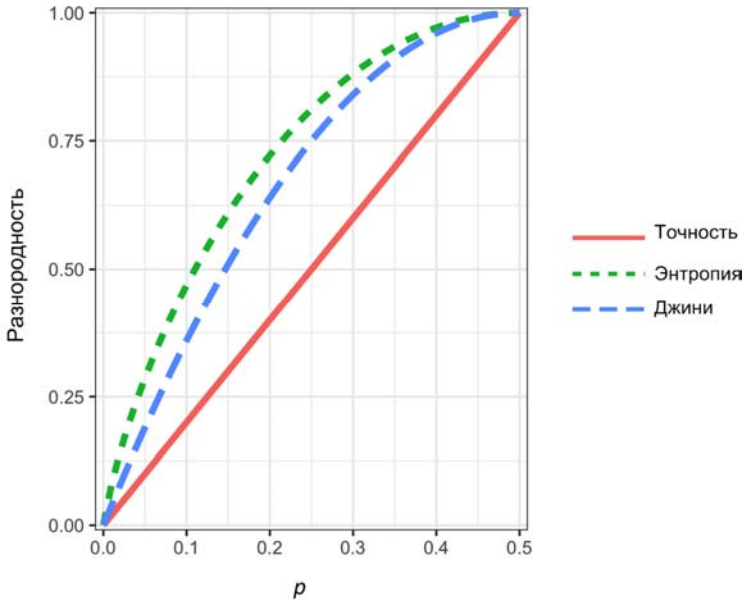


Рис. 6.5. Меры разнородности Джини и энтропии

Остановка роста дерева

По мере того как дерево разрастается, правила разбиения становятся более подробными, и дерево постепенно смещается от распознавания "больших" правил, которые идентифицируют реальные и надежные связи в данных в сторону "крошечных" правил, которые отражают только шум. Полностью выращенное дерево приводит к абсолютно чистым листам и, следовательно, 100%-й точности в классификации данных, на которых оно натренировано.

Эта точность, конечно, иллюзорная — мы выполнили чрезмерно близкую подгонку *(см. примечание "Компромисс между смещением и дисперсией" в разд. "Выбор K" ранее в этой главе)* к данным, приспособившись к шуму в тренировочных данных, а не к тому сигналу, который хотим идентифицировать в новых данных.

Подрезание

Простой и интуитивно понятный метод сокращения размера дерева состоит в подрезании терминальных и более мелких ветвей дерева, в результате чего остается уменьшенное дерево. Как далеко должен заходить процесс подрезания ветвей? Общепринятый прием состоит в подрезании дерева до точки, где ошибка на контрольной выборке с отложенными данными минимальная. Вместе с тем, когда мы будем объединять предсказания, полученные от множественных деревьев *(см. разд. "Бэггинг и случайный лес" далее в этой главе)*, нам понадобится способ остановки роста дерева. Подрезание ветвей играет свою роль в процессе перекрестной проверки и нужно для того, чтобы определить, насколько высоко (вернее, глубоко) выращивать деревья, которые используются в ансамблевых методах.

Нам нужен какой-нибудь способ определения, когда следует прекратить выращивание дерева на этапе, который будет обобщать выводы на новые данные. Существует два общепринятых способа прекратить разбиение данных.

- Не допускать разбиение сегмента, если результирующий подсегмент либо терминальный лист слишком мал. В `rpart` эти ограничения контролируются отдельно соответственно параметрами `minsplit` и `minbucket` со значениями по умолчанию `20` и `7`.
- Не разбивать сегмент, если только новый сегмент "значительно" не уменьшает разнородность. В `rpart` это контролируется *параметром сложности* `cp`, т. е. мерой того, насколько сложным дерево является — чем сложнее, тем больше значение `cp`. На практике `cp` используется для ограничения роста дерева путем наложения штрафа на дополнительную сложность (дополнительные разбиения) в дереве.

Первый метод предполагает произвольные правила и может быть полезным для работы на этапе разведочного анализа, но мы не можем с легкостью определить оптимальные значения (т. е. значения, которые максимизируют предсказательную точность с новыми данными). При помощи параметра сложности `cp` мы можем оценить, какой размер дерева будет давать наилучшую результативность с новыми данными.

Если параметр сложности `cp` окажется слишком малым, то дерево будет переподогнано к данным, приспосабливаясь к шуму, а не к сигналу. С другой стороны, если `cp` будет слишком большим, то дерево окажется слишком малым и будет иметь малую предсказательную силу. В `rpart` значение по умолчанию равно 0,01, хотя в случае больших наборов данных вы, вероятно, обнаружите, что оно слишком большое. В предыдущем примере `cp` был установлен в 0,005, поскольку значение по умолчанию привело к дереву с единственным разбиением. В разведочном анализе достаточно просто испытать несколько значений.

Определение оптимального параметра `cp` является примером компромисса между смещением и дисперсией *(см. примечание "Компромисс между смещением и дисперсией" в разд. "Выбор K" ранее в этой главе)*. Самый общепринятый способ вычислить приблизительную оценку подходящего значения параметра `cp` лежит через перекрестную проверку *(см. разд. "Перекрестная проверка" главы 4)*:

1. Разделить данные на тренировочный и проверочный (контрольная выборка с отложенными данными) наборы.
2. Вырастить дерево с тренировочными данными.
3. Подрезать его последовательно, шаг за шагом, на каждом шаге записывая `cp` (используя *тренировочные* данные).
4. Отметить `cp`, который соответствует минимальной ошибке (потере) на *проверочных* данных.
5. Повторно разделить данные на тренировочный и проверочный наборы и повторить процесс выращивания дерева, подрезания ветвей и записи `cp`.

6. Выполнять этот процесс снова и снова и усреднить параметры `cp`, которые отражают минимальную ошибку для каждого дерева.
7. Вернуться к исходным данным или будущим данным и вырастить дерево, остановившись на полученном оптимальном значении параметра `cp`.

В `rpart` можно использовать аргумент `cptable` с целью создания таблицы значений `cp` и связанной с ними ошибки перекрестной проверки (`xerror` в R), из которой можно определить значение `cp`, имеющее самую низкую ошибку перекрестной проверки.

Предсказывание непрерывной величины

Предсказывание непрерывной величины (т. е. регрессия) на основе дерева следует той же самой логике и процедуре, за исключением того, что разнородность измеряется квадратическими отклонениями от среднего (квадратическими ошибками) в каждом подсегменте, и предсказательная результативность оценивается квадратным корнем из среднеквадратической ошибки (RMSE) *(см. разд. "Диагностика модели" главы 4)* в каждом сегменте.

Каким образом деревья используются

Одно из самых больших препятствий, с которыми сталкиваются разработчики моделей в организациях, — это феномен "черного ящика", который приписывается используемым ими методам, что дает основания для оппозиции со стороны других элементов организации. В этом отношении древовидная модель имеет два привлекательных аспекта.

- Древовидные модели обеспечивают визуальный инструмент обследования данных для получения представления о том, какие переменные важны и как они друг с другом связаны. Деревья могут захватывать нелинейные связи среди предикторных переменных.
- Древовидные модели обеспечивают набор правил, которые могут быть эффективным образом переданы неспециалистам для реализации либо для "продажи" проекта глубинного анализа данных.

Однако, что касается предсказания, то использование результатов из множественных деревьев, как правило, эффективнее, чем использование всего одного дерева. В частности, алгоритмы случайного леса и бустированных деревьев почти всегда обеспечивают превосходящие предсказательную точность и результативность *(см. разд. "Бэггинг и случайный лес" и "Бустинг" далее в этой главе)*, но вышеупомянутые преимущества для одиночного дерева пропадают.

Ключевые идеи для древовидных моделей

- Деревья решений порождают набор правил классификации или предсказывают исход.
- Правила соответствуют последовательному разбиению данных на сегменты.
- Каждый сегмент, или разбиение, соотнесен с определенным значением предикторной переменной и делит данные на записи, где значение этого предиктора выше или ниже значения в точке разбиения.
- На каждом этапе древовидный алгоритм выбирает точку разбиения, которая минимизирует разнородность исхода в каждом подсегменте.
- Когда никакие дальнейшие разбиения сделать невозможно, дерево считается полностью выращенным, и каждый терминальный узел, или лист, имеет записи с единственным классом; новым случаям, которые следуют этим путем правил (разбиения), назначается этот класс.
- Полностью выращенное дерево переподогнано к данным и должно быть подрезано так, чтобы оно получало сигнал вместо шума.
- Алгоритмы множественных деревьев, такие как случайные леса и бустированные деревья, дают более хорошую предсказательную результативность, но теряют в основанной на правилах коммуникативной способности одиночных деревьев.

Дополнительные материалы для чтения

- "Полное учебное руководство по моделированию на основе деревьев с нуля на Python и R" (A Complete Tutorial on Tree Based Modeling from Scratch (in R & Python) // 2016. — April 12) от группы Analytics Vidya Content Team (**https://www.analyticsvidhya.com/blog/2016/04/complete-tutorial-tree-based-modeling-scratch-in-python/**).
- "Введение в рекурсивное сегментирование с использованием подпрограмм RPART" (Therneau T. M., Atkinson E. J. An Introduction to Recursive Partitioning Using the RPART Routines // Mayo Foundation. — 2015. — June 29) (**https://cran.r-project.org/web/packages/rpart/vignettes/longintro.pdf**).

Бэггинг и случайный лес

В 1907 г. статистик сэр Фрэнсис Гальтон (Francis Galton) посещал окружную ярмарку в Англии, на которой проводился конкурс по угадыванию убойного веса вола, демонстрируемого на выставке. Поступило 800 предположений, и, хотя отдельные предположения значительно варьировались, среднее и медиана вышли в пределах 1% от истинной массы вола. Джеймс Суровецки (James Suroweicki) исследовал этот феномен в своей книге "Мудрость толпы" (The Wisdom of Crowds. — Doubleday, 2004). Данный принцип также применяется к предсказательным моде-

лям: усреднение (или взятие большинства голосов) многочисленных моделей — *ансамбля*, состоящего из моделей — оказывается точнее, чем выбор всего одной модели.

Ключевые термины

Ансамбль (ensemble)

Формирование предсказания при помощи набора моделей.

Синоним: усреднение моделей.

Бэггинг (bagging)

Общая методика формирования набора моделей путем бутстрапирования данных.

Синонимы: агрегирование бутстраповских выборок, бутстрап-агрегирование.

Случайный лес (random forest)

Тип бутстрап-агрегированной оценки на основе моделей деревьев решений.

Синоним: бутстрап-агрегированные деревья решений.

Важность переменной (variable importance)

Метрический показатель важности предикторной переменной для результативности модели.

Ансамблевый подход применяется в многочисленных и разнообразных методах моделирования, наиболее наглядно это выразилось в конкурсе Netflix Contest, в котором компания Netflix предложила приз в 1 млн долларов любому конкурсанту, придумавшему модель, дающую 10%-е улучшение предсказания рейтинга, которым клиент Netflix наградит кинофильм. Простая версия ансамблей имеет следующий вид:

1. Разработать предсказательную модель и записать предсказания для набора конкретных данных.
2. Повторить для многочисленных моделей на тех же данных.
3. Для каждой предсказываемой записи взять среднее (либо среднее взвешенное) предсказаний или выбрать предсказание мажоритарным голосованием.

Ансамблевые методы наиболее системно и эффективно применяются к деревьям решений. Ансамблевые древовидные модели настолько мощны, что обеспечивают способ создания хороших предсказательных моделей с относительно небольшими усилиями.

За рамками простого ансамблевого алгоритма существуют два главных варианта ансамблевых моделей: *бэггинг* и *бустинг*. Если иметь в виду ансамблевые древовидные модели, то они называются моделями *случайного леса* и *бутстированными древовидными* моделями. Данный раздел посвящен бэггингу; бустинг рассматривается в одноименном разделе *далее в этой главе*.

Бэггинг

Бэггинг (bagging, как сокр. от *bootstrap aggregating* (в русском языке — *бутстрап-агрегирование* или агрегация бутстраповских выборок)) был представлен Лео Брейманом в 1994 г.[4] Предположим, мы имеем отклик Y и P предикторных переменных $\mathbf{X} = X_1, X_2, ..., X_P$ с n записями.

Бэггинг похож на базовый алгоритм для ансамблей за одним исключением — вместо подгонки различных моделей к одинаковым данным, каждая новая модель подгоняется к повторно отобранной бутстраповской выборке. Ниже данный алгоритм представлен более формально:

1. Инициализировать M, число моделей для подгонки, и n, число записей, из которых делается выборка ($n < N$). Установить итерацию в $m = 1$.
2. Взять повторную бутстраповскую выборку (т. е. с возвратом) из n записей из тренировочных данных, чтобы сформировать подвыборку Y_m и $\mathbf{X}_m$ (пакет).
3. Натренировать модель, используя Y_m и $\mathbf{X}_m$, чтобы создать набор правил решения $\hat{f}_m(\mathbf{X})$.
4. Прирастить счетчик моделей $m = m + 1$. Если $m \leq M$, перейти к шагу 1.

В случае, где f_m предсказывает вероятность $Y = 1$, оценка на основе бэггинга (bagged estimate — пакетная оценка) задается следующей формулой:

$$\hat{f} = \frac{1}{M}\left(\hat{f}_1(\mathbf{X}) + \hat{f}_2(\mathbf{X}) + ... + \hat{f}_M(\mathbf{X})\right).$$

Случайный лес

Случайный лес основывается на применении бэггинга к деревьям решений с одним важным расширением: в дополнение к отбору записей алгоритм также отбирает переменные[5]. В традиционных деревьях решений, для того чтобы определить, как создать подсегмент сегмента A, алгоритм делает выбор переменной и точки разбиения путем минимизации критерия, в частности, коэффициента разнородности Джини *(см. разд. "Измерение однородности или разнородности" ранее в этой главе)*. При использовании случайных лесов на каждом этапе алгоритма выбор переменной ограничен *случайным подмножеством переменных*. По сравнению с базовым древовидным алгоритмом *(см. разд. "Алгоритм рекурсивного сегментирования" ранее в этой главе)*, алгоритм случайного леса добавляет еще два шага: обсуждавшийся

[4] Термин bagging — это своего рода технический каламбур, который, с одной стороны, является аббревиатурой для бутстрап-агрегирования и, с другой, означает упаковывание в пакеты, поскольку, в сущности, характер работы алгоритма бэггинга в этом и состоит — он отбирает бутстраповские пакеты данных и затем их агрегирует. — *Прим. пер.*

[5] Термин *"случайный лес"* является товарным знаком Лео Бреймана и Адель Катлер с лицензией, выданной Salford Systems. Стандартное наименование нетоварного знака отсутствует, и термин *"случайный лес"* является таким же синонимом алгоритма, как и "Клинекс" для косметических салфеток.

ранее бэггинг *(см. разд. "Бэггинг и случайный лес" ранее в этой главе)* и бутстрапированный отбор переменных в каждой точке разбиения:

1. Взять из *записей* бутстраповскую подвыборку (с возвратом).
2. Для первого разбиения отобрать в произвольном порядке $p < P$ *переменных* без возврата.
3. Для каждой из отобранных переменных $X_{j(1)}, X_{j(2)}, ..., X_{j(p)}$ применить алгоритм разбиения:
   - для каждого значения $s_{j(k)}$ из X:
     - разбить записи в сегменте A с $X_{j(k)} < s_{j(k)}$ на один сегмент и оставшиеся записи, где $X_{j(k)} \geq s_{j(k)}$, на другой сегмент;
     - измерить однородность классов внутри каждого подсегмента A;
   - выбрать значение $s_{j(k)}$, которое порождает максимальную внутрисегментную однородность класса.
4. Выбрать переменную $X_{j(k)}$ и значение в точке разбиения $s_{j(k)}$, которое порождает максимальную внутрисегментную однородность класса.
5. Перейти к следующему разбиению и повторить предыдущие шаги, начиная с шага 2.
6. Продолжить дополнительные разбиения, следуя той же процедуре, пока дерево не будет выращено.
7. Вернуться к шагу 1, взять еще одну бутстраповскую подвыборку и начать процесс заново.

Сколько переменных отбирать на каждом шаге? Эмпирическое правило говорит о выборе $\sqrt{P}$, где P — это число предикторных переменных. Программный пакет `randomForest` реализует случайный лес в R. Приведенный далее фрагмент применяет этот пакет к данным о ссудах (*см. разд. "К ближайших соседей" ранее в этой главе* относительно описания данных).

```
> library(randomForest)
> rf <- randomForest(outcome ~ borrower_score + payment_inc_ratio,
                     data=loan3000)
Call:

randomForest(formula = outcome ~ borrower_score + payment_inc_ratio,
             data = loan3000)
              Type of random forest: classification
                    Number of trees: 500
No. of variables tried at each split: 1

        OOB estimate of error rate: 38.53%
Confusion matrix:
         paid off default class.error
```

```
paid off     1089     425   0.2807133
default       731     755   0.4919246
```

По умолчанию натренировано 500 деревьев. Поскольку в наборе предикторов имеется всего две переменные, алгоритм в произвольном порядке отбирает переменную, по которой можно выполнить разбиение на каждом этапе (т. е. бутстраповскую подвыборку размера 1).

Внепакетная оценка OOB (out-of-bag — не вошедший в пакет) ошибки — это коэффициент ошибок для натренированных моделей, применяемый к данным, отложенным в сторону из тренировочного набора для этого дерева. Используя выходные данные из модели, ошибку OOB можно отобразить на графике против числа деревьев в случайном лесе:

```
error_df = data.frame(error_rate = rf$err.rate[,'OOB'],
                      num_trees = 1:rf$ntree)
ggplot(error_df, aes(x=num_trees, y=error_rate)) +
geom_line()
```

Результат показан на рис. 6.6. Коэффициент ошибок быстро уменьшается примерно с 0,44 до стабилизации на уровне 0,385. Предсказанные значения могут быть получены из функции `predict` и отображены на графике следующим образом:

```
pred <- predict(loan_lda)
rf_df <- cbind(loan3000, pred_default=pred[,'default']>.5)
ggplot(data=rf_df, aes(x=borrower_score, y=payment_inc_ratio,
                       color=pred_default, shape=pred_default)) +
  geom_point(alpha=.6, size=2) +
  scale_shape_manual(values=c(46, 4))
```

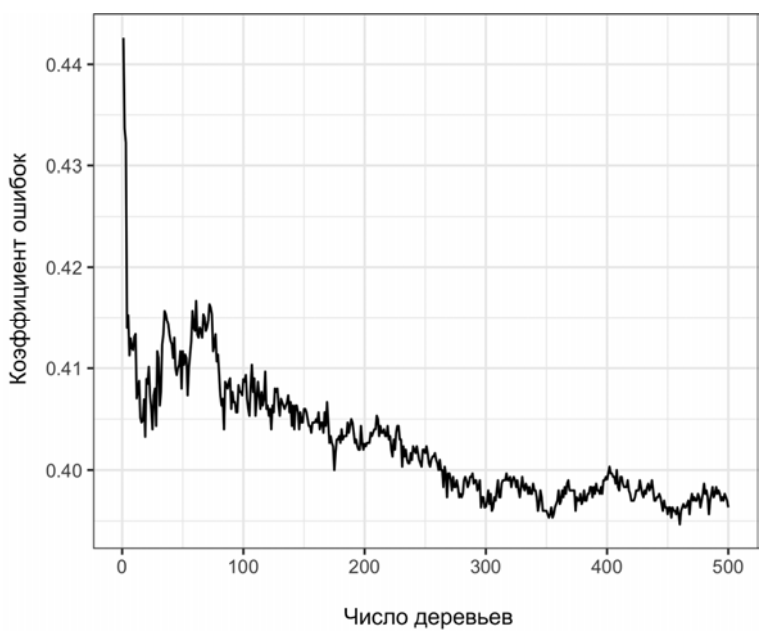


Рис. 6.6. Улучшение точности случайного леса с добавлением большего числа деревьев

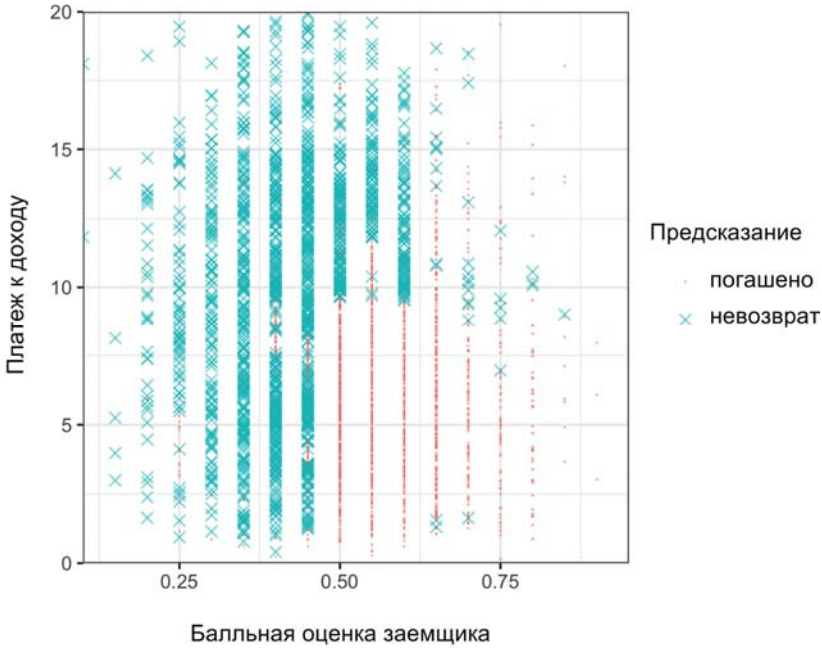


Рис. 6.7. Предсказанные исходы из случайного леса применительно к данным о невозвратных ссудах

График, представленный на рис. 6.7, весьма показателен в отношении природы случайного леса.

Метод случайного леса — это метод "черного ящика". Он производит более точные предсказания, чем простое дерево, но интуитивно понятные правила решения простого дерева теряются. Предсказания также несколько шумные: отметим, что некоторые заемщики с очень высоким баллом оценки, говорящим о высокой кредитоспособности, по-прежнему в итоге получают предсказание невозврата ссуды. Это является результатом нескольких необычных записей в данных и демонстрирует опасность переподгонки, вызванной случайным лесом *(см. примечание "Компромисс между смещением и дисперсией" в разд. "Выбор K" ранее в этой главе)*.

Важность переменных

Мощь алгоритма случайного леса проявляет себя при создании предсказательных моделей для данных с большим количеством признаков и записей. Алгоритм способен автоматически определять, какие предикторы важны, и обнаруживать сложные связи между предикторами, которые соответствуют членам взаимодействия *(см. разд. "Взаимодействия и главные эффекты" главы 4)*. Например, выполним подгонку модели к данным о невозвратных ссудах, включив столбцы:

```
> rf_all <- randomForest(outcome ~ ., data=loan_data, importance=TRUE)
> rf_all

Call:
 randomForest(formula = outcome ~ ., data = loan_data, importance = TRUE)
```

```
                Type of random forest: classification
                      Number of trees: 500
No. of variables tried at each split: 3

        OOB estimate of error rate: 34.38%
Confusion matrix:
         paid off default class.error
paid off    15078    8058   0.3482884
default      7849   15287   0.3392548
```

Аргумент `importance=TRUE` запрашивает, чтобы `randomForest` сохранил дополнительную информацию о важности разных переменных. Функция `varImpPlot` построит график относительной результативности переменных:

```
varImpPlot(rf_all, type=1)
varImpPlot(rf_all, type=2)
```

Результат показан на рис. 6.8.

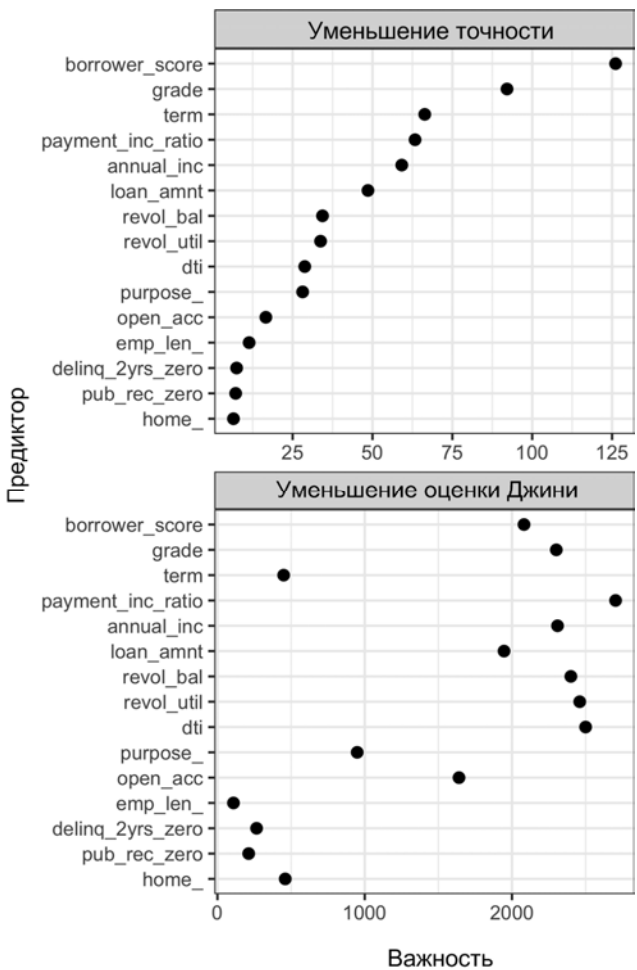


Рис. 6.8. Важность переменных для полной подгонки модели к данным о ссудах

Существует два способа оценить важность переменных.

- Путем уменьшения точности модели, если значения переменной в произвольном порядке перестановлены (`type=1`). Произвольная перестановка значений имеет эффект удаления всей предсказательной силы для этой переменной. Точность вычисляется из внепакетных данных (поэтому данная мера является практически перекрестно-проверочной оценкой).
- Путем среднего уменьшения в оценке разнородности Джини *(см. разд. "Измерение однородности или разнородности" ранее в этой главе)* для всех узлов, которые были разбиты по переменной (`type=2`). Это демонстрирует, какой вклад переменная вносит в улучшение чистоты узлов. Мера основывается на тренировочном наборе, и поэтому менее надежная, чем мера, вычисленная на внепакетных данных.

Верхние и нижние части рис. 6.8 показывают важность переменных согласно уменьшению точности и разнородности Джини в указанном порядке. Переменные в обеих частях ранжированы по уменьшению точности. Оценки важности переменных, произведенные этими двумя мерами, очень различаются.

Поскольку уменьшение точности является более надежным метрическим показателем, зачем нужно использовать меру уменьшения разнородности Джини? По умолчанию `randomForest` вычисляет только оценку Джини: мера разнородности Джини является побочным продуктом алгоритма, тогда как точность модели в зависимости от переменной требует дополнительных вычислений (произвольная перестановка данных и предсказание этих данных). В случаях, где вычислительная сложность имеет значение, например в эксплуатационной среде, где выполняется подгонка тысяч моделей, она (сложность) не будет стоить дополнительных вычислительных усилий. Кроме того, уменьшение меры Джини проливает свет на то, какие переменные используются случайным лесом для создания своих правил разбиения (вспомним, что данная информация, легко видимая в простом дереве, практически теряется в случайном лесе). Исследование разницы между уменьшением разнородности Джини и важностью переменных за счет точности модели может подсказать способы улучшения модели.

Гиперпараметры

Случайный лес, как и в случае многих статистических алгоритмов машинного обучения, может рассматриваться, как алгоритм "черного ящика" с кнопками для настройки характера работы ящика. Эти кнопки называются *гиперпараметрами*, т. е. параметрами, которые необходимо настроить прежде, чем приступить к подгонке модели; они не оптимизируются как составная часть тренировочного процесса. В то время как традиционные статистические модели требуют выбора (например, выбора предикторов для использования в модели регрессии), гиперпараметры случайного леса имеют большее значение, в особенности, чтобы предотвратить переподгонку. В частности, два самых важных гиперпараметра случайного леса следующие:

- `nodesize` — минимальный размер терминальных узлов (листьев в дереве), значение по умолчанию равно 1 для классификации и 5 для регрессии;
- `maxnodes` — максимальное количество узлов в каждом дереве решений. По умолчанию предел отсутствует, и самое большое дерево будет подогнано к ограничениям, установленным в `nodesize`.

Может возникнуть соблазн проигнорировать эти параметры и просто начать со значениями по умолчанию. Однако использование значения по умолчанию может привести к переподгонке, когда вы применяете случайный лес к шумным данным. Когда вы увеличите `nodesize` или установите `maxnodes`, алгоритм будет выполнять подгонку меньших деревьев и с меньшей вероятностью создаст мнимые предсказательные правила. Для проверки эффектов принятия гиперпараметрами разных значений может использоваться перекрестная проверка *(см. разд. "Перекрестная проверка" главы 4)*.

Ключевые идеи для бэггинга и случайного леса

- Ансамблевые модели улучшают точность модели путем объединения результатов многих моделей.
- Бэггинг — это особый тип ансамблевой модели, опирающийся на подгонку большого числа моделей к бутстраповским выборкам из данных, и усреднения моделей.
- Случайный лес — это специальный тип бэггинга, применяемый к деревьям решений. В дополнение к повторному отбору данных алгоритм случайного леса отбирает предикторные переменные при разбиении деревьев.
- Полезными данными на выходе из случайного леса является мера важности переменных, которая ранжирует предикторы с точки зрения их вклада в точность модели.
- Случайный лес имеет ряд гиперпараметров, которые следует настроить при помощи перекрестной проверки для предотвращения переподгонки.

Бустинг

Ансамблевые модели стали стандартным инструментом для предсказательного моделирования. Бустинг — это общая методика создания ансамбля моделей. Она была разработана примерно в то же время, что и бэггинг *(см. разд. "Бэггинг и случайный лес" ранее в этой главе)*. Подобно бэггингу, бустинг очень широко используется с деревьями решений. Несмотря на их общие черты, в бустинге принят совсем другой подход, который сопровождается многими излишними аксессуарами. В результате в то время как бэггинг можно применять с относительно небольшой донастройкой, бустинг требует в своем применении намного большей внимательности. Если бы эти два метода были автомобилями, то бэггинг можно было бы рассматривать как автомобиль Honda модели Accord (надежный и устойчивый), тогда как бустинг был бы Porsche (мощный, но требует большего ухода).

В линейных регрессионных моделях нередко остатки обследуют, чтобы посмотреть, можно ли улучшить подгонку *(см. разд. "Графики частных остатков и нелинейность" главы 4)*. Бустинг продвинул эту концепцию гораздо дальше и выполняет подгонку серии моделей, где каждая последующая модель подгоняется с целью минимизировать ошибку предыдущих моделей. Широко используются несколько вариантов данного алгоритма: *Adaboost, градиентный бустинг* и *стохастический градиентный бустинг*. Последний из перечисленных, стохастический градиентный бустинг, является самым общим. Фактически при правильном выборе параметров этот алгоритм может эмулировать случайный лес.

Ключевые термины

Ансамбль (ensemble)

Формирование предсказания за счет использования набора моделей.

Синонимы: усреднение моделей, композиция моделей.

Бустинг (boosting)

Общая методика подгонки последовательности моделей путем предоставления большего веса записям с большими остатками для каждого последующего цикла.

Adaboost

Ранняя версия бустинга, опирающаяся на перевесовку данных на основе остатков.

Градиентный бустинг (gradient boosting)

Более общая форма бустинга, которая создана с точки зрения минимизации функции стоимости.

Стохастический градиентный бустинг (stochastic gradient boosting)

Самый общий алгоритм бустинга, который предусматривает повторный отбор записей и столбцов в каждом цикле.

Регуляризация (regularization)

Метод предотвращения переподгонки путем добавления штрафного члена в функцию стоимости на ряде параметров в модели.

Гиперпараметры (hyperparameters)

Параметры, которые необходимо установить перед подгонкой алгоритма.

Алгоритм бустинга

Основная идея, которая лежит в основе различных алгоритмов бустинга, в сущности одна и та же. Самым простым для понимания является алгоритм Adaboost, который работает следующим образом:

1. Инициализировать M — максимальное число моделей, подлежащих подгонке, и установить счетчик итераций в $m = 1$. Инициализировать веса наблюдений $w_i = 1/N$ для $i = 1, 2, ..., N$. Инициализировать ансамблевую модель $\hat{F}_0 = 0$.
2. Натренировать модель, используя $\hat{f}_M$ с применением весов наблюдений $w_1, w_2, ..., w_N$, которые минимизируют взвешенную ошибку e_M, определяемую путем суммирования весов неправильно классифицированных наблюдений.
3. Добавить в ансамбль модель: $\hat{F}_m = \hat{F}_{m-1} + \alpha_m \hat{f}_m$, где $\alpha_m = \frac{\log 1 - e_m}{e_m}$.
4. Обновить веса $w_1, w_2, ..., w_N$ таким образом, чтобы веса наблюдений, которые были классифицированы неправильно, были увеличены. Размер увеличения зависит от α_M, при этом более крупные значения α_M приводят к бóльшим весам.
5. Нарастить счетчик моделей $m = m + 1$. Если $m \leq M$, то перейти к шагу 1.

Бустированная оценка задается следующей формулой:

$$\hat{F} = \alpha_1 \hat{f}_1 + \alpha_2 \hat{f}_2 + ... + \alpha_M \hat{f}_M.$$

Путем увеличения весов наблюдений, которые были классифицированы неправильно, алгоритм побуждает модели более активно тренироваться на данных, для которых он показывал плохую результативность. Фактор α_m гарантирует, что модели с более низкой ошибкой будут иметь больший вес.

Градиентный бустинг похож на Adaboost, но представляет задачу как оптимизацию функции стоимости. Вместо того чтобы корректировать веса, градиентный бустинг выполняет подгонку моделей к псевдоостатку, что имеет эффект более активной тренировки на более крупных остатках. В духе случайного леса стохастический градиентный бустинг добавляет в алгоритм произвольность, отбирая наблюдения и предикторные переменные на каждом этапе.

XGBoost

XGBoost — это наиболее широко используемый бесплатный пакет программ с реализацией стохастического градиентного бустинга, который первоначально был разработан Тьянси Ченом (Tianqi Chen) и Карлосом Гестрином (Carlos Guestrin) в Вашингтонсоком университете. Его вычислительно эффективная реализация со многими опциями доступна в качестве программной библиотеки для большинства основных языков программирования, используемых в науке о данных. В R XGBoost доступен в виде программного пакета `xgboost`.

Функция `xgboost` имеет много параметров, которые могут и должны корректироваться *(см. разд. "Гиперпараметры и перекрестная проверка" далее в этой главе)*. Двумя очень важными параметрами являются `subsample`, управляющий долей наблюдений, которые должны отбираться во время каждой итерации, и `eta` — фактор сжатия, применяемый к α_m в алгоритме бустинга *(см. разд. "Алгоритм бустинга" ранее в этой главе)*. Использование `subsample` заставляет бустинг действовать как случайный лес за исключением того, что отбор выполняется без возврата. Параметр

сжатия `eta` полезен для предотвращения переподгонки путем сокращения изменений в весах (меньшее изменение в весах означает, что алгоритм с меньшей вероятностью будет переподогнан к тренировочному набору). Представленный далее фрагмент кода применяет `xgboost` к данным о ссудах всего с двумя предикторными переменными:

```
library(xgboost)
predictors <- data.matrix(loan3000[, c('borrower_score',
                                       'payment_inc_ratio')])
label <- as.numeric(loan3000[,'outcome'])-1
xgb <- xgboost(data=predictors, label=label,
               objective = "binary:logistic",
               params=list(subsample=.63, eta=0.1), nrounds=100)
```

Отметим, что `xgboost` не поддерживает формульный синтаксис, поэтому предикторы нужно конвертировать в матрицу данных `data.matrix`, а отклик — в двоичные переменные в формате 0/1. Аргумент `objective` сообщает `xgboost` тип решаемой задачи; опираясь на эти данные `xgboost` выберет для оптимизации метрический показатель.

Предсказанные значения можно получить из функции `predict`, и поскольку переменных всего две, их можно вывести на графике против предикторов:

```
pred <- predict(xgb, newdata=predictors)
xgb_df <- cbind(loan3000, pred_default=pred>.5, prob_default=pred)
ggplot(data=xgb_df, aes(x=borrower_score, y=payment_inc_ratio,
                        color=pred_default, shape=pred_default)) +
       geom_point(alpha=.6, size=2)
```

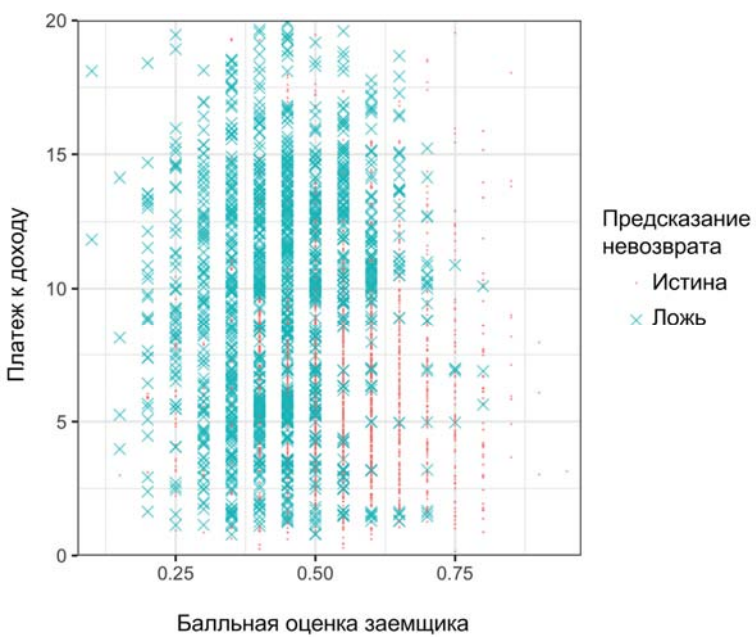


Рис. 6.9. Предсказанные исходы из XGBoost применительно к данным о невозвратных ссудах

Результат показан на рис. 6.9. В качественном плане он аналогичен предсказаниям из случайного леса (см. рис. 6.7). Предсказания несколько шумные в том, что некоторые заемщики с очень высокой балльной оценкой заемщика по-прежнему получают предсказание невозврата ссуды.

Регуляризация: предотвращение переподгонки

Слепое применение `xgboost` может привести к нестабильным моделям в результате переподгонки к тренировочным данным. Проблема с переподгонкой имеет две составляющие:

- точность модели на новых данных не из тренировочного набора ухудшится;
- получаемые из модели предсказания весьма вариабельны, что приводит к нестабильным результатам.

Любой прием моделирования потенциально подвержен переподгонке. Например, если слишком много переменных включены в уравнение регрессии, то в конечном итоге модель может прийти к мнимым предсказаниям. Однако в отношении большинства статистических приемов переподгонку можно предотвратить разумным выбором предикторных переменных. Даже случайный лес обычно порождает разумную модель без настройки параметров. Это, однако, не относится к `xgboost`. Выполним подгонку `xgboost` к данным о ссудах для тренировочного набора со всеми включенными в модель переменными:

```
> predictors <- data.matrix(loan_data[,-which(names(loan_data) %in%
                                        'outcome')])
> label <- as.numeric(loan_data$outcome)-1
> test_idx <- sample(nrow(loan_data), 10000)
> xgb_default <- xgboost(data=predictors[-test_idx,],
                         label=label[-test_idx],
                         objective = "binary:logistic", nrounds=250)
> pred_default <- predict(xgb_default, predictors[test_idx,])
> error_default <- abs(label[test_idx] - pred_default) > 0.5
> xgb_default$evaluation_log[250,]
   iter train_error
1:  250    0.145622
> mean(error_default)
[1] 0.3715
```

Проверочный набор состоит из 10 000 в произвольном порядке отобранных записей из полных данных, а тренировочный набор — из оставшихся записей. Бустинг приводит к коэффициенту ошибок, равному всего 14,6% для тренировочного набора. Проверочный набор, однако, имеет намного более высокий коэффициент ошибок — 36,2%. Это является результатом переподгонки: в то время как бустинг способен очень хорошо объяснить вариабельность в тренировочном наборе, правила предсказания не применимы к новым данным.

Бустинг предоставляет несколько параметров для предотвращения переподгонки, в том числе параметры `eta` и `subsample` *(см. разд. "XGBoost" ранее в этой главе)*.

Другой подход заключается в *регуляризации*, методе, который модифицирует функцию стоимости, чтобы *оштрафовать* сложность модели. Деревья решений подгоняются путем минимизации критериев стоимости, в частности, оценки разнородности Джини *(см. разд. "Измерение однородности или разнородности" ранее в этой главе)*. В `xgboost` можно модифицировать функцию стоимости путем добавления члена, который измеряет сложность модели.

В `xgboost` имеется два параметра для регуляризации модели: `alpha` и `lambda`, которые представляют собой соответственно манхэттенское расстояние и квадратическое евклидово расстояние *(см. разд. "Метрические показатели расстояния" ранее в этой главе)*. Увеличение этих параметров оштрафует более сложные модели и уменьшит размер подгоняемых деревьев. Например, посмотрим, что произойдет, если установить `lambda` в 1000:

```
> xgb_penalty <- xgboost(data=predictors[-test_idx,],
                         label=label[-test_idx],
                         params=list(eta=.1, subsample=.63, lambda=1000),
                         objective = "binary:logistic", nrounds=250)
> pred_penalty <- predict(xgb_penalty, predictors[test_idx,])
> error_penalty <- abs(label[test_idx] - pred_penalty) > 0.5
> xgb_penalty$evaluation_log[250,]
   iter train_error
1:  250    0.332405
> mean(error_penalty)
[1] 0.3483
```

Теперь ошибка тренировки только немного ниже ошибки на проверочном наборе.

Метод `predict` предлагает удобный аргумент `ntreelimit`, который заставляет использовать в предсказании только первые *i* деревьев. Это позволяет нам непосредственно сопоставлять внутривыборочный коэффициент ошибок с вневыборочным по мере включения большего числа моделей:

```
> error_default <- rep(0, 250)
> error_penalty <- rep(0, 250)
> for(i in 1:250){
  pred_def <- predict(xgb_default, predictors[test_idx,], ntreelimit=i)
  error_default[i] <- mean(abs(label[test_idx] - pred_def) >= 0.5)
  pred_pen <- predict(xgb_penalty, predictors[test_idx,], ntreelimit = i)
  error_penalty[i] <- mean(abs(label[test_idx] - pred_pen) >= 0.5)
}
```

В данных на выходе из модели содержится ошибка для тренировочного набора в компоненте `xgb_default$evaluation_log`. Объединив ее с вневыборочными ошибками, мы можем отобразить на графике ошибки против числа итераций:

```
> errors <- rbind(xgb_default$evaluation_log,
                  xgb_penalty$evaluation_log,
                  data.frame(iter=1:250, train_error=error_default),
                data.frame(iter=1:250, train_error=error_penalty))
```

```
> errors$type <- rep(c('default train', 'penalty train',
                       'default test', 'penalty test'), rep(250, 4))
> ggplot(errors, aes(x=iter, y=train_error, group=type)) +
  geom_line(aes(linetype=type, color=type))
```

Результат, изображенный на рис. 6.10, показывает, как заданная по умолчанию модель неуклонно улучшает точность для тренировочного набора, но фактически становится хуже для проверочного набора. Оштрафованная модель такое поведение не показывает.

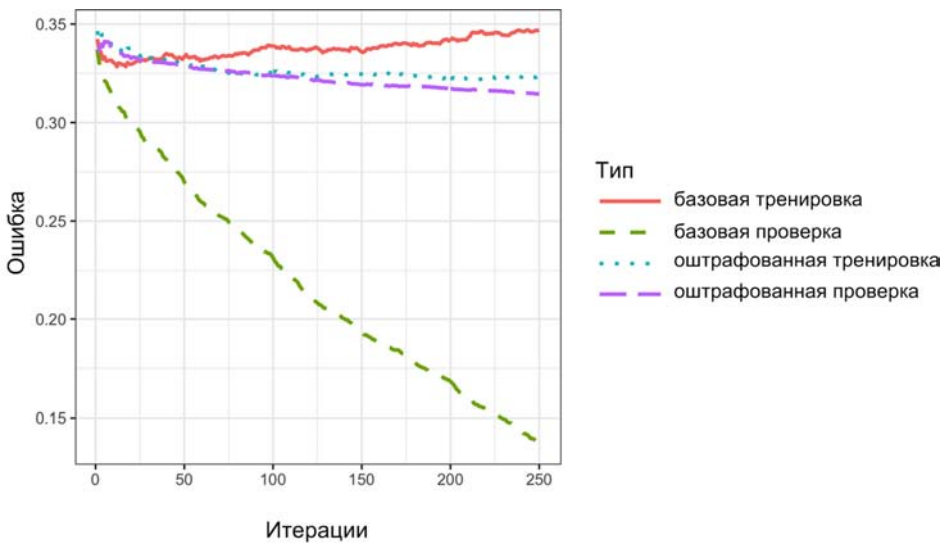


Рис. 6.10. Коэффициент ошибок модели XGBoost, заданной по умолчанию, против оштрафованной версии XGBoost

Гребневая регрессия и лассо-регрессия

Метод наложения штрафа на сложность модели, чтобы помочь предотвратить переподгонку, берет начало с 1970-х гг. Регрессия наименьших квадратов минимизирует остаточную сумму квадратов (RSS) *(см. разд. "Наименьшие квадраты" главы 4)*. *Гребневая регрессия* минимизирует сумму квадратов остатков плюс штраф на число и размер коэффициентов:

$$\sum_{i=1}^{n}\left(Y_i - \hat{b}_0 - \hat{b}_1 x_1 - \ldots - \hat{b}X_p\right)^2 + \lambda\left(\hat{b}_1^2 + \ldots + \hat{b}_p^2\right).$$

Значение X определяет, насколько много коэффициенты штрафуются; более крупные значения порождают модели, которые с меньшей вероятностью будут переподогнаны к данным. *Лассо*-регуляризация аналогична за исключением того, что в качестве штрафного члена она использует манхэттенское расстояние вместо евклидова:

$$\sum_{i=1}^{n}\left(Y_i - \hat{b}_0 - \hat{b}_1 x_1 - \ldots - \hat{b}X_p\right)^2 + \alpha\left(\left|\hat{b}_1\right| + \ldots + \left|\hat{b}_p\right|\right).$$

Параметры `lambda` и `alpha` в `xgboost` действуют аналогичным образом.

Гиперпараметры и перекрестная проверка

Пакет `xgboost` имеет просто пугающий массив гиперпараметров *(см. врезку "Гиперпараметры XGBoost" далее в этой главе*, в которой обсуждается этот вопрос). Как явствует из *разд. "Регуляризация: предотвращение переподгонки" ранее в этой главе*, конкретный выбор может существенно изменить подгонку модели. Чем следует руководствоваться при выборе с учетом огромного количества сочетаний гиперпараметров на выбор? Стандартное решение этой проблемы состоит в том, чтобы использовать *перекрестную проверку (см. разд. "Перекрестная проверка" главы 4)*. Перекрестная проверка в произвольном порядке разбивает данные на K разных групп, так называемые *блоки*. Для каждого блока выполняется тренировка модели на данных, которые не находятся в блоке, и затем модель оценивается на данных в блоке. Это приводит к мере точности модели на вневыборочных данных. Лучшим набором гиперпараметров является тот, который получен моделью с самой низкой общей ошибкой согласно расчетам по усреднению ошибок из каждого блока.

Для того чтобы проиллюстрировать данный прием, применим его к набору параметров для `xgboost`. Мы обследуем два параметра: параметр сжатия `eta` *(см. разд. "XGBoost" ранее в этой главе)* и `max_depth`. Параметр `max_depth` — это максимальная глубина от листового узла до корня дерева, чье значение по умолчанию равно 6. Он дает нам еще один способ управлять переподгонкой: глубокие деревья имеют тенденцию к большей сложности и могут чрезмерно подстраиваться под данные. Сначала мы задаем блоки и список параметров:

```
> N <- nrow(loan_data)
> fold_number <- sample(1:5, N, replace = TRUE)
> params <- data.frame(eta = rep(c(.1, .5, .9), 3),
                       max_depth = rep(c(3, 6, 12), rep(3,3)))
```

Теперь мы применим приведенный выше алгоритм вычисления ошибки для каждой модели и каждого блока, используя пять блоков:

```
> error <- matrix(0, nrow=9, ncol=5)
> for(i in 1:nrow(params)){
>   for(k in 1:5){
>     fold_idx <- (1:N)[fold_number == k]
>     xgb <- xgboost(data=predictors[-fold_idx,], label=label[-fold_idx],
                     params = list(eta = params[i, 'eta'],
                                   max_depth = params[i, 'max_depth']),
                   objective = "binary:logistic", nrounds=100, verbose=0)
>     pred <- predict(xgb, predictors[fold_idx,])
>     error[i, k] <- mean(abs(label[fold_idx] - pred) >= 0.5)
>   }
> }
```

Поскольку мы выполняем подгонку в общей сложности 45 моделей, этот процесс займет немного времени. Ошибки хранятся в виде матрицы, при этом модели расположены вдоль строк, блоки — вдоль столбцов. При помощи функции `rowMeans` мы можем сравнить коэффициент ошибок для разных наборов параметров:

```
> avg_error <- 100 * rowMeans(error)
> cbind(params, avg_error)
  eta max_depth avg_error
1 0.1         3     35.41
2 0.5         3     35.84

3 0.9         3     36.48
4 0.1         6     35.37
5 0.5         6     37.33
6 0.9         6     39.41
7 0.1        12     36.70
8 0.5        12     38.85
9 0.9        12     40.19
```

Перекрестная проверка позволяет предположить, что использование более мелких деревьев с меньшим значением `eta` приводит к более точным результатам. Поскольку эти модели также более стабильны, лучшими параметрами являются `eta=0.1` и `max_depth=3` (либо, возможно, `max_depth=6`).

Гиперпараметры XGBoost

Гиперпараметры в XGBoost используют преимущественно для того, чтобы сбалансировать переподгонку с точностью и вычислительной сложностью. Полное описание параметров см. в документации по XGBoost.

- `eta` — фактор сжатия в диапазоне 0–1, применяемый к α в алгоритме бустинга. Значение по умолчанию равно 0,3, но для шумных данных рекомендуются меньшие значения (например, 0,1).
- `nrounds` — число циклов бустинга. Если параметр `eta` установлен в малое значение, важно увеличить число циклов, поскольку алгоритм медленнее обучается. При условии что для предотвращения переподгонки вводятся несколько параметров, наличие большего количества циклов вреда не причинит.
- `max_depth` — максимальная глубина дерева (значение по умолчанию равняется 6). В отличие от случайного леса, который выполняет подгонку очень глубоких деревьев, бустинг обычно осуществляет подгонку мелких деревьев. Это дает преимущество предотвращения мнимых сложных взаимодействий в модели, которые могут возникнуть из-за шумных данных.
- `subsample` и `colsample_bytree` — доля записей для отбора без возврата и доля предикторов для отбора с целью использования в подгонке деревьев. Данные параметры аналогичны параметрам в случайных лесах и помогают предотвратить переподгонку.
- `lambda` и `alpha` — параметры регуляризации для помощи в управлении переподгонкой *(см. разд. "Регуляризация: предотвращение переподгонки" ранее в этой главе)*.

Ключевые идеи для бустинга

- Бустинг — это класс ансамблевых моделей, основанных на подгонке последовательности моделей, где в последующих циклах больший вес придается записям с крупными ошибками.
- Стохастический градиентный бустинг — это самый общий тип бустинга, который обеспечивает наилучшую результативность. Общепринятая форма стохастического градиентного бустинга использует древовидные модели.
- XGBoost — это популярный и вычислительно эффективный пакет программ для стохастического градиентного бустинга; он доступен на всех общепринятых языках программирования, используемых в науке о данных.
- Бустинг подвержен переподгонке данных, и для ее предотвращения необходимо настраивать гиперпараметры.
- Регуляризация — это один из способов предотвратить переподгонку путем применения штрафного члена уравнения к набору параметров (например, размеру дерева) в модели.
- Перекрестная проверка в особенности важна для бустинга из-за большого числа гиперпараметров, которые необходимо настроить.

Резюме

В данной главе были рассмотрены два метода классификации и предсказания, которые гибко и локально "учатся" на данных без структурной модели (как например, в линейной регрессии), которая подгоняется ко всему набору данных. *K* ближайших соседей — это простой процесс, который смотрит вокруг на схожие записи и назначает предсказываемой записи ее мажоритарный класс (или усредненное значение). Испытывая разные пороговые значения предикторных переменных, древовидные модели многократно делят данные на сегменты и подсегменты, которые становятся все более гомогенными относительно класса. Самые эффективные пороговые значения формируют путь, а также "правило", для выполнения классификации или предсказания. Древовидные модели являются очень мощным и популярным прогнозирующим инструментом, часто превосходящим другие методы по результативности. Они дали начало различным ансамблевым методам (случайного леса, бэггинга, бустинга), которые заостряют предсказательную силу деревьев.

Обучение без учителя

Термин *"обучение без учителя"* относится к статистическим методам, которые извлекают смысл из данных без тренировки модели на помеченных данных (данных, где целевой исход известен). *Главы 4* и *5* были посвящены построению модели (набора правил) для предсказания ответа из набора предикторных переменных. Обучение без учителя тоже создает модель данных, но при этом не разграничивает переменную отклика и предикторные переменные.

Обучение без учителя может преследовать самые разные возможные цели. В некоторых случаях оно может использоваться для создания предсказательного правила в отсутствие помеченного отклика. Методы *кластеризации* могут применяться для идентификации содержательных групп данных. Например, отслеживая нажатия на веб-страницах и имея демографические сведения о посетителях веб-сайта, нам, возможно, удастся сгруппировать разные типы пользователей. После этого веб-сайт может быть индивидуально настроен под эти разные типы.

В других случаях цель может состоять в *сокращении размерности* данных до большего управляемого набора переменных. Этот сокращенный набор затем может использоваться в качестве входа в предсказательную модель, такую как регрессия или классификация. Например, у нас могут быть тысячи датчиков контроля производственного процесса. Сократив данные до меньшего набора признаков, нам, возможно, будет под силу построить более мощную и поддающуюся интерпретации модель для предсказания сбоя в процессе, чем модель с потоками данных от тысяч датчиков.

Наконец, обучение без учителя может рассматриваться как расширение разведочного анализа данных *(см. главу 1)* до ситуаций, где вы сталкиваетесь с большим количеством переменных и записей. Задача состоит в том, чтобы проникнуть вглубь набора данных и узнать, каким образом разные переменные друг с другом соотносятся. Приемы обучения без учителя обеспечивают способы отсеивания и анализа этих переменных, а также обнаружения глубинных связей.

Обучение без учителя и предсказание

Обучение без учителя может играть важную роль в предсказании как для задач регрессии, так и для задач классификации. В некоторых случаях мы хотим предсказать категорию в отсутствии каких-либо помеченных данных. Например, мы, возможно, захотим предсказать тип растительности в конкретном географическом районе исходя из набора спутниковых сенсорных данных. Поскольку у нас нет переменной отклика, чтобы натренировать модель, разбивка данных на кластеры предоставляет нам способ идентифицировать общие закономерности и распределить районы по категориям.

Кластеризация является особенно важным инструментом для задачи "холодного старта". В задачах такого типа, как запуск новой маркетинговой кампании либо идентификация потенциально новых типов мошенничества или спама, у нас первоначально может не оказаться какого-либо отклика, чтобы натренировать модель. С течением времени, когда данные уже собраны, мы можем узнать о системе больше и построить традиционную предсказательную модель. Но кластеризация помогает нам запустить процесс обучения быстрее путем идентификации сегментов популяции.

Обучение без учителя имеет также особое значение, как структурный элемент методологии регрессии и классификации. Если в условиях больших данных небольшая субпопуляция представлена в общей популяции не очень хорошо, то натренированная модель может не показать хорошую результативность для этой субпопуляции. При помощи кластеризации есть возможность идентифицировать и промаркировать субпопуляцию. Отдельные модели в дальнейшем могут быть подогнаны к разным субпопуляциям. Как вариант, субпопуляция может быть представлена собственным признаком, заставляя общую модель явно рассматривать идентичность субпопуляции в качестве предиктора.

Анализ главных компонент

Нередко переменные варьируются вместе (ковариативно), и часть вариации в одной переменной практически дублируется вариацией в другой. Анализ главных компонент (PCA, principal components analysis) — это метод поиска способа, которым числовые переменные соварьируются.

Ключевые термины

Главная компонента (principal component)

Линейная комбинация предикторных переменных.

Нагрузка (loadings)

Веса, которые позволяют трансформировать предикторы в компоненты.

Синоним: веса.

График каменистой осыпи (screeplot)

График дисперсий компонент, показывающий относительную важность компонент.

Идея PCA состоит в том, чтобы объединить многочисленные числовые предикторные переменные в меньший набор переменных, т. е. взвешенные линейные комбинации исходного набора. Меньший набор переменных (главные компоненты) "объясняет" бóльшую часть вариабельности полного набора переменных, сокращая размерность данных. Веса, используемые для формирования главных компонент, показывают относительные вклады исходных переменных в новые главные компоненты.

PCA-анализ был впервые предложен Карлом Пирсоном (см. **http://stat.smmu.edu.cn/history/pearson1901.pdf**). В своей статье, которая, возможно, стала первой работой, посвященной обучению без учителя, Пирсон признавал, что во многих задачах в предикторных переменных имеется вариабельность, и поэтому он разработал метод моделирования этой вариабельности. PCA можно рассматривать в качестве версии линейного дискриминантного анализа без учителя *(см. разд. "Дискриминантный анализ" главы 5)*.

Простой пример

Для двух переменных X_1 и X_2 имеются две главные компоненты Z_i ($i = 1$ или 2):

$$Z_i = w_{i,1}X_1 + w_{i,2}X_2.$$

Веса ($w_{i,1}$, $w_{i,2}$) называются *нагрузками* компонент. Они преобразуют исходные переменные в главные компоненты. Первая главная компонента, Z_1 — это линейная комбинация, которая лучше всего объясняет общую вариацию. Вторая главная компонента, Z_2, объясняет оставшуюся вариацию (это тоже линейная комбинация, но которая является худшим приближением).

Общепринято также вычислять главные компоненты на отклонениях от среднего предикторных переменных, а не на самих значениях.

Вы можете вычислить главные компоненты в R, используя функцию `princomp`. Представленный далее фрагмент кода выполняет PCA на доходности курса акций Chevron (CVX) и ExxonMobil (XOM):

```
oil_px <- sp500_px[, c('CVX', 'XOM')]
pca <- princomp(oil_px)
pca$loadings

Loadings:
    Comp.1 Comp.2
CVX -0.747 0.665
XOM -0.665 -0.747
```

Для акций CVX и XOM веса для первой главной компоненты равняются $-0,747$ и $-0,665$, а веса для второй главной компоненты составляют 0,665 и –0,747. Как это

интерпретировать? Первая главная компонента — это по существу среднее от CVX и XOM, отражающее корреляцию между этими двумя энергетическими компаниями. Вторая компонента измеряет, когда курсы акций CVX и XOM дивергируют.

Весьма поучительно отобразить на графике главные компоненты вместе с данными:

```
loadings <- pca$loadings
ggplot(data=oil_px, aes(x=CVX, y=XOM)) +
  geom_point(alpha=.3) +
  stat_ellipse(type='norm', level=.99) +
  geom_abline(intercept = 0, slope = loadings[2,1]/loadings[1,1]) +
  geom_abline(intercept = 0, slope = loadings[2,2]/loadings[1,2])
```

Результат показан на рис. 7.1.

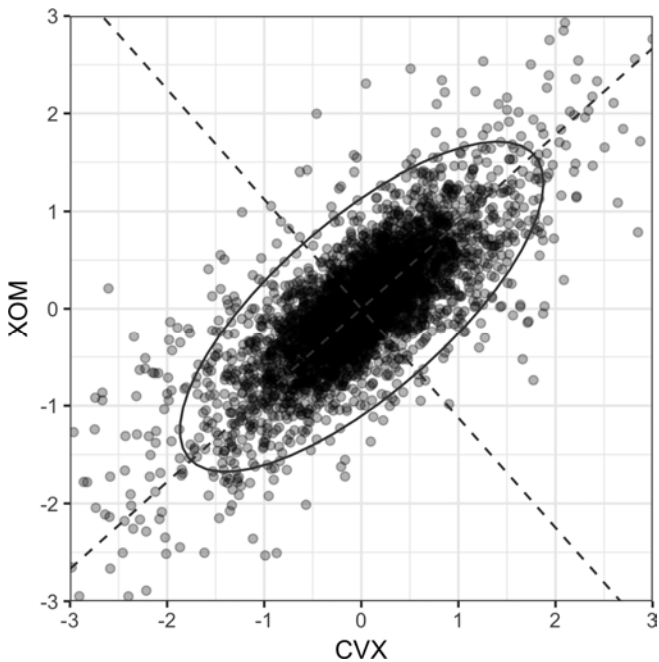


Рис. 7.1. Главные компоненты для доходности акций Chevron и ExxonMobil

Пунктирные линии показывают эти две главные компоненты: первая проходит вдоль длинной оси эллипса и вторая — вдоль короткой оси. Вы видите, что большинство вариабельности в двух доходностях акций объясняется первой главной компонентой. Это имеет смысл, поскольку курсы энергетических акций имеют тенденцию перемещаться всей группой.

Оба веса первой главной компоненты отрицательны, но инвертирование знака всех весов не изменяет главную компоненту. Например, использование весов 0,747 и 0,665 для первой главной компоненты эквивалентно отрицательным весам, так же как бесконечная линия, определенная началом координат и точкой с координатами (1, 1), одинакова с линией, определенной началом координат и точкой (–1, –1).

Вычисление главных компонент

Переход от двух переменных к большему их количеству довольно прямолинеен. В отношении первой компоненты нужно просто внести в линейную комбинацию дополнительные предикторные переменные, назначая веса, которые оптимизируют коллекцию сопряженной изменчивости (ковариантности), всех предикторных переменных в эту первую главную компоненту (*ковариация* — это статистический термин; *см. разд. "Ковариационная матрица" главы 5*). Вычисление главных компонент — это классический статистический метод, который опирается на корреляционную матрицу данных либо на ковариационную матрицу и выполняется очень быстро, не завися от итерации. Как отмечено ранее, он работает только с числовыми переменными, не категориальными. Полную процедуру вычисления можно описать следующим образом:

1. При создании первой главной компоненты метод PCA приходит к линейной комбинации предикторных переменных, которая максимизирует процент общей объясненной дисперсии.
2. Эта линейная комбинация далее становится первым "новым" предиктором, Z_1.
3. PCA повторяет этот процесс, используя те же переменные, но с разными весами, чтобы создать второй новый предиктор, Z_2. Взвешивание выполняется таким образом, чтобы Z_1 и Z_2 не коррелировались.
4. Процесс продолжается, пока не будет столько новых переменных, или компонент Z_i, сколько имеется исходных переменных X_i.
5. Оставить столько компонент, сколько нужно, чтобы была охвачена бóльшая часть дисперсии.
6. В этом месте получится набор весов для каждой компоненты. Последний шаг состоит в конвертации исходных данных в новые оценки главных компонент с помощью применения весов к исходным значениям. Эти новые оценки можно далее использовать в качестве сокращенного набора предикторных переменных.

Интерпретация главных компонент

Природа главных компонент часто позволяет сделать очевидной информацию о структуре данных. Существует несколько стандартных форм наглядного отображения, чтобы помочь вам вникнуть в суть главных компонент. Одним таким методом является *график каменистой осыпи* (Screeplot), предназначенный для визуализации относительной важности главных компонент (название графика происходит от сходства графика с каменистым откосом на боковой поверхности дорожного полотна). Представленный далее фрагмент кода является примером для нескольких ведущих компаний фондового индекса S&P 500:

```
syms <- c( 'AAPL', 'MSFT', 'CSCO', 'INTC', 'CVX', 'XOM',
   'SLB', 'COP', 'JPM', 'WFC', 'USB', 'AXP', 'WMT', 'TGT', 'HD', 'COST')
```

```
top_sp <- sp500_px[row.names(sp500_px)>='2005-01-01', syms]
sp_pca <- princomp(top_sp)
screeplot(sp_pca)
```

Как явствует из рис. 7.2, дисперсия первой главной компоненты довольно большая (как это часто и бывает), но другие верхние главные компоненты имеют важное значение.

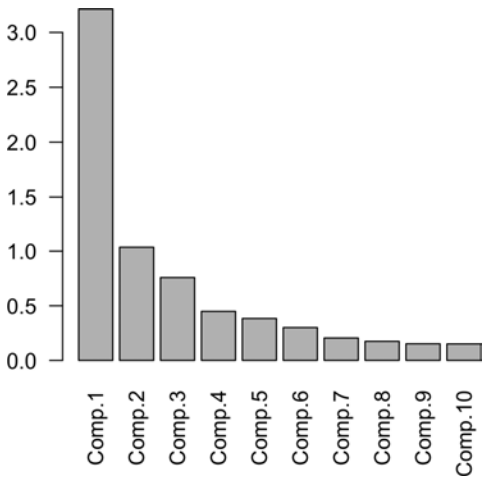


Рис. 7.2. График каменистой осыпи для PCA лидирующих акций S&P 500

В особенности очевидным может быть отображение весов верхних главных компонент. Один из способов состоит в использовании функции `gather` из программного пакета `tidyr` в сочетании с `ggplot`:

```
library(tidyr)
loadings <- sp_pca$loadings[,1:5]
loadings$Symbol <- row.names(loadings)
loadings <- gather(loadings, "Component", "Weight", -Symbol)
ggplot(loadings, aes(x=Symbol, y=Weight)) +
  geom_bar(stat='identity') +
  facet_grid(Component ~ ., scales='free_y')
```

Нагрузки для верхних пяти компонент показаны на рис. 7.3. Нагрузки для первой главной компоненты имеют одинаковый знак: это типично для данных, в которых все столбцы имеют общий множитель (в данном случае, общий тренд фондового рынка). Вторая компонента захватывает ценовые изменения энергетических акций по сравнению с другими акциями. Третья компонента преимущественно противопоставляет динамику цен Apple и CostCo. Четвертая компонента противопоставляет динамику цен Schlumberger другим энергетическим акциям. Наконец, в пятой компоненте доминирующие позиции занимают главным образом финансовые компании.

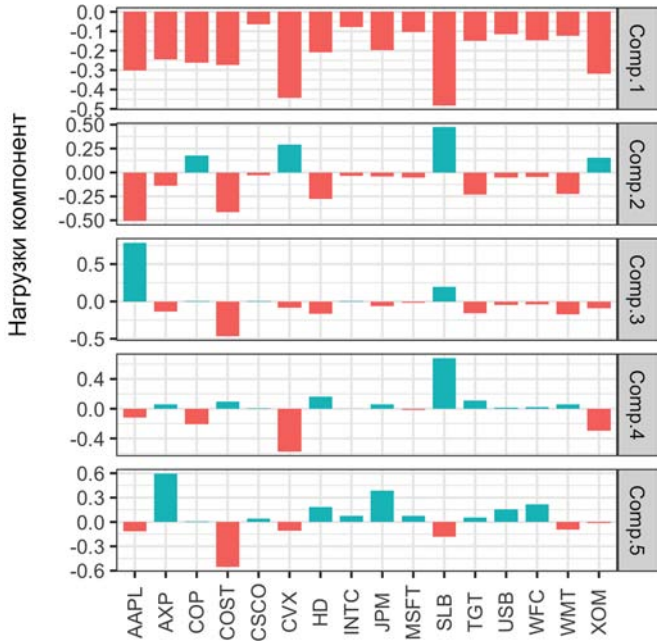


Рис. 7.3. Нагрузки для верхних пяти главных компонент доходности курса акций

Сколько компонент выбрать?

Если ваша цель состоит в том, чтобы сократить размерность данных, то вы должны принять решение, сколько главных компонент выбрать. Общепринятый подход состоит в использовании оперативного (ad hoc) правила отбирать компоненты, которые объясняют "бóльшую часть" дисперсии. Это можно сделать визуально посредством графика каменистой осыпи; например, на рис. 7.2 было бы вполне естественно ограничить анализ верхними пятью компонентами. Как вариант, вы можете отобрать верхние компоненты таким образом, чтобы совокупная дисперсия превышала порог, скажем, 80%. Кроме того, можно обследовать нагрузки, чтобы определить, имеет ли компонента интуитивно понятную интерпретацию. Перекрестная проверка предоставляет более формальный метод для отбора числа значимых компонент (*см. разд. "Перекрестная проверка" главы 4* для получения дополнительной информации).

Ключевые идеи для главных компонент

- Главные компоненты — это линейные комбинации предикторных переменных (только с числовыми данными).
- Их вычисляют с целью минимизировать корреляцию между компонентами, сокращая при этом избыточность.
- Предельное количество компонент, как правило, будет объяснять бóльшую часть дисперсии в переменной исхода.
- Этот предельный набор главных компонент далее можно использовать вместо (более многочисленных) исходных предикторов, тем самым сокращая размерность.

Дополнительные материалы для чтения

Для подробного обзора использования перекрестной проверки в анализе главных компонент см. статью "Перекрестная проверка компонентных моделей: критический взгляд на существующие методы" (Bro R., Kjeldahl K., Smilde A. K., Kiers H. A. L. Cross-validation of component models: a critical look at current methods // Analytical and Bioanalytical Chemistry. — 2008. — № 5. — P. 390), **http://bit.ly/2oW00Fl**.

Кластеризация на основе *K* средних

Кластеризация — это метод деления данных на разные группы, такие, что записи в каждой группе сходны друг с другом. Цель кластеризации состоит в том, чтобы идентифицировать значительные и содержательные группы данных. Группы могут использоваться непосредственно, анализироваться более углубленно либо передаваться как признак или исход в предсказательную модель регрессии или классификации. *K средних*, как метод кластеризации, был разработан самым первым; он по-прежнему широко используется и обязан своей популярностью относительной простоте алгоритма и его способности масштабироваться до больших наборов данных.

Ключевые термины

Кластер (cluster)

Группа схожих записей.

Центр кластера (cluster mean)

Вектор средних значений переменных для записей в кластере.

Синонимы: центроид, кластерное среднее, центр масс.

K

Количество кластеров.

Метод *K* средних делит данные на *K* кластеров путем минимизации суммы квадратических расстояний каждой записи до *среднего значения* назначенного ей кластера. Это среднее значение, также именуемое центром кластера, или центроидом, выражается *внутрикластерной суммой квадратов*, или *внутрикластерной SS*. *K* средних не гарантирует, что кластеры будут иметь одинаковый размер, но он находит кластеры, которые разделены наилучшим образом.

Нормализация

Как правило, непрерывные переменные нормализуются (стандартизируются) путем вычитания среднего значения и деления на стандартное отклонение. В противном случае переменные с крупной шкалой значений будут доминировать над процессом кластеризации *(см. разд. "Стандартизация (нормализация, z-оценки)" главы 6)*.

Простой пример

Начнем с того, что рассмотрим набор данных с n записями и всего двумя переменными, x и y. Предположим, что мы хотим разделить данные на $K = 4$ кластеров. Это значит, что нужно отнести каждую запись (x_i, y_i) к кластеру k. С учетом отнесения n_k записей к кластеру k, центром кластера $(\bar{x}_k, \bar{y}_k)$ является среднее значение точек в кластере:

$$\bar{x}_k = \frac{1}{n_k} \sum_{i \in \text{кластер } k} x_i;$$

$$\bar{y}_k = \frac{1}{n_k} \sum_{i \in \text{кластер } k} y_i.$$

Центр кластера

В кластеризации записей с многочисленными переменными (типичный случай) термин *"центр кластера"* обозначает не одно число, а вектор средних значений переменных.

Сумма квадратов внутри кластера задается следующей формулой:

$$SS_k = \sum_{i \in \text{кластер } k} \left(x_i - \bar{x}_k\right)^2 + \left(y_i - \bar{y}_k\right)^2.$$

Метод K средних находит такое отнесение записей к кластерам, которое минимизирует внутрикластерную сумму квадратов по всем четырем кластерам $SS_1 + SS_2 + SS_3 + SS_4$.

$$\sum_{k=1}^{4} SS_i.$$

Кластеризация методом K средних можно использовать, чтобы лучше понять динамику цен на акции относительно ее тенденции группироваться в кластеры. Отметим, что о доходности акций сообщается в виде, который практически стандартизирован, и поэтому нам не нужно выполнять нормализацию данных. В R кластеризация по методу K средних можно выполнить при помощи функции `kmeans`. Например, приведенный далее фрагмент кода находит четыре кластера на основе двух переменных — доходности акций ExxonMobil (`XOM`) и Chevron (`CVX`):

```
df <- sp500_px[row.names(sp500_px)>='2011-01-01', c('XOM', 'CVX')]
km <- kmeans(df, centers=4)
```

Отнесение к кластеру каждой записи возвращается в компоненте `cluster`:

```
> df$cluster <- factor(km$cluster)
> head(df)
                 XOM        CVX cluster
2011-01-03 0.73680496  0.2406809       2
2011-01-04 0.16866845 -0.5845157       1
2011-01-05 0.02663055  0.4469854       2
2011-01-06 0.24855834 -0.9197513       1
```

```
2011-01-07 0.33732892  0.1805111       2
2011-01-10 0.00000000 -0.4641675       1
```

Первые шесть записей отнесены к кластеру 1 либо к кластеру 2. Также возвращаются центры (средние значения) этих кластеров:

```
> centers <- data.frame(cluster=factor(1:4), km$centers)
> centers
  cluster        XOM        CVX
1       1 -0.3284864 -0.5669135
2       2  0.2410159  0.3342130
3       3 -1.1439800 -1.7502975
4       4  0.9568628  1.3708892
```

Кластеры 1 и 3 представляют "падающие" рынки, в то время как кластеры 2 и 4 — рынки "растущие". В этом примере всего с двумя переменными визуализация кластеров и их центров довольно прямолинейна:

```
ggplot(data=df, aes(x=XOM, y=CVX, color=cluster, shape=cluster)) +
  geom_point(alpha=.3) +
  geom_point(data=centers, aes(x=XOM, y=CVX), size=3, stroke=2)
```

Результирующий график, приведенный на рис. 7.4, демонстрирует отнесения к кластерам и центры кластеров.

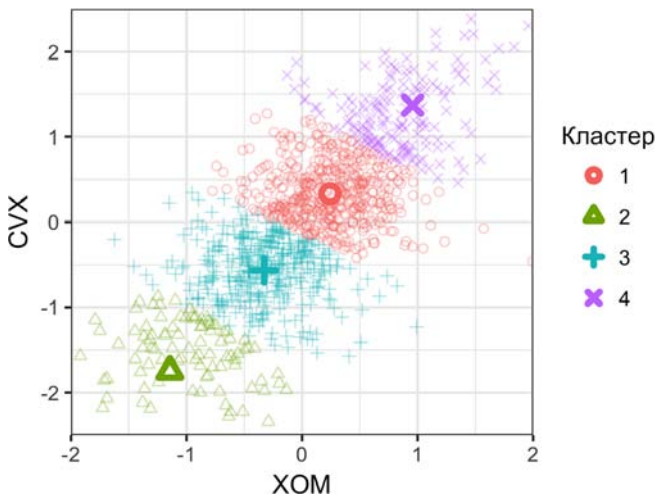


Рис. 7.4. Кластеры *K* средних применительно к данным курсов акций ExxonMobil и Chevron (центры двух кластеров в плотной области трудно различить)

Алгоритм *K* средних

В целом алгоритм *K* средних может применяться к набору данных с p переменными X_1, ..., X_p. Хотя найти точное решение *K* средних в вычислительном плане очень трудно, имеется возможность эффективно вычислить локальное оптимальное решение с привлечением эвристических алгоритмов.

Алгоритм начинается с определенного пользователем количества *K* и первоначального набора центров кластеров, затем многократно выполняет следующие шаги:

1. Отнести каждую запись к центру ближайшего кластера согласно измеренному квадратическому расстоянию.
2. Пересчитать центры кластеров по-новому на основе отнесения записей к кластерам.

Алгоритм сходится, когда отнесение записей к кластерам не изменяется.

Для первой итерации вам нужно указать исходный набор центров кластеров. Обычно это делается путем отнесения в произвольном порядке каждой записи к одному из *K* кластеров, а затем отыскивания средних этих кластеров.

Поскольку этот алгоритм не гарантирует нахождение лучшего решения, рекомендуется выполнять алгоритм несколько раз с использованием разных случайных выборок для инициализации алгоритма. Когда используется больше одного набора итераций, результат *K* средних задается итерацией, которая имеет самую низкую внутрикластерную сумму квадратов.

Параметр `nstart` для функции R `kmeans` позволяет определять число случайных запусков для попыток. Например, приведенный далее фрагмент кода выполняет алгоритм *K* средних, чтобы найти 5 кластеров с использованием 10 разных начальных центров кластеров:

```
syms <- c( 'AAPL', 'MSFT', 'CSCO', 'INTC', 'CVX', 'XOM', 'SLB', 'COP',
           'JPM', 'WFC', 'USB', 'AXP', 'WMT', 'TGT', 'HD', 'COST')
df <- sp500_px[row.names(sp500_px)>='2011-01-01', syms]
km <- kmeans(df, centers=5, nstart=10)
```

Функция автоматически возвращает лучшее решение из 10 разных начальных точек. Вы можете использовать параметр `iter.max` для определения максимального числа итераций, которое дается алгоритму для каждого случайного запуска.

Интерпретация кластеров

Важная часть кластерного анализа может быть сопряжена с интерпретацией кластеров. Двумя самыми важными элементами данных на выходе из `kmeans` являются размеры кластеров и центры кластеров. Например, в предыдущем подразделе размеры результирующих кластеров задаются такой командой R:

```
km$size
[1] 186 106 285 288 266
```

Размеры кластеров относительно сбалансированы. Несбалансированные кластеры могут быть вызваны дальними выбросами либо группами записей, которые сильно отличаются от остальной части данных — обе причины могут нуждаться в дальнейшем изучении.

Вы можете построить график центров кластеров при помощи функции `gather` в сочетании с `ggplot`:

```
centers <- as.data.frame(t(centers))
names(centers) <- paste("Cluster", 1:5)
centers$Symbol <- row.names(centers)
centers <- gather(centers, "Cluster", "Mean", -Symbol)
centers$Color = centers$Mean > 0
ggplot(centers, aes(x=Symbol, y=Mean, fill=Color)) +
  geom_bar(stat='identity', position = "identity", width=.75) +
  facet_grid(Cluster ~ ., scales='free_y')
```

Результирующий график показан на рис. 7.5 и демонстрирует природу каждого кластера. Например, кластеры 1 и 2 соответствуют дням, в которые рынок падает и растет. Кластеры 3 и 5 характеризуются соответственно днями растущего рынка акций потребительского рынка и днями падающего рынка энергетических акций. Наконец, кластер 4 фиксирует дни, в которые энергетические акции росли, а акции потребительского рынка падали.

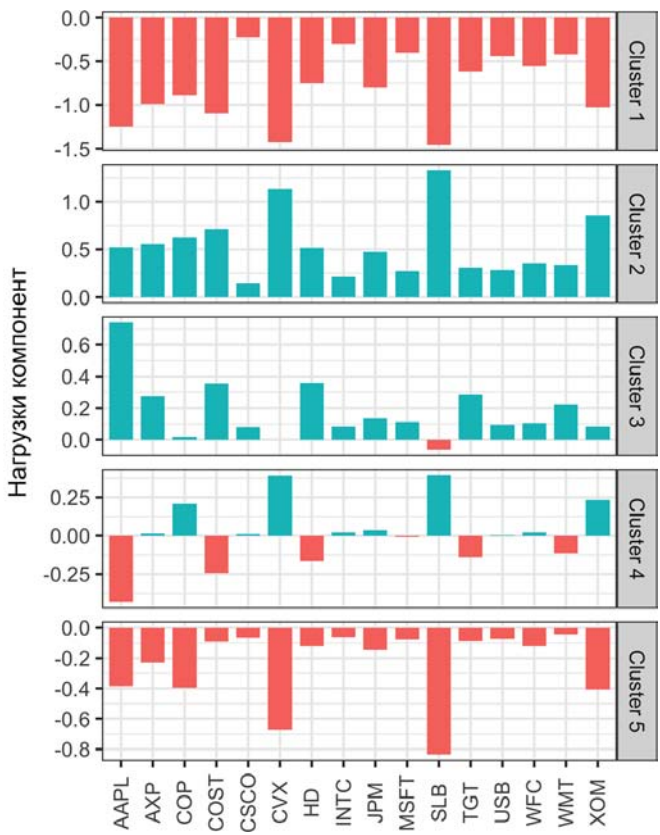


Рис. 7.5. Средние значения переменных в каждом кластере ("центроиды")

Кластерный анализ против PCA

График центров кластеров в сущности аналогичен рассмотрению нагрузок в анализе главных компонент (PCA); *см. разд. "Интерпретация главных компонент" ранее в этой главе*. Главное отличие в том, что в PCA знак кластерных центров имеет значение. PCA-анализ идентифицирует главные направления вариации, тогда как кластерный анализ находит группы записей, расположенных рядом друг к другу.

Выбор количества кластеров

Алгоритм *K* средних требует, чтобы вы определили количество кластеров *K*. Иногда количество кластеров обусловлено применением. Например, компания, организующая работу отдела продаж, может захотеть сгруппировать клиентов в "типажи", на которых сосредоточить свою деятельность и звонить им с коммерческими предложениями. В таком случае организаторские соображения будут диктовать количество планируемых потребительских сегментов — например, два сегмента могут не привести к полезному разделению клиентов, тогда как восемь может быть слишком много, чтобы с ними справиться.

В отсутствие количества кластеров, продиктованного практическими или организаторскими соображениями, можно воспользоваться статистическим подходом. Не существует единого стандартного метода нахождения "лучшего" количества кластеров.

Общепринятый подход, который называется *методом локтя* (elbow method), состоит в идентификации точки, когда набор кластеров объясняет "большинство" дисперсии в данных. Добавление новых кластеров вне этого набора вносит относительно малый инкрементный вклад в объясненную дисперсию. Локоть — это точка, где совокупная объясненная дисперсия выравнивается после крутого подъема, отсюда и название данного метода.

На рис. 7.6 отображен совокупный процент дисперсии, объясненной для данных о невозвратах для количества кластеров в пределах от 2 до 15. Где же локоть в этом примере? Очевидный кандидат отсутствует, поскольку инкрементное увеличение дисперсии постепенно падает. Это довольно типичная ситуация для данных, которые не имеют четко определенных кластеров. Возможно, в этом заключается недостаток метода локтя, но он действительно показывает природу данных.

В R функция `kmeans` не обеспечивает единую команду для применения метода локтя, но его можно с легкостью применить на основе данных на выходе из `kmeans`, как показано ниже:

```
pct_var <- data.frame(pct_var = 0,
                      num_clusters=2:14)
totalss <- kmeans(df, centers=14, nstart=50, iter.max = 100)$totss
for(i in 2:14){
  pct_var[i-1, 'pct_var'] <- kmeans(df, centers=i, nstart=50, iter.max = 100)
    $betweenss/totalss
}
```

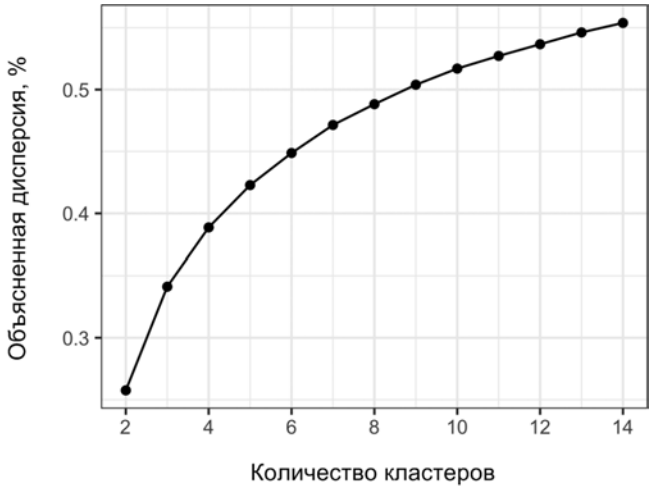


Рис. 7.6. Метод локтя применительно к данным об акциях

В оценке того, сколько кластеров должно остаться, возможно, самая важная проверка состоит в следующем: какова вероятность, что кластеры будут повторены на новых данных? Поддаются ли кластеры интерпретации и связаны ли они с общими характеристиками данных или же они просто отражают конкретный экземпляр? Вы можете это частично квалифицировать при помощи перекрестной проверки *(см. разд. "Перекрестная проверка" главы 4)*.

В целом единое правило, которое будет надежно инструктировать относительно того, сколько кластеров создавать, отсутствует.

Существует несколько более формальных способов определения количества кластеров, которые основываются на статистической теории либо теории информации. Например, Роберт Тибширани (Robert Tibshirani), Гюнтер Уолтер (Guenther Walther) и Тревор Хейсти (Trevor Hastie) (**http:// www.stanford.edu/ ~hastie/Papers/gap.pdj**) предлагают статистику "gap" (скачок), опираясь на статистическую теорию при идентификации локтя. Для большинства приложений теоретический подход, вероятно, не понадобится, или даже не будет обоснован.

Ключевые идеи для кластеризации по методу *K* средних

- Число требующихся кластеров *K* выбирается пользователем.
- Алгоритм совершенствует кластеры путем итеративного отнесения записей к наиближайшему центру кластера, пока отнесения к кластерам не перестанут изменяться.
- Над выбором *K* обычно доминируют соображения практического характера; статистически обусловленного оптимального количества кластеров не существует.

Иерархическая кластеризация

Иерархическая кластеризация — это альтернативный для *K* средних метод, который может порождать совсем другие кластеры. Иерархическая кластеризация гибче *K* средних и проще воспринимает нечисловые переменные. Она более чувствительна в обнаружении отдаленных или отклоняющихся групп либо записей. Иерархическая кластеризация также поддается интуитивно понятному графическому отображению, приводя к более простой интерпретации кластеров.

Ключевые термины

Дендограмма (dendrogram)

Визуальное представление записей и иерархии кластеров, которым они принадлежат.

Расстояние (distance)

Метрический показатель степени близости одной записи к другой.

Различие (dissimilarity)

Метрический показатель степени близости одного кластера к другому.

Синонимы: неподобие, несхожесть.

За гибкость иерархической кластеризации нужно платить — иерархическая кластеризация хорошо не масштабируется до крупных наборов данных с миллионами записей. Даже для данных скромного размера всего с десятками тысяч записей иерархическая кластеризация может потребовать интенсивных вычислительных ресурсов. И действительно, большинство приложений иерархической кластеризации сосредоточены на относительно небольших наборах данных.

Простой пример

Иерархическая кластеризация работает на наборе данных с n записями и p переменными и основывается на двух главных структурных элементах:

- метрическом показателе расстояния $d_{i,j}$, который измеряет расстояние между двумя записями i и j;
- метрическом показателе различия $D_{A,B}$, который измеряет разницу между двумя кластерами A и B, основываясь на расстояниях $d_{i,j}$ между членами каждого кластера.

Для приложений, сопряженных с числовыми данными, наиболее важным решением является выбор метрического показателя различия. Иерархическая кластеризация начинается с установки каждой записи в качестве своего собственного кластера (т. е. создаются одноэлементные кластеры по числу записей) и многократно выполняется с целью объединения наименее несхожих кластеров.

В R функция `hclust` может использоваться для выполнения иерархической кластеризации. Одно большое отличие `hclust` от `kmeans` состоит в том, что данная функция оперирует на попарных расстояниях $d_{i,j}$ нежели самих данных, как таковых. Вы можете их вычислить при помощи функции `dist`. Например, приведенный далее фрагмент кода применяет иерархическую кластеризацию к доходностям акций для ряда компаний:

```
syms1 <- c('GOOGL', 'AMZN', 'AAPL', 'MSFT', 'CSCO', 'INTC', 'CVX',
           'XOM', 'SLB', 'COP', 'JPM', 'WFC', 'USB', 'AXP',
            'WMT', 'TGT', 'HD', 'COST')
# транспонировать: чтобы кластеризовать компании, нужно
# разместить акции в строках
df <- t(sp500_px[row.names(sp500_px)>='2011-01-01', syms1])
d <- dist(df)
hcl <- hclust(d)
```

Алгоритм кластеризации будет распределять записи (строки) кадра данных по кластерам. Поскольку мы хотим кластеризовать компании, мы должны *транспонировать* кадр данных и распределить акции по строкам, а даты — по столбцам.

Дендограмма

Иерархическая кластеризация поддается естественному графическому отображению в виде дерева, которое называется *дендограммой*. Это название происходит от греческих слов *dendro* (дерево) и *gramma* (рисунок). В R вы можете легко произвести дендограмму при помощи команды `plot`:

```
plot(hcl)
```

Результат показан на рис. 7.7. Листья дерева соответствуют записям. Длина ветви дерева говорит о степени различия между соответствующими кластерами. Доходности акций Google и Amazon довольно отличаются от доходности других акций. Другие акции попадают в естественные группы: энергетические акции, финансовые акции и акции потребительского сектора — все они распределены по собственным поддеревьям.

В отличие от метода *K* средних, нет необходимости предварительно определять количество кластеров. Для того чтобы извлечь конкретное количество кластеров, вы можете воспользоваться функцией `cutree`:

```
cutree(hcl, k=4)
GOOGL AMZN AAPL MSFT CSCO INTC CVX XOM SLB COP JPM WFC
    1    2    3    3    3    3   4   4   4   4   3   3
  USB  AXP  WMT  TGT   HD COST
    3    3    3    3    3    3
```

Количество извлекаемых кластеров принимается равным 4, и вы видите, что акции Google и Amazon каждые в отдельности принадлежат своему кластеру. Нефтяные акции (XOM, CVS, SLB, COP) принадлежат другому кластеру. Оставшиеся акции находятся в четвертом кластере.

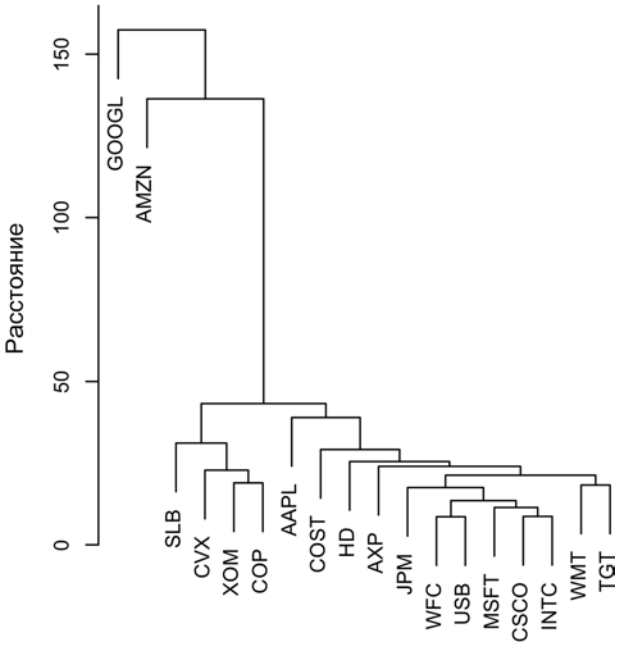


Рис. 7.7. Дендограмма акций

Агломеративный алгоритм

Главным алгоритмом иерархической кластеризации является *агломеративный* алгоритм, который итеративно объединяет схожие кластеры. Агломеративный алгоритм начинает свою работу с того, что делает каждую запись ее собственным одноэлементным кластером, затем достраивает кластеры все бóльших и бóльших размеров. Первый шаг состоит в вычислении расстояния между всеми парами записей.

Для каждой пары записей $(x_1, x_2, ..., x_p)$ и $(y_1, y_2, ..., y_p)$ мы измеряем расстояние между двумя записями, $d_{x,y}$, используя метрический показатель расстояния *(см. разд. "Метрические показатели расстояния" главы 6)*. Например, мы можем использовать евклидово расстояние:

$$d(x, y) = \sqrt{(x_1 - y_1)^2 + (x_2 - y_2)^2 + ... + (x_p - y_p)^2}.$$

Теперь мы переходим к межкластерному расстоянию. Рассмотрим два кластера A и B, каждый с характерным набором записей, $A = (a_1, a_2, ..., a_m)$ и $B = (b_1, b_2, ..., b_m)$. Мы можем измерить различие между кластерами $D(A, B)$ при помощи расстояний между членами A и членами B.

Одной из мер различия является метод *полной связи*, т. е. максимальное расстояние по всем парам записей между A и B:

$$D(A, B) = \max d(a_i, b_j) \text{ для всех пар } i, j.$$

Эта формула определяет различие, как самую большую разницу между всеми парами.

Главные шаги агломеративного алгоритма следующие:

1. Создать исходный набор кластеров для всех записей в данных, где каждый кластер состоит из единственной записи.
2. Вычислить различие $D(C_k, C_l)$ между всеми парами кластеров *k*, *l*.
3. Объединить два кластера C_k и C_l, которые наименее отличаются, согласно измерению при помощи $D(C_k, C_l)$.
4. Если осталось более одного кластера, то вернуться к шагу 2. В противном случае работа завершена.

Меры различия

Существует четыре общепринятые меры различия: *полная связь, одиночная связь, средняя связь* и *минимальная дисперсия*. Все перечисленные (плюс другие меры) поддерживаются большинством программных систем иерархической кластеризации, включая программный пакет `hclust`. Определенный ранее метод полной связи порождает кластеры со схожими членами. Метод одиночной связи минимизирует расстояние между записями в двух кластерах:

$$D(A, B) = \min d(a_i, b_j) \text{ для всех пар } i, j.$$

Это "жадный" метод, и он порождает кластеры, которые могут содержать довольно разрозненные элементы. Метод средней связи — это среднее всех расстояний рассматриваемых пар, и он представляет компромисс между методами одиночной и полной связи. Наконец, метод минимальной дисперсии, который также называется методом Уорда (Ward), аналогичен методу *K* средних, поскольку он минимизирует внутрикластерную сумму квадратов *(см. разд. "Кластеризация на основе K средних" ранее в этой главе)*.

На рис. 7.8 отражена иерархическая кластеризация с привлечением четырех мер для доходностей акций ExxonMobil и Chevron. Для каждой меры оставлены четыре кластера.

Результаты поразительно отличаются: мера одиночной связи относит почти все точки в один кластер. За исключением метода минимальной дисперсии (метода Уорда), все меры заканчивают по крайней мере одним кластером всего с несколькими отдаленными точками. Метод минимальной дисперсии наиболее близкий кластеру *K* средних; сравните с рис. 7.4.

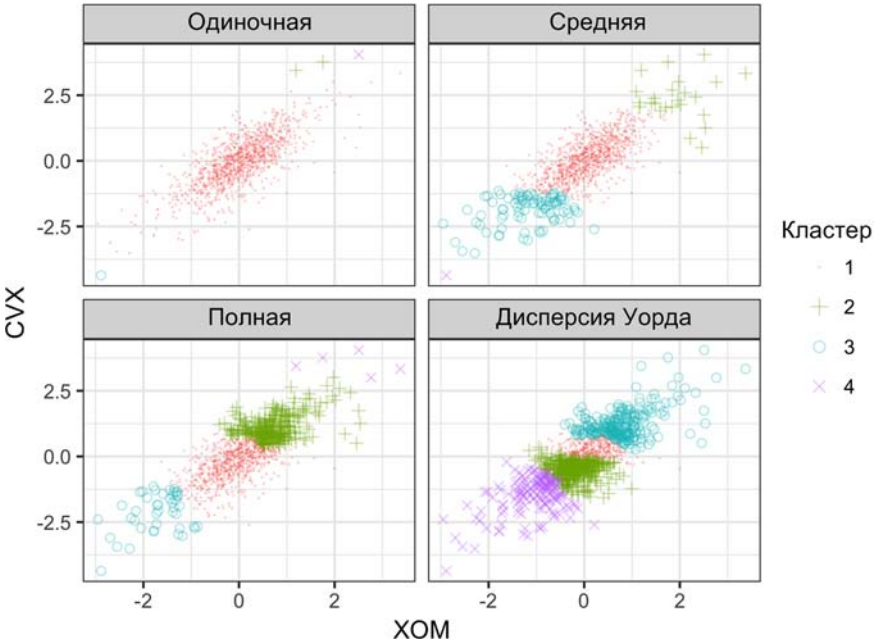


Рис. 7.8. Сравнение мер различия применительно к данным об акциях

Ключевые идеи для иерархической кластеризации

- Следует начать с того, что поместить каждую запись в собственный кластер.
- Кластеры постепенно соединяются с соседними кластерами, пока все записи не будут принадлежать единственному кластеру (агломеративный алгоритм).
- Последовательность агломерирования сохраняется и наносится на график, а пользователь (без предварительного указания количества кластеров) может наглядно увидеть количество и структуру кластеров на разных этапах.
- Межкластерные расстояния вычисляются по-разному, все они опираются на набор всех расстояний между записями.

Модельно-ориентированная кластеризация

Методы кластеризации, такие как иерархическая кластеризация и *K* средних, базируются на эвристиках и преимущественно опираются на нахождение кластеров, члены которых находятся близко друг к другу согласно измерениям, полученным непосредственно при помощи данных (вероятностная модель не участвует). За прошедшие 20 лет много сил было потрачено на разработку методов *кластеризации на основе моделей*. Эдриан Рэфтери (Adrian Raftery) и другие исследователи из Вашингтонского университета внесли весомый вклад в методы модельно-ориентированной кластеризации, включая и теорию, и программное обеспечение. Данные приемы опираются на статистическую теорию и обеспечивают более стро-

гие способы определения природы и количества кластеров. Они могут использоваться, например, в случаях, где может иметься одна группа записей, которые схожи друг с другом, но не обязательно близко расположены друг к другу (например, технологические акции с высокой дисперсией доходности), и другая группа записей, которые схожи и близко расположены (например, акции компаний коммунального хозяйства с низкой дисперсией).

Многомерное нормальное распределение

Наиболее широко используемые методы модельно-ориентированной кластеризации опираются на *многомерное нормальное распределение*. Это обобщение нормального распределения на набор из p переменных $X_1, X_2, ..., X_p$. Распределение определяется набором средних $\mu = \mu_1, \mu_2, ..., \mu_p$ и ковариационной матрицей $\mathbf{\Sigma}$. Ковариационная матрица является мерой того, как переменные коррелируют друг с другом (*см. разд. "Ковариационная матрица" главы 5*, чтобы узнать подробнее о ковариации). Ковариационная матрица $\mathbf{\Sigma}$ состоит из p дисперсий $\sigma_1^2, \sigma_2^2, ..., \sigma_p^2$ и ковариаций $\sigma_{i,j}$ для всех пар переменных $i \neq j$. С переменными, распределенными по строкам и продублированными по столбцам, матрица выглядит следующим образом:

$$\mathbf{\Sigma} = \begin{bmatrix} \sigma_1^2 & \sigma_{1,2} & \cdots & \sigma_{1,p} \\ \sigma_{2,1} & \sigma_2^2 & \cdots & \sigma_{2,p} \\ \vdots & \vdots & \ddots & \vdots \\ \sigma_{p,1} & \sigma_{p,2} & \cdots & \sigma_p^2 \end{bmatrix}.$$

Поскольку ковариационная матрица симметрична, и $\sigma_{i,j} = \sigma_{j,i}$, существуют всего $p \times (p-1) - p$ членов ковариации. В общей сложности, ковариационная матрица имеет $p \times (p-1)$ параметров. Распределение обозначается следующим:

$$\left(X_1, X_2, ..., X_p\right) \tilde{N}_p\left(\mu, \mathbf{\Sigma}\right).$$

Это аналитический способ сказать, что все переменные нормально распределены, и общее распределение полностью описывается вектором средних значений переменных и ковариационной матрицей.

На рис. 7.9 показаны контуры вероятностей для многомерного нормального распределения двух переменных *X* и *Y* (0,5 контур вероятности, например, содержит 50% распределения).

Средние равняются $\mu_x = 0,5$ и $\mu_y = -0,5$, и ковариационная матрица имеет вид:

$$\mathbf{\Sigma} = \begin{bmatrix} 1 & 1 \\ 1 & 2 \end{bmatrix}.$$

Поскольку ковариация σ_{xy} положительная, *X* и *Y* коррелируются положительно.

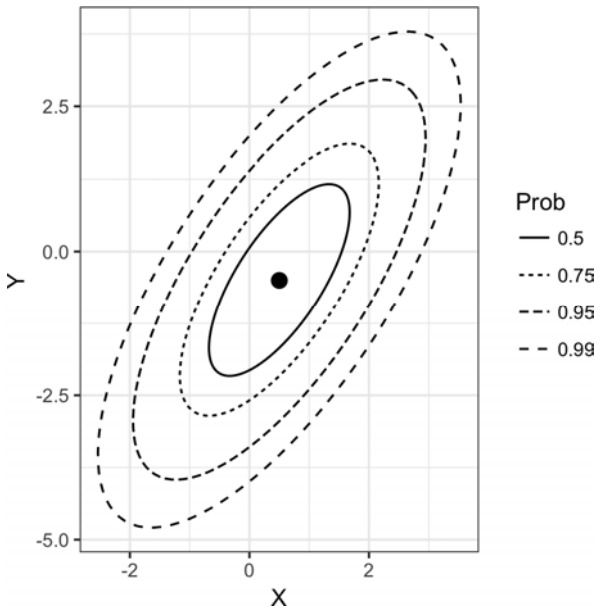


Рис. 7.9. Контуры вероятностей для двумерного нормального распределения

Смеси нормальных распределений

В основе модельно-ориентированной кластеризации лежит ключевая идея, которая состоит в том, что каждая запись принимается распределенной, как одно из K многомерных нормальных распределений, где K — количество кластеров. Каждое распределение имеет разное среднее μ и ковариационную матрицу $\mathbf{\Sigma}$. Например, если имеется две переменные X и Y, тогда каждая строка (X_i, Y_i) моделируется, как отобранная одного из K распределений $\left(N_1(\mu_1), \mathbf{\Sigma}_1\right)$, $\left(N_1(\mu_2), \mathbf{\Sigma}_2\right)$, ..., $\left(N_1(\mu_K), \mathbf{\Sigma}_K\right)$.

R имеет очень богатый программный пакет для модельно-ориентированной кластеризации, который называется `mclust`, первоначально разработанный Крисом Фрейли (Chris Fraley) и Эрианом Рэфтери (Adrian Raftery). При помощи данного пакета мы можем применить модельно-ориентированную кластеризацию к данным доходности акций, которые мы ранее проанализировали при помощи методов K средних и иерархической кластеризации:

```
> library(mclust)
> df <- sp500_px[row.names(sp500_px)>='2011-01-01', c('XOM', 'CVX')]
> mcl <- Mclust(df)
> summary(mcl)
Mclust VEE (ellipsoidal, equal shape and orientation) model with 2 components:
 log.likelihood    n df       BIC       ICL
      -2255.134 1131  9 -4573.546 -5076.856

Clustering table:
  1   2
963 168
```

Если выполнить этот фрагмент кода, то можно заметить, что вычисление требует значительно больше времени, чем другие процедуры. После извлечения кластерных назначений при помощи функции `predict` мы можем визуализировать кластеры:

```
cluster <- factor(predict(mcl)$classification)
ggplot(data=df, aes(x=XOM, y=CVX, color=cluster, shape=cluster)) +
  geom_point(alpha=.8)
```

Результирующий график показан на рис. 7.10. Имеется два кластера: один кластер посередине данных и второй кластер во внешнем крае данных. Это сильно отличается от кластеров, полученных при помощи *K* средних (см. рис. 7.4) и иерархической кластеризации (см. рис. 7.8), которые находят кластеры, которые имеют компактный вид.

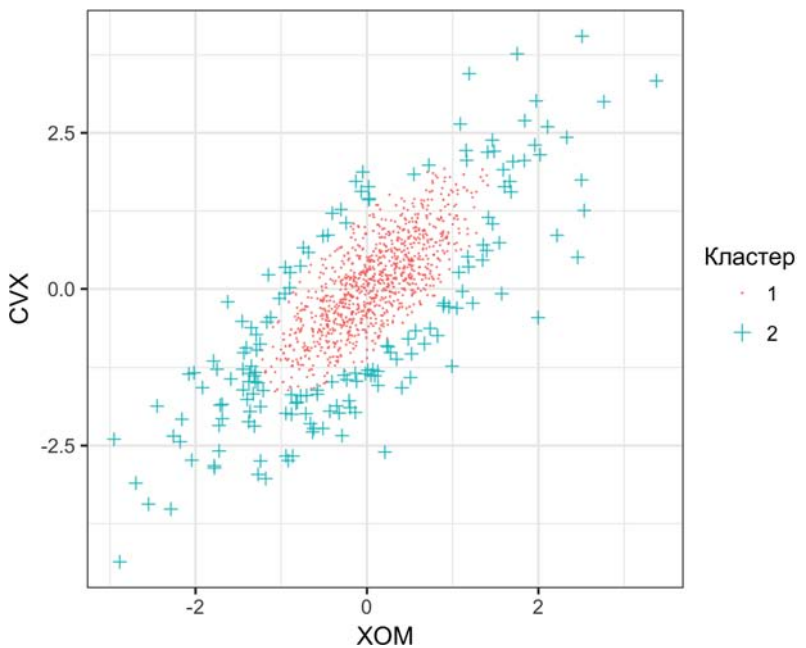


Рис. 7.10. Два кластера, полученные для данных доходности акций при помощи пакета `mclust`

Мы можем извлечь параметры нормальных распределений при помощи функции `summary`:

```
> summary(mcl, parameters=TRUE)$mean
          [,1]        [,2]
XOM 0.05783847 -0.04374944
CVX 0.07363239 -0.21175715
> summary(mcl, parameters=TRUE)$variance
, , 1
          XOM       CVX
```

```
XOM 0.3002049 0.3060989
CVX 0.3060989 0.5496727
, , 2
         XOM      CVX
XOM 1.046318 1.066860
CVX 1.066860 1.915799
```

Распределения имеют схожие средние и корреляции, но у второго распределения намного более крупные дисперсии и ковариации.

Кластеры из `mclust` могут показаться удивительными, но фактически они иллюстрируют статистическую природу метода. Цель модельно-ориентированной кластеризации состоит в том, чтобы найти набор оптимально подогнанных многомерных нормальных распределений. Данные об акциях, похоже, внешне имеют нормальную форму (см. контуры на рис. 7.9). Фактически, тем не менее, доходности акций имеют распределение с более длинным хвостом, чем нормальное распределение. Для того чтобы уладить это, `mclust` выполняет подгонку распределения к основной части данных, но затем осуществляет подгонку второго распределения с большей дисперсией.

Выбор количества кластеров

В отличие от *K* средних и иерархической кластеризации, `mclust` отбирает количество кластеров автоматически (в данном случае, два). Он делает это путем выбора количества кластеров, для которых *байесовский информационный критерий* (bayesian information criteria, BIC) имеет самое большое значение. BIC (аналогичный AIC) — это общий инструмент для нахождения наилучшей модели среди набора возможных моделей. Например, AIC (или BIC) широко используется для отбора модели в шаговой регрессии *(см. разд. "Отбор модели и шаговая регрессия" главы 4)*. BIC работает путем отбора оптимально подогнанной модели со штрафом за число параметров в модели. В случае модельно-ориентированной кластеризации добавление большего количества кластеров будет всегда улучшать подгонку за счет введения дополнительных параметров в модель.

Вы можете отобразить на графике значения BIC для каждого размера кластера, используя функцию в `hclust`:

```
plot(mcl, what='BIC', ask=FALSE)
```

Количество кластеров — или количество разных многомерных нормальных моделей (компонент) — показано на оси *x* (рис. 7.11).

Данный график аналогичен графику, построенному по методу локтя, используемому для идентификации количества кластеров для *K* средних, за исключением того, что на графике отображается значение BIC, а не процент объясненной дисперсии (см. рис. 7.6). Одно большое отличие состоит в том, что вместо одной линии `mclust` показывает 14 разных линий! Это вызвано тем, что `mclust` выполняет подгонку 14 разных моделей для каждого размера кластера и в конечном счете выбирает оптимально подогнанную модель.

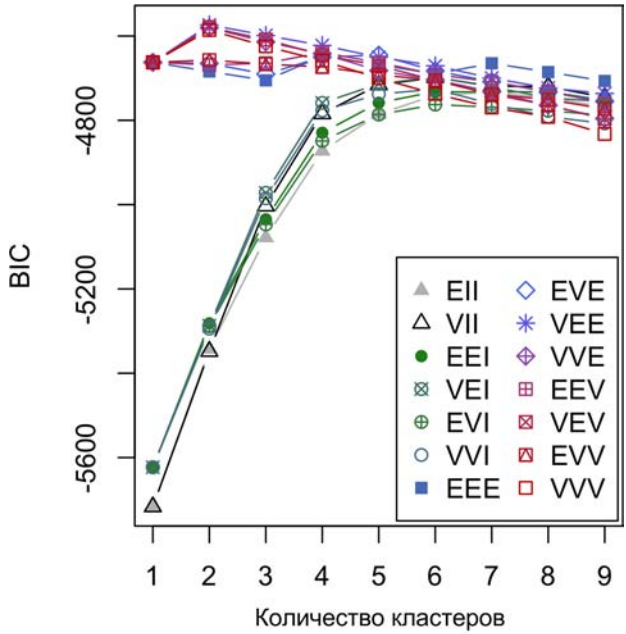


Рис. 7.11. BIC-оценки для данных о доходностях акций для разных количеств кластеров (компонент)

Почему `mclust` подгоняет столько много моделей для определения лучшего набора многомерных нормальных распределений? Потому что существуют разные способы параметризации ковариационной матрицы $\mathbf{\Sigma}$ для подгонки модели. В большинстве случаев вам не нужно беспокоиться по поводу деталей моделей, и вы можете спокойно использовать модель, выбранную пакетом `mclust`. В данном примере, согласно BIC, три разных модели (именуемые VEE, VEV и VVE) дают оптимальную подгонку с использованием двух компонентов.

Модельно-ориентированная кластеризация является богатой и стремительно развивающейся областью исследования, и освещение данной области в этом тексте охватывает лишь небольшую ее часть. Действительно, справочный файл `mclust` в настоящее время имеет 154 страницы. Подробное изучение нюансов в модельно-ориентированной кластеризации, вероятно, выходит за рамки того, что необходимо для большинства задач, с которыми встречаются аналитики данных.

Приемы модельно-ориентированной кластеризации, правда, имеют несколько ограничений. Данные методы требуют принятия основного допущения о характере модели для данных, и кластерные результаты очень зависят от этого допущения. Ее вычислительные потребности выше, чем даже при иерархической кластеризации, затрудняя масштабирование на большие данные. Наконец, ее алгоритм более сложен и менее доступен, чем в других методах.

Ключевые идеи для модельно-ориентированной кластеризации

- Предполагаются, что кластеры происходят из различных процессов, порождающих данные, с разными вероятностными распределениями.
- Далее выполняются подгонки разных моделей, принимая разные количества (обычно нормальных) распределений.
- Данный метод выбирает модель (и связанное с ней количество кластеров), которая хорошо подходит к данным без использования слишком большого числа параметров (т. е. переподгонки).

Дополнительные материалы для чтения

Подробнее о модельно-ориентированной кластеризации см. документацию по `mclust` (**http://www.stat.washington.edu/research/reports/2012/tr597.pdf**).

Шкалирование и категориальные переменные

Приемы обучения без учителя обычно требуют, чтобы данные были соответствующим образом прошкалированы. В этом состоит отличие от многих приемов регрессии и классификации, в которых шкалирование не имеет значения (исключением является метод *K* ближайших соседей; *см. разд. "K ближайших соседей" главы 6*).

Ключевые термины

Шкалирование (scaling)

Сплющивание или расширение данных обычно для приведения многочисленных переменных к одинаковой шкале измерения.

Нормализация (normalization)

Один из методов шкалирования — вычитание среднего и деление на стандартное отклонение.

Синоним: стандартизация.

Расстояние Говера (Gower's distance)

Алгоритм шкалирования, применяемый к смешанным числовым и категориальным данным для приведения всех переменных к диапазону 0–1.

Например, если говорить о данных персональной ссуды, переменные имеют самые разные единицы измерения и величины. У некоторых переменных относительно малые значения (например, число используемых лет), в то время как у других — очень большие (например, сумма кредита в долларах). Если данные не прошкалировать, то переменные с большими значениями будут доминировать над PCA,

K средних и другими методами кластеризации, а переменные с малыми значениями будут ими проигнорированы.

Для некоторых процедур кластеризации категориальные данные могут представлять особую проблему. Как и в случае с *K* ближайшими соседями, непорядковые факторные переменные обычно конвертируются в набор двоичных (0/1) переменных с использованием кодирования с одним активным состоянием *(см. разд. "Кодировщик с одним активным состоянием" главы 6)*. У двоичных переменных шкала измерения отличается от других шкал данных, и вдобавок факт, что двоичные переменные имеют всего два значения, может создать проблемы применения таких приемов, как PCA и *K* средних.

Шкалирование переменных

Переменные с совсем другой шкалой и единицами измерения необходимо соответствующим образом нормализовать перед применением процедуры кластеризации. Например, давайте применим `kmeans` к набору данных о невозвратных ссудах без нормализации:

```
df <- defaults[, c('loan_amnt', 'annual_inc', 'revol_bal', 'open_acc',
                   'dti', 'revol_util')]
km <- kmeans(df, centers=4, nstart=10)
centers <- data.frame( size=km$size, km$centers)
round(centers, digits=2)
   size loan_amnt annual_inc revol_bal open_acc   dti revol_util
1    55  23157.27  491522.49  83471.07    13.35  6.89      58.74
2  1218  21900.96  165748.53  38299.44    12.58 13.43      63.58
3  7686  18311.55   83504.68  19685.28    11.68 16.80      62.18
4 14177  10610.43   42539.36  10277.97     9.60 17.73      58.05
```

Переменные `annual_inc` и `revol_bal` доминируют над кластерами, и кластеры имеют очень разные размеры. Кластер 1 имеет всего 55 членов со сравнительно высоким сальдо годового дохода и возобновляемого кредита.

Общепринятый подход к шкалированию переменных состоит в их конвертировании в *z*-оценки путем вычитания среднего значения и деления на стандартное отклонение. Данная процедура называется стандартизацией или нормализацией (*см. разд. "Стандартизация (нормализация, z-оценки)" главы 6*, где использование *z*-оценок рассматривается подробнее):

$$z = \frac{x - \overline{x}}{s}.$$

Посмотрим, что произойдет с кластерами, когда `kmeans` применяется к нормализованным данным:

```
df0 <- scale(df)
km0 <- kmeans(df0, centers=4, nstart=10)
centers0 <-scale(km0$centers, center=FALSE,
                 scale=1/attr(df0, 'scaled:scale'))
```

```
centers0 <- scale(centers0, center=-attr(df0, 'scaled:center'), scale=F)
data.frame(size=km0$size, centers0)
  size loan_amnt annual_inc revol_bal       open_acc  dti revol_util
1 5429  10393.60   53689.54   6077.77     8.69 11.35           30.69
2 6396  13310.43   55522.76  16310.95    14.25 24.27           59.57
3 7493  10482.19   51216.95  11530.17     7.48 15.79           77.68
4 3818  25933.01  116144.63  32617.81    12.44 16.25           66.01
```

Размеры кластеров более сбалансированы, обе переменные, `annual_inc` и `revol_bal`, не доминируют над кластерами, раскрывая более интересную структуру в данных. Отметим, что в приведенном выше фрагменте кода центры перешкалированы в исходные единицы измерения. Если оставить их не шкалированными, то результирующие значения будут в *z*-оценках, и поэтому менее интерпретируемыми.

Шкалирование также имеет серьезное значение для PCA. Использование *z*-оценок эквивалентно использованию корреляционной матрицы *(см. разд. "Корреляция" главы 1)* вместо ковариационной матрицы в вычислениях главных компонент. Программные системы для вычислений PCA обычно имеют опцию использования корреляционной матрицы (в R функция `princomp` имеет аргумент `cor`).

Доминантные переменные

Даже в случаях, где переменные находятся на одинаковой шкале измерения и точно отражают относительную важность (например, динамику курсов акций), иногда может быть полезным переменные перешкалировать.

Предположим, что мы добавляем Alphabet (GOOGL) и Amazon (AMZN) к анализу из *разд. "Интерпретация главных компонент" ранее в этой главе*.

```
syms <- c('AMZN', 'GOOGL' 'AAPL', 'MSFT', 'CSCO', 'INTC', 'CVX', 'XOM',
   'SLB', 'COP', 'JPM', 'WFC', 'USB', 'AXP', 'WMT', 'TGT', 'HD', 'COST')
top_sp1 <- sp500_px[row.names(sp500_px)>='2005-01-01', syms]
sp_pca1 <- princomp(top_sp1)
screeplot(sp_pca1)
```

График каменистой осыпи отображает дисперсии для верхних главных компонент. В данном случае график каменистой осыпи на рис. 7.12 обнаруживает, что дисперсии первой и второй компонент намного крупнее других. Это часто говорит о том, что одна или две переменные доминируют над нагрузками. Именно так и в данном примере:

```
round(sp_pca1$loadings[,1:2], 3)
      Comp.1 Comp.2
GOOGL  0.781  0.609
AMZN   0.593 -0.792
AAPL   0.078  0.004
MSFT   0.029  0.002
CSCO   0.017 -0.001
INTC   0.020 -0.001
```

```
CVX     0.068 -0.021
XOM     0.053 -0.005
...
```

Первые две главные компоненты почти полностью доминируются GOOGL и AMZN. Это вызвано тем, что динамика курса акций GOOGL и AMZN доминирует над вариабельностью.

Для того чтобы справиться с этой ситуацией, вы можете либо включить их, как есть, перешкалировать переменные *(см. разд. "Шкалирование переменных" ранее в этой главе)*, либо исключить доминантные переменные из анализа и обработать их отдельно. "Правильный" подход отсутствует, и решение зависит от применения.

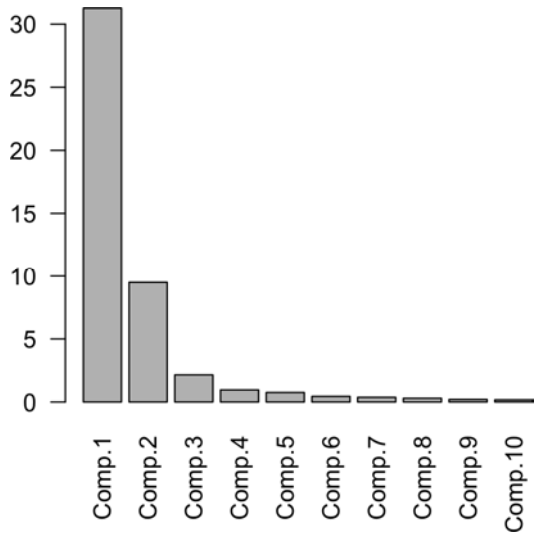


Рис. 7.12. График каменистой осыпи для PCA доходности акций в S&P 500, включая GOOGL и AMZN

Категориальные данные и расстояние Говера

В случае категориальных данных вы должны конвертировать их в числовые данные путем ранжирования (для порядкового фактора) либо путем кодирования как набор двоичных (фиктивных) переменных. Если данные будут состоять из смешанных непрерывных и двоичных переменных, то вам обычно нужно будет перешкалировать переменные так, чтобы диапазоны были схожими *(см. разд. "Шкалирование переменных" ранее в этой главе)*. Один из популярных методов состоит в использовании *расстояния Говера*.

В основе расстояния Говера лежит идея применения разных метрических показателей расстояния к каждой переменной в зависимости от типа данных:

- для числовых переменных и порядковых факторов расстояние вычисляется как абсолютное значение разницы между двумя записями *(манхэттенское расстояние)*;

- для категориальных переменных расстояние равняется 1, если категории между двумя записями отличаются, и 0, если категории одинаковые.

Расстояние Говера вычисляется следующим образом:

1. Вычислить расстояние $d_{i,j}$ для всех пар переменных *i* и *j* для каждой записи.
2. Прошкалировать каждую пару $d_{i,j}$ таким образом, чтобы минимум равнялся 0 и максимум равнялся 1.
3. Соединить вместе попарно прошкалированные расстояния между переменными, используя простое либо взвешенное среднее, для создания матрицы расстояний.

Для того чтобы проиллюстрировать расстояния Говера, возьмем несколько строк из данных о ссудах:

```
> x = defaults[1:5, c('dti', 'payment_inc_ratio', 'home', 'purpose')]
> x
# A table: 5 × 4
    dti payment_inc_ratio   home            purpose
  <dbl>             <dbl> <fctr>             <fctr>
1  1.00           2.39320   RENT                car
2  5.55           4.57170    OWN     small_business
3 18.08           9.71600   RENT              other
4 10.08          12.21520   RENT debt_consolidation
5  7.06           3.90888   RENT              other
```

Функция `daisy` в программном пакете `cluster` может использоваться для вычисления расстояния Говера:

```
> library(cluster)
> daisy(x, metric='gower')
Dissimilarities :
          1         2         3         4
2 0.6220479
3 0.6863877 0.8143398
4 0.6329040 0.7608561 0.4307083
5 0.3772789 0.5389727 0.3091088 0.5056250
```

Все расстояния находятся между 0 и 1. Пара записей с самым большим расстоянием равняется 2 и 3: ни одно не имеет одинаковых значений для `home` (дом) или `purpose` (цель), и у них совсем разные уровни `dti` (соотношение долга к доходу) и `payment_inc_ratio` (соотношение платежей к доходу). Записи 3 и 5 имеют самое маленькое расстояние, потому что они разделяют одни и те же значения для `home` или `purpose`.

Вы можете использовать иерархическую кластеризацию *(см. разд. "Иерархическая кластеризация" ранее в этой главе)* для результирующей матрицы расстояний, применив `hclust` к данным на выходе из функции `daisy`:

```
df <- defaults[sample(nrow(defaults), 250),
               c('dti', 'payment_inc_ratio', 'home', 'purpose')]
```

```
d = daisy(df, metric='gower')
hcl <- hclust(d)
dnd <- as.dendrogram(hcl)
plot(dnd, leaflab='none')
```

Результирующая дендограмма показана на рис. 7.13. Индивидуальные записи неразличимы на оси *x*, но мы можем обследовать записи в одном из поддеревьев (слева, используя "отсечение" 0,5) при помощи представленного далее фрагмента кода:

```
> df[labels(dnd_cut$lower[[1]]),]
# A tablle: 9 × 4
    dti payment_inc_ratio  home   purpose
  <dbl>             <dbl>  <fctr>  <fctr>
1 24.57           0.83550    RENT   other
2 34.95           5.02763    RENT   other
3  1.51           2.97784    RENT   other
4  8.73          14.42070    RENT   other
5 12.05           9.96750    RENT   other
6 10.15          11.43180    RENT   other
7 19.61          14.04420    RENT   other
8 20.92           6.90123    RENT   other
9 22.49           9.36000    RENT   other
```

Это поддерево полностью состоит из арендаторов с целью предоставления ссуды, помеченной как "другая" (other). Хотя нельзя сказать, что для всех поддеревьев есть строгое разделение, график иллюстрирует, что категориальные переменные имеют тенденцию группироваться в кластерах.

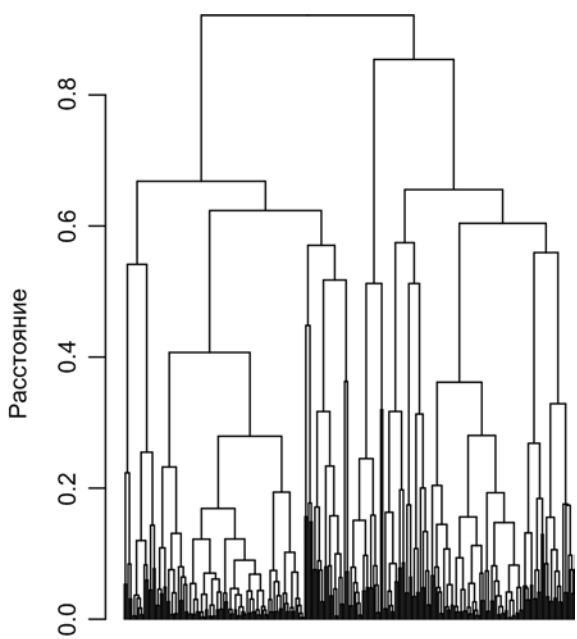


Рис. 7.13. Дендограмма hclust применительно к выборке данных о невозвратных ссудах с типами смешанных переменных

Проблемы кластеризации смешанных данных

Методы *K* средних и PCA лучше всего подходят для непрерывных переменных. Для меньших наборов данных лучше использовать иерархическую кластеризацию с расстоянием Говера. В принципе нет причин отказываться от применения *K* средних к двоичным или категориальным данным. Вы обычно будете использовать представление в виде "кодирования с одним активным состоянием" *(см. разд. "Кодировщик с одним активным состоянием" главы 6)* для конвертирования категориальных данных в числовые значения. На практике, однако, использование методов *K* средних и PCA с двоичными данными может быть сопряжено с трудностями.

Если используются стандартные *z*-оценки, то двоичные переменные будут доминировать над определением кластеров. Это вызвано тем, что переменные в формате 0/1 принимают всего два значения, и *K* средних могут получить малую внутрикластерную сумму квадратов за счет отнесения всех записей с 0 или 1 к единственному кластеру. Например, применим `kmeans` к данным о невозвратных ссудах, включив факторные переменные `home` и `pub_rec_zero`:

```
df <- model.matrix(~ -1 + dti + payment_inc_ratio + home + pub_rec_zero,
                   data=defaults)
df0 <- scale(df)
km0 <- kmeans(df0, centers=4, nstart=10)
centers0 <-scale(km0$centers, center=FALSE,
                 scale=1/attr(df0, 'scaled:scale'))
scale(centers0, center=-attr(df0, 'scaled:center'), scale=F)
    dti payment_inc_ratio homeMORTGAGE homeOWN homeRENT pub_rec_zero
1 17.02              9.10         0.00       0     1.00         1.00
2 17.47              8.43         1.00       0     0.00         1.00
3 17.23              9.28         0.00       1     0.00         0.92
4 16.50              8.09         0.52       0     0.48         0.00
```

Верхние четыре кластера являются по существу эрзацами для разных уровней факторных переменных. Для того чтобы предотвратить такое поведение, вы можете прошкалировать двоичные переменные и получить дисперсию меньше, чем у других переменных. Как вариант, если речь идет об очень крупных наборах данных, вы можете применить кластеризацию к разным подмножествам данных, принимающим конкретные категориальные значения. Например, вы можете применить кластеризацию отдельно к тем ссудам, которые были выданы физическим лицам, выплачивающим ипотеку, владеющим домом напрямую или его арендующим.

Ключевые идеи для шкалирования данных

- Переменные разных измерительных шкал должны быть приведены к единой шкале, чтобы их влияние на алгоритмы не определялось главным образом шкалой их измерения.
- Общепринятым методом шкалирования является нормализация (стандартизация) — вычитание среднего значения и деление на стандартное отклонение.
- Еще один метод — расстояние Говера — шкалирует все переменные, приводя к диапазону 0–1 (он часто используется со смешанными числовыми и категориальными данными).

Резюме

Для сокращения размерности числовых данных основным инструментом является анализ главных компонент или кластеризация по методу *K* средних. Оба метода требуют уделять должное внимание надлежащему шкалированию данных с тем, чтобы гарантировать содержательное сокращение данных.

Что касается кластеризации с очень структурированными данными, в которых кластеры хорошо разделены, все методы скорее всего будут приводить к аналогичному результату. Каждый метод обладает собственным преимуществом. *K* средних масштабируется до очень больших данных и понятен. Иерархическая кластеризация может применяться к смешанным типам данных — числовым и категориальным — и поддается интуитивно понятному отображению (дендограмма). Модельно-ориентированная кластеризация основана на статистической теории и обеспечивает более строгий подход, в противоположность эвристическим методам. В случае очень крупных данных, однако, главным образом используется метод *K* средних.

В отношении шумных данных, таких как данные о ссудах и акциях (и большая часть данных, с которыми столкнется аналитик данных), выбор еще более спартанский. *K* средних, иерархическая кластеризация и, в особенности, модельно-ориентированная кластеризация будут порождать весьма разные решения. Как аналитику данных в такой ситуации действовать дальше? К сожалению, нет никакого простого эмпирического правила, которое могло бы помочь сделать выбор. Используемый метод будет зависеть от размера данных и цели применения.

Библиография

[bokeh] Bokeh: Python library for interactive visualization [Electronic resource]. URL: http://www.bokeh.pydata.org (date of access: 20.11.2017).

[Deng-Wickham-2011] Deng H., Wickham H. Density estimation in R [Electronic resource]. URL: http://vita.had.co.nz/papers/density-estimation.pdf (date of access: 20.11.2017).

[Donoho-2015] Donoho D. 50 Years of Data Science [Electronic resource]. URL: http://courses.csail.mit.edu/18.337/2015/docs/50YearsDataScience.pdf (date of access: 20.11.2017).

[Duong-2001] Duang T. An introduction to kernel density estimation [Electronic resource]. URL: http://www.mvstat.net/tduong/research/seminars/seminar-2001-05.pdf (date of access: 20.11.2017).

[Few-2007] Few S. Save the Pies for Dessert [Electronic resource] // Visual Intelligence Newsletter. Perceptual Edge. — 2007. URL: https://www.perceptualedge.com/articles/visual_business_intelligence/save_the_pies_for_dessert.pdf (date of access: 20.11.2017).

[Galton-1886] Galton F. Regression towards mediocrity in Hereditary stature // The Journal of the Anthropological Institute of Great Britain and Ireland. — № 5. — P. 246–273.

[ggplot2] Wickham H. ggplot2: Elegant Graphics for Data Analysis. — Springer–Verlag New York, 2009. URL: http://www.springer.com/gp/book/9780387981413.

[Hintze-Nelson-1998] Hintze J., Nelson R. Violin Plots: A Box Plot-Density Trace Synergism // The American Statistician. — 1998. — May. — P. 181–184.

[Hyndman-Fan-1996] Hyndman R. J., Fan, Y. Sample quantiles in statistical packages // American Statistician. — 1996. — № 50. — P. 361–365.

[lattice] Sarkar D. Lattice: Multivariate Data Visualization with R. — Springer, 2008, ISBN 978-0-387-75968-5. URL: http://lmdvr.r-forge.r-project.org.

[Legendre] Legendre A.-M. Nouvelle methodes pour la determination des orbites des cometes. — F. Didot, Paris, 1805.

[NIST-Handbook-2012] NIST/SEMATECH e-Handbook of Statistical Methods, 2012 [Electronic resource] . URL http://www.itl.nist.gov/div898/handbook/eda/section3/eda35b.htm (date of access: 27.01.2018).

[R-base-2015] R: A Language and Environment for Statistical Computing R Foundation for Statistical Computing (2015) [Electronic resource]. URL: http://www.R-project.org/.

[seaborne] Wasdom M. Seaborn: statistical data visualization [Electronic resource]. URL: http://stanford.edu/~mwaskom/software/seaborn/# (date of access: 20.11.2017).

[Stigler-Gauss] Stigler S. M. Gauss and the Invention of Least Squares // Ann. Stat. — 1981. — Vol. 9. — № 3. — P. 465–474.

[Trellis-Graphics] Becker R., Cleveland,W, Shyu M., Kaluzny S. A Tour of Trellis Graphics [Electronic resource] . — 1996. URL: http://polisci.msu.edu/jacoby/icpsr/graphics/manuscripts/Trellis_tour.pdf (date of access: 20.11.2017).

[Tukey-1962] Tukey J. W. The Future of Data Analysis [Electronic resource] // Ann. Math. Statist. — 1962. — Vol. 33. — N 1. — P. 1–67. URL: https://projecteuclid.org/download/pdf_1/euclid.aoms/1177704711 (date of access: 20.11.2017).

[**Tukey-1977]** Tukey J. W. Exploratory Data Analysis. — Pearson, 1977.

[Tukey-1987] Tukey J. W. The collected works of John W. Tukey: Philosophy and Principles of Data Analysis 1965–1986: Vol. IV / edited by Jones L. V. — Chapman and Hall/CRC, 1987.

[UCLA] R Library: Contrast Coding Systems for Categorical Variables, UCLA: Statistical Consulting Group [Electronic resource]. URL: http://www.ats.ucla.edu/stat/rflibrary/contrast_coding.htm (date of access: 27.01.2018).

[Wikipedia-2016] "Diving" // Wikipedia: The Free Encyclopedia. Wikimedia Foundation, Inc. 10 Mar 2016. Web. 19 Mar 2016.

[Zhang-Wang-2007] Zhang Q., Wang W. 19th International Conference on Scientific and Statistical Database Management. — IEEE Computer Society, 2007. (19-я международная конференция по вопросам управления научными и статистическими базами данных, IEEE Computer Society.)

Предметный указатель

Л

М

Н

О

П

Р

С

Т

У

Ф

Х

Ч

Ш

Э