Содержание

Вступительное слово 10

Об авторе 12

О рецензентах 13

Предисловие 14

Глава 1. Вероятностное мышление 19

Статистика, модели и подход, принятый в этой книге 19
- Работа с данными 21
- Байесовское моделирование 22

Теория вероятностей 23
- Объяснение смысла вероятностей 23
- Определение вероятности 25

Байесовский вывод с одним параметром 34
- Задача о подбрасывании монеты 35

Взаимодействие с байесовским анализом 46
- Нотация и визуализация модели 46
- Обобщение апостериорного распределения 47

Проверки апостериорного прогнозируемого распределения 49

Резюме 50

Упражнения 52

Глава 2. Вероятностное программирование 54

Вероятностное программирование 55

Основы использования библиотеки PyMC3 56
- Решение задачи о подбрасывании монет с использованием библиотеки PyMC3 57

Обобщение апостериорного распределения 59
- Решения на основе апостериорного распределения 61

Гауссова модель в подробном изложении 67
- Гауссовы статистические выводы 68
- Надежные статистические выводы 73

Сравнение групп 79
- d-мера Коэна 81
- Вероятность превосходства 82
- Набор данных tips 82

Иерархические модели 86

Редуцирование....91
Еще один пример....94
Резюме....96
Упражнения....99

Глава 3. Моделирование с использованием линейной регрессии....101
Простая линейная регрессия....102
Связь с машинным обучением....102
Сущность моделей линейной регрессии....103
Линейные модели и сильная автокорреляция....108
Интерпретация и визуальное представление апостериорного распределения....111
Коэффициент корреляции Пирсона....114
Робастная линейная регрессия....118
Иерархическая линейная регрессия....122
Корреляция, причинно-следственная связь и беспорядочность жизни....128
Полиномиальная регрессия....130
Интерпретация параметров полиномиальной регрессии....131
Является ли полиномиальная регрессия конечной моделью....132
Множественная линейная регрессия....133
Спутывающие переменные и избыточные переменные....137
Мультиколлинеарность или слишком сильная корреляция....140
Маскировочный эффект переменных....144
Добавление взаимодействий....146
Дисперсия переменной....147
Резюме....150
Упражнения....151

Глава 4. Обобщение линейных моделей....154
Обобщенные линейные модели....154
Логистическая регрессия....156
Логистическая модель....157
Набор данных iris....157
Множественная логистическая регрессия....163
Граница решения....163
Реализация модели....164
Интерпретация коэффициентов логистической регрессии....165
Обработка коррелирующих переменных....167
Работа с несбалансированными классами....169
Регрессия с использованием функции softmax....171
Дискриминативные и порождающие модели....173
Регрессия Пуассона....176
Распределение Пуассона....176
Модель Пуассона с дополнением нулевыми значениями....178

Регрессия Пуассона и модель Пуассона с дополнением нулевыми значениями ... 179
Робастная логистическая регрессия ... 181
Модуль GLM ... 183
Резюме ... 184
Упражнения ... 185

Глава 5. Сравнение моделей ... 188
Проверки прогнозируемого апостериорного распределения ... 188
Лезвие Оккама – простота и точность ... 194
Лишние параметры приводят к переподгонке ... 196
Недостаточное количество параметров приводит к недоподгонке ... 197
Баланс между простотой и точностью ... 197
Измерения прогнозируемой точности ... 198
Информационные критерии ... 200
Логарифмическая функция правдоподобия и отклонение ... 201
Информационный критерий Акаике ... 202
Часто применяемый информационный критерий ... 202
Парето-сглаженная выборка по значимости для перекрестной проверки LOOCV ... 203
Другие информационные критерии ... 203
Сравнение моделей с помощью библиотеки PyMC3 ... 204
Усреднение моделей ... 207
Коэффициенты Байеса ... 210
Некоторые дополнительные замечания ... 212
Коэффициенты Байеса и информационные критерии ... 216
Регуляризация априорных распределений ... 220
Более подробно об информационном критерии WAIC ... 222
Энтропия ... 222
Расхождение Кульбака–Лейблера ... 224
Резюме ... 227
Упражнения ... 228

Глава 6. Смешанные модели ... 230
Смешанные модели ... 231
Конечные смешанные модели ... 232
Категориальное распределение ... 234
Распределение Дирихле ... 235
Неидентифицируемость смешанных моделей ... 238
Как правильно выбрать число K ... 241
Смешанные модели и кластеризация ... 245
Смешанные модели с бесконечной размерностью ... 246
Процесс Дирихле ... 246
Непрерывные смешанные модели ... 253

Биномиальное бета-распределение и отрицательное биномиальное распределение ... 254
t-распределение Стьюдента ... 255
Резюме ... 255
Упражнения ... 257

Глава 7. Гауссовы процессы ... 258
Линейные модели и нелинейные данные ... 258
Функции моделирования ... 259
Многомерные гауссовы распределения и функции ... 261
Ковариационные функции и ядра ... 261
Гауссовы процессы ... 264
Регрессия на основе гауссовых процессов ... 265
Регрессия с пространственной автокорреляцией ... 270
Классификация с использованием гауссова процесса ... 277
Процессы Кокса ... 283
Модель катастроф в угледобывающей промышленности ... 284
Набор данных redwood ... 286
Резюме ... 289
Упражнения ... 289

Глава 8. Механизмы статистического вывода ... 291
Механизмы статистического вывода ... 292
Немарковские методы ... 293
Грид-вычисления ... 293
Метод квадратической аппроксимации ... 296
Вариационные методы ... 298
Марковские методы ... 301
Метод Монте-Карло ... 303
Цепи Маркова ... 305
Алгоритм Метрополиса–Гастингса ... 305
Метод Монте-Карло с использованием механики Гамильтона ... 310
Последовательный метод Монте-Карло ... 312
Диагностирование выборок ... 314
Сходимость ... 316
Ошибка метода Монте-Карло ... 319
Автокорреляция ... 320
Эффективный размер выборки ... 321
Расхождения ... 322
Резюме ... 326
Упражнения ... 326

Глава 9. Что дальше? ... 328

Предметный указатель ... 332

Вступительное слово

Вероятностное программирование – это программная среда, которая позволяет создавать гибкие байесовские статистические модели в программном коде. После создания такой модели для обработки в ней данных могут быть использованы мощные алгоритмы логического вывода, работающие независимо. Такое сочетание гибкого определения модели и механизма автоматического логического вывода предоставляет исследователю мощный инструмент для быстрого создания, анализа и постепенного усовершенствования новых статистических моделей. Подобный итеративный подход абсолютно противоположен ранее применявшемуся способу подгонки байесовских моделей к данным: ранее используемые алгоритмы логического вывода обычно работали только с одной конкретной моделью. При этом требовались глубокие и прочные математические знания и навыки для формирования модели и разработки схемы логического вывода, что существенно замедляло итеративный цикл: изменение модели, модификация процесса логического вывода. Таким образом, вероятностное программирование делает статистическое моделирование доступным практически для всех, значительно снижая требования к уровню математической подготовки и сокращая время, требуемое для успешного создания новых моделей и нового, ранее недоступного, глубокого понимания исследуемых данных.

Сама идея вероятностного программирования не нова: BUGS, самый первый инструмент такого типа, появился в 1989 году. Количество моделей, для которых этот инструмент успешно применялся, было крайне ограниченным, а логический вывод выполнялся медленно, поэтому первое поколение языков этого типа не получило широкого распространения на практике. В наши дни существует множество специализированных языков вероятностного программирования, которые широко используются как для академических научных исследований, так и в компаниях Google, Microsoft, Amazon, Facebook и Uber для решения крупномасштабных и сложных задач. Что же изменилось? Главным фактором роста значимости вероятностного программирования и эволюции от состояния занимательной игрушки до мощного механизма, способного решать сложнейшие крупномасштабные задачи, стало появление алгоритма выборки на основе гамильтонова метода Монте-Карло, на несколько порядков более мощного, чем предыдущие алгоритмы выборки. Несмотря на то что этот алгоритм был разработан в 1987 году, только в последнее время системы вероятностного программирования Stan и PyMC3 сделали эту методику выборки широко доступной и удобной в практическом применении.

Предлагаемая книга представляет собой практический вводный курс по использованию этого чрезвычайно мощного и гибкого инструментального средства. Она, несомненно, окажет большое воздействие на ваш образ мышления

и на понимание путей решения сложных аналитических задач. Лишь немногие люди подошли бы для написания такой книги лучше, чем один из основных разработчиков системы PyMC3 Освалдо Мартин (Osvaldo Martin). Освалдо обладает редким талантом подробного постепенного объяснения сложных тем, упрощая их понимание. Его глубокое знание и понимание этих тем, основанное на солидном практическом опыте, позволяет вести читателя по наиболее эффективному пути освоения этой области, которая иначе могла бы показаться недоступной. Наглядные иллюстрации и схемы, примеры программного кода делают эту книгу в высшей степени полезным практическим ресурсом, с помощью которого вы сможете в полной мере овладеть всеми необходимыми теоретическими основами.

Читатели, которые приобрели данную книгу, сделали правильный выбор. Это не простой и не быстрый путь. В наше время, когда широко рекламируется глубокое обучение, как методика решения всех текущих и будущих аналитических задач, более осмотрительный и взвешенный подход к созданию специализированных моделей для конкретной цели, возможно, не выглядит столь привлекательным. Но вы сможете решать задачи, которые трудно решаются любыми другими способами.

Это не говорит о том, что глубокое обучение не является весьма перспективной методикой. В действительности само по себе вероятностное программирование не ограничено классическими статистическими моделями. Изучая современную литературу по машинному обучению, вы наверняка обнаружите, что байесовская статистика определяется как мощный инструментальный комплекс для формирования и исследования следующего поколения глубоких нейронных сетей. Таким образом, эта книга вооружит читателя не только знаниями и навыками решения трудных аналитических задач, но также позволит создать более широкомасштабную основу для одного из самых великих достижений человечества: разработки искусственного интеллекта. Желаю успеха.

Томас Виецки (Thomas Wiecki),
д-р философии (PhD),
руководитель отдела исследований
в компании Quantopian (Бостон, США)

Об авторе

Освалдо Мартин (Osvaldo Martin) – ученый-исследователь агентства The National Scientific and Technical Research Council (CONICET) (Аргентина), занимающийся разработками в области структурной биоинформатики протеина, полисахаридов и молекул РНК. Обладает большим опытом использования цепей Маркова с применением метода Монте-Карло для имитации молекулярных систем, предпочитает пользоваться языком программирования Python для решения задач анализа данных.

Освалдо являлся преподавателем курсов по структурной биоинформатике, науке о данных и байесовскому анализу данных. Также возглавлял организационный комитет PyData в Сан-Луисе (Аргентина) в 2017 году. Освалдо – один из основных разработчиков программных систем PyMC3 и ArviZ.

«Я благодарен Ромине за ее постоянную поддержку. Также хочу поблагодарить Уолтера Лападула (Walter Lapadula), Билла Энгелса (Bill Engels), Эрика Х Ма (Eric J Ma) и Остина Рошфора (Austin Rochford) за бесценные замечания, поправки, комментарии и предложения к черновому варианту книги. Особая благодарность основным разработчикам и всем участникам проектов PyMC3 и ArviZ. Создание этой книги стало возможным благодаря поддержке, позитивному отношению и упорному труду, который все они вложили и вкладывают в эти библиотеки, а также в формирование великолепного сообщества пользователей».

О рецензентах

Эрик Х Ма (Eric J Ma) – ученый в области обработки данных в институте биомедицинских исследований корпорации Novartis. Занимается исследованиями биомедицинских данных с применением в основном байесовских статистических методов с целью улучшения качества медицинского обслуживания пациентов. До работы в корпорации Novartis был действительным членом научного общества Insight Health Data летом 2017 года, защитил докторскую диссертацию весной 2017 года.

Кроме того, Эрик является разработчиком ПО с открытым исходным кодом, ранее возглавлял разработку nxviz, пакета визуализации для NetworkX, и pujanitor, открытого API для очистки данных на языке Python. Также участвовал в разработке ряда инструментальных средств с открытым исходным кодом, включая PyMC3, Matplotlib, bokeh и CuPy.

Основной жизненный принцип (девиз) Эрика можно найти в Евангелии от Луки 12:48.

Остин Рошфор (Austin Rochford) – главный научный сотрудник в Monetate Labs, где он занимается разработкой продуктов, позволяющих розничным продавцам персонализировать свой рынок с учетом миллиардов событий, происходящих ежегодно. По образованию Остин математик, являющийся активным пропагандистом байесовских методов.

Предисловие

Методы байесовской статистики разрабатываются уже более 250 лет. Не только признание и одобрение, но не меньшее пренебрежение и даже неприятие постоянно сопровождали эту ветвь математики. На протяжении нескольких последних десятилетий байесовская статистика стала привлекать все большее внимание специалистов, занимающихся статистикой, и почти всех других ученых, инженеров и даже людей, не принадлежащих к миру науки. Подобный рост интереса стал возможным благодаря теоретическим и вычислительным разработкам, выполненным в основном во второй половине XX века. Разумеется, современная байесовская статистика является главным образом вычислительной статистикой. Необходимость в создании гибких и прозрачных моделей и более глубокая и подробная интерпретация статистических моделей и методов анализа также внесли свой вклад в тенденцию роста.

В предлагаемой книге применяется прагматический подход к изучению байесовской статистики, здесь не уделяется слишком много внимания другим статистическим парадигмам и их взаимосвязям с байесовской статистикой. Главная цель книги – научить практическому выполнению байесовского анализа данных. Философские теоретические дискуссии интересны, но на страницах этой книги вы их не обнаружите, попробуйте поискать в других, более подходящих местах.

В книге излагается методический подход моделирования в статистике, она поможет научиться мыслить в терминах вероятностных моделей и применять теорему Байеса для вывода логических следствий из используемых моделей и данных. Такой подход также является вычислительным, модели кодируются с использованием PyMC3, библиотеки поддержки байесовской статистики, которая скрывает от конечного пользователя большинство математических подробностей и вспомогательных вычислений, и ArviZ, пакета языка Python для исследовательского анализа байесовских моделей.

Байесовские методы с теоретической точки зрения основаны на теории вероятностей, поэтому неудивительно, что многие книги по байесовской статистике содержат множество математических формул, требующих определенного уровня математической подготовки. Изучение математических основ статистики может оказать немалую помощь при создании более качественных моделей и улучшить интуитивное понимание задач, моделей и результатов. Тем не менее такие библиотеки, как PyMC3, позволяют изучать и применять методы байесовской статистики даже при скромном объеме математических знаний, в чем вы сможете убедиться сами, читая эту книгу.

Для кого предназначена эта книга

Если вы студент, специалист по обработке данных, исследователь в области естественных или общественных наук или разработчик, начинающий изучать байесовский анализ данных и вероятностное программирование, то эта книга предназначена для вас. Книга представляет собой введение в байесовский анализ, поэтому не требует предварительных знаний в области статистики, хотя некоторый практический опыт использования языка Python и библиотеки NumPy был бы полезен.

Краткое содержание книги

В главе 1 «Вероятностное мышление» рассматриваются основные концепции байесовской статистики и ее практическое применение для анализа данных. Здесь содержится большинство базовых понятий и положений, которые используются в следующих главах книги.

Глава 2 «Вероятностное программирование» дает обзор концепций из предыдущей главы с точки зрения вычислительной практики. Здесь представлена ознакомительная информация о библиотеке вероятностного программирования PyMC3, а также о библиотеке ArviZ (пакет Python), предназначенной для исследовательского анализа байесовских моделей. На нескольких конкретных примерах рассматриваются иерархические модели.

В главе 3 «Моделирование с использованием линейной регрессии» рассматриваются основные элементы линейной регрессии, весьма широко применяемой модели и структурного компонента более сложных моделей.

В главе 4 «Обобщение линейных моделей» описывается, как расширить область применения линейных моделей для распределений, отличающихся от распределения Гаусса, открывая путь к решению многих задач анализа данных.

В главе 5 «Сравнение моделей» обсуждаются методы сравнения, выбора и усреднения моделей с использованием факторов Байеса, WAIC, LOO, а также основные характерные особенности и детали применения этих методов.

В главе 6 «Смешанные модели» рассматриваются способы повышения гибкости моделей с помощью объединения простых распределений для создания более сложных. Здесь представлена первая непараметрическая модель, рассматриваемая в книге: процесс Дирихле.

В главе 7 «Гауссовы процессы» описывается теоретическая концепция, лежащая в основе гауссовых процессов, а также способы ее применения для создания непараметрических моделей для решения широкого спектра задач.

В главе 8 «Механизмы логического вывода» представлено введение в методы числовой аппроксимации апостериорного распределения (вероятности), а также весьма важная с практической точки зрения тема: как точно определить надежность аппроксимированного апостериорного распределения.

В главе 9 «Что дальше» предлагается список информационных ресурсов, которыми можно воспользоваться для более глубокого изучения байесовского анализа, а также очень короткое итоговое резюме автора.

Максимально эффективное использование книги

Код в этой книге написан на языке Python версии 3.6. Для установки программной среды Python и всех необходимых библиотек рекомендуется воспользоваться Anaconda, специализированным для научных вычислений дистрибутивом. Получить более подробную информацию о дистрибутиве Anaconda и скачать его можно на сайте **https://www.anaconda.com/download/**. При этом в вашей системе будет установлено множество полезных пакетов на языке Python. После этого потребуется установить еще два пакета. Для установки библиотеки PyMC3 используйте утилиту `conda`:

```
conda install -c conda-forge pymc3
```

Чтобы установить пакет ArviZ, можно выполнить следующую команду:

```
pip install arviz
```

Другой способ установки необходимых пакетов, после того как дистрибутив Anaconda установлен в системе, – перейти по адресу **https://github.com/aloctavodia/BAP** и загрузить файл описания среды *bap.yml*. С помощью этого файла можно установить все необходимые пакеты следующей командой:

```
conda env create -f bap.yml
```

Все пакеты языка Python, использованные при написании этой книги, перечислены ниже:

- IPython 7.0;
- Jupyter 1.0 (или Jupyter-lab 0.35);
- NumPy 1.14.2;
- SciPy 1.1;
- pandas 0.23.4;
- Matplotlib 3.0.2;
- Seaborn 0.9.0;
- ArviZ 0.3.1;
- PyMC3 3.6.

При выполнении кода, приведенного в каждой главе, предполагается, что вы установили и импортировали, по крайней мере, некоторые из перечисленных выше пакетов. Вместо копирования и вставки кода из книги рекомендуется скачать файлы исходного кода из репозитория **https://github.com/aloctavodia/BAP** и запускать их с помощью Jupyter Notebook или Jupyter Lab. Автор регулярно обновляет этот репозиторий после выхода новых версий PyMC3 и ArviZ.

Если при выполнении кода из книги возникает техническая проблема или обнаружена опечатка либо какая-либо другая ошибка, то передайте информацию об этом в указанный репозиторий, и автор попытается устранить проблему как можно быстрее.

Большинство иллюстраций в книге сгенерировано при выполнении программного кода. Основная схема такова: сначала приводится блок кода, за которым сразу же следует соответствующая иллюстрация (сгенерированная при выполнении приведенного выше кода). Автор надеется, что такая схема окажется привычной для тех, кто использует Jupyter Notebook/Lab, и не вызовет никаких затруднений у других читателей.

Скачивание исходного кода примеров

Скачать файлы с дополнительной информацией для книг издательства «ДМК Пресс» можно на сайте **www.dmkpress.com** или **www.дмк.рф** на странице с описанием соответствующей книги.

Загрузка цветных иллюстраций

Мы также предоставляем файл в формате PDF, содержащий цветные изображения снимков экрана и схем из данной книги. Этот файл доступен по адресу **https://www.packtpub.com/sites/default/files/downloads/9781789341652_ColorImages.pdf**.

Типографские соглашения, принятые в книге

В этой книге используется несколько стилей выделения некоторых элементов текста.

`Фрагмент кода в тексте` – ключевые слова, операторы, имена переменных и функций непосредственно в тексте. Пример: «Большая часть приведенного выше кода предназначена для построения и вывода графической схемы, вероятностные вычисления выполняются в строке `y = stats.norm(mu, sd).pdf(x)`».

Блок кода отображается в следующем формате:

```
μ = 0.
σ = 1.
X = stats.norm(μ, σ)
x = X.rvs(3)
```

Курсив – имена файлов, каталогов и прочих объектов.

Полужирный шрифт – важные (ключевые) слова, элементы пользовательского интерфейса или слова, которые выводятся на экран.

Отзывы и пожелания

Мы всегда рады отзывам наших читателей. Расскажите нам, что вы думаете об этой книге – что понравилось или, может быть, не понравилось. Отзывы важны для нас, чтобы выпускать книги, которые будут для вас максимально полезны.

Вы можете написать отзыв на нашем сайте **www.dmkpress.com**, зайдя на страницу книги и оставив комментарий в разделе «Отзывы и рецензии». Также можно послать письмо главному редактору по адресу **dmkpress@gmail.com**; при этом укажите название книги в теме письма.

Если вы являетесь экспертом в какой-либо области и заинтересованы в написании новой книги, заполните форму на нашем сайте по адресу **http://dmkpress.com/authors/publish_book/** или напишите в издательство по адресу **dmkpress@gmail.com**.

Список опечаток

Хотя мы приняли все возможные меры для того, чтобы обеспечить высокое качество наших текстов, ошибки все равно случаются. Если вы найдете ошибку в одной из наших книг – возможно, ошибку в основном тексте или программном коде, – мы будем очень благодарны, если вы сообщите нам о ней. Сделав это, вы избавите других читателей от недопонимания и поможете нам улучшить последующие издания этой книги.

Если вы найдете какие-либо ошибки в коде, пожалуйста, сообщите о них главному редактору по адресу **dmkpress@gmail.com**, и мы исправим это в следующих тиражах.

Нарушение авторских прав

Пиратство в интернете по-прежнему остается насущной проблемой. Издательства «ДМК Пресс» и Springer очень серьезно относятся к вопросам защиты авторских прав и лицензирования. Если вы столкнетесь в интернете с незаконной публикацией какой-либо из наших книг, пожалуйста, пришлите нам ссылку на интернет-ресурс, чтобы мы могли применить санкции.

Ссылку на подозрительные материалы можно прислать по адресу электронной почты **dmkpress@gmail.com**.

Мы высоко ценим любую помощь по защите наших авторов, благодаря которой мы можем предоставлять вам качественные материалы.

Глава 1

Вероятностное мышление

> «Теория вероятностей – это не что иное, как здравый смысл, сведенный к вычислениям».
>
> *– Пьер Симон Лаплас*

В этой главе мы начнем изучать основные концепции байесовской статистики и познакомимся с некоторыми инструментами из байесовского арсенала. Здесь приводятся небольшие фрагменты кода на языке Python, но глава в основном теоретическая, поэтому почти все концепции, описываемые в ней, будут многократно использоваться на протяжении всей книги. Из-за обилия теоретического материала глава может показаться немного усложненной для программистов-кодеров, но я полагаю, что такой подход способствует упрощению эффективного применения байесовской статистики при решении практических задач.

В этой главе рассматриваются следующие темы:

- статистическое моделирование;
- вероятность и неопределенность;
- теорема Байеса и статистический вывод;
- статистический вывод с одним параметром и классическая задача о подбрасывании монеты;
- выбор априорного распределения вероятностей и почему людям часто не нравится эта процедура, хотя она обязательна;
- взаимодействие с байесовским анализом (представление результатов).

Статистика, модели и подход, принятый в этой книге

Статистика занимается сбором, организацией (упорядочением), анализом и интерпретацией данных, следовательно, знание статистики чрезвычайно важно для анализа данных. При анализе данных используются два основных статистических метода:

- разведочный анализ данных (РАД)[1] (Exploratory Data Analysis – EDA) – используются числовые обобщающие характеристики, такие как среднее

[1] Термин взят из русской версии Википедии, хотя более подходящим кажется вариант «исследовательский анализ данных». – *Прим. перев.*

значение, мода, стандартное отклонение и вероятные отклонения (этот раздел разведочного анализа данных также называют описательной статистикой). Кроме того, при разведочном анализе данные исследуются визуально с применением широко известных инструментальных средств, таких как гистограммы и диаграммы рассеяния;

- статистический вывод (inferential statistics) – вывод утверждений на основе текущих данных. Это может быть необходимо для понимания некоторых конкретных явлений, или для прогнозирования в будущем точек данных (которые ранее не наблюдались), либо для выбора одного из нескольких альтернативных объяснений результатов наблюдений. Статистический вывод – это набор методов и инструментов, которые помогают ответить на типы вопросов, перечисленные выше.

В этой книге главное внимание сосредоточено на выполнении байесовского статистического вывода и последующем применении метода разведочного анализа данных для обобщения, интерпретации, проверки и предъявления результатов байесовского статистического вывода.

В большинстве вводных курсов по статистике, по крайней мере для людей, незнакомых со статистикой, материал излагается как набор готовых рецептов, более или менее похожих на следующий: войти в статистическую кладовую, выбрать одну из банок и открыть ее, добавить данные по вкусу, перемешивать до получения твердого P-значения (P-критерия), предпочтительно меньшего 0.05. Главная цель курсов с таким подходом – научить выбирать правильную банку в статистической кладовой. Мне никогда не нравился подобный подход, главным образом потому, что наиболее частым результатом такого обучения становится группа сбитых с толку людей, неспособных хорошо понимать, даже на концептуальном уровне, совокупность различных изучаемых методов. Мы будем придерживаться другой методики: также будем изучать некоторые конкретные рецепты, но при этом предпочитая домашнее приготовление, а не консервы из банок. То есть мы будем учиться смешивать свежие ингредиенты, наиболее подходящие для разнообразных блюд, и, что более важно, такой подход позволит применять изученные концепции в последующих примерах, содержащихся в книге.

Применение подобного подхода возможно по двум причинам:

- онтологическая причина – статистика является формой моделирования, унифицированной с помощью математического аппарата, основанного на теории вероятностей. Использование вероятностного подхода обеспечивает единую универсальную точку зрения на все, что может показаться весьма несопоставимыми методами, – статистические методы и методы машинного обучения выглядят намного более похожими друг на друга при взгляде с вероятностной точки зрения;
- техническая причина – современное программное обеспечение, например библиотека PyMC3, позволяет специалистам-практикам, таким как

вы и я, определять и создавать модели решений относительно простым способом. Многие из этих моделей оставались неразрешимыми (следовательно, неприменимыми на практике) буквально несколько лет назад или требовали слишком высокого уровня математической и технической подготовки.

Работа с данными

Данные – это важнейший ингредиент в статистике и науке о данных (датологии). Данные поступают из различных источников, таких как эксперименты, компьютерные имитации, опросы и полевые наблюдения. Если мы являемся ответственными за генерацию или сбор данных, то всегда в первую очередь необходимо тщательно продумать и сформулировать вопросы, на которые нужно получить ответы, и определить используемые для этого методы, и только после этого приступать к обработке данных. В действительности существует целая область статистики, занимающаяся сбором данных, – планирование эксперимента (experimental design). В наше время, когда поток данных достиг невероятных размеров, мы иногда можем забыть о том, что сбор данных не всегда является простым и дешевым делом. Например, всем известно, что Большой адронный коллайдер генерирует сотни терабайтов данных в день, но не все помнят о том, что для его создания потребовались годы ручного и умственного труда.

В качестве обобщенного правила можно интерпретировать процесс генерации данных как случайный (стохастический), поскольку в этом процессе существует онтологическая, техническая и/или эпистемологическая неопределенность, то есть система по своей внутренней сущности является случайной, также существуют технические проблемы, добавляющие шум или ограничивающие наши возможности измерения с произвольной точностью, а кроме того, некоторые концептуальные теоретические ограничения, скрывающие от нас подробности. Из-за всех вышеперечисленных причин всегда необходимо интерпретировать данные в контексте используемых моделей, включая ментальные и формальные. Данные без моделей ни о чем не говорят.

В книге предполагается, что все необходимые данные уже собраны. Кроме того, данные считаются предварительно подготовленными и очищенными, что чрезвычайно редко встречается в реальной практике. Эти исходные предположения сделаны для того, чтобы полностью сосредоточиться на основной теме книги. Очень существенное замечание, особенно для начинающих изучать анализ данных: даже несмотря на то, что подготовка и очистка данных не рассматривается в нашей книге, это весьма важные практические навыки, которыми вы должны овладеть и развивать их для успешной работы с данными.

Очень полезной способностью при анализе данных является умение написать код на каком-либо языке программирования, например на Python. Обработка данных является неизбежной необходимостью с учетом того, что мы живем в беспорядочном мире с еще более беспорядочными данными, поэтому

умение писать программный код помогает решать эти задачи. Даже если вы настолько удачливы, что находящиеся в вашем распоряжении данные предварительно обработаны и очищены, умение писать код все равно останется полезным навыком, потому что современная байесовская статистика реализована в основном с помощью языков программирования, таких как Python или R.

Если вы хотите более подробно узнать, как использовать Python для очистки и обработки данных, рекомендуется изучить превосходную книгу «Python Data Science Handbook» Джейка Ван-дер-Пласа (Jake VanderPlas).

Байесовское моделирование

Модели – это упрощенные описания конкретной системы или процесса, которые, по тем или иным причинам, нас интересуют. Эти описания преднамеренно формулируются так, чтобы отобразить самые важные аспекты системы, но не уделять внимание каждой малозначимой подробности. Это одна из причин, по которой более сложная модель не всегда является лучшим вариантом.

Существует множество различных типов моделей, но в этой книге мы ограничимся байесовскими моделями. В общем случае процесс байесовского моделирования включает три основных этапа.

1. Выбираются некоторые данные и делаются предварительные предположения о том, как эти данные могли быть сгенерированы (извлечены), затем формируется модель с помощью объединения структурных компонентов, известных под названием распределения вероятностей. В основном эти модели представляют собой достаточно грубые приближения (аппроксимации), но в большинстве случаев это именно то, что нам нужно.
2. Используется теорема Байеса для добавления данных в сформированные модели, и выполняются логические выводы по совокупности данных и наших предварительных предположений. Обычно это называют формированием или уточнением условий работы модели по имеющимся данным.
3. Модель критически оценивается посредством проверки степени осмысленности результатов по различным критериям, включая данные, уровень наших экспертных знаний в области исследований, а иногда путем сравнения нескольких моделей.

Вообще говоря, три перечисленных выше этапа в большинстве случаев выполняются итеративно и без соблюдения строгого порядка. Мы будем возвращаться для повторения любого из этих этапов в произвольные моменты времени: возможно, была совершена ошибка при написании кода, или был найден способ изменить и усовершенствовать модель, или возникла необходимость добавления новых данных или сбора данных другого типа.

Байесовские модели также называют вероятностными моделями, потому что они создаются с использованием вероятностей. Почему сделан именно такой выбор? Потому что вероятности являются самым правильным мате-

матическим инструментом для моделирования неопределенности. Поэтому необходимо поближе познакомиться с этой «новой землей», покрытой сетью разветвляющихся дорог.

Теория вероятностей

Название этого раздела может показаться излишне претенциозным, ведь мы не собираемся изучить теорию вероятностей всего на нескольких страницах, да это и не было моим намерением. Я хотел лишь представить несколько общих и наиболее важных концепций, необходимых для лучшего понимания байесовских методов, достаточных для освоения материала данной книги. При необходимости мы будем более подробно рассматривать или вводить новые концепции, относящиеся к теории вероятности. Для более глубокого изучения теории вероятности настоятельно рекомендую книгу «Introduction to Probability» Джозефа К Блитцштайна (Joseph K Blitzstein) и Джессики Хван (Jessica Hwang). Другой весьма полезной книгой может оказаться «Mathematical Theory of Bayesian Statistics» Сумио Ватанабе (Sumio Watanabe), поскольку по названию понятно, что эта книга в большей степени ориентирована на байесовские методы, чем первая, но она более сложна с математической точки зрения.

Объяснение смысла вероятностей

Несмотря на то что теория вероятностей является вполне сформировавшейся и прочно обоснованной математической дисциплиной, существуют и другие интерпретации смысла термина «вероятность». С байесовской точки зрения вероятность – это мера, которая определяет в числовом выражении уровень неопределенности высказывания. С учетом этого определения вероятности абсолютно допустимо и даже естественно задать вопрос о вероятности существования жизни на Марсе, о вероятности того, что масса электрона составляет 9.1×10^{-31} кг, или о вероятности того, что 9 июля 1816 года в Буэнос-Айресе был солнечный день. Но при этом следует отметить, например, что жизнь на Марсе либо существует, либо не существует, то есть итоговый результат бинарный, это тип вопроса с ответом да-нет. Но, учитывая то, что мы не уверены в самом факте существования жизни на Марсе, разумным образом действий является попытка определить, насколько вероятна жизнь на Марсе. Поскольку приведенное выше определение вероятности связано с эпистемологическими, то есть познавательными, функциями нашего мышления, его часто называют субъективным определением вероятности. Однако отметим, что любой человек с научным складом ума не будет использовать свои личные верования или «информацию, полученную от ангела», для ответа на вопросы такого рода, а воспользуется всеми доступными геофизическими данными о Марсе, обратится к своим знаниям в области биомеханики, чтобы определить необходимые условия для жизни, и т. д. Таким образом, байесовские вероятности, следовательно, и вся байесовская статистика, так же субъективны (или объективны),

как и любой другой прочно обоснованный научный метод, имеющийся в нашем распоряжении.

Если у нас нет информации о задаче, то вполне разумно утверждать, что любое возможное событие одинаково вероятно (правдоподобно), а с формальной точки зрения это равносильно предположению об одинаковой вероятности наступления любого возможного события. При отсутствии информации неопределенность максимальна. Но если нам известно, что наступление каких-либо событий более правдоподобно, то это можно формально представить предположением о более высокой вероятности наступления этих событий и более низкой вероятности для прочих событий. Отметим, что когда мы говорим о событиях в контексте статистики, то не ограничиваемся «событиями, которые могут произойти», такими как падение астероида на Землю или вечеринка по поводу 60-летия моей тетушки. Событие – это любое из возможных значений (или набора значений), которые может принимать некоторая переменная, например событие (event), соответствующее утверждению, что вам больше 30 лет, или цена Sachertorte, или количество велосипедов, проданных в прошлом году по всему миру. Концепция вероятности также связана с понятиями логики. Пользуясь аристотелевой или классической логикой, мы можем формулировать утверждения, значениями которых могут быть только истина или ложь. С учетом байесовского определения вероятности определенность является всего лишь особым случаем: истинному утверждению соответствует вероятность 1, а ложному – вероятность 0. Присвоить значение вероятности 1 утверждению «жизнь на Марсе существует» можно было бы только после получения убедительных данных о том, что какие-то объекты развиваются и воспроизводятся, а также выполняют другие действия, которые мы считаем присущими живым организмам. Но следует также отметить, что присваивание значения вероятности 0 сложнее, потому что мы всегда можем считать, что некоторые области Марса остаются неисследованными, или что мы совершили ошибки в некотором эксперименте, либо какие-то другие причины могли привести к ложному выводу об отсутствии жизни на Марсе, хотя этот факт не доказан. Здесь уместно применение правила Кромвеля, гласящего, что необходимо избегать применения априорных вероятностей 0 или 1 к логически истинным или ложным утверждениям. Весьма любопытно, что Ричард Кокс (Richard Cox) математически доказал, что при необходимости расширения логики с включением неопределенности мы непременно должны использовать значения вероятности и теорию вероятностей. Теорема Байеса представляет собой логический вывод на основе правил вычисления вероятности, и мы вскоре убедимся в этом сами. Таким образом, другим способом интерпретации байесовской статистики является расширение логики с учетом неопределенности, то есть нечто, явно не влияющее на субъективное обоснование в уничижительном смысле, именно в этом смысле люди часто употребляют термин «субъективный».

Если обобщить все вышесказанное, то использование вероятности для описания неопределенности модели не обязательно связано с дискуссиями о том,

является ли природа детерминированной или случайной на самом базовом, фундаментальном уровне, кроме того, не связано и с субъективными личными верованиями. Это чисто методологический подход к моделированию неопределенности. Мы считаем большинство явлений трудными для глубокого понимания, потому что в основном имеем дело с неполными и/или зашумленными (загрязненными) данными. Кроме того, существует внутреннее ограничение, зависящее от сформированного в процессе эволюции мозга приматов, в частности человеческого мозга, или от любой другой важной причины, которую можно найти. Вследствие этого мы используем методику моделирования, которая явно принимает во внимание неизбежную неопределенность.

С практической точки зрения большинство важных элементов информации в этом разделе подтверждает использование байесовских вероятностей как инструментального средства для числового выражения неопределенности.

После обсуждения байесовской интерпретации вероятности рассмотрим подробнее несколько математических свойств вероятностей.

Определение вероятности

Вероятности – это числа в интервале [0,1], то есть числа между 0 и 1, включая оба граничных значения. Вероятности подчиняются нескольким правилам. Одно из них – правило умножения:

$$p(A,B) = p(A|B)p(b). \tag{1.1}$$

Это читается следующим образом: вероятность A и B равна вероятности A при условии B, умноженной на вероятность B. Выражение p(A,B) представляет совместную вероятность A и B. Выражение p(A|B) используется для обозначения условной вероятности. Это название указывает на то, что вероятность A обусловлена знанием B и зависит от него. Например, вероятность того, что автодорога мокрая, отличается от вероятности влажности автодороги, если мы знаем (или предполагаем), что идет дождь. Условная вероятность может быть больше, меньше или равна безусловной вероятности. Если знание B не дает нам информацию об A, то p(A|B) = p(A). Это выражение будет истинным, только если A и B независимы друг от друга. С другой стороны, если знание B дает нам полезную информацию об A, то условная вероятность может быть больше или меньше безусловной вероятности в зависимости от того, делает ли знание B более или менее возможным A. Рассмотрим простой пример с использованием обычного игрального кубика (кости) с шестью гранями. Какова вероятность выпадения числа 3 при броске кости p(die=3)? Она равна 1/6, так как каждое из шести чисел имеет одинаковый шанс для обычной шестигранной кости. А чему равна вероятность выпадения числа 3, при условии что мы получаем нечетное число, то есть p(die=3|die=odd)? Вероятность равна 1/3, поскольку если мы знаем, что получаем нечетное число, то возможно выпадение только чисел {1, 3, 5}, и каждое из этих чисел имеет равные шансы. Наконец,

какова вероятность p(die=3|die=even)? Она равна 0, так как если известно, что выпадает четное число, то возможны лишь варианты {2, 4, 6}, следовательно, появление числа 3 невозможно.

Из этого простого примера можно видеть, что при назначении условий для исследуемых данных мы действительно изменяем вероятность наступления событий, а вместе с тем и присущую им неопределенность. Условные вероятности можно назвать сердцем статистики независимо от того, какая задача поставлена перед нами: бросание игральных костей или создание самоуправляемых автомобилей.

Распределения вероятностей

Распределение вероятностей – это математический объект, который описывает, насколько возможными (вероятными) являются различные события. Вообще говоря, эти события ограничены каким-либо образом, то есть представляют собой набор возможных событий, например набор возможных чисел {1, 2, 3, 4, 5, 6} для игральной кости (исключая неопределенные случаи). Общепринятым и полезным представлением концепции в статистике является интерпретация данных как генерируемых из некоторого истинного распределения вероятностей с неизвестными параметрами. Далее логический вывод – это процесс поиска значений этих параметров с использованием только частичной выборки (также называемой набором данных) из истинного распределения вероятностей. В обобщенном случае у нас нет доступа к такому истинному распределению вероятностей, поэтому необходимо примириться с действительностью и создать модель, чтобы попытаться как-то аппроксимировать это распределение. Вероятностные модели создаются при помощи правильного объединения распределений вероятностей.

Следует отметить, что в обобщенном случае мы не можем быть уверены в том, является ли наша модель корректной, следовательно, необходимо критически оценивать создаваемые модели, чтобы повысить степень уверенности и убедить других людей в том, что предлагаемые модели соответствуют задаче, которую нужно исследовать или решить.

Если переменная X может быть описана с помощью распределения вероятностей, то ее называют случайной величиной (переменной). Существует общепринятое правило, согласно которому для обозначения объектов случайных величин используются заглавные буквы, например X, а для обозначения экземпляра случайной величины принято использовать строчные буквы, например x. Экземпляр x может быть вектором и содержать множество элементов или отдельных значений $x = (x_1, x_2, \ldots, x_n)$. Рассмотрим пример с применением языка Python. Для этого примера в качестве истинного распределения вероятностей принято нормальное, или гауссово, распределение со средними значениями $\mu = 0$ и $\sigma = 1$ – эти два параметра полностью и недвусмысленно определяют нормальное распределение. Используя библиотеку SciPy, можно определить случайную переменную X, записав выражение `stats.norm(μ, σ)`, после чего мы можем получить экземпляр x этой переменной с помощью метода `rvs` (сокра-

щение от random variates). В приведенном ниже примере запрашиваются три значения:

```
μ = 0.
σ = 1.
X = stats.norm(μ, σ)
x = X.rvs(3)
```

Обратите внимание на то, что при каждом выполнении этого кода (в терминах статистики: при каждом испытании) вы получаете различные случайные результаты. Отметим, что поскольку значения параметров распределений известны, то вероятность каждого значения x также известна. Случайность состоит в тех самых значениях x, которые мы получаем при каждом испытании. Весьма часто возникает неправильная интерпретация случайности в том смысле, что можно получать любые возможные значения из случайной переменной или что все значения равновозможны. Допустимые значения случайной переменной и вероятности их появления строго управляются распределением вероятностей, а случайность возникает только из того факта, что мы не можем предсказать точные значения, которые будут выдаваться при каждом испытании. При каждом выполнении приведенного выше кода мы будем получать три различных числа, но если повторить выполнение этого кода несколько тысяч раз, то получим возможность опытным путем проверить тот факт, что среднее значение выбираемых экземпляров чисел приближается к нулю, а также что в 95 % выборок содержатся значения в интервале [–1.96, +1.96]. Не следует принимать мое высказывание на веру, проверьте его на практике в среде программирования Python. К точно такому же выводу можно прийти, изучая математические свойства нормального распределения.

Ниже приведена общепринятая форма записи, используемая для обозначения того факта, что переменная имеет нормальное распределение с параметрами μ и σ:

$$x \sim N(\mu, \sigma). \tag{1.2}$$

В этом контексте символ $\sim$ (тильда) читается как «имеет распределение» или «описывается распределением».

Достаточно часто в публикациях можно встретить обозначение нормального распределения, выраженное в терминах дисперсии, а не стандартного отклонения. То есть записывается выражение $N(\mu, \sigma^2)$. В этой книге параметризация нормального распределения будет выполняться с использованием стандартного отклонения, во-первых, потому что оно проще для интерпретации, во-вторых, потому что так работает библиотека PyMC3.

В этой книге будут встречаться и некоторые другие распределения вероятностей, в таких случаях будет приводиться краткое описание соответствующего распределения. Мы начинаем с нормального распределения, потому что оно является своеобразным родоначальником всех распределений вероятностей. Переменная X описывается гауссовым распределением, если его значения определяются следующим выражением:

$$p(x|\mu,\sigma) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}. \tag{1.3}$$

Это функция плотности распределения вероятностей для нормального распределения. Нет необходимости запоминать эту формулу, она приведена здесь только для того, чтобы показать, откуда берутся числовые значения. Как уже было сказано ранее, здесь μ и σ являются параметрами распределения, поэтому, задавая их значения, мы полностью определяем конкретное распределение. Это можно понять из формулы 1.3, так как все прочие члены выражения являются константами. Параметр μ может принимать любое действительное значение, то есть $\mu \in \mathbb{R}$, и определяет среднее значение распределения (а также срединное значение или медиану и моду, которые равны). Параметр σ – это стандартное отклонение, которое может быть только положительным числом и определяет размах распределения. Чем больше значение σ, тем больше размах распределения. Поскольку существует бесконечное число возможных сочетаний значений μ и σ, возможно бесконечное количество экземпляров гауссова распределения, но все они принадлежат к одному семейству распределения Гаусса.

Математические формулы лаконичны и недвусмысленны, некоторые люди даже называют их красивыми, но необходимо признать, что при первой встрече они могут показаться непонятными и даже пугающими, особенно для людей, которые не очень любят математику. Неплохим способом преодоления этого препятствия является использование языка программирования Python для исследования и реализации математических формул на практике:

```
mu_params = [-1, 0, 1]
sd_params = [0.5, 1, 1.5]
x = np.linspace(-7, 7, 200)
_, ax = plt.subplots(len(mu_params), len(sd_params), sharex=True,
                     sharey=True,
                     figsize=(9, 7), constrained_layout=True)
for i in range(3):
    for j in range(3):
        mu = mu_params[i]
        sd = sd_params[j]
        y = stats.norm(mu, sd).pdf(x)
        ax[i,j].plot(x, y)
        ax[i,j].plot([], label="μ = {:3.2f}\nσ = {:3.2f}".format(mu,
                     sd), alpha=0)
        ax[i,j].legend(loc=1)
ax[2,1].set_xlabel('x')
ax[1,0].set_ylabel('p(x)', rotation=0, labelpad=20)
ax[1,0].set_yticks([])
```

Большая часть приведенного выше кода предназначена для построения и вывода графических схем, а вероятностные вычисления выполняются в строке

`y = stats.norm(mu, sd).pdf(x)`. В этой строке вычисляется значение функции плотности вероятности для нормального распределения с передачей параметров `mu` и `sd` для набора значений `x`. Приведенный выше код генерирует схемы, показанные на рис. 1.1. На каждом отдельном графике изображена темно-серая кривая, представляющая гауссово распределение с заданными параметрами μ и σ, указанными в описании (легенде) каждого графика:

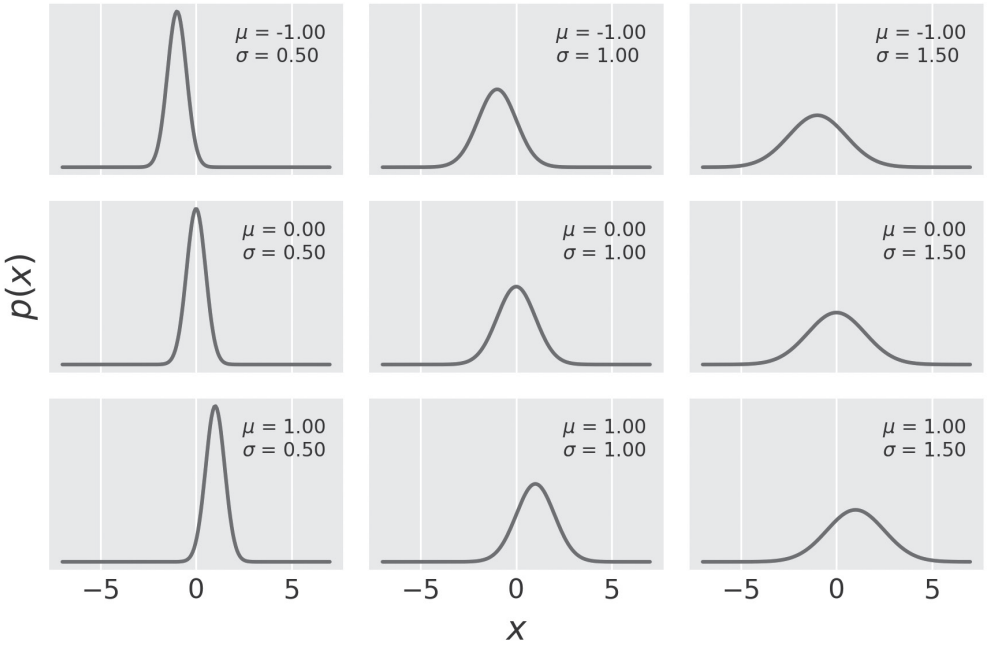

Рис. 1.1

Большинство иллюстраций в этой книге сгенерировано непосредственно из приведенного перед каждой иллюстрацией программного кода, во многих случаях даже без предварительного описания, связывающего код с иллюстрацией. Такой стиль должен быть хорошо знаком тем, кто работает с Jupyter Notebooks или Jupyter Lab.

Существуют два типа случайных переменных: непрерывные и дискретные. Непрерывные переменные могут принимать любое значение из некоторого интервала (для их представления можно воспользоваться числами с плавающей точкой (float) языка Python). Дискретные переменные могут иметь только определенные значения (их можно представить целыми числами (integer) языка Python). Нормальное распределение является непрерывным распределением.

Отметим, что на схемах рис. 1.1 не указаны значения `yticks`, но это сделано преднамеренно. На основе своего опыта могу сказать, что эти значения не

добавляют какой-либо важной информации и даже могут запутать некоторых людей. Более подробное объяснение: числа, указанные по оси y, в действительности не имеют особого смысла – важны лишь их относительные значения. Если взять два значения из x, например x_i и x_j, и обнаружить, что $p(x_i) = 2p(x_j)$ (то есть на графике второе значение выше в два раза), то можно с уверенностью сказать, что вероятность значения x_i в два раза больше вероятности значения x_j. Большинство людей понимает это интуитивно, и, к счастью, это правильная интерпретация. Сложность возникает, когда мы имеем дело с непрерывными распределениями, а значения по оси y являются не вероятностями, а плотностями вероятности. Для получения правильного значения вероятности необходимо выполнить интегрирование по заданному интервалу, то есть требуется вычислить площадь под кривой распределения в заданном интервале. Вероятность не может быть больше единицы, но плотности вероятности могут превышать это значение, поэтому общая площадь под кривой плотности вероятности ограничена значением 1. Понимание различия между вероятностями и плотностями вероятностей чрезвычайно важно с математической точки зрения. При практическом подходе, принятом в этой книге, можно допустить небольшую неточность, поскольку различие между этими понятиями не столь важно, если вы правильно поняли, как интерпретировать схемы на рис. 1.1 с учетом относительности значений.

Независимые одинаково распределенные случайные величины

Во многих моделях предполагается, что все последовательные значения случайных переменных выбраны из одного и того же распределения и независимы друг от друга. В этом случае их называют независимыми одинаково распределенными случайными величинами (переменными) (иногда для краткости используют аббревиатуру нор или iid – independently and identically distributed). Используя математическую нотацию, можно показать, что две переменные являются независимыми, если $p(x,y) = P(x)p(y)$ для любых значений x и y.

В качестве обобщенного примера случайных величин, которые не являются независимыми одинаково распределенными (не-нор), можно привести временны́е ряды (temporal series), в которых временная зависимость, существующая в случайной переменной, представляет собой главную характеристику, которую необходимо принимать во внимание. Например, возьмем следующие данные с сайта **http://cdiac.esd.ornl.gov**. Это данные измерений содержания CO_2 в атмосфере с 1959 по 1997 год. Загрузим эти данные (и прилагаемый к ним исходный код) и построим по ним график:

```
data = np.genfromtxt('../data/mauna_loa_CO2.csv', delimiter=',')
plt.plot(data[:,0], data[:,1])
plt.xlabel('year')
plt.ylabel('$CO_2$ (ppmv)')
plt.savefig('B11197_01_02.png', dpi=300)
```

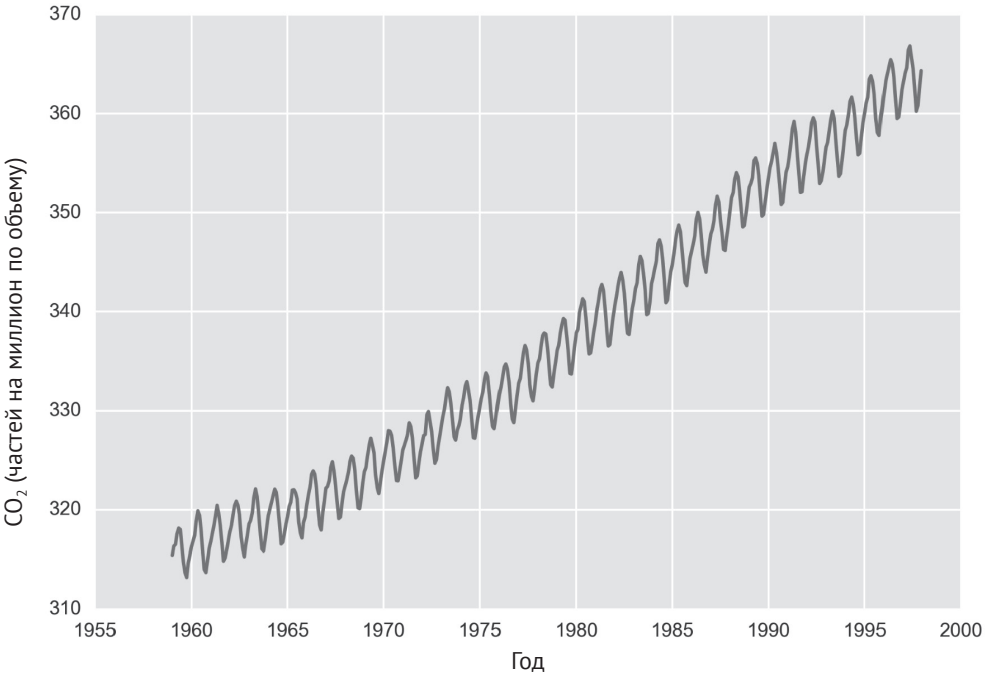

Рис. 1.2

Каждая точка данных соответствует измеренному уровню содержания CO_2 в атмосфере за месяц. Временную зависимость точек данных легко видеть на этом графике. В действительности здесь наблюдаются два тренда: сезонный (связанный с циклами роста и снижения) и глобальный, отображающий постоянный рост концентрации CO_2 в атмосфере.

Теорема Байеса

После ознакомления с некоторыми основными концепциями и терминами теории вероятностей можно перейти к главному предмету наших ожиданий. Без лишних церемоний позвольте представить теорему Байеса во всем ее величии:

$$p(\theta|y) = \frac{p(y \mid \theta)p(\theta)}{p(y)}. \tag{1.4}$$

Возможно, выглядит не слишком впечатляюще. Больше похоже на простую формулу из курса средней школы, тем не менее, перефразируя Ричарда Фейнмана (Richard Feynman), это все, что вам необходимо знать о байесовской статистике.

Происхождение теоремы Байеса и логические ходы, приводящие к ее формулировке, помогут нам понять ее смысл.

В соответствии с правилом произведения получаем:

$$p(\theta, y) = p(\theta|y)p(y). \tag{1.5}$$

Это выражение можно также записать в следующем виде:

$$p(\theta, y) = p(y|\theta)p(\theta). \tag{1.6}$$

С учетом того, что члены в левых частях выражений 1.5 и 1.6 равны, можно объединить эти выражения и записать:

$$p(\theta|y)p(y) = p(y|\theta)p(\theta). \tag{1.7}$$

После простого преобразования выражения 1.7 (перенос члена p(y) в правую часть) получим формулу 1.4, то есть теорему Байеса.

Теперь рассмотрим, что означает формула 1.4 и почему она так важна. Во-первых, необходимо отметить, что p(θ|y) не обязательно является тем же самым, что p(y|θ). Это чрезвычайно важный факт, который легко упустить из вида в повседневных ситуациях даже людям, вполне сведущим в статистике и теории вероятностей. Рассмотрим простой пример для разъяснения того факта, что эти числовые характеристики не обязательно могут быть одинаковыми. Вероятность того, что человек является римским папой, при условии что этот человек аргентинец, не равна вероятности того, что человек является аргентинцем, принимая во внимание, что этот человек является римским папой. Поскольку в настоящее время в Аргентине проживает около 44 млн человек и один из них является действующим римским папой, получаем $p(\text{Pope}|\text{Argentinian}) \cong 1/44\,000\,000$, но при этом $p(\text{Argentinian}|\text{Pope}) = 1$.

Если заменить элемент θ на «предположение» (гипотезу), а элемент y на «данные», то теорема Байеса показывает, как вычислить вероятность предположения θ при наличии данных y. Именно такое объяснение смысла практического применения теоремы Байеса вы найдете во многих местах. Но как превратить предположение в некий объект, который можно поместить в формулу теоремы Байеса? Это делается с использованием распределений вероятностей. Вообще говоря, наше предположение (или гипотеза) представляет собой предположение в чрезвычайно узком смысле, если говорить более точно, то мы ищем наиболее подходящее значение для параметров выбранных нами моделей, то есть для параметров распределений вероятностей. Кстати, не следует пытаться в качестве предположения θ подставить выражения типа «единороги существуют», если только вы не намерены создать реалистичную вероятностную модель существования единорогов.

Теорема Байеса занимает центральное место в байесовской статистике, и как мы увидим в главе 2 «Вероятностное программирование», использование таких инструментов, как PyMC3, освобождает от необходимости всякий раз явно записывать теорему Байеса при создании байесовской модели. Тем не менее важно знать название каждого элемента теоремы, поскольку мы постоянно будем ссылаться на эти элементы, и весьма важно понимать, что означает

каждый элемент, так как это помогает теоретическому обоснованию (концептуализации) моделей. Ниже приведены обозначения и соответствующие названия элементов теоремы Байеса:

- $p(\theta)$ – априорная вероятность;
- $p(y|\theta)$ – правдоподобие;
- $p(\theta|y)$ – апостериорная вероятность;
- $p(y)$ – предельное правдоподобие.

Априорная вероятность должна соответствовать тому, что нам известно о значении параметра θ перед рассмотрением данных y. Если нам ничего неизвестно, как Джону Сноу[1], то можно использовать постоянные фиксированные априорные вероятности, которые не содержат сколько-нибудь значимого объема информации. Вообще говоря, можно выбрать более удачный вариант, чем фиксированные априорные вероятности, как мы увидим далее в этой книге. Использование априорных вероятностей – это основная причина, по которой некоторые люди продолжают называть байесовскую статистику субъективной, даже если априорные вероятности представляют собой всего лишь другой способ предположений, формулируемых при моделировании, следовательно, являются субъективными (или объективными) в той же мере, что и любые другие предположения, такие как правдоподобия.

Правдоподобие определяет, как будут представлены данные в дальнейшем анализе. Это выражение правдоподобности данных с учетом принятых параметров. В некоторых публикациях используются термины «сэмплинг-модель», «статистическая модель» или просто «модель». Мы продолжим использовать термин «правдоподобие» и будем обозначать им комбинацию априорных вероятностей и модели правдоподобия.

Апостериорная вероятность – это результат байесовского анализа, который отображает все, что известно о задаче (проблеме) с учетом имеющихся данных и используемой модели. Апостериорная вероятность – это распределение вероятностей для θ параметров в используемой модели, и это не единственное значение. Такое распределение представляет собой баланс между априорной вероятностью и правдоподобием. Существует широко известная шутка: «Байесовский аналитик – это тот, кто смутно надеется увидеть лошадь, но, заметив быстро промелькнувшего осла, твердо верит, что видел мула». Превосходным способом ликвидации впечатления от этой шутки является объяснение того, что если правдоподобие и априорные вероятности неясны и смутны, то вы получите апостериорную вероятность, отражающую «смутную веру» в то, что видели мула, а не твердую уверенность. В любом случае мне по душе эта шутка и мне нравится, как точно она выражает мысль о том, что апостериорная вероятность является в некоторой степени компромиссом между априорной вероятностью и правдоподобием. С теоретической концептуальной точки зрения

[1] Джон Сноу (Jon Snow) – один из главных персонажей цикла романов Джорджа Р. Р. Мартина «Песнь льда и огня» и телесериала «Игра престолов». – *Прим. перев.*

можно воспринимать апостериорную вероятность как обновленную априорную вероятность в свете (новых) данных. В действительности апостериорная вероятность, полученная в результате одного процесса анализа, может использоваться как априорная вероятность для нового процесса анализа. Это свойство делает байесовский анализ особенно подходящим для анализа данных, которые становятся доступными в определенном последовательном порядке. Примерами могут служить системы раннего оповещения о природных катастрофах, которые обрабатывают в режиме онлайн данные, поступающие с метеорологических станций и спутников. Более подробно об этом можно узнать в публикациях о методах онлайнового машинного обучения.

Последний элемент и термин – предельное правдоподобие, также называемое обоснованностью. Формально предельное правдоподобие – это вероятность исследуемых данных, усредненная по всем возможным значениям, которые могут принимать параметры (в соответствии с предварительно описанной априорной вероятностью). В любом случае практически во всей этой книге мы не уделяем особого внимания предельному правдоподобию и будем считать его простым фактором нормализации. Такой подход принят потому, что при анализе распределения апостериорной вероятности нас будут интересовать только относительные, а не абсолютные значения параметров. Возможно, вы помните, что мы упоминали это обстоятельство при обсуждении методики интерпретации графиков распределений вероятности в предыдущем разделе. Если не принимать во внимание предельное правдоподобие, то можно записать теорему Байеса как прямое пропорциональное отношение:

$$p(\theta, y) \propto p(y|\theta)p(\theta). \tag{1.8}$$

Точное понимание роли каждого элемента (и смысл соответствующего термина) теоремы Байеса занимает некоторое время и требует определенной практики. Также потребуется работа с рядом примеров в следующих главах книги.

Байесовский вывод с одним параметром

В двух предыдущих разделах рассматривались некоторые важные концепции, но две из них представляют собой самые важные компоненты ядра байесовской статистики, поэтому необходимо кратко напомнить их смысл.

Вероятности используются для измерения неопределенности, присущей параметрам, а теорема Байеса – это механизм правильного обновления этих вероятностей при поступлении новых данных в расчете на уменьшение существующей неопределенности.

Теперь, когда мы знаем, что такое байесовская статистика, рассмотрим, как ее применять на практике, с помощью простого примера. Начнем с байесовского вывода с одним неизвестным параметром.