O'REILLY®

Data Science

ИНСАЙДЕРСКАЯ ИНФОРМАЦИЯ ДЛЯ НОВИЧКОВ.
ВКЛЮЧАЯ ЯЗЫК R

Кэти О'Нил, Рэйчел Шатт

ПИТЕР®

Doing Data Science

Cathy O'Neil and Rachel Schutt

Beijing • Cambridge • Farnham • Köln • Sebastopol • Tokyo O'REILLY®

Кэти О'Нил, Рэйчел Шатт

Data Science

ИНСАЙДЕРСКАЯ ИНФОРМАЦИЯ ДЛЯ НОВИЧКОВ.
ВКЛЮЧАЯ ЯЗЫК R

ПИТЕР®
Санкт-Петербург • Москва • Екатеринбург • Воронеж
Нижний Новгород • Ростов-на-Дону • Самара • Минск

2019

ББК 32.972.2-018
УДК 004.3
О-58

О'Нил Кэти, Шатт Рэйчел
О-58 Data Science. Инсайдерская информация для новичков. Включая язык R. — СПб.: Питер, 2019. — 368 с.: ил. — (Серия «Библиотека программиста»).

ISBN 978-5-4461-0622-6

Data Science (исследование данных) — одна из самых востребованных специализаций нашего времени. Изучение данных позволяет преобразить любую традиционную или инновационную бизнес-модель. Эта книга основана на вводном курсе по Data Science из Колумбийского университета, и начинающему специалисту-аналитику она совершенно необходима.

Эта книга увлекательно и доступно рассказывает о:

- байесовском методе;
- статистических алгоритмах;
- финансовом моделировании;
- рекомендательных движках;
- визуализации данных;
- MapReduce.

16+ (В соответствии с Федеральным законом от 29 декабря 2010 г. № 436-ФЗ.)

ББК 32.972.2-018
УДК 004.3

ISBN 978-1449358655 англ. Authorized Russian translation of the English edition of Doing Data Science ISBN9781449358655 © 2013 by Cathy O'Neil, Rachel Schutt
This translation is published and sold by permission of O'Reilly Media, Inc., which owns or controls all rights to publish and sell the same.
ISBN 978-5-4461-0622-6 © Перевод на русский язык ООО Издательство «Питер», 2019
© Издание на русском языке, оформление ООО Издательство «Питер», 2019
© Серия «Библиотека программиста», 2019

Оглавление

Светлой памяти Келли Фини

Предисловие Рэйчел Шатт

Наука о данных (даталогия, Data Science) — зарождающаяся отрасль промышленности, которая еще не полностью определена как академическая дисциплина. Эта книга представляет собой незавершенное исследование главного вопроса: «Что такое наука о данных?» Книга основана на дисциплине под названием «Введение в науку о данных», которую я разработала и впервые преподавала в Колумбийском университете осенью 2012 года.

Если вы поймете меня и мотивы, сподвигшие на создание указанной дисциплины, то это может помочь вам понять книгу и ее первоистоки.

Мотивация

Если коротко, то я создала курс, который хотела видеть в числе прочих, когда училась в колледже. Но дело было в 1990-х годах, и еще не было такого «взрыва» данных, как сегодня, поэтому в тот момент такая дисциплина попросту не могла существовать. Когда я училась, моей основной специализацией была математика, с направлением теоретическим и ориентированным на доказательность. Хотя теперь я рада, что пошла по этому пути, и считаю, что смогла научиться строгому отношению к решению задач, в тот момент я бы не возражала, если бы мне показали, как эти навыки можно использовать для решения задач реального мира.

По окончании колледжа, еще до получения степени доктора наук (PhD), я забрела в статистику, изо всех сил пытаясь найти свою сферу и место — место, в которое я могла бы вложить свою любовь к поиску закономерностей и решению задач и в котором нашла бы ей применение. Я рассказываю об этом, поскольку сейчас

многие студенты хотят знать, чем же им заниматься в жизни. Будучи студенткой, я не планировала работать в науке о данных, ведь тогда она даже еще не была отдельной областью. Мой совет студентам (и любому, кто хочет его послушать): вы не должны понимать все здесь и сейчас. Менять направление — это нормально. Кто знает, что вы можете для себя найти? После того как я получила докторскую степень, я несколько лет проработала в Google примерно в то время, когда в Кремниевой долине происходило становление терминов «наука о данных» и «исследователь данных».

Мир открывает множество возможностей для людей, имеющих математический склад ума и заинтересованных в том, чтобы работать над решением мировых проблем. Я считаю своей целью помочь этим ученикам стать критическими мыслителями, творческими решателями проблем (даже еще не определенных) и авторами любопытных вопросов. В то время как я сама никогда не смогу построить математическую модель, которая поможет в лечении рака, или раскроет тайну аутизма, или предотвратит террористическую атаку, мне нравится думать, что я вношу свой вклад, обучая студентов, способных однажды сделать это. И написанием книги я расширяю свой доступ к еще более широкой аудитории исследователей данных, которые, надеюсь, вдохновятся изданием или изучат описанные в нем инструменты, что позволит им сделать мир лучше, но не хуже.

Создание моделей и работа с данными — вид деятельности, который зависит от личных ценностей. Вы выбираете проблемы, над которыми будете работать, делаете предположения в своих моделях, выбираете метрики и разрабатываете алгоритмы.

Решения всех мировых проблем могут заключаться не в данных и технологиях, а на самом деле в даре хорошего исследователя данных, то есть того, кто способен идентифицировать проблемы, которые *можно* решить с помощью данных, и того, кто хорошо разбирается в инструментах моделирования и написания программного кода. Но я верю, что многопрофильные команды людей, имеющие в своем составе человека, который видит здравый смысл в данных, обладает математическим складом ума, умеет писать программный код и является грамотным решателем проблем (назовем такого человека исследователем данных), могут очень далеко пойти.

Происхождение курса

Я предложила этот курс в марте 2012-го. Тогда на то были три основные причины. Для объяснения первой потребуется больше всего времени.

Первая причина: я хотела дать студентам понимание того, каково это — быть исследователем данных на производстве, а также привить им ряд навыков, которыми обладают подобные специалисты.

Я работала в команде исследователей данных Google+ — коллективе докторов наук. В нее входили я (статистик), социолог, инженер, физик и специалист в области

компьютерных наук. Мы были частью более крупной команды, имеющей в своем составе талантливых инженеров-аналитиков, которые создавали конвейеры данных, инфраструктуру и информационные панели, а также экспериментальную инфраструктуру (A/B-тестирование).

Объединив наши профессиональные навыки, мы могли выполнять удивительные вещи с массивными наборами данных, в том числе прогнозное моделирование, создание прототипов алгоритмов и обнаружение закономерностей в данных, которые оказывали огромное влияние на программы.

Мы предлагали руководству идеи принятия решений, основанные на данных, а также создали новые методологии и способы понимания причинности. Наша способность делать это зависела от первоклассной инженерии и инфраструктуры. Каждый из нас привносил в команду целый комплекс навыков из таких областей, как программирование, разработка программного обеспечения, статистика, математика, машинное обучение, коммуникации, визуализация, разведочный анализ данных (РАД), чувство данных и интуицию, а также экспертный опыт в соцсетях и социальном пространстве.

Поясню: никто из нас по отдельности не преуспел во всех указанных областях, но вместе мы распознали ценность всех навыков и именно поэтому добились успеха. Общими у нас были верность своим принципам и искренняя заинтересованность в решении интересных задач, при этом всегда со здоровой долей скептицизма и воодушевления от научных открытий. Мы ценили свою работу и любили обнаруживать закономерности в данных.

Я живу в Нью-Йорке, поэтому хотела передать свой опыт работы в Google студентам Колумбийского университета: считаю, что им это нужно знать (а мне нравится преподавать). Я хотела научить их тому, что узнала на работе. Я понимала: на нью-йоркской технологической сцене появилось новое сообщество исследователей данных — и надеялась, что мои ученики научатся чему-то и от них.

Одна из особенностей этого курса состояла в том, что у нас были гостевые лекции исследователей данных, работающих в промышленности и науке, каждый из которых обладал разными профессиональными знаниями. Нам представили многообразие точек зрения и взглядов, способствовавших целостному пониманию Data Science.

Вторая причина: наука о данных может представлять собой глубокую и сложную исследовательскую дисциплину, влияющую на все стороны нашей жизни. Колумбийский университет и мэр Блумберг (Bloomberg) объявили об открытии Института наук о данных и инженерии в июле 2012 года. Это позволило развивать теорию даталогии и формализовать ее как полноценную науку.

Третья причина: я продолжала слышать от исследователей данных, занятых на производстве, что преподавать Data Science в классе или университете невозмож-

но, — и приняла этот вызов. Я думала о своем классе как об инкубаторе команд исследователей данных. Мои студенты выдавали впечатляющие результаты и превращались в первоклассных специалистов. На самом деле благодаря их стараниям эта книга стала больше на целую главу.

Как появилась эта книга

Курс не стал бы книгой, не повстречай я Кэти О'Нил (Cathy O'Neil), математика, ставшего исследователем данных, выдающегося и прямолинейного блогера на https://mathbabe.org/, где в разделе «О себе» она говорит, что надеется когда-нибудь найти лучший ответ на вопрос: «Что неакадемический математик может сделать для улучшения мира?» Мы с Кэти встретились примерно в то время, когда я предложила этот курс, а она работала исследователем данных в одном стартапе. Она поощряла и поддерживала мои усилия по созданию этого курса и предлагала написать о нем в блоге. Поскольку я довольно закрытый человек, то изначально не обрадовалась этой идее. Но Кэти убедила меня, указав, что это возможность перевести идеи о науке о данных в публичную сферу, выступить в роли голоса, противоречащего маркетингу и шумихе, которая происходит вокруг самой науки.

Кэти посещала каждое мое занятие и сидела в первом ряду, задавая вопросы, а также была приглашенным лектором (см. главу 6). Помимо документирования в своем блоге, она внесла ценный вклад в учебный материал курса, включая напоминание об этических компонентах моделирования. Кроме того, она поощряла меня вести блог. В результате параллельно с тем, как она документировала занятия, я вела его (http://ww1.columbiadatascience.com/blog/), чтобы напрямую общаться со студентами, а также отражать свой опыт преподавания науки о данных в надежде, что он будет полезен другим профессорам. Все записи блога Кэти для курса и некоторые из моих стали сырьем для этой книги. Мы добавили дополнительный материал, пересмотрели и отредактировали, сделали его гораздо более надежным, чем то, что написано в блогах, и теперь это полноценная книга.

Чего следует ждать от книги

Здесь мы хотим *описать* текущее состояние даталогии, приведя в пример некоторых профессионалов, характеризующих свою работу и рассказывающих, каково это — «делать науку о данных». Мы также хотим *прописать*, что наука может быть академической дисциплиной.

Не ожидайте учебника по машинному обучению. Вместо этого вам предстоит полное погружение в многогранные аспекты науки о данных с разных точек зрения. Это обзор существующего ландшафта науки, попытка картировать эту новую область.

Книга написана с надеждой на то, что окажется в руках того (в ваших?), кто с пользой применит изложенный в ней материал и продолжит решать важные проблемы.

После того как курс закончился, я слышала, что его характеризовали как целостный гуманистический подход к науке о данных — мы сосредоточились не просто на инструментах, математике, моделях, алгоритмах и коде, но и на человеческой стороне. Мне нравится такое определение гуманиста: «Лицо, проявляющее большой интерес и заботу о человеческом благосостоянии, ценностях и достоинстве». Быть гуманистом в контексте науки о данных означает признать роль, которую играет ваша собственная личность в построении моделей и алгоритмов с учетом имеющихся у вас как у человека качеств, которых нет у компьютера (что включает в себя и способность принимать этические решения), — думать о людях, на чьи жизни вы влияете, перенося свою модель на мир.

Структура издания

Эта книга устроена как учебный курс. Мы начнем с некоторых вводных материалов по основному вопросу «Что такое наука о данных?» и введем в качестве организующего принципа процесс научного изучения данных. В главах 2 и 3 мы начнем с обзора статистического моделирования и алгоритмов машинного обучения в качестве базы для остальной части книги. Затем в главах 4–6 и 8 рассмотрим конкретные примеры моделей и алгоритмов в контексте. Из главы 7 узнаем, как извлечь смысл из данных и создать признаки для включения в модель. В главах 9 и 10 представлены две области, которые традиционно не входят (но это временно) в академическое образование: визуализация данных и социальные сети. Затем в главах 11 и 12 перейдем от прогнозирования к причинности. Главы 13 и 14 посвящены подготовке данных и инжинирингу. В главе 15 вы прочитаете высказывания студентов, которые прошли курс, о том, каково это — научиться науке о данных. А затем в главе 16 мы закончим рассказом о том, что (как мы надеемся) случится с этой наукой в будущем.

Как читать книгу

Вообще говоря, эта книга будет иметь больше смысла, если вы прочтете ее от начала и до конца по порядку, поскольку многие из обсуждаемых концепций основаны друг на друге. Возможно также, что вам нужно будет прочитать ее, вооружившись дополнительными материалами, если у вас есть пробелы в понимании теории вероятности и статистики или вы никогда раньше не писали программный код. Мы попытались предложить дополнительные материалы в тексте книги. Надеемся, что когда вы чего-то не поймете (вероятно, из-за пробелов в ваших знаниях или недостаточного объяснения с нашей стороны), то воспримете это как возможность для дальнейшего изучения указанных концепций.

Как в книге используется код

Это не практическое руководство, поэтому код служит только в качестве примера, но во многих случаях может потребоваться, чтобы вы реализовали его самостоятельно и немного с ним поэкспериментировали, дабы по-настоящему понять его суть.

Для кого это издание

Поскольку СМИ своеобразно освещают науку о данных и характеризуют исследователей данных как «рок-звезд», то у вас может возникнуть чувство, что войти в эту сферу не получится. Если вы принадлежите к типу людей, которые любят решать головоломки и находить закономерности, независимо от того, считаете ли себя специалистом по количественному анализу, то это издание для вас.

Эта книга предназначена для людей из самых разных слоев общества. Мы надеемся и ожидаем, что каждый читатель получит от нее разные впечатления в зависимости от своих сильных и слабых сторон.

- Опытные исследователи данных, возможно, придут посмотреть и увидеть себя и то, что они делают, в новом свете.
- Статистики могут получить представление о взаимосвязи между наукой о данных и статистикой или продолжать поддерживать мнение, что «это всего лишь статистика». И в таком случае мы бы хотели, чтобы этот аргумент был четко сформулирован.
- Специалисты по количественному анализу, математики, физики и доктора других наук, которые думают о переходе к даталогии или наращивании своих навыков в области научных знаний, получат видение того, что это от них потребует.
- Студенты и новички в науке о данных сразу оказываются в самой гуще событий, так что если вы относитесь к их числу и не всегда понимаете, о чем речь, — не волнуйтесь, это часть процесса.
- Те, кто ранее не программировал на R или Python, наверняка захотят получить руководство по изучению этих языков. Мы рекомендуем книгу *The Art of R Programming* («Искусство программирования на R») Нормана Матлоффа (Norman Matloff), издательство No Starch Press. Студентам, которые прослушали этот курс, также оказалась весьма полезна книга *R for Everyone: Advanced Analytics and Graphics* («R для всех»: расширенная аналитика и графика) Джареда Лендера (Jared Lander), издательство Addison-Wesley. Кроме того, все упражнения можно выполнить с помощью пакетов в Python.

- Для тех, кто никогда раньше не программировал, предыдущий совет остается в силе. Можете также обратиться к книгам *Learning Python* («Изучаем Python») Марка Лутца (Mark Lutz) и Дэвида Ашера (David Ascher) (http://shop.oreilly.com/product/063 69200 28154.do) и *Python for Data Analysis* («Python для анализа данных») Уэса Маккинни (Wes McKinney), обе издательства O'Reilly (http://shop.oreilly.com/product/063 69200 23784.do).

Что вы уже должны знать

Мы предполагаем, что вы знакомы с линейной алгеброй, имеете представление о теории вероятности и статистике, а также у вас есть некий опыт программирования на любом языке. Тем не менее мы постараемся сделать книгу максимально самодостаточной; вам понадобится прочитать некие дополнительные материалы, только если не будет хватать каких-то знаний. Мы попытаемся указать места в книге, где дополнительное чтение может помочь более глубокому пониманию.

Дополнительная литература

Эта книга представляет собой обзор новой развивающейся области, корни которой уходят во многие другие дисциплины, такие как статистический анализ, алгоритмы, статистическое моделирование, машинное обучение, экспериментальная разработка, оптимизация, теория вероятности, искусственный интеллект, визуализация данных и их разведочный анализ. Проблема в написании книги заключалась в том, что сама по себе каждая из указанных дисциплин соответствует нескольким академическим курсам или книгам. Иногда для устранения пробелов в знаниях требуется обратиться к дополнительным материалам.

- *Математика.*
  - Strang G. *Linear Algebra and Its Applications*. Cengage Learning.
  - Boyd S., Vandenberghe L. *Convex Optimization*. Cambridge University Press.
  - Ross S. *A First Course in Probability*. Pearson.
  - Ross S. *Introduction to Probability Models*. Academic Press.
- *Программирование.*
  - Adler J. *R in a Nutshell*. O'Reilly (http://shop.oreilly.com/product/063 69200 22008.do).
  - Lutz M., Ascher D. *Learning Python by*. O'Reilly (http://shop.oreilly.com/product/063 69200 28154.do).
  - Lander J. *R for Everyone: Advanced Analytics and Graphics*. Addison-Wesley.

- Matloff N. *The Art of R Programming: A Tour of Statistical Software Design*. No Starch Press.
- McKinney W. *Python for Data Analysis*. O'Reilly (http://shop.oreilly.com/product/0636920023784.do).

❑ *Анализ данных и статистический анализ.*

- Casella G, Berger R. L. *Statistical Inference*. Cengage Learning.
- Gelman A. et al. *Bayesian Data Analysis*. Chapman & Hall.
- Gelman A., Hill J. *Data Analysis Using Regression and Multilevel/Hierarchical Models by*. Cambridge University Press.
- Shalizi C. *Advanced Data Analysis from an Elementary Point of View*. Cambridge University Press (http://www.stat.cmu.edu/~cshalizi/ADAfaEPoV/).
- Hastie T., Tibshirani R., Friedman J. *The Elements of Statistical Learning: Data Mining, Inference and Prediction*. Springer.

❑ *Искусственный интеллект и машинное обучение.*

- Bishop C. *Pattern Recognition and Machine Learning*. Springer.
- Barber D. *Bayesian Reasoning and Machine Learning*. Cambridge University Press.
- Segaran T. *Programming Collective Intelligence*. O'Reilly (http://shop.oreilly.com/product/9780596529321.do).
- Russell S., Norvig P. *Artificial Intelligence: A Modern Approach*. Prentice Hall.
- Mohri M., Rostamizadeh A., Talwalkar A. *Foundations of Machine Learning*. MIT Press.
- Alpaydin E. *Introduction to Machine Learning (Adaptive Computation and Machine Learning)*. MIT Press.

❑ *Экспериментальная разработка.*

- Gerber A. S., Green D. P. *Field Experiments*. Norton.
- Box G. E. P. et al. *Statistics for Experimenters: Design, Innovation, and Discovery*. Wiley-Interscience.

❑ *Визуализация.*

- Cleveland W. *The Elements of Graphing Data*. Hobart Press.
- Yau N. *Visualize This: The FlowingData Guide to Design, Visualization, and Statistics*. Wiley.
- Tufte E. *The Visual Display of Quantitative Information*. Graphics Press.

О тех, кто внес вклад в книгу

Курс не был бы успешным без многих приглашенных лекторов, которые пришли поговорить с классом. Я читала лишь несколько лекций, а львиную долю читали гости из стартапов и технологических компаний, а также профессора из Колумбийского университета. Большинство глав книги основаны на этих лекциях. Вообще говоря, приглашенные лекторы не писали книгу, но были инициаторами многих идей, они просмотрели главы и оставили отзывы, и мы им благодарны. Ни курс, ни книга не существовали бы без этих людей. Я пригласила их выступить в классе, поскольку для меня они являются примерами исследователей данных.

Условные обозначения

В этой книге используются следующие условные обозначения.

Курсив

Курсивом выделены новые термины и важные слова.

`Моноширинный шрифт`

Применяется для листингов программ, а также внутри абзацев, чтобы обратиться к элементам программы вроде переменных, функций и типов данных. Им также выделены имена и расширения файлов.

`Моноширинный полужирный шрифт`

Показывает команды или другой текст, который пользователь должен ввести самостоятельно.

`Моноширинный курсивный шрифт`

Показывает код, который должен быть заменен значениями, введенными пользователем, или значениями, определяемыми контекстом.

Шрифт без засечек

Используется для обозначения URL, адресов электронной почты.

Этот рисунок указывает на общее замечание.

Этот рисунок указывает на предупреждение.

Использование примеров кода

Дополнительный материал (базы данных, упражнения и т. д.) доступен для загрузки по ссылке https://github.com/oreillymedia/doing_data_science.

Эта книга призвана помочь вам выполнить свою работу. Как правило, предлагаемые здесь примеры кода вы можете задействовать в своих программах и документации. Вам не нужно обращаться к нам за разрешением, если вы не воспроизводите значительную часть кода. Так, для написания программы, которая использует несколько фрагментов кода из этой книги, не требуется разрешение, в отличие от продажи или распространения CD-ROM с примерами из книг O'Reilly. Ответ на вопрос, в котором цитируется эта книга и пример кода, не требует разрешения, чего не скажешь о включении большого количества примеров кода из данной книги в документацию вашего продукта.

Мы ценим, но не требуем ссылки на источник. Обычно таковая включает название, автора, издателя и ISBN. Например: «О'Нил К., Шатт Р. Data Science. Инсайдерская информация для новичков. Включая язык R. — СПб.: Питер, 2018. 978-5-4461-0622-6». Если вам кажется, что заимствование примеров кода выходит за рамки правомерного использования или данного ранее разрешения, не стесняясь связывайтесь с нами по адресу permissions@oreilly.com.

Благодарности

Рэйчел хотела бы выразить благодарность людям, оказавшим на нее влияние во время работы в Google, а именно Дэвиду Хаффакеру (David Huffaker), Макото Учида (Makoto Uchida), Эндрю Томкинсу (Andrew Tomkins), Абхийит Бос (Abhijit Bose), Дэрилу Прегибону (Daryl Pregibon), Дайан Ламберт (Diane Lambert), Джошу Уиллсу (Josh Wills), Дэвиду Кроушоу (David Crawshaw), Дэвиду Гибсону (David Gibson), Коринне Кортес (Corinna Cortes), Заку Йескелю (Zach Yeskel) и Джорджи Коссинетсу (Gueorgi Kossinetts). Представителям факультета статистики Колумбийского университета: Эндрю Гельману (Andrew Gelman) и Дэвиду Мэдигану (David Madigan), а также лаборанту и ассистенту преподавателя по данному курсу Джареду Ландеру (Jared Lander) и Бену Редди (Ben Reddy).

Рэйчел благодарит семью и друзей за любовь и поддержку, особенно Эран Голдштейн (Eran Goldshtein), Барбару (Barbara) и Шутта (Schutt), Бэки (Becky), Сьюзи (Susie) и Алекса (Alex), Ника (Nick), Лайлу (Lilah), Бэлль (Belle), Шахеда (Shahed) и семью Фини (Feeney).

Кэти хотела бы поблагодарить семью и друзей, в том числе своих замечательных детей и мужа, который отпускал ее раз в неделю для публикации в блоге вечерних занятий.

Мы обе хотели бы выразить благодарность:

- группе экспертов, собиравшихся в квартире Кэти: Крису Виггинсу (Chris Wiggins), Дэвиду Мэдигану, Марку Хансену (Mark Hansen), Джейку Хофману (Jake Hofman), Ори Стайтельману (Ori Stitelman) и Брайану Далессандро (Brian Dalessandro);

- нашим редакторам Кортни Нэш (Courtney Nash) и Майку Лукидесу (Mike Loukides);
- участникам и организаторам конференции по моделированию на уровне пользователя IMA, где случилось несколько предварительных разговоров;
- студентам;
- кафе Coppelia, где Кэти и Рэйчел часто встречались за завтраком.

Мы также хотели бы поблагодарить Джона Джонсона (John Johnson) и Дэвида Парка (David Park) из исследовательской лаборатории Johnson Research Labs за их щедрость и драгоценное время, проведенное за написанием этой книги.

1 Введение: что такое наука о данных

В последние несколько лет вокруг «науки о данных» или «больших данных» было много шумихи. Первая, вполне обоснованная, реакция на все это — смесь скептицизма и смущения. Действительно, наши реакции (Кэти и Рэйчел) были именно такими.

И мы потакали своим заблуждениям. Сначала каждая сама по себе. А позже вместе, собираясь по средам за завтраком. Однако нас не покидало ощущение, будто в этом *действительно* есть нечто, возможно, глубокое и мудрое, представляющее новую парадигму мышления относительно сферы данных. Вероятно, появилось ощущение, что изменение парадигмы сделает нас сильнее. И вместо того, чтобы игнорировать все это, мы решили разобраться более тщательно.

Но прежде, чем двигаться дальше, разберемся, что именно вызывает сомнения и недопонимание. Вероятно, у вас они тоже есть. Затем мы расскажем, каким образом пришли к заключениям, в результате которых Рэйчел составила курс по Data Science в Колумбийском университете, Кэти опубликовала его в своем блоге, а вы сейчас читаете эту книгу.

Большие данные и наука о данных

Сразу определим наши позиции, так как многие из вас уже испытывают определенный скептицизм в отношении Data Science по тем же причинам, которые были и у нас. Мы хотим сообщить: *мы по одну сторону баррикад*. Если вы тоже скептик, то, вероятно, сможете привнести нечто полезное в процесс легитимизации науки о данных.

Итак, что же вызывает удивление при упоминании больших данных или Data Science? Перечислим главное.

1. Существуют сложности в определении терминологии. Что есть большие данные? Что такое наука о данных? Что общего между первым и вторым? Является ли

Data Science наукой о больших данных? Является ли она всего лишь логическим развитием Google или Facebook, развитием их технологической платформы? Почему многие люди относят большие данные к стыку наук (астрономия, финансы, технология и т. д.), а Data Science считают просто технологией? Насколько велико *большое*? Или это просто удобный термин? Приведенные термины настолько неоднозначны, что почти бессмысленны.

2. Со стороны академической и прикладной наук наблюдается откровенное пренебрежение к исследованиям в этой области, несмотря на то что подобные исследования базируются на десятилетиях (в некоторых случаях — столетиях) работы статистиков, специалистов в области компьютерных наук, математиков, инженеров и специалистов других наук. СМИ твердят о том, что алгоритмы машинного обучения изобрели совсем недавно, буквально «на прошлой неделе», а данные никогда не были «большими», пока не появилась Google. Это совершенно не так. Множество современных методов и технологий (а также современных проблем) — частичная эволюция происходящего ранее. Это не значит, что нет ничего нового, достойного восхищения, но нам кажется важным указать на некоторые достижения прошлого, вызывающие уважение.
3. «Бесполезная шумиха», — бросают люди устало. Так описывают специалистов, работающих в области науки о данных, и это не сулит ничего хорошего. Чем больше шума, тем больше отвернувшихся и тем труднее разглядеть — есть что-то хорошее или его нет вовсе.
4. Статистики (ученые в области математической статистики) уверены, что уже работают в области «науки о данных». Это их хлеб с маслом. Может быть, вы, дорогой читатель, не статистик и вообще не имеете никакого отношения к этой области; просто представьте, что статистики воспринимают развитие отдельной науки о данных как покушение на их идентификацию, так же как это восприняли бы вы. Но мы намерены показать, что Data Science — не просто ребрендинг статистики или технологий машинного обучения, а, наоборот, самостоятельная наука. Мы хотим сделать это в противовес СМИ, которые часто описывают науку о данных просто как статистические методы или машинное обучение, применяемые в индустриальных технологиях.
5. Существует утверждение: «Нечто, вынужденное само себя называть наукой, таковой не является». В известной степени это так. *Сам по себе* термин «наука о данных» ничего не значит. А то, что скрывается под ним, не наука, а скорее искусство.

За пологом шумихи

Трудясь над получением степени PhD, Рэйчел приобрела значительный опыт в обработке статистических данных Google. Именно он иллюстрирует, несмотря на все вышеизложенное, те причины, по которым у нас появились подозрения, что в полемике, посвященной «науке о данных», возможно, есть здравое зерно. Вот ее слова:

> «Довольно быстро мне стало ясно, что реальная работа в Google совершенно не похожа на то, чему учили в школе. Я не хочу сказать, будто все ранее полученные знания оказались бесполезны. Наоборот, все изученное в школе явилось твердым, совершенно необходимым фундаментом, позволяющим выполнять мою работу.
>
> Однако множеству навыков, которые потребовались при решении рабочих задач, в школе *совершенно* не обучали. Конечно, мой опыт несколько специфичен, поскольку я владею навыками компьютерной обработки статистических данных, программирования, визуализации информации и обладаю знаниями предметной области Google. Любому другому, являющемуся только специалистом в области компьютерных или общественных наук, гуманитарием либо физиком, придется восполнить имеющиеся пробелы в знаниях. Однако здесь важно то, что, имея различные по глубине и охвату знания, мы сумели объединиться для решения проблем, возникших при работе с данными».

Вот так и возникла вся эта история. Всем известна прописная истина: приступая к реальной работе сразу после учебы, мы ощущаем разрыв между тем, чему нас учили, и тем, что требуется на работе. Другими словами, мы сталкиваемся с различием между академической статистикой и статистикой на производстве.

Несколько замечаний по этому поводу.

- Разумеется, между наукой и производством есть разница. Но действительно ли ей надлежит быть? Почему множество учебных курсов должны быть оторваны от действительности?
- Даже в таком случае отрыв не обусловлен различием между академической статистикой и производственной. Главное достижение адептов науки о данных в том, что их методология создает процессы, позволяющие работать *с бóльшим объемом знаний*, чем процессы, создаваемые фундаментальными методами статистики и компьютерных наук. Именно такие процессы мы определяем как *процессы науки о данных* (подробнее — в главе 2).

Ну а всю шумиху можно подытожить правдивым заключением: появилось нечто новенькое. Однако есть риск дальнейшего отвержения этой хрупкой, зарождающейся идеи. Прежде всего из-за ее позиционирования как некого чудодейственного средства, что порождает совершенно нереалистичные ожидания и, несомненно, приведет к разочарованию.

Рэйчел поставила перед собой задачу понять этот культурный феномен — науку о данных — и то, как даталогию воспринимают другие. Она стала встречаться с работниками Google, начинающими бизнесменами, представителями технологических компаний, университетов, преимущественно занятых в сфере статистики.

На этих встречах начала формироваться новая картина видения. Окончательное понимание вылилось в составление курса для Колумбийского университета под

названием «Введение в науку о данных», опубликованного Кэти в ее блоге. Мы решили, что по окончании семестра наряду с наиболее активными студентами будем понимать, что все это значит на самом деле. И теперь с помощью книги надеемся привести к этому же пониманию множество людей.

Почему именно сейчас

Мы накопили значительный объем данных о разных аспектах нашей жизни и в то же время имеем обилие недорогих компьютерных мощностей. Шопинг, общение, чтение новостей, прослушивание музыки, поиск информации — все это происходит онлайн и знакомо большинству людей.

А вот чего люди могут не знать: одновременно началась «датафикация» нашего поведении офлайн, и это обратная сторона онлайн-сбора информации (подробнее — ниже). Достаточно сложить два и два, чтобы получить четкое представление о нашем поведении и даже больше — о том, что мы вообще за вид.

Это касается не только интернет-данных, но и финансов, медицины, фармацевтики, социологии, правительства, образования, недвижимости; список можно продолжить. Значимость информации растет в большинстве секторов промышленности. В одних случаях объемы собираемой информации достаточно велики, чтобы называть их «большими» (подробнее — в главе 2), в других случаях это не так.

Но не только объемность этих новых данных делает их интересными (или создает проблемы). Данные, часто в режиме реального времени, служат строительными блоками для создания *информационного продукта*. В Интернете это рекомендательная система Amazon, рекомендации друзей в Facebook, советы по поводу фильмов, музыки и т. д. В финансах это представлено кредитными рейтингами, торговыми стратегиями и моделями. В образовании это приводит к пониманию динамики персонального обучения и оценки, проводимых, например, в Академии Хана. Для правительства это политика, основанная на информации.

Мы являемся свидетелями начала огромного цикла, насыщенного обратными связями, в котором наше поведение изменяет конечный продукт, а продукт изменяет наше поведение. Технологии делают это возможным: создается инфраструктура для крупномасштабной обработки данных, увеличиваются память и пропускная способность, а также культура использования технологий в построении нашей жизни. Еще десять лет назад это было невозможно.

Учитывая влияние обратной связи на весь цикл, следует серьезно подумать о том, как он проводится, а также об этических и технических обязательствах людей, ответственных за данный процесс. И первейшая цель нашей книги — начать именно этот разговор.

Датафикация. В мае — июне 2013 года издательство Foreign Affairs опубликовало статью *The Rise of Big Data* («Возникновение больших данных») Кеннета Нейла Кюкера (Kenneth Neil Cukier) и Виктора Майер-Шенбергера (Viktor Mayer-Schoenberger) (https://www.foreignaffairs.com/articles/2013-04-03/rise-big-data). В ней обсуждается концепция датафикации на примере того, как мы оцениваем дружественный контент с помощью лайков. Конечный вывод таков: все, что мы делаем онлайн, заканчивается записью в чье-либо хранилище данных для последующего изучения или продажи.

Авторы определят датафикацию как процесс «представления всех аспектов жизни в виде данных». Примером датафикации видения служат очки дополненной реальности компании Google. Twitter — образец датафикации случайных мыслей. LinkedIn — сеть для датафикации профессионализма.

Датафикация — очень интересная концепция, она вынудила нас признать ее важность в плане совместного использования информации разными людьми. Нас — или скорее наши действия, когда мы ставим лайки или запрашиваем какие-либо сведения, — датафицируют. Ну или как минимум мы должны быть готовы к этому. Едва мы открываем браузер, тут же (пусть непреднамеренно и неосознанно) датафицируемся с помощью cookie-файлов, о которых можем и не знать. Даже когда мы гуляем по магазину или просто по улице, мы непреднамеренно датафицируемся через сенсорные датчики, камеры наблюдения или очки Google.

Уровень преднамеренности колеблется в очень широком спектре: от восторженного участия в экспериментах, проводимых в социальных сетях, чем мы порой гордимся, до скрытого наблюдения и преследования. Но это все — датафикация. Наши намерения различны, но результат один.

В статье обращает на себя внимание строка, в которой авторы говорят о перспективах ценностей:

> «В результате датафикации вещей изменится их значимость, а информация превратится в новую форму ценностей».

Возникает важный вопрос, который мы будем поднимать на протяжении всей книги: кто «мы» есть в таком случае? Какого типа *ценности* имеются в виду? Большая часть приведенных примеров говорит о том, что «мы» — это модели собственников, которые зарабатывают деньги, побуждая людей покупать их вещи. Тогда под ценностью понимается нечто увеличивающее эффективность продаж с помощью автоматизации процесса.

Но если мы хотим мыслить шире, если хотим говорить о «нас» как людях вообще, то придется плыть против течения.

Сегодняшняя картина (и немного истории)

Итак, что такое наука о данных? Нечто новенькое или ребрендинг статистики и аналитики? Есть в ней зерно или это просто шумиха? И если это нечто новое и оно настоящее, то в чем оно заключается?

Это давняя дискуссия, и единственный способ понять, что происходит в данной индустрии, — обратиться к Интернету и посмотреть, в какой стадии находится обсуждение. Это не обязательно даст представление о даталогии, но зато мы увидим, что думают люди по данному поводу и как ее воспринимают. Так, на Quora[1] обсуждение *What is Data Science* («Что такое наука о данных?») длится с 2010 года. А вот пример ответа на этот вопрос, который дает генеральный директор Metamarket Майк Дрисколл (Mike Driscoll) (https://www.quora.com/What-is-data-science):

> «Практически наука о данных — это смесь хакерства, которое заряжается пивом Red Bull, и статистики, вдохновляющейся кофе эспрессо.
>
> Однако Data Science — не совсем хакерство ввиду того, что когда хакеры отлаживают свои однотипные приложения типа Bash или Pig, едва ли их волнует неевклидова геометрия.
>
> Но это и не совсем статистика, ведь когда статистики заканчивают теоретизировать какую-либо модель, едва ли кто-то из них способен прочесть файл на языке R, который составлен по результатам их работы.
>
> Даталогия — гражданское проектирование данных. Ее адепты обладают не только практическими знаниями инструментов и материалов, но и теоретическим пониманием того, что возможно».

Далее Дрисколл ссылается на диаграмму Венна о науке о данных, разработанную Дрю Конвеем (Drew Conway) в 2010 году (http://drewconway.com/zia/?p=2378) и приведенную на рис. 1.1.

Он также упоминает статью Натана Яу (Nathan Yau) *Rise of the Data Scientist* («Возникновение науки о данных»), опубликованную в 2009 году, и перечисляет замечательные навыки фанатов науки о данных в таких областях, как:

- статистика (традиционный анализ, о котором вы привыкли думать);
- очистка данных (разбор, очистка и форматирование);
- визуализация (графики, инструменты и т. д.).

Но позвольте, значит, наука о данных — просто набор специфических приемов? Или это логическое расширение других областей, таких как статистика и машинное обучение?

---

[1] Популярный калифорнийский сайт. — *Примеч. пер.*

Рис. 1.1. Диаграмма Венна о науке о данных, составленная Дрю Конвеем

Один из аргументов см. в статьях Космы Шализи и постах Кэти (https://mathbabe.org/2011/09/25/why-and-how-to-hire-a-data-scientist-for-your-business/ и https://www.nakedcapitalism.com/2012/07/cathy-oneil-data-science-the-problem-isnt-statisticians-its-too-many-poseurs.html), в которых постоянно обсуждается разница между статистиком и исследователем данных. Косма в основном утверждает, что любой статистический отдел, заслуживающий уважения, выполняет все работы, описываемые наукой о данных, и поэтому последняя — всего лишь ребрендинг и нежелательное поглощение статистики.

Для ознакомления с другой точкой зрения см. опубликованную в Amstat News в 2011 году статью президента ASA (American Statistical Association, Американская статистическая ассоциация) Нэнси Геллер (Nancy Geller) *Don't shun the 'S' word* («Не избегайте слова на S») (http://magazine.amstat.org/blog/2011/08/01/prescorneraug11/), в которой она защищает статистику:

> «Нам нужно объяснить людям, что статистики одни из немногих, кто четко разбирается в лавине данных, порождаемых наукой, техникой и медициной; что статистика предоставляет методы анализа данных во всех областях: от истории искусств до зоологии; что в XXI веке увлекательно быть статистиком из-за множества проблем, вызванных взрывным ростом объемов данных во всех этих областях».

Представляя свою точку зрения, Нэнси сама себе ставит подножку, выбрав эти примеры, поскольку они не соответствуют высокотехнологичному миру, в котором

происходит взрывной рост объемов данных. Большая часть развития происходит в области промышленности, а не в академических кругах.

Как следствие, в производственных компаниях есть люди, чья специализация называется «наука о данных», но нет профессоров науки о данных в академических кругах (хотя это может измениться).

Не так давно Дханурджей DJ Патил (DJ Patil) рассказал (http://radar.oreilly.com/2011/09/building-data-science-teams.html), как он и Джефф Хаммербахер (Jeff Hammerbacher), а затем LinkedIn и Facebook соответственно придумали термин «исследователь данных» в 2008 году. Вот когда этот термин стал названием профессии. (В 2012 году в «Википедии» наконец появилась статья о Data Science.)

Для нас совершенно очевидно, что после того, как набор навыков, необходимых для стремительного роста Google (командная работа над задачами, требующими совместных навыков в статистике и компьютерных науках, таких индивидуальных качеств, как любознательность и упорство), стал востребован во всех технологических компаниях Силиконовой долины, появилась необходимость в новой профессии с новым названием. Когда нечто обретает закономерность, оно заслуживает имени. И как только обретает его, все хотят стать родоначальником этого «нечто». И стало еще хуже, после того как в Harvard Business Review объявили, что исследователь данных — «самая притягательная профессия XXI века» (https://hbr.org/2012/10/data-scientist-the-sexiest-job-of-the-21st-century).

РОЛЬ СОЦИОЛОГОВ В НАУКЕ О ДАННЫХ

И LinkedIn, и Facebook — компании, руководящие социальными сетями. Зачастую описание или определение исследователя данных заключается в совмещении навыков: статистика, инженера-программиста и социолога. Это понятно в контексте компаний, производящих *социальный* (общественный) продукт, и сохраняет смысл, когда речь идет о поведении человека вообще или конкретного пользователя. Но если обратиться к диаграмме Венна, разработанной Дрю Конвеем, то можно увидеть следующее: задачи науки о данных лежат на стыке различных дисциплин, что приводит к новому, независимому опыту.

Другими словами, все зависит от конкретного содержания проблемы, подлежащей решению. Если речь о проблемах социологии, таких как рекомендации друзей или знакомых либо сегментирование пользователей, то это работа для социологов! Последние склонны проводить опросы в целях изыскательской аналитики, поэтому социолог, обладающий навыками количественного анализа и программирования, может быть великолепным исследователем данных.

Но это почти «историческое» (в кавычках, поскольку 2008 год был не так давно) искусственное ограничение концепции исследователя данных как человека, работающего исключительно с данными об онлайн-поведении пользователей. Сегодня появилась новая область, называемая вычислительной социологией, которую можно рассматривать как подмножество Data Science.

Но можно заглянуть глубже в прошлое. В 2001 году Уильям Кливленд опубликовал статью о науке о данных *Data Science: An action plan to expand the field of statistics* («Наука о данных: действия по расширению области статистики»).

Таким образом, даталогия существовала до появления исследователей данных?

Это смысловое или семантическое совпадение? Возникает вопрос: наука о данных — то, чем *занимаются* исследователи данных? Кем все-таки определяется поле деятельности? Вокруг темы много шумихи (http://business.time.com/2012/07/31/big-data-knows-what-youre-doing-right-now/). Можно ли полагаться на мнение СМИ либо самозваных исследователей данных? Или существуют некие авторитеты? Пока оставим вопрос открытым, хотя будем к нему возвращаться на протяжении книги.

Наука о данных — профессия. В 2013 году в Колумбии при содействии агентства «Блумберг» (Bloomberg) (https://www.mikebloomberg.com/index.cfm?objectid=D867EFB0-C29C-7CA2-F4B1FEBC8B06249D) было решено создать Институт науки о данных и инжиниринга (http://datascience.columbia.edu/). По проверенным сведениям, в Нью-Йорке насчитывается 465 вакансий для исследователей данных. Это много. Таким образом, даже если Data Science — ненастоящая область науки, то реальная профессия.

И вот что примечательно: в большинстве описаний должностных обязанностей исследователей данных содержатся требования знаний в области информатики, статистики, коммуникации, визуализации данных и обширных знаний предметной области. Нет специалистов, которые были бы экспертами во всех этих областях одновременно, так что есть смысл создавать команды людей, имеющих разные профили и знания, а вместе, как команда, они могут быть специалистами во всех указанных сферах. Мы поговорим об этом подробнее после того, как рассмотрим набор навыков, которые необходимы современному исследователю данных, чтобы быть востребованным.

Профиль науки о данных

В своем классе Рэйчел раздавала карточки и просила всех оценить (в процентах) уровень навыков в таких областях, как:

- компьютерные науки;
- математика;
- статистика;
- машинное обучение;
- знание предметной области;

- коммуникативные и презентационные навыки;
- визуализация данных.

Для примера на рис. 1.2 показан профиль науки о данных, составленный Рэйчел.

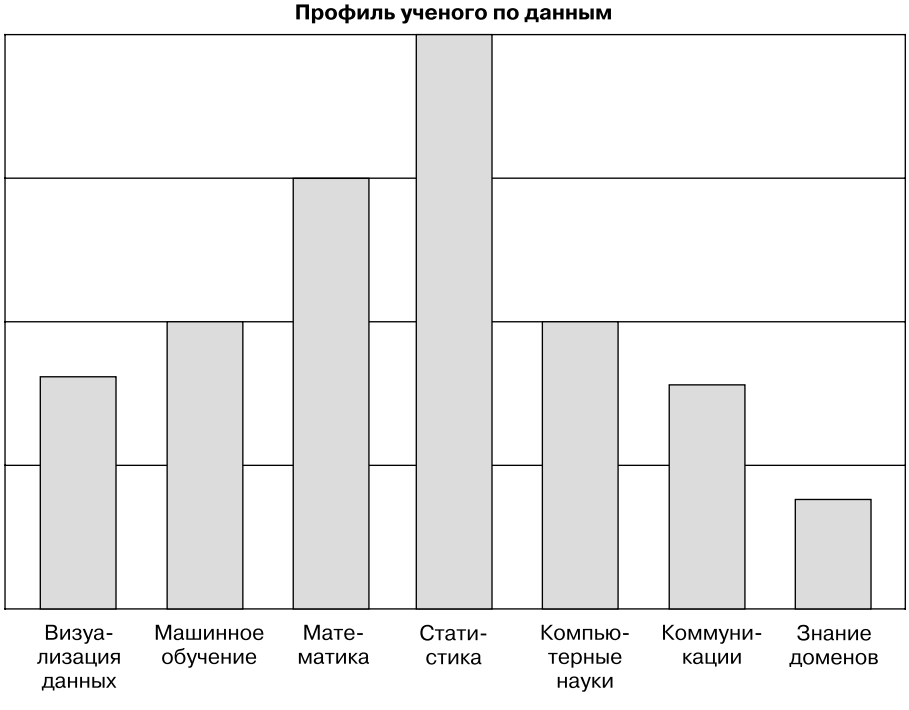


Рис. 1.2. Профиль даталогии, который создала Рэйчел, чтобы проиллюстрировать попытку представить себя исследователем данных. Она хотела, чтобы студенты и вольнослушатели прошли этот риф, пополнив или сократив список навыков, используя различные методы оценки, визуализации, и сами определили собственные недостатки

Мы приклеили карточки на доску, чтобы рассмотреть общую картину того, как люди оценили сами себя. Примечательно, что различия были незначительными, например, в классе присутствовало много социологов.

Каков ваш профиль исследователя данных сейчас и каким бы вы хотели видеть его через несколько месяцев или лет?

Как мы упоминали ранее, команда исследователей данных работает максимально эффективно, когда в нее входят разные люди с различными навыками, поскольку нет людей, совершенных во всем. Это заставило нас задуматься о том, что, возможно, целесообразнее говорить о команде исследователей данных, как показано на рис. 1.3, чем об одном таком специалисте.

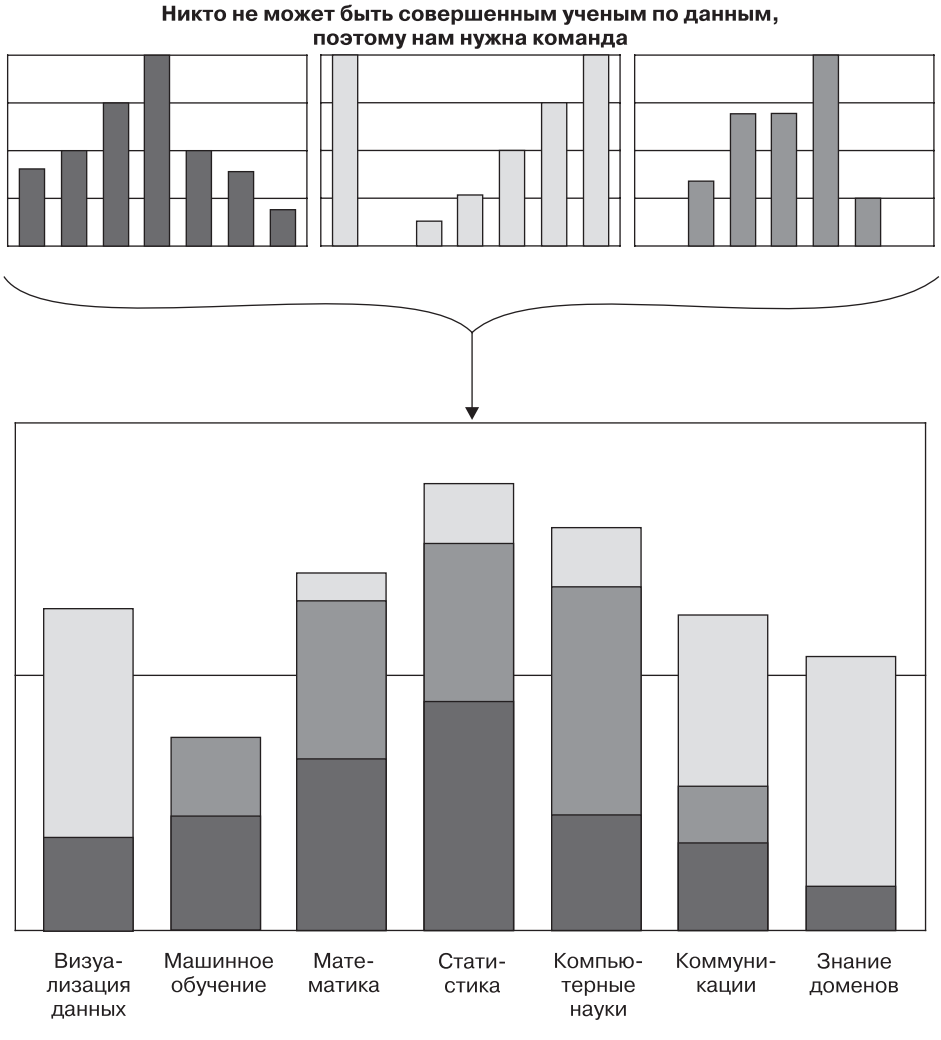


Рис. 1.3. Профиль команды исследователей данных можно составить из профилей отдельных специалистов, поскольку профиль команды должен соответствовать профилю задач, которые ей предстоит решать

Мысленный эксперимент: метаопределение

В каждом классе проводился как минимум один мысленный эксперимент, который студенты обсуждали в составе группы. Большинство таких экспериментов были не ограничены во времени и имели целью вызвать дискуссию по широкому кругу тем даталогии. В первом классе темой эксперимента был вопрос: *можем ли мы, используя науку о данных, дать ей определение?*

Класс разделился на несколько небольших групп, внутри которых и проводилось обсуждение. Приведем отдельные выдержки из обсуждений, содержащие наиболее интересные мысли.

- *Начнем с модели интеллектуального анализа текста.* Мы можем ввести в поисковик Google запрос `data science` и выполнить интеллектуальный анализ этого текста. Предположим, *вольное обращение со словоформами* для нас более приемлемо, чем *строгое следование канонам языка*. Тогда мы позволим массам дать определение науке о данных (под массами подразумеваются все те, кто использует для поиска Google). Но если мы предпочитаем языковой консерватизм, то захотим обратиться к *Большому Оксфордскому словарю*. Но, к сожалению, словарь пока не имеет записи на этот счет, а времени ждать у нас нет. Увы, приходится признать, что существуют вещи, непонятные ни массам, ни авторитетам.
- *Тогда, возможно, стоит попробовать алгоритм кластеризации?* Как насчет того, чтобы понаблюдать за исследователями данных и посмотреть, как они описывают то, чем занимаются? Затем понаблюдать, как представители других областей, например статистики, физики или экономисты, описывают свою деятельность. И вот теперь мы можем попытаться использовать алгоритм кластеризации (будет применяться в главе 3) или какую-либо другую модель, которая в качестве входных данных получает информацию о том, «что человек делает», и на выходе делает прогнозирование, в какой области работает этот человек.

Просто для сравнения посмотрим, что относит к Data Science Харлан Харрис (Harlan Harris): он провел опрос и методом кластеризации выделил науку о данных как подмножество (http://www.datacommunitydc.org/blog/2012/08/data-scientists-survey-results-teaser/) (рис. 1.4).

Итак, кто же такой исследователь данных

Наверное, самое правильное — оставить определение Data Science тем, кто ею непосредственно занимается, то есть пусть исследователи данных дадут определение тому, что они делают. Исходя из этого, опишем, чем занимаются исследователи данных. Мы немного схитрим, говоря о наличии таких специалистов в академических кругах.

В академических кругах

Реальность такова: в настоящее время в академических кругах никто не называет себя исследователем данных. Этот термин используется лишь как дополнение к званию, дабы подчеркнуть принадлежность к «институту даталогии», или для подачи заявки на грант, который предоставляет деньги для исследований в области науки о данных.

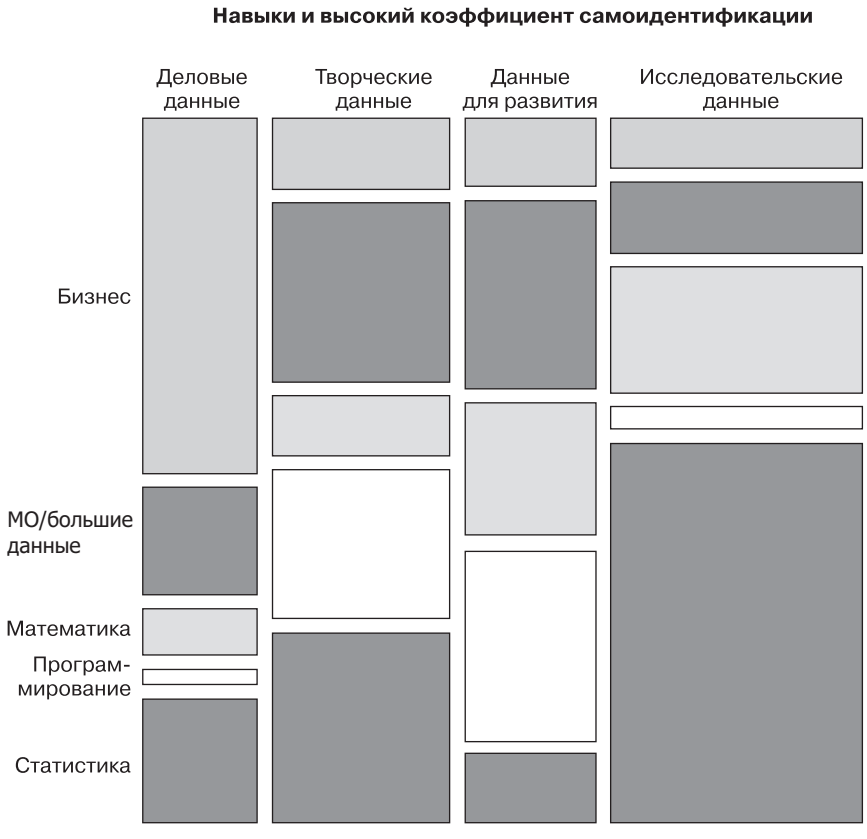


Рис. 1.4. Кластеризация и визуализация подмножества «наука о данных» из книги Analyzing the Analyzers («Анализ анализаторов») (O'Reilly). Харлан Харрис, Сэн Мерфи и Марк Вайсман в середине 2012 года проводили опрос среди нескольких сотен специалистов-практиков

Зададим другой вопрос: кто в этих кругах собирается *стать* исследователем данных? В Колумбийском университете в классе Intro 60 студентов изучали науку о данных. Когда Рэйчел предложила свой курс, она предполагала привлечь преимущественно студентов, специализирующихся в статистике, прикладной математике и компьютерных науках. Фактически в конечном итоге среди студентов оказались: социологи, журналисты, политологи, студенты биомедицинской информатики, студенты из правительственных учреждений Нью-Йорка и некоммерческих организаций, связанные с социальным обеспечением, некто из архитектурной школы, специалисты по экологии, чистые математики, студенты, специализирующиеся в бизнес-маркетинге, и люди, которые уже работали исследователями данных. Оказалось, им всем интересно разобраться с методами решения проблем обработки социально значимых данных.

Для того чтобы термин «наука о данных» в академических кругах стал названием факультета или определял область научных исследований, его необходимо формализовать. Отметим, что уже существует множество проблем, которые могли бы стать темой диссертации на степень PhD.

Вот в чем главная проблема: академический исследователь данных, специализирующийся в какой-либо области, от социологии до биологии, который работает с большими объемами информации, должен бороться с вычислительными проблемами, создаваемыми структурой, размером, беспорядочностью данных, а также их сложностью и характером, одновременно решая реальную задачу.

Все вышеописанное коротко можно сформулировать следующим образом: академические дисциплины, проблемы фундаментальной науки и вычисления имеют много общего. Когда исследователи из смежных областей объединяют усилия, они способны решать самые разноплановые задачи.

В промышленности

Как выглядит исследователь данных в промышленности? Это зависит от уровня в иерархии подчинения и в первую очередь касается интернет-индустрии. Исследователь данных не обязательно должен быть технарем, но именно технари создали этот термин; так что в рамках нашего разговора скажем, о чем речь.

Руководящий исследователь данных должен задавать стратегию данных компании, которая включает в себя множество аспектов. Сюда входит настройка всего, начиная от инженерной инфраструктуры и инфраструктуры для сбора данных, ведение журналов, проблемы конфиденциальности, определение типов данных, то есть решение, какие данные будут ориентированы на пользователя, а какие — служить для принятия решений и того, как они будут встроены в конечный продукт. Руководящий исследователь должен управлять командой инженеров, ученых и аналитиков, обмениваться информацией с главными лицами компании, включая генерального и технического директоров и менеджеров продукта, а также заниматься патентами инновационных решений и определять цели исследований.

В целом, исследователь данных — тот, кто знает, как извлечь смысл и интерпретировать данные, что требует знаний инструментов и методов статистики и машинного обучения. Он проводит много времени в процессе сбора, очистки и обработки данных, поскольку те почти всегда избыточны и содержат много мусора. Этот процесс требует настойчивости, навыков программирования, статистических вычислений и отладки кода, понимания особенностей работы с данными.

Критически важная часть работы такого специалиста — приведение данных к виду, удобному для исследований, то есть визуализация смысла данных. Исследователь данных находит закономерности, создает модели и алгоритмы. Одна часть из них служит для достижения максимальной эффективности

в использовании продукта, его общего состояния, а другая выступает в роли прототипов, которые в конечном итоге будут интегрированы в продукт. Исследователь данных — ключевое звено в процессе принятия решений, основанных на данных. Он общается с членами команды, инженерами и руководством на понятном для них языке, визуализируя имеющуюся информацию, чтобы коллеги, не знакомые с данными, понимали их смысл.

Но это уже очень высокий уровень, и наша книга призвана помочь вам его достичь. Здесь мы завершаем *разговор* о науке о данных; пойдем вперед и *сделаем* нечто большее!

2 Статистический анализ, разведочный анализ данных и процесс их научного исследования

Мы начнем эту главу с обсуждения статистического анализа и статистического мышления. Затем исследуем то, что, как мы считаем, должен делать каждый специалист, располагающий информацией по любому проекту, связанному с данными: разведочный анализ данных (Exploratory Data Analysis, EDA).

Далее перейдем к более подробному рассмотрению того, что определяем как процесс научного исследования данных. И закончим мысленным экспериментом и практическим примером.

Статистическое мышление в век больших данных

> «Большие данные» — размытый термин, который часто и свободно используется в наши дни. Но проще говоря, эта шаблонная фраза означает три вещи: во-первых, набор технологий; во-вторых, потенциальная революция в системах измерения; и в-третьих, точка зрения или философия о том, как станут и, возможно, должны приниматься решения в будущем.
>
> *Стив Лор, The New York Times*

Когда вы развиваете навыки исследователя данных, иногда непонятно, чему нужно отдать приоритет: статистике, линейной алгебре, программированию... Даже при

наличии подобных навыков часть проблемы состоит в том, что вы будете одновременно и параллельно развивать несколько наборов приемов: подготовку и очистку данных, моделирование, кодирование, визуализацию и коммуникацию — и все эти навыки взаимозависимы. По мере продвижения по книге они будут переплетаться. Тем не менее нам нужно от чего-то оттолкнуться, и начнем мы с азов навыка статистического анализа.

Мы ожидаем, что читатели этой книги имеют неодинаковый опыт и образование. Например, одни из вас уже могут быть отличными разработчиками программного обеспечения, способными создавать потоки данных и код с лучшими из этих потоков, но мало знают о статистике, другие могут быть маркетинговыми аналитиками, совершенно не умеющими программировать, а остальные — просто любопытными умными люди, которые хотят знать, что такое наука о данных.

Поэтому, хоть и рассчитываем на то, что читатели уже имеют определенный опыт и знания, мы не можем прийти к вам домой и посмотреть вашу зачетку или диплом с целью убедиться, что вы действительно прошли курс статистики или хотя бы читали книги по данной теме. И даже если вы прошли «Введение в статистику», оно, скорее всего, не дало вам почувствовать вкус, глубину и красоту статистического анализа.

Но даже если вы и почувствовали вкус и, возможно, вы статистик со степенью доктора наук, всегда полезно вернуться к основам и вспомнить, что такое статистический анализ и мышление. Кроме того, в эпоху больших данных классические методы статистики необходимо пересмотреть и переосмыслить в новых контекстах.

Статистический анализ

Мир, в котором мы живем, сложный, он полон случайностей и неопределенности. В то же время это одна большая машина для генерации данных.

Когда мы отправляемся на работу на метро или автомобилях, когда наша кровь перемещается по нашим телам, когда мы занимаемся покупками, отправляем сообщения по электронной почте, бездельничаем на работе, просматривая Интернет и наблюдая за фондовым рынком, принимаем пищу, общаясь с нашими друзьями и семьей, а фабрики производят продукты — все это потенциально создает данные.

Представьте, вы 24 часа глядите в окно и каждую минуту подсчитываете и записываете количество проходящих мимо людей. Или собираете всех, кто живет в радиусе одной мили от вашего дома, и просите их информировать вас, сколько сообщений электронной почты они будут получать каждый день на протяжении следующего года. Представьте, что направляетесь в местную больницу с целью изучить образцы крови на предмет неких закономерностей в ДНК. Это прозвучало жутко, но мы не хотели вас напугать. Дело здесь в том, что процессы в нашей жизни являются фактически процессами генерации данных.

Мы хотели бы иметь способы, позволяющие описывать, понимать и осознавать эти процессы, отчасти потому, что, как ученые, просто хотим лучше понять мир, однако зачастую понимание этих процессов является частью решения стоящих перед нами задач.

Данные представляют собой следы процессов, происходящих в реальном мире, и именно то, какие следы мы собираем, определяет используемый нами метод сбора или выборка данных. Вы исследователь данных, наблюдатель, превращаете мир в данные, и этот процесс совершенно субъективный, а не объективный.

После отделения этого процесса от сбора данных мы ясно видим, что существует два источника случайности и неопределенности. А именно, случайность и неопределенность, лежащие в основе самого процесса, а также неопределенность, связанная с используемыми методами сбора данных.

Получив всю информацию, вы каким-то образом запечатлели мир или, точнее, его определенные следы. Но вы не можете ходить с огромной таблицей Excel или базой данных миллионов транзакций и понять мир и процесс, сгенерировавшие их, просто взглянув на таблицы и щелкнув пальцами.

Таким образом, вам нужна новая идея, а именно — как превратить эти запечатленные следы в нечто более понятное, каким-то образом улавливающее все аспекты гораздо более лаконичным и выразительным способом. В роли этого «нечто» могут выступать математические модели или функции данных, известные как алгоритм статистического оценивания.

Этот общий процесс перехода от мира к данным, а затем из данных обратно в мир является сферой *статистического анализа*.

Говоря точнее, статистический анализ — дисциплина, которая касается разработки процедур, методов и теорем, позволяющих извлекать смысл и информацию из данных, сгенерированных стохастическими (случайными) процессами.

Генеральные совокупности и выборки

Сверим терминологию и концепции, дабы убедиться в том, что говорим об одном и том же.

В классической статистической литературе проводится различие между популяцией и образцом. Слово *population*, используемое в англоязычной литературе для обозначения «генеральной совокупности», сразу вынуждает нас думать о том, что население США составляет 300 млн человек, а население всего мира — 7 млрд человек. Но выбросьте этот образ из головы, поскольку в статистическом анализе слово *population* не применяется для описания только людей: это может быть любой набор объектов или единиц, таких как твиты, фотографии или звезды.

Если бы мы могли измерить или извлечь признаки (характеристики) всех этих объектов, то получили бы полный набор *наблюдений*, а по принятому соглашению для обозначения общего числа наблюдений в генеральной совокупности используется латинская буква *N*.

Предположим, ваша генеральная совокупность — это все электронные письма, отправленные в прошлом году сотрудниками огромной корпорации BigCorp. Тогда единственным наблюдением может быть следующий список: имя отправителя, список получателей, дата отправки, текст письма, количество в письме символов, предложений, глаголов и время, прошедшее до момента получения первого ответа.

Для *выборки* мы берем подмножество единиц размера *n*, чтобы исследовать наблюдения с целью подвести итоги и сделать выводы о генеральной совокупности. Существуют разные способы получения этого подмножества данных, и вы должны знать механизм выборки, поскольку он может *искажать* данные и притом делать это так, что подмножество не будет являться сжатой мини-версией генеральной совокупности. Если такое произойдет, то любые сделанные вами выводы просто будут ошибочными и искаженными.

В примере электронной почты сотрудников BigCorp вы можете составить список всех сотрудников и *случайным образом* выбрать из них 1/10, взять все электронные письма, которые они когда-либо отправляли, — и это будет вашей выборкой. В качестве альтернативы вы могли бы случайным образом отбирать 1/10 всех сообщений, отправляемых каждый день, и это была бы ваша выборка. Оба описанных метода представляются разумными и дают одинаковый размер выборки. Но если бы вы с помощью этих методов подсчитали, сколько сообщений отправил каждый сотрудник, а затем использовали эти данные для оценки основного *распространения* электронных писем, отправленных всеми сотрудниками компании BigCorp, то могли бы получить кардинально различающиеся ответы.

Таким образом, если даже получение неких базовых данных (например, подсчет) может искажаться, когда вы используете разумно звучащий метод выборки, то представьте, что может случиться с более сложными алгоритмами и моделями при неучтенном процессе, который передал вам данные.

Генеральные совокупности и выборки больших данных

Но подождите! В эпоху больших данных, когда мы можем постоянно записывать все действия пользователей, разве мы не наблюдаем *все*? Действительно ли еще актуальны понятия генеральной совокупности и выборки? Прежде всего, если у нас с самого начала были все электронные письма, то зачем нам нужна выборка?

С этими вопросами мы дошли до сути дела. Есть несколько аспектов, которые необходимо рассмотреть.

- *Выборка решает некоторые инженерные задачи.* В популярном ныне обсуждении больших данных основное внимание уделяется корпоративным решениям, таким как Hadoop, являющимся достойным решением инженерных и вычислительных задач, которые вызваны слишком большим количеством данных, не учитываемых в выборках. Так, в Google разработчики программного обеспечения, исследователи данных и статистики регулярно составляют выборки.

  Сколько нужно данных, в действительности зависит от целей: для анализа или вывода обычно не нужно хранить все данные постоянно. С другой стороны, это может потребоваться для удобства применения; например, для отображения корректной информации в пользовательском интерфейсе необходимо иметь всю информацию, актуальную для каждого конкретного пользователя.

- *Искажение.* Даже имей мы доступ ко всему корпусу данных Facebook, Google или Twitter, не следует экстраполировать любые выводы из этих данных на людей, которые не входят в число их пользователей, или даже на этих же пользователей, но в какой-либо другой день.

  Кейт Кроуфорд (Kate Crawford), старший научный сотрудник Microsoft Research, рассказывает в своем разговоре со Strata Hidden Biases of Big Data («Скрытые искажения больших данных»), что если бы вы проанализировали твиты непосредственно перед ураганом Сэнди и сразу после него, то подумали бы, что большинство людей совершали покупки в супермаркетах до Сэнди и устраивали вечеринки после. Однако большинство этих твитов пришло от жителей Нью-Йорка. Во-первых, они более преданные пользователи Twitter, чем, скажем, жители прибрежного Нью-Джерси; и во-вторых, жители прибрежного Нью-Джерси беспокоились о других вещах, таких как сохранность их падающих домов, — и у них не было времени на твиты.

  Иными словами, если вы используете данные твитов, то подумаете, что ураган Сэнди был не так уж и ужасен. Единственный вывод, который вы можете на самом деле сделать, — каков был ураган Сэнди для подмножества пользователей Twitter (которые сами не являются представителями всего населения США), чья ситуация была не настолько плохой, чтобы не иметь времени на твиты.

  Обратите также внимание на то, что в этом случае если бы вы *не располагали* контекстом и не знали об урагане Сэнди, то не имели бы достаточно знаний для правильной интерпретации этих данных.

- *Составление выборок.* Переосмыслим, что такое генеральная совокупность и выборка в различных контекстах.

  В статистике мы часто моделируем связь между популяцией и образцом с помощью лежащего в основе математического процесса. Поэтому делаем упрощающие *допущения* об основополагающей истине, математической структуре и форме лежащего в основе генерирующего процесса, который создал эти данные. Мы наблюдаем только одну конкретную реализацию порождающего процесса, то есть выборку.

Как следствие, если мы воспринимаем все электронные письма в почте компании BigCorp как генеральную совокупность и случайным образом составим выборку на основе этой генеральной совокупности, прочитав лишь некоторые, но не все электронные письма, то этот процесс выборки создаст одну конкретную выборку. Однако при повторной выборке мы получим другой набор наблюдений.

Неопределенность, создаваемая таким процессом выборки, имеет имя: *распределение выборки*. Но, как показано в фильме 2010 года «Начало» (https://ru.wikipedia.org/wiki/Начало_(фильм,_2010)) с Леонардо Ди Каприо, где он погружается в сон во сне, можно воспринимать полный корпус электронных писем сотрудников BigCorp не как совокупность, но как выборку.

Подобный набор электронных писем (и здесь мы начинаем философствовать, но в этом вся суть) действительно может быть только одной реализацией из какой-то более крупной *сверхсовокупности*, и если бы Великий Подбрасыватель Монет на небесах снова бросил монету в этот день, то мы бы наблюдали другой набор писем.

В этой интерпретации мы рассматриваем указанный набор писем как выборку, которую будем использовать, чтобы делать выводы о базовом генерирующем процессе — привычкой всех сотрудников BigCorp писать электронные письма.

- *Новые виды данных*. Прошли те времена, когда данные были всего лишь группой чисел и категориальных переменных. Сильный исследователь данных должен быть универсальным и комфортно себя чувствовать, обрабатывая различные типы данных, в том числе:
  - традиционные — числовые, категориальные или бинарные;
  - текст — электронные письма, твиты, статьи The New York Times (см. главу 4 или 7);
  - записи — данные уровня пользователя, данные о событиях с метками даты/времени, файлы журнала в формате JSON (см. главу 6 или 8);
  - данные, основанные на геолокации, — немного затрагиваются в этой главе в контексте жилья в городе Нью-Йорк;
  - сеть (см. главу 10);
  - данные с датчиков (здесь не рассматриваются);
  - изображения (здесь не рассматриваются).

  Эти новые виды данных требуют более тщательного осмысления того, что значит выборка в подобных контекстах.

  Например, данные реального времени меняются со скоростью воды, бьющей из брандспойта. Вы пытаетесь проанализировать набор данных на уровне пользователя Facebook за неделю активности, которую агрегировали из журналов событий с метками времени. Будут ли какие-либо выводы, сделанные из этого набора данных, релевантными для следующих недели или года?

Как вы составляете выборку из сети, сохраняя при этом сложную структуру последней?

Многие из приведенных вопросов представляют собой открытые исследовательские вопросы для сообществ статистиков и информатиков. Это передовая! Учитывая, что некоторые из них являются открытыми исследовательскими проблемами, на практике ученые-исследователи делают все возможное, и часто изобретение новых методов — часть их работы.

ТЕРМИНОЛОГИЯ: БОЛЬШИЕ ДАННЫЕ

Мы уже много раз упоминали термин «большие данные», но ни разу четко не определили его.

Существует несколько способов воспринимать большие данные.

«Большой» — это движущаяся цель. Построение порогового объема больших данных, таких как 1 петабайт, бессмысленно, поскольку оно делает эту границу абсолютной. Только когда размер становится проблемой, стоит упомянуть его как «большой». Таким образом, это относительный термин, ссылающийся на то, что размер данных опережает возможности доступных современных вычислительных решений (с точки зрения памяти, хранения, сложности и скорости обработки). Поэтому в 1970-х годах «большой» означало нечто иное, чем сегодня.

«Большой» — это когда вы не можете поместить все данные на одной машине. Разные люди и компании обладают отличными вычислительными ресурсами, поэтому для одного специалиста данные являются большими, если он не может поместить их на одной машине, поскольку такое событие потребует научиться целому новому множеству инструментов и методов.

Большие данные — это культурный феномен. В нем описывается, сколько данных является частью нашей жизни, которые концентрируются благодаря все ускоряющемуся прогрессу в области технологий.

Четыре V: объем (volume), разнообразие (variety), скорость (velocity) и значимость (value). Многие люди распространяют эту мысль как способ характеристики больших данных. Возьмите из нее что желаете.

Большие данные могут означать большие допущения

В главе 1 мы упоминали статью «Происхождение больших данных» Кюкера и Майер-Шенбергера. В ней авторы утверждают, что революция больших данных состоит из трех аспектов, таких как:

- сбор и использование большого количества данных вместо небольших выборок;
- допущение беспорядка в данных;
- допущение незнания причин.

Они описывают эти шаги достаточно подробно, утверждая, что большим данным не нужно понимать причину, учитывая их огромность. Не стоит беспокоиться об ошибках составления выборки, поскольку она буквально *отслеживает истину*. В статье утверждается, что новый подход к большим данным позволяет сделать допущение *N* = ВСЕ.

Может ли N = ВСЕ?

Тут вот какое дело: *N* почти никогда не равно ВСЕ. И мы зачастую пропускаем именно то, на что должны обратить внимание в первую очередь.

Например, как сказано в статье в InfoWorld (https://www.infoworld.com/article/2614523/federal-regulations/why-internet-surveillance-will-never-work.html), системы слежения в Интернете никогда не будут работать, поскольку самые умные и искусные преступники, которых мы больше всего хотим поймать, — это те, кого мы никогда не сможем поймать, потому что они всегда на шаг впереди.

Пример из той же статьи — опрос в ночь после выборов. Сам по себе это отличный контрпример: даже если мы опросим полностью всех выходящих из избирательных участков, мы все равно не учтем людей, решивших вообще не голосовать. А это как раз могут быть именно те люди, с которыми нужно поговорить, чтобы понять проблемы выборов в стране.

На самом деле мы готовы утверждать, что допущение *N* = ВСЕ — одна из самых больших проблем, с которыми мы сталкиваемся в эпоху больших данных. Это прежде всего способ исключить голоса людей, не располагающих временем, энергией или доступом к голосованию во всех формах неофициальных, возможно, необъявленных выборов.

Люди, занятые на двух работах и тратящие время на ожидание автобусов, становятся невидимыми, когда мы подсчитываем голоса без них. Для вас это может означать следующее: рекомендации, которые вы получаете на Netflix, выглядят не очень подходящими, поскольку большинство из тех, кто потрудился оценить шоу на Netflix, молоды и могут иметь вкусы, отличные от ваших, что приводит к перекосу механизма рекомендаций в их сторону. Но есть множество гораздо более коварных последствий, вытекающих из этой основной мысли.

Данные необъективны

Другой вариант, в контексте которого допущение *N* = ВСЕ может иметь значение, состоит в следующем: это допущение часто трансформируется в мысль, будто данные *объективны*. Неверно полагать, что данные объективны или они «говорят», и остерегайтесь людей, утверждающих иное.

Недавно нам напомнили об этом ужасным образом с помощью статьи в The New York Times, посвященной большим данным и практикам набора персонала рекру-

терами (https://www.nytimes.com/2013/04/28/technology/how-big-data-is-playing-recruiter-for-specialized-workers.html?pagewanted=4&ref=business&pagewanted=all&_r=1&). В какой-то момент исследователь данных сказал: «Введем все в систему и позволим данным говорить самим за себя».

Если вы прочтете всю статью, то узнаете, что алгоритм пытается найти на работу человека с большими задатками. Это достойная попытка, но ее нужно как следует продумать.

Скажем, вы решили сравнить женщин и мужчин с абсолютно одинаковыми квалификациями, которые были наняты в прошлом, но затем, изучая то, что произошло дальше, вы узнаете, что эти женщины чаще увольняются, чаще получают повышение и дают больше отрицательных отзывов об условиях труда по сравнению с мужчинами.

Вероятно, в следующий раз ваша модель может нанять мужчину, а не женщину, когда появятся два похожих кандидата, вместо того, чтобы рассмотреть возможность негативного отношения компании к женщинам.

Другими словами, игнорирование причинности может быть недостатком, а не характеристикой. Модели, которые игнорируют причинность, могут прибавить исторических проблем вместо того, чтобы их решить (мы рассмотрим это более подробно в главе 11). И данные не говорят сами за себя. Данные — просто количественное, бледное эхо событий, происходящих в нашем обществе.

N = 1

На противоположном от N = ВСЕ конце спектра находится утверждение n = 1, под которым мы подразумеваем размер выборки 1. В старые времена размер выборки 1 был бы смешным: вы никогда не захотите делать выводы обо всем населении, глядя лишь на одного человека. Не волнуйтесь: это по-прежнему смешно. Но понятие n = 1 приобретает новый смысл в эпоху больших данных, где для одного человека мы фактически можем записывать тонны информации и на самом деле могли бы даже сделать выборку из всех событий или действий, которые предпринял этот человек (например, телефонные звонки или нажатия клавиш), и сделать выводы о нем. Это то, что называется моделированием на уровне пользователя.

Моделирование

В следующей главе мы рассмотрим, как модели создаются на основе собираемых данных, но сначала хотели бы обсудить, что подразумевается под этим термином.

Рэйчел говорила с кем-то по телефону о семинаре *по моделированию* и через несколько минут поняла, что слово «модель» означает для собеседника совершенно другую вещь. Он применял этот термин для обозначения моделей

данных — представление, используемое для хранения своих данных, что является компетенцией администраторов баз данных, тогда как Рэйчел говорила о *статистических моделях*, о которых идет речь в нашей книге. Одной из статей блога Эндрю Гельмана (Andrew Gelman) по моделированию недавно поделились в Twitter люди из индустрии моды, но это совсем другая проблема.

Даже если вы использовали термины *«статистическая модель»* или *«математическая модель»* в течение многих лет, понятно ли вам самим и людям, с которыми вы говорите, о чем речь? Что делает модель *моделью*? Кроме того, хотя мы задаем такие фундаментальные вопросы, какова разница между статистической моделью и алгоритмом машинного обучения?

Прежде чем мы углубимся в тему, добавим немного контекста с помощью этой преднамеренно провокационной статьи из журнала Wired: *The End of Theory: The Data Deluge Makes the Scientific Method Obsolete* («Конец теории: поток данных делает научный метод устаревшим») (https://www.wired.com/2008/06/pb-theory/), опубликованной в 2008 году Крисом Андерсоном (Chris Anderson), тогда главным редактором.

Андерсон приравнивает огромное количество данных к полной информации и утверждает: никакие модели не нужны и «корреляции достаточно»; например, что в контексте огромных объемов данных «они [Google] не должны соглашаться на модели вообще».

В самом деле? Мы придерживаемся другого мнения и не думаем, что вы будете так считать, дочитав эту книгу до конца. Но подобный ход рассуждений напоминает статью Кюкера и Майер-Шенбергера, которую мы обсуждали чуть выше в контексте N = ВСЕ, так что вы уже можете почувствовать глубокое замешательство, которое мы наблюдаем вокруг нас.

Хвала прессе, ведь она в настоящее время освещает эти вопросы и проблемы, и кто-то должен это делать. Тем не менее это трудно осуществить, когда люди, создающие мнение, фактически не работают с данными. Внимательно подумайте о том, соглашаетесь ли вы с мнением Андерсона: где вы согласны, где не согласны и где вам нужна дополнительная информация для формирования собственной точки зрения.

Учитывая, что именно так популярная пресса в настоящее время описывает проблему и влияет на общественное восприятие науки о данных и моделировании, нам как исследователям данных стоит знать об этом и наполнять эфир комментариями от людей, владеющих информацией.

Что мы имеем в виду в таком контексте, когда говорим слово *«модели»*? И как мы используем их, будучи исследователями данных? Чтобы получить ответы на эти вопросы, погрузимся глубже.

Что такое модель

Люди пытаются понять окружающий мир, представляя его по-разному. Архитекторы отражают атрибуты зданий через чертежи и трехмерные уменьшенные версии. Молекулярные биологи фиксируют структуру белка с помощью трехмерной визуализации связей между аминокислотами. Статистики и исследователи данных отражают неопределенность и случайность процессов генерации данных с помощью математических функций, которые выражают форму и структуру самих данных.

Модель — наша попытка понять и представить природу реальности через конкретную линзу, будь то архитектура, биология или математика.

Модель представляет собой искусственную конструкцию, в которой все посторонние детали были удалены или абстрагированы. После анализа модели всегда следует уделять внимание этим абстрагированным деталям, чтобы увидеть возможные упущения.

В случае белков модель белковой цепи с боковыми связями сама по себе удаляется из законов квантовой механики, определяющих поведение электронов, которые в конечном счете диктуют структуру и действия белков. В случае статистической модели мы, вероятно, ошибочно отмели ключевые переменные, при этом добавили нерелевантные переменные или предположили, что математическая структура отделена от реальности.

Статистическое моделирование

Прежде чем вы глубоко погрузитесь в данные и начнете писать программный код, полезно нарисовать процесс, который, по вашему мнению, будет лежать в основе вашей модели. Что на первом месте? Что на что влияет? Какие последствия вызывает тот или иной элемент? Как это проверить?

Но все люди думают по-разному. Некоторые предпочитают выражать такие отношения на языке математики. Математические выражения достаточно общие, поэтому должны включать параметры, но значения последних пока неизвестны.

В математических выражениях по соглашению используются буквы греческого алфавита для параметров и латинских букв для данных. Например, если у вас есть два столбца данных, x и y, и вы думаете, что существует линейная связь, то должны записать $y = \beta_0 + \beta_{1x}$. Вы еще не знаете, что такое β_0 и β_1 с точки зрения фактических чисел, поэтому они и называются параметрами.

Другие предпочитают фотографии и сначала рисуют диаграмму потока данных, возможно со стрелками, показывая, как одни элементы влияют на другие или что происходит с течением времени. Так появляется абстрактная картина отношений, предваряющая выбор уравнений для выражения этих отношений.

Но как создать модель?

Как определить функциональную форму, которую должны принимать данные? Правда в том, что это отчасти искусство и отчасти наука. И к сожалению, именно по этому аспекту учебники дают меньше всего указаний, несмотря на то что это и есть ключ ко всему процессу. В конце концов, это часть процесса моделирования, в которой вам нужно сделать много предположений о базовой структуре реальности, и должны быть стандарты относительно того, как делать выбор и объяснить его. Но глобальных стандартов нет, поэтому мы создаем их по мере продвижения и, надеемся, вдумчиво.

Скажем прямо: непонятно, с чего начать. Будь иначе, люди знали бы смысл жизни. Однако мы сделаем все возможное, чтобы продемонстрировать на протяжении всей книги, как это делается.

Начать можно с разведочного анализа данных (РАД), который мы рассмотрим ниже. Он влечет создание графиков и тренировку интуиции для вашего конкретного набора данных. РАД очень помогает, как и метод проб, ошибок и итераций.

Честно говоря, пока вы не набили руку, все это кажется очень загадочным. Лучше всего начать с простого, а затем увеличить сложность. Совершите самое глупое действие, которое только можете себе представить. Наверное, оно не такое и глупое.

Например, вы можете (и должны) рисовать гистограммы и смотреть на диаграммы рассеивания, чтобы начать ощущать данные. Затем просто попытайтесь записать что-то, даже если сначала это что-то неправильное (вероятно, записанное вами сначала будет неверным, но это не имеет значения).

Поэтому попробуйте записать линейную функцию (подробнее о ней в следующей главе). Когда вы будете ее записывать, заставьте себя думать: а эта функция *вообще* имеет смысл? Если нет, то почему? Что будет иметь *больше* смысла? Вы начинаете с простого и постоянно увеличиваете сложность, делаете предположения и записываете их. Вы можете задействовать полноценные предложения, если это помогает, например: «Я допускаю, что мои пользователи естественным образом группируются примерно в пять групп, поскольку, когда я слышу, как торговые представители говорят о них, в своих обсуждениях они выделяют около пяти разных типов людей». Затем вы берете слова и пытаетесь выразить их в форме уравнения или программного кода.

Запомните: всегда полезно начинать с простого. Существует компромисс в моделировании между простым и безошибочным. Простые модели могут быть проще в интерпретации и понимании. Зачастую грубая простая модель сделает за вас 90 % работы и ее создание и отладка займет всего пару часов, тогда как более сложную модель можно строить несколько месяцев и получить в результате 92 % эффективности.

В нашей книге вы начнете создавать арсенал заготовок моделей. Некоторые из строительных блоков этих моделей являются *распределениями вероятностей*.

Распределения вероятностей

Распределение вероятностей является основой статистических моделей. Когда мы доберемся до линейной регрессии и наивного классификатора Байеса, вы увидите, как это происходит на практике. Курс теории вероятности может занять несколько семестров, поэтому очень сложно ужать его до одного небольшого раздела.

Давным-давно, до появления компьютеров, ученые наблюдали реальные явления, делали измерения и замечали, что некоторые математические формы возникают вновь и вновь. Классический пример — рост человека, который подчиняется *нормальному* распределению — колоколообразной кривой, также называемой гауссовым распределением, названной в честь немецкого математика Карла Гаусса.

Другие общие формы были названы также в честь их наблюдателей (например, распределение Пуассона или распределение Вейбулла), в то время как другие формы, такие как гамма- или экспоненциальные распределения, называются в честь связанных математических объектов.

Естественные процессы имеют тенденцию генерировать измерения, чья эмпирическая форма подходит для выражения математическими функциями с несколькими параметрами, которые можно оценить по данным.

Не *все* процессы (но многие) генерируют данные, которые похожи на *именные* распределения. Мы можем использовать эти функции в качестве строительных блоков наших моделей. Детальное описание каждого распределения выходит за рамки нашей книги, но на рис. 2.1 мы приводим их в качестве иллюстрации различных общих фигур и напоминаем: у них есть имена только потому, что кто-то их наблюдал достаточное количество раз, благодаря чему подумал, будто эти распределения заслуживают имен. На самом же деле существует бесконечное количество возможных распределений.

Они должны восприниматься как назначающие *вероятности* подмножеству вероятных исходов и имеющие соответствующие функции. Например, нормальное распределение записывается так:

$$N(x \mid \mu, \sigma) \approx \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}.$$

Параметр μ является средним и медианой и управляет тем, где центрируется распределение (поскольку оно симметричное), а параметр σ контролирует, как то распространяется. Это общая функциональная форма, но для конкретного явления реального мира указанные параметры имеют фактические числа в виде значений, которые мы можем оценить по данным.

Случайную переменную, обозначенную как x или y, можно считать имеющей соответствующее распределение вероятностей $p(x)$, которое отображает x в положительное

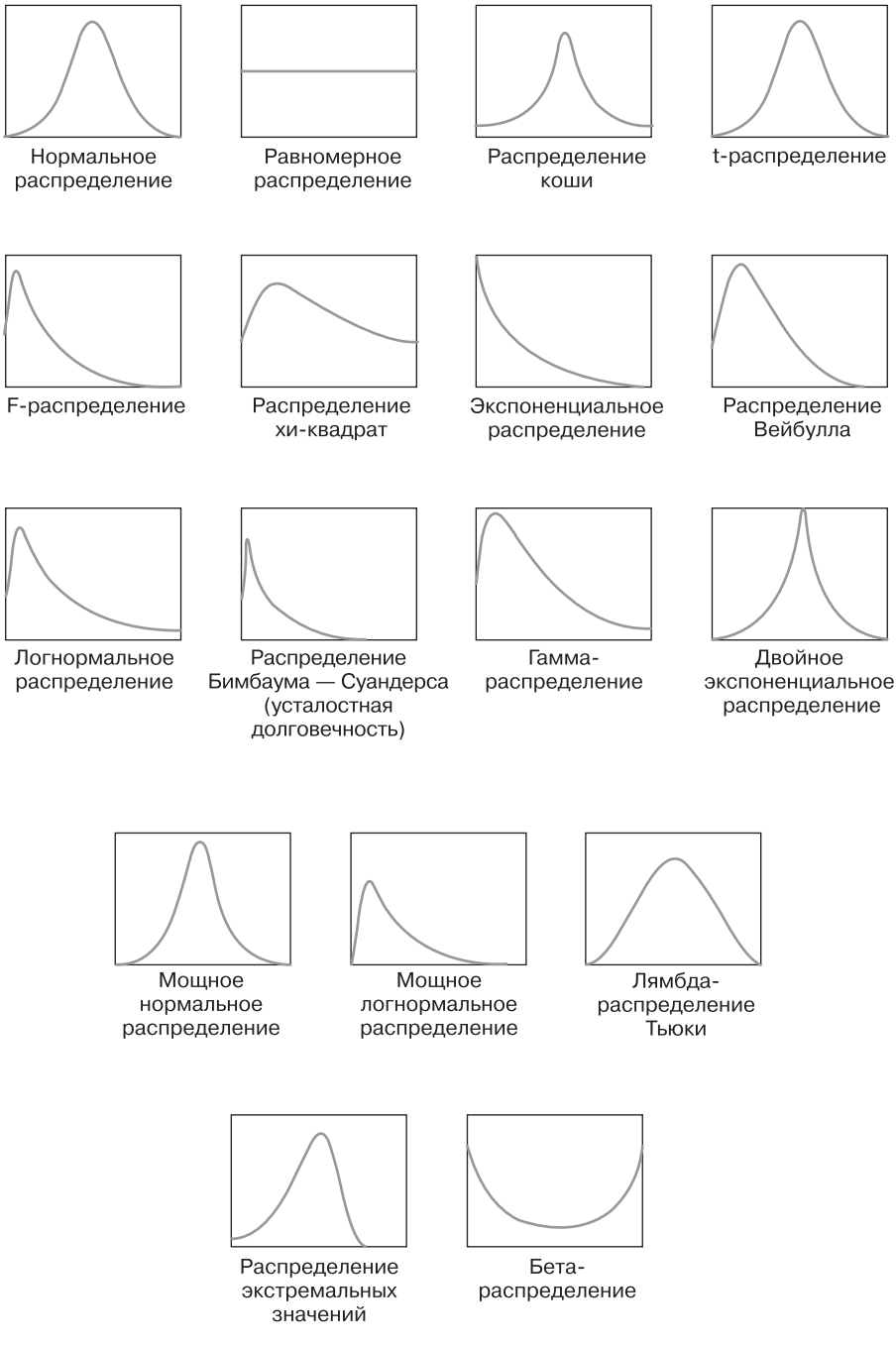


Рис. 2.1. Примеры непрерывных функций плотности (или распределения вероятности)

вещественное число. Мы ограничены в своем выборе возможных функций: чтобы можно было интерпретировать нашу функцию как вероятность и получить должную функцию плотности вероятности, площадь под кривой (которую можно узнать путем интегрирования $p(x)$) должна равняться 1.

Например, пусть x — количество времени до прибытия следующего автобуса (измеряется в минутах). Это случайная величина, поскольку существует определенная вариативность и неопределенность в количестве времени, которое проходит до прибытия следующего автобуса.

Предположим, мы знаем (чисто теоретически), что время до прибытия следующего автобуса имеет функцию плотности вероятности $p(x) = 2e^{-2x}$. Если мы хотим знать, какова вероятность того, что следующий автобус прибудет в промежутке между 12 и 13 минутами, то находим область под кривой между 12 и 13 с помощью выражения $\int_{12}^{13} 2e^{-2x}$.

Как узнать, что это распределение подходит для использования? Есть два возможных способа проверить. Первый: провести эксперимент, в ходе которого мы появляемся на остановке в случайное время, измеряем время до прибытия следующего автобуса и повторяем этот эксперимент снова и снова. Затем посмотрим на измерения, построим график и приблизим функцию, как уже обсуждалось ранее. Второй способ: поскольку мы знакомы с тем, что «время ожидания» — достаточно распространенное явление реального мира, для описания которого было предложено распределение, называемое экспоненциальным, то знаем, что оно принимает вид $p(x) = \lambda e^{-\lambda x}$.

Помимо обозначения распределений отдельных случайных величин с функциями одной переменной, мы используем многомерные функции, называемые *совместными распределениями*, которые выполняют одно и то же преобразование нескольких случайных величин. Например, при двух случайных величинах мы могли бы обозначить наше распределение функцией $p(x, y)$, эта функция примет значения в системе координат и даст неотрицательные значения. В соответствии с интерпретацией данной функции как вероятности ее (двойной) интеграл по всей системе координат будет равен 1.

Есть также то, что называется *условным распределением*, $p(x|y)$; оно должно интерпретироваться как функция плотности x при заданном значении y.

Когда мы работаем с данными, установка условий соответствует выделению подмножеств. Так, предположим, у нас есть набор данных пользовательского уровня сайта Amazon.com, на котором для каждого пользователя распечатывается сумма денег, потраченная им в прошлом месяце на торговой площадке, независимо от того, является ли пользователь мужчиной или женщиной, и того, сколько товаров тот просмотрел до добавления первой позиции в корзину.

Если мы рассмотрим x как случайную величину, которая представляет собой потраченную сумму денег, то можем посмотреть распределение денег, потраченных всеми пользователями, и представить его как функцию $p(x)$.

Затем мы можем взять подмножество пользователей, которые просмотрели более пяти позиций, прежде чем купить что-либо, и оценить распределение денег, потраченных этими пользователями. Пусть y — случайная величина, представляющая количество рассмотренных товарных позиций, тогда $p(x|\, y > 5)$ будет соответствующим *условным распределением*. Обратите внимание: условное распределение имеет те же свойства, что и регулярное, в случае его интеграции оно суммируется до 1 и принимает неотрицательные значения.

Наблюдая точки данных, например, (x_1, y_1), (x_2, y_2)... (x_n, y_n), мы видим *реализации* пары случайных величин. При наличии полного набора данных с n-количеством строк и k-количеством столбцов мы наблюдаем n реализаций совместного распределения этих k случайных величин.

Узнать больше о распределении вероятностей можно в книге *A First Course in Probability* («Первый курс по вероятности») Шелдона Росса (Sheldon Ross), издательство Pearson.

Обучение модели

Обучение модели означает, что вы оцениваете параметры модели с помощью наблюдаемых данных. Вы используете свои данные в качестве доказательства, помогающего вам приблизить математический процесс реального мира, который и сгенерировал данные. Обучение модели часто включает методы и алгоритмы оптимизации, такие как *оценка максимального правдоподобия*, которые позволяют получить параметры.

Фактически, когда вы оцениваете параметры, они на самом деле выступают *оценщиками*, то есть сами являются *функциями* данных. Как только вы обучите модель, можете записать ее, например, как $y = 7{,}2 + 4{,}5x$. Это значит, что лучше всего предположить следующее: приведенное уравнение или форма функции выражает отношения между двумя переменными, основываясь на предположении о том, что данные следуют линейной закономерности.

Обучение модели — это когда вы фактически начинаете писать программный код: он будет считывать данные, вы укажете форму функции, которую записали на листе бумаги. Затем язык R или Python станут использовать встроенные методы оптимизации, чтобы дать наиболее вероятные значения параметров этих данных.

По мере наращивания опыта и знаний или если это одна из ваших областей специализации, вы сами будете углубляться в методы оптимизации. Изначально вам следует понимать, что производится оптимизация и как она работает, но не нужно самостоятельно программировать эту часть — она базируется на функциях языка R или Python.

Переобучение

На протяжении всей книги мы будем неоднократно предупреждать о *переобучении*. Переобучение — термин, служащий для обозначения того, что вы использовали

набор данных для оценки параметров вашей модели, но она не очень хороша в отражении реальности, находящейся вне ваших данных.

Возможно, вы сталкивались с переобучением, поскольку пытались прогнозировать с его помощью метки для другого набора данных, не использовавшегося в целях подгонки модели, однако дополнительная мера оценки, скажем безошибочность, показала, что модель работает плохо.

Разведочный анализ данных

> «Разведочный анализ данных» — это отношение, состояние гибкости, готовность искать то, что считается отсутствующим, а также то, что считается существующим.
>
> *Джон Тьюки (John Tukey)*

Ранее мы говорили о разведочном анализе данных как о первом шаге к построению модели. РАД часто приводят в первой главе (здесь мы подразумеваем «самый простой» и самый низкий уровень) стандартных учебников по введению в статистику, а затем в остальной части книги забывают о нем.

Данный метод традиционно представляет собой набор гистограмм и диаграмм «стебель — листья». Ему обучают детей в пятом классе, поэтому кажется тривиальным, не так ли? Неудивительно, что никто не думает о разведочном анализе.

Но РАД является важной частью процесса научных исследований данных, а также представляет собой философию и способ демонстрировать статистику, практикуемые школой статистиков, которая происходит из традиции лаборатории Bell Labs.

Джон Тьюки, математик из этой лаборатории, разработал разведочный анализ данных как противоположность подтверждающему анализу, рассматривающему моделирование и гипотезы, как описано в предыдущем разделе. В РАД нет никакой гипотезы, как нет и модели. «Разведочный» аспект означает, что ваше понимание проблемы, которую вы решаете или можете решить, меняется по мере продвижения.

Основные инструменты РАД — диаграммы, графики и сводная статистика. Вообще говоря, это метод систематического прохождения по данным, построения графиков распределений всех переменных (с помощью блочных диаграмм), временны*х* рядов данных, трансформации переменных, просмотра с использованием матриц рассеивания всех попарных отношений между переменными, а также формирования сводной статистики для всех этих элементов. По крайней мере, это означало бы вычисление среднего, минимального, максимального, верхнего и нижнего квартилей, как и выявление выбросов.

ИСТОРИЧЕСКАЯ СПРАВКА: BELL LABS

Bell Labs — исследовательская лаборатория, восходящая к 1920-м годам. В ней родились инновации в физике, информатике, статистике и математике, разработаны такие языки, как C++, и многие ее сотрудники — лауреаты Нобелевской премии.

В Bell Labs работала очень успешная и продуктивная статистическая группа, и среди ее многочисленных заметных членов был Джон Тьюки, математик, который трудился над множеством статистических проблем. Тьюки считается отцом РАД и языка R (начинавшегося как язык S в этой лаборатории, R — версия с открытым исходным кодом), он также интересовался попытками визуализации высокоразмерных данных.

Мы считаем Bell Labs одним из мест, где была рождена наука о данных благодаря сотрудничеству научных дисциплин и огромного количества сложных данных, доступных людям, работающих там. Это была виртуальная игровая площадка для статистиков и специалистов в области компьютерных наук, так же как и Google сегодня.

В 2001 году Билл Кливленд (Bill Cleveland) написал монографию *Data Science: An Action Plan for expanding the technical areas of the field of statistics* («Наука о данных: план действий по расширению технических областей в статистике»). В ней он описывал многодисциплинарные исследования, модели и методы для данных (традиционные прикладные показатели), вычисления с данными (аппаратное и программное обеспечение, алгоритмы, написание программного кода), педагогику, оценку инструментов (то, как оставаться в тренде современных тенденций в технологии), а также теорию (математические основы данных). Вы можете больше узнать о Bell Labs из книги *The Idea Factory* («Фабрика идей») Джона Гертнера (Jon Gertner) (издательство Penguin Books).

Но насколько РАД — набор инструментов, настолько же он и мышление. И оно касается ваших отношений с данными. Вы хотите понять их, то есть наработать интуицию, понять форму данных и попытаться связать свое понимание процесса, который сгенерировал данные, с ними самими. РАД происходит между вами и данными и ни в коем случае не доказывает ничего кому-либо еще.

Философия разведочного анализа данных

> Задолго до того как начать волноваться о том,
> как убедить других, вы сначала должны сами понять,
> что происходит.
>
> *Эндрю Гельман*

Во время работы в Google Рэйчел повезло иметь в коллегах двух бывших аналитиков из Bell Labs/AT&T — Дэрила Прегибона (Daryl Pregibon) и Диану Ламберт (Diane Lambert), которые также работают в этом направлении прикладной статистики, — и они подсказали ей сделать РАД частью ее передовых приемов.

Даже с очень большими данными масштаба Google они делали РАД. В контексте данных в интернет-инжиниринговой компании такой анализ выполняется по тем же причинам, что и с меньшими наборами информации, но в такой компании есть дополнительные причины делать разведочный анализ данных, полученных из журналов протоколирования.

Существуют важные причины, по которым любой человек, работающий с данными, должен делать РАД. А именно, чтобы наработать интуицию относительно данных; проводить сравнения между распределениями; контролировать корректность данных (убедиться, что они находятся в ожидаемом масштабе и в том формате, в котором должны быть); выяснить, где отсутствуют данные или имеются ли выбросы, и обобщить данные.

В контексте данных, генерируемых из журналов протоколирования, разведочный анализ также помогает отлаживать процесс ведения такого журнала. Например, «закономерности», обнаруживаемые в данных, действительно могут быть некой ошибкой протоколирования, которую необходимо устранить. Если вы не займетесь проблемой отладки, то продолжите думать, что ваши закономерности реальны. Инженеры, с которыми мы работали, всегда благодарны за помощь в этой области.

В конце концов, РАД помогает убедиться, что продукт работает так, как должен.

Хотя в разведочном анализе много визуализации, мы различаем РАД и визуализацию данных: РАД выполняется в начале анализа, а визуализация (которую мы рассмотрим в главе 9), как мы говорим на нашем жаргоне, производится к концу анализа, чтобы сообщить свои выводы. В случае с РАД графики предназначены исключительно для *вас*, чтобы помочь понять происходящее.

РАД также позволяет использовать полученный опыт для информирования и улучшения создания алгоритмов. Предположим, вы пытаетесь построить алгоритм ранжирования, который оценивает контент, показываемый пользователям. Для этого вам может понадобиться разработать понятие «популярный».

Прежде чем вы решите, как количественно оценить популярность (например, самую высокую частоту переходов, или сообщение с наибольшим количеством комментариев, или с количеством комментариев выше определенного порога, или некоторое средневзвешенное большого количества других метрик), вам нужно понять, как данные ведут себя, и лучший способ сделать это — изучить их.

Построение графиков данных и их сравнение позволит продвинуться очень далеко, и гораздо лучше сделать это, чем получить набор данных и сразу запустить регрессию только потому, что вы знаете, как это делается. Аналитики и исследователи данных допустили большое упущение, не применяя РАД в качестве критичной части процесса работы с данными. Воспользуйтесь этой возможностью, чтобы сделать разведочный анализ частью вашего процесса!

Вот несколько источников, которые помогут понять передовые приемы и исторический контекст.

1. Tukey J. *Exploratory Data Analysis*. Pearson.
2. Tufte E. *The Visual Display of Quantitative Information*. Graphics Press.
3. Cleveland W. S. *The Elements of Graphing Data*. Hobart Press.
4. Jacoby W. G. *Statistical Graphics for Visualising Multivariate Data*. Sage.
5. Gelman A. *Exploratory Data Analysis for Complex Models*. American Statistical Association.
6. Tukey J. *The Future of Data Analysis* // Annals of Mathematical Statistics, Volume 33, Number 1. 1962. 1–67 (https://web.stanford.edu/~gavish/documents/Tukey_the_future_of_data_analysis.pdf).
7. Brillinger D. *Data Analysis, Exploratory* // International Encyclopedia of Political Science. Sage.

Упражнение: РАД

На сайте https://github.com/oreillymedia/doing_data_science вы найдете 31 набор данных с именами `nyt1.csv`, `nyt2.csv`, ..., `nyt3.csv`.

Каждый из них демонстрирует один (симулированный) день показов объявлений и переходов по ним, записанных на главной странице газеты The New York Times в мае 2012 года. Каждая строка представляет одного пользователя. Существует пять столбцов: возраст, пол (0 = женщина, 1 = мужчина), количество показов, количество переходов и статус авторизации.

Для обработки этих данных вы будете использовать язык программирования R. Он создан специально для анализа данных и интуитивно понятен, что облегчает его применение. Он доступен для скачивания здесь: https://www.r-project.org/. После установки можете загрузить один файл в R с помощью следующей команды:

```
data1 <- read.csv(url("http://stat.columbia.edu/~rachel/datasets/nyt1.csv"))
```

После загрузки данных можно выполнить разведочный анализ.

1. Создайте новую переменную `age_group`, которая классифицирует пользователей как `<18`, `18–24`, `25–34`, `35–44`, `45–54`, `55–64` и `65+`.
2. Для одного дня сделайте следующее.
   - Зафиксируйте на диаграмме количество показов и показатель переходов (CTR = # clicks / # inmpressions) для этих шести возрастных категорий.
   - Определите новую переменную для сегментации или категоризации пользователей в зависимости от переходов.
   - Изучите данные и проведите визуальные и количественные сравнения между сегментами пользователей/демографическими группами (например,

мужчины старше 18 лет по сравнению с женщинами старше 18 лет или авторизованные и неавторизованные пользователи).

- Создайте метрики/измерения/статистику, которые суммируют данные. Примеры возможных метрик включают CTR, квантили, среднее значение, медиану, дисперсию и максимальное значение. Эти показатели можно рассчитать по различным сегментам пользователей. Будьте избирательны. Подумайте об элементах, которые важно отслеживать с течением времени — что сжимает данные, но по-прежнему захватывает поведение пользователя.

3. Теперь проводите анализ в течение нескольких дней. Визуализируйте некоторые показатели и распределения во времени.
4. Опишите и интерпретируйте любые закономерности, которые найдете.

Пример кода

В этом пункте будет приведено начало варианта решения данного упражнения. Реальность такова, что в рамках одной книги мы не можем научить вас науке о данных и тому, как написать программный код для всего. Изучение кода на новом языке требует многочисленных проб и ошибок, а также выхода в Интернет и поиска в Google или stackoverflow.

Скорее всего, если вы хотите выяснить, как нарисовать что-то или построить модель в R, существует вероятность, что другие люди тоже попытались это сделать, и вместо того, чтобы изобретать велосипед, можно посмотреть в Интернете. Мы предлагаем не смотреть на код, приведенный ниже, пока вы не попробуете немного поработать самостоятельно:

```
# Автор: Маура Фитцджеральд (Maura Fitzgerald)
data1 <- read.csv(url("http://stat.columbia.edu/~rachel/datasets/nyt1.csv"))

# категоризация
head(data1)
data1$agecat <-cut(data1$Age,c(-Inf,0,18,24,34,44,54,64,Inf))

# просмотр
summary(data1)

# скобки
install.packages("doBy")
library("doBy")
siterange <- function(x){c(length(x), min(x), mean(x), max(x))}
summaryBy(Age~agecat, data =data1, FUN=siterange)

# только у авторизованных пользователей есть пол и возраст
summaryBy(Gender+Signed_In+Impressions+Clicks~agecat,data =data1)

# диаграмма
install.packages("ggplot2")
```

О ПРОГРАММИРОВАНИИ

В авторской колонке за май 2013 года под названием *How to be a Woman Programmer* («Как быть женщиной-программистом») Эллен Ульман (Ellen Ullman) довольно хорошо описывает то необходимое, позволяющее стать программистом (откладывая пока в сторону женский аспект): «Первое требование для программирования — это страсть к работе, глубокая потребность исследовать загадочное пространство между мыслями человека и тем, что способна понять машина; между человеческими желаниями и тем, как машины могут их удовлетворить».

Второе требование — высокий уровень терпимости к неудачам. Программирование — это искусство разработки алгоритмов и отладки ошибок кода. Как сказал великий Джон Бэкус (John Backus), создатель языка программирования Фортран, *«вам нужно быть готовыми все время терпеть неудачу. Вы должны генерировать много идей, и тогда вам следует очень много работать, чтобы обнаружить, что они не работают. И вы продолжаете делать это снова и снова, пока не найдете то, что действительно работает»*.

```
library(ggplot2)
ggplot(data1, aes(x=Impressions, fill=agecat))+geom_histogram(binwidth=1)
ggplot(data1, aes(x=agecat, y=Impressions, fill=agecat))+geom_boxplot()

# показатель переходов
# нас не интересуют переходы, если не было показов
# если есть переходы, но нет показов, то я допускаю, что данные ошибочны
data1$hasimps <-cut(data1$Impressions,c(-Inf,0,Inf))
summaryBy(Clicks~hasimps, data =data1, FUN=siterange)
ggplot(subset(data1, Impressions>0), aes(x=Clicks/Impressions,
colour=agecat)) + geom_density()
ggplot(subset(data1, Clicks>0), aes(x=Clicks/Impressions,
colour=agecat)) + geom_density()
ggplot(subset(data1, Clicks>0), aes(x=agecat, y=Clicks,
fill=agecat)) + geom_boxplot()
ggplot(subset(data1, Clicks>0), aes(x=Clicks, colour=agecat))
+ geom_density()

# создать категории
data1$scode[data1$Impressions==0] <- "NoImps"
data1$scode[data1$Impressions >0] <- "Imps"
data1$scode[data1$Clicks >0] <- "Clicks"

# конвертировать столбец в фактор
data1$scode <- factor(data1$scode)
head(data1)

# посмотреть на уровни
clen <- function(x){c(length(x))}
etable<-summaryBy(Impressions~scode+Gender+agecat, data = data1, FUN=clen)
```

Подсказка для остальной части задания: не считывайте все данные в память. После того как вы довели до совершенства код для обработки данных за один день, считывайте наборы данных по одному за раз, обработайте их, выведите любые соответствующие показатели и переменные и сохраните их в кадре данных, затем удалите текущий набор перед считыванием следующего. Это позволит вам подумать о том, как обрабатывать данные, распределенные по нескольким компьютерам.

Процесс научных исследований данных

Объединим вышеописанное в то, что мы определяем как процесс передачи данных. Чем больше вы видите примеров людей, занимающихся даталогией, тем больше обнаруживаете, что они вписываются в общую структуру, показанную на рис. 2.2. По мере продвижения по книге мы неоднократно и с разных сторон обсудим этапы данного процесса и примеры.

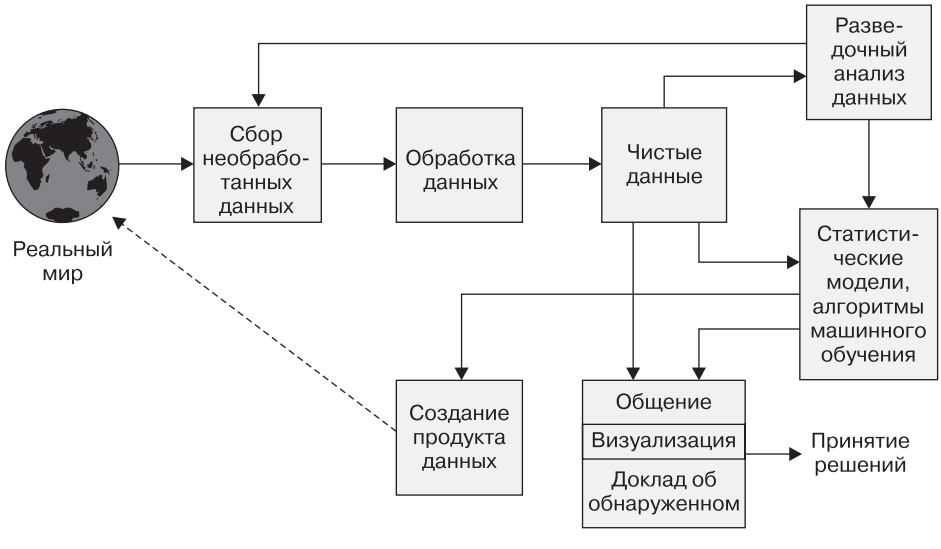


Рис. 2.2. Процесс научных исследований данных

Изначально у нас есть реальный мир. Внутри него много людей, занятых различными видами деятельности. Некоторые люди используют Google+, другие соревнуются на Олимпийских играх, есть спамеры и люди, сдающие кровь. Скажем, у нас есть данные по одному из этих аспектов.

В частности, мы начнем с необработанных данных — протоколов, записей Олимпийских игр, электронных писем сотрудников компании Enron или записанных генетических материалов (обратите внимание, что многие из аспектов этих действий уже потеряны еще на этапе необработанных данных). Мы хотим обработать

данные, чтобы очистить для анализа. Таким образом, создаем и используем конвейеры очистки данных: объединение, агрегацию, перебор — называйте как хотите. Для этого применяем такие инструменты, как Python, сценарии оболочки, R, SQL или все вышеперечисленное.

В конце концов мы приводим данные к какому-то компактному формату, например такому:

имя | событие | год | пол | время события

Обычно на уроках статистики все начинается с этого момента — чистый, упорядоченный набор данных. Но в реальном мире, как правило, все начинается по-другому.

Как только у нас будет этот чистый набор данных, мы должны сделать нечто наподобие РАД. Во время выполнения анализа мы можем понять, что набор на самом деле не является чистым из-за дубликатов, отсутствующих значений, абсурдных выбросов и данных, которые на деле были запротоколированы неправильно или вовсе не прошли протоколирование. Если это так, то, возможно, придется вернуться и собрать больше данных или потратить больше времени на очистку имеющегося набора.

Затем мы разрабатываем нашу модель, в которой будет использоваться некий алгоритм, например *k*-ближайшие соседи (*k*-БС), линейная регрессия, наивный классификатор Байеса или что-то еще. Выбор модели зависит от типа проблемы, которую мы пытаемся решить; конечно, это может быть проблема классификации, прогнозирования или базовая проблема описания.

Затем мы можем интерпретировать, визуализировать, передать или доложить наши результаты. Действия способны принять форму сообщения результатов начальству либо коллегам или публикации статьи в журнале, выхода в люди и дачи академических интервью.

В качестве альтернативы нашей целью может быть создание или прототипирование «продукта данных»; например, классификатор спама либо алгоритм ранжирования поиска или рекомендательная система. Теперь ключевой момент, который делает науку о данных особой и отличной от статистики: этот продукт данных затем *инкорпорируется обратно* в реальный мир, пользователи взаимодействуют с указанным продуктом и генерируют больше данных, что создает цикл обратной связи.

Описанное весьма отличается от, скажем, прогнозирования погоды, где ваша модель вообще не влияет на результат. Например, вам под силу предсказать, что на следующей неделе будет дождь, и если у вас нет определенных способностей, о которых мы не знаем, то вы не сможете *вызвать* дождь. Но, создав рекомендательную систему, которая, скажем, генерирует доказательства того, что «много людей любят эту книгу», вы узнаете, что вызвали этот цикл обратной связи.

При проведении любого анализа учитывайте этот цикл с поправкой на любые искажения, вызванные вашей моделью. Ваши модели не просто прогнозируют будущее, но и *вызывают* его!

Продукт данных, выводимый в линию и взаимодействующий с пользователями, находится на одном полюсе, а погода — на другом. Однако независимо от типа данных, с которыми вы работаете, и «продукта», создаваемого на основе этих данных (будь то публичная политика, определяемая статистической моделью, медицинская страховка или избирательные опросы, широко освещаемые и, возможно, влияющие на мнения зрителей), вы должны учитывать, насколько ваша модель воздействует на тот феномен, который вы пытаетесь пронаблюдать и понять.

Роль исследователя данных в этом процессе. На фоне нашего повествования может сложиться впечатление, будто моделирование предполагает, что все произойдет волшебным образом без вмешательства человека. Под человеком здесь подразумевается исследователь данных. Кому-то нужно принимать решения о том, какие данные собирать и почему. Этот человек должен формулировать вопросы и гипотезы и составлять план того, с какого фланга подойти к проблеме. И этот кто-то — исследователь данных или наша любимая команда таких специалистов.

Повторим или по крайней мере добавим информации, чтобы дать понять: исследователь данных должен всецело участвовать в этом процессе, то есть такие специалисты вносят свою лепту при написании программного кода и принимают участие в процессах более высокого уровня, как показано на рис. 2.3.

СВЯЗЬ С НАУЧНЫМ МЕТОДОМ

Мы можем воспринимать процесс научного изучения данных как расширение или вариацию научного метода:

- задать вопрос;
- провести предварительное исследование;
- сконструировать гипотезы;
- экспериментально проверить гипотезы;
- проанализировать данные и прийти к выводу;
- передать свои результаты.

Как в процессе обработки данных, так и в научном методе не каждая проблема требует прохождения всех этапов, но почти все проблемы можно решить с помощью некой комбинации этапов. Например, если ваша конечная цель — визуализация данных (которая сама по себе может считаться продуктом данных), то, вероятно, вам не нужно заниматься машинным обучением или статистическим моделированием, но вы бы хотели пройти весь путь до получения чистого набора данных, провести некий РАД, а затем создать визуализацию.

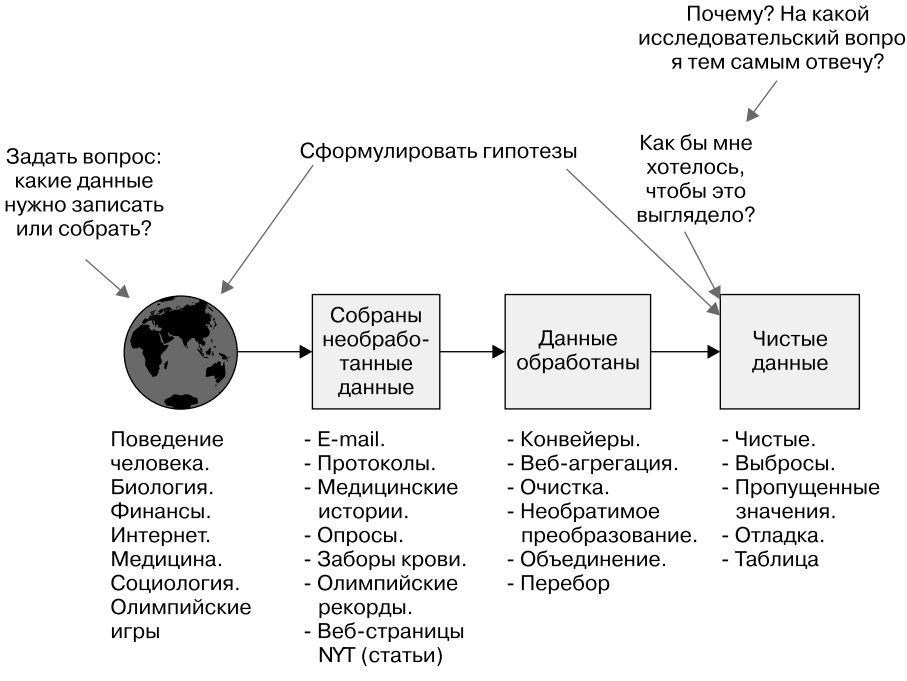


Рис. 2.3. Исследователь данных принимает участие во всех этапах данного процесса

Мысленный эксперимент: как бы вы имитировали хаос?

Большинство проблем с данными начинаются с определенного количества грязных данных, неправильно поставленных вопросов и срочности. Мы, исследователи данных, в некотором смысле пытаемся создать порядок из хаоса. Класс отвлекся от лекции, чтобы обсудить, как имитировать хаос. Вот несколько идей из обсуждения.

- Лоренцианское водяное колесо представляет собой колесо наподобие колеса обозрения с одинаково разнесенными ведрами воды, вращающимися по кругу. Теперь представьте, что вода стекает в систему на самом верху. В каждом ведре есть течь, поэтому некая часть воды попадает в то ведро, которое находится прямо под течью. В зависимости от скорости поступления воды эта система проявляет хаотический процесс, зависящий от молекулярных взаимодействий молекул воды по сторонам ведер. Узнайте больше об этом в статье «Википедии»: https://en.wikipedia.org/wiki/User:Pankajgarg_india/The_lorenzian_waterwheel.

- Многие системы могут проявлять внутренний хаос. Филипп М. Байндер (Philippe M. Binder) и Родерик В. Дженсен (Roderick V. Jensen) написали статью *Simulating Chaotic Behaviour with Finite-State Machines* («Имитация хаотического поведения с конечными автоматами») (https://journals.aps.org/pra/abstract/10.1103/PhysRevA.34.4460), которая посвящена компьютерному моделированию хаоса.
- Многопрофильная программа с участием Массачусетского технологического института, университетов Гарварда и Тафтса включала обучение методу, озаглавленному *Simulating chaos to teach order* («Имитация хаоса для обучения порядку») (https://news.harvard.edu/gazette/). Они имитировали чрезвычайную ситуацию в спорном районе Дарфур на границе Чада и Судана. При этом студенты играли роль членов организации «Врачи без границ», Международного медицинского корпуса и других гуманитарных организаций.
- См. также эссе Джоэля Гаскоина (Joel Gascoigne) *Creating Order from Chaos in a Startup* («Создание порядка из хаоса в стартапе») (http://joel.is/creating-order-from-chaos-in-a-startup/).

ПРИМЕЧАНИЯ ПРЕПОДАВАТЕЛЯ

- Опыт исследователя данных часто хаотичен, так что его задача — попытаться создать порядок из этого хаоса. Я хотела смоделировать этот хаотический опыт для своих учеников на протяжении всего семестра. Но я также хотела, чтобы они знали: вещи будут несколько хаотичными вследствие педагогических причин, а не из-за моей неумелости!
- Я хотела использовать различные интерпретации слова «хаос» как средство подумать о важности лексики и о трудностях, возникающих в общении, когда люди либо не знают, что означает это слово, либо имеют разные представления о его значении. Исследователи данных могут общаться с экспертами предметной области, которые, например, действительно не понимают значения выражения «логистическая регрессия», но будут притворяться, будто знают, из-за нежелания показаться глупыми или потому, что думают, будто должны это знать, и поэтому не спрашивают. Но общение на самом деле не является успешным, если хотя бы один из собеседников не понимает, о чем речь. Точно так же исследователи данных должны задавать вопросы с целью убедиться, что понимают терминологию, используемую экспертом предметной области (будь то астрофизик, специалист по социальным сетям или климатолог). Нет ничего плохого в том, чтобы не знать значения того или иного слова; но не спрашивать неправильно! Вероятно, вы обнаружите, что, задавая уточняющие вопросы о лексике, получаете еще больше информации о проблеме данных, лежащей в основе.
- Моделирование — полезный метод в науке о данных. Может быть полезно моделировать ложные наборы данных из модели, чтобы, например, лучше понять генеративный процесс, а также отладить код.

Практический пример: RealDirect

Дуг Перлсон (Doug Perlson), генеральный директор RealDirect (https://www.realdirect.com/), имеет опыт в области имущественного права, стартапов и онлайн-рекламы. Его цель в RealDirect — использовать все данные, которые он может получить о недвижимости, чтобы улучшить способы продажи и покупки домов.

Обычно люди продают свои дома примерно раз в семь лет и делают это с помощью профессиональных риелторов и современных данных. Но есть проблема как с системой риелторов, так и с качеством данных. RealDirect ориентирована и на тех и на других.

Итак, риелторы. Они, как правило, «свободные агенты», действующие самостоятельно; воспринимайте их как консультантов по продажам недвижимости. Это значит, что они защищают свои данные агрессивно, а действительно хорошие риелторы обладают большим опытом. Но на самом деле опытный риелтор отличается от неопытного лишь тем, что у него чуть больше данных.

RealDirect решает эту проблему, нанимая команду лицензированных агентов по недвижимости, которые работают вместе и объединяют свои знания. Чтобы этого достичь, компания создала интерфейс для продавцов, предоставив им полезные советы о том, как продать дом. Она также использует данные взаимодействия, чтобы дать рекомендации в реальном времени о дальнейших действиях.

Риелторы, работающие в подобной группе, также становятся экспертами по данным, обучаются использовать инструменты сбора информации, чтобы следить за новыми и релевантными данными или получать доступ к общедоступным сведениям. Например, теперь вы можете получить данные о продаже кооператива (определенный вид квартиры в Нью-Йорке), но это относительно недавнее изменение.

Одна из проблем с общедоступными данными заключается в том, что это старые новости — существует трехмесячная задержка между моментом продажи и тем, когда данные о продаже становятся доступны. RealDirect работает в режиме реального времени на следующих потоках информации: когда люди начинают искать дом; каково первоначальное предложение; время между предложением и закрытием сделки; как люди ищут дом через Интернет.

В конце концов, хорошая информация помогает и продавцу, и покупателю. По крайней мере, если они честны.

Как RealDirect зарабатывает деньги

Во-первых, компания предлагает продавцам подписку для доступа к инструментам продажи — около 395 долларов в месяц. Во-вторых, сайт позволяет продавцам

использовать агентов RealDirect по сниженной комиссии, обычно 2 % от продажи вместо обычных 2,5 или 3 %. Здесь и случается волшебство объединения данных: оно позволяет RealDirect брать меньшую комиссию, поскольку их сайт более оптимизирован и, следовательно, получает больше объема.

Сам сайт лучше всего рассматривать как платформу для покупателей и продавцов для управления процессом продажи или покупки. На сайте реализована система статусов для каждого человека: активные; сделано предложение; предложение отклонено; демонстрация; в процессе подписания договора и т. д. На основе вашего статуса программное обеспечение предлагает различные действия.

Конечно, есть и проблемы, с которыми RealDirect также приходится иметь дело. В Нью-Йорке есть закон, гласящий, что вы не можете показывать все текущие списки жилья незарегистрированным посетителям сайта, поэтому RealDirect требует регистрации. С одной стороны, это служит препятствием для покупателей, но люди, настроенные серьезно, скорее всего, захотят его устранить. Более того, площадки, которые не требуют регистрации, например Zillow (https://www.zillow.com/), не являются настоящими конкурентами RealDirect, поскольку просто показывают списки без предоставления каких-либо дополнительных услуг. Дуг отметил, что также необходимо зарегистрироваться для использования Pinterest (https://www.pinterest.com/), однако, несмотря на это, у данного сервиса много пользователей.

Команда RealDirect состоит из лицензированных риелторов из различных зарегистрированных риелторских ассоциаций, но даже несмотря на это, компания получила свою долю ненависти риелторов, не понимающих подхода RealDirect к сокращению комиссионных расходов. В этом смысле компания вломилась непосредственно в гильдию. С другой стороны, если риелтор отказался показывать дома, потому что они продаются на RealDirect, то потенциальные покупатели могут увидеть эти же предложения где-то в другом месте и пожаловаться. Таким образом, традиционным риелторам ничего не остается, кроме как сотрудничать с RealDirect, даже если им это не нравится. Другими словами, сами списки предложений достаточно прозрачны, так что традиционные риелторы не могут держать своих покупателей вдали от этих домов.

Дуг рассказал о ключевых моментах, которые заботят покупателя: находящиеся по соседству парки, метро и школы, а также сравнение цен за квадратный фут квартир, продаваемых в здании или жилом комплексе. Это тот тип данных, который они хотят еще больше охватывать как часть сервиса RealDirect.

Упражнение. Стратегия по данным RealDirect

Вы были наняты в качестве главного научного сотрудника по данным на https://www.realdirect.com/ и подотчетны непосредственно генеральному директору. Компания (предположительно) еще не имеет своего плана данных. Она хочет, чтобы вы

разработали стратегию. Вот несколько способов, с помощью которых это можно выполнить.

1. Исследуйте существующий сайт, поразмыслите о том, как покупатели и продавцы ориентируются на нем и как он структурирован/организован. Попытайтесь понять существующую бизнес-модель и подумайте о том, каким образом анализ данных поведения пользователя RealDirect может служить для информирования о принятии решений и разработке продукта. Придумайте список вопросов для исследования, на которые, по вашему мнению, могут ответить данные.
   - Какие данные вы посоветовали бы протоколировать инженерам и как бы выглядели ваши идеальные наборы данных?
   - Как эти данные будут использоваться для информирования и мониторинга применения продукта?
   - Как данные будут встроены обратно в продукт/сайт?
2. Поскольку пока еще нет данных для анализа (что типично для стартапа, когда компания все еще строит свой продукт), то вы должны получить некоторые вспомогательные данные, чтобы немного сориентироваться в ситуации на этом рынке. Например, перейдите на страницу https://github.com/oreillymedia/doing_data_science и найдите ссылку Rolling Sales Update.

   Здесь вы можете использовать любой набор данных или все. Начните с Манхэттена, август-2012 — август-2013.
   - Первая задача: загрузить и очистить данные. Затем выполните разведочный анализ, чтобы узнать, где имеются выбросы или отсутствуют значения, решите, как вы будете их обрабатывать, убедитесь, что даты отформатированы правильно, значения, которые вы считаете числовыми, рассматриваются как таковые и т. д.
   - После того как данные будут приведены в хорошее состояние, выполните анализ разведочных данных для визуализации и сопоставления по жилым массивам и по времени. Если у вас есть время, то начните искать осмысленные закономерности в этом наборе.
3. Соберите выводы в небольшой доклад для генерального директора.
4. Работа исследователя данных часто связана с людьми, которые не являются такими же специалистами, поэтому было бы идеально иметь набор коммуникационных стратегий для получения необходимой информации о данных. Можете ли вы представить других людей, с которыми вам следует поговорить?
5. Большинство из вас — не «отраслевые эксперты» в недвижимости и онлайн-бизнесе.
   - Дают ли вам выход из зоны комфорта, попытки представить ваше общение с людьми и «сбор данных» в новой для вас среде представление о том, как вы станете делать это в своей сфере?

- Иногда у «отраслевых экспертов» есть свой лексикон. Использовал ли Дуг жаргон, специфичный для его отрасли, который вы не поняли (comps, open houses, CPC)? Иногда непонимание словарного запаса, используемого экспертом, может помешать постичь проблему. Задавать вопросы — хорошая привычка, поскольку в конце концов вы усвоите то, что не понимаете. Это требует настойчивости и является привычкой, которую нужно развивать.

6. Дуг упомянул, что его компания не обязательно имела стратегию данных. Для создания такой стратегии нет отраслевого стандарта. По мере выполнения этого задания подумайте о том, существует ли набор рекомендаций по разработке стратегии данных для онлайн-бизнеса или вашей отрасли, которые вы могли бы дать.

Пример кода на языке R

Далее мы приводим пример кода на языке R, который берет данные о жилье в Бруклине из предыдущего упражнения, немного очищает их и исследует. (В упражнении мы просим вас сделать то же самое и для Манхэттена.)

```
# Автор: Бенжамин Редди (Benjamin Reddy)

library(plyr)

require(gdata)
bk <- read.xls("rollingsales_brooklyn.xls",pattern="BOROUGH")
head(bk)
summary(bk)

bk$SALE.PRICE.N <- as.numeric(gsub("[^[:digit:]]","", bk$SALE.PRICE))
count(is.na(bk$SALE.PRICE.N))

names(bk) <- tolower(names(bk))

## очистить/отформатировать данные с помощью регулярных выражений
bk$gross.sqft <- as.numeric(gsub("[^[:digit:]]","", bk$gross.square.feet))
bk$land.sqft <- as.numeric(gsub("[^[:digit:]]","", bk$land.square.feet))
bk$sale.date <- as.Date(bk$sale.date)
bk$year.built <- as.numeric(as.character(bk$year.built))

## провести небольшое исследование с целью убедиться,
## что с ценами не происходит ничего странного
attach(bk)

hist(sale.price.n)
hist(sale.price.n[sale.price.n>0])
hist(gross.sqft[sale.price.n==0])

detach(bk)

## сохранить только реальные продажи
```

```
bk.sale <- bk[bk$sale.price.n!=0,]

plot(bk.sale$gross.sqft,bk.sale$sale.price.n)
plot(log(bk.sale$gross.sqft),log(bk.sale$sale.price.n))

## сейчас взглянем на дома на 1, 2 и 3 семьи
bk.homes <- bk.sale[which(grepl("FAMILY",
bk.sale$building.class.category)),]
plot(log(bk.homes$gross.sqft),log(bk.homes$sale.price.n))

bk.homes[which(bk.homes$sale.price.n<100000),]
[order(bk.homes[which(bk.homes$sale.price.n<100000),]
$sale.price.n),]

## удалить выбросы, непохожие на действительные продажи
bk.homes$outliers <- (log(bk.homes$sale.price.n) <=5) + 0
bk.homes <- bk.homes[which(bk.homes$outliers==0),]

plot(log(bk.homes$gross.sqft),log(bk.homes$sale.price.n))
```

3 Алгоритмы

В предыдущей главе мы обсудили использование моделей в науке о данных. В текущей главе мы погрузимся в алгоритмы.

Алгоритм — процедура либо набор шагов или правил для выполнения задания. Алгоритмы — одно из фундаментальных понятий (иначе строительных блоков) информатики: основа конструкции элегантного и эффективного кода, подготовки и обработки данных, а также разработки программного обеспечения.

Алгоритмы позволяют решить такие базовые задачи, как сортировка, поиск и вычисления на графах. Несмотря на то что задание, подобное сортировке объектов, можно решить несколькими возможными алгоритмами, существует понятие «наилучшего» алгоритма, измеряемого эффективностью и временем вычисления. Эти два аспекта особенно важны при работе с огромными объемами данных и создании пользовательских продуктов.

Эффективные алгоритмы, работающие последовательно или параллельно, — это основа каналов для обработки и подготовки данных. Касательно науки о данных существует как минимум три класса алгоритмов, о которых следует знать.

1. Алгоритмы очистки, подготовки и обработки данных, такие как сортировка, MapReduce и Pregel.

   Мы бы охарактеризовали эти типы алгоритмов как проектирование данных, но, хотя посвящаем этому целую главу, сама тема не является основной для нашей книги. Мы не хотим сказать, что не займемся выпасом и очисткой данных: мы просто не делаем упор на алгоритмический аспект этих действий.
2. Оптимизация алгоритмов для оценки параметров, в том числе стохастический градиентный спуск, метод Ньютона и наименьшие квадраты. Мы упоминаем эти алгоритмы по всему тексту книги, они также лежат в основе многих функций языка R.
3. Алгоритмы машинного обучения составляют существенную часть этой книги. Далее мы обсудим их более подробно.

Алгоритмы машинного обучения

Алгоритмы машинного обучения в основном используются для прогнозирования, классификации и кластеризации.

Погодите! Разве в предыдущей главе мы не говорили, что для прогнозирования и классификации может использоваться моделирование? Это правда. И между ними есть несколько различий, которые только все запутывают, причем желательно понимать, в чем именно они состоят.

Статистическое *моделирование* пришло с факультетов статистики, тогда как *алгоритмы* машинного обучения — с факультетов информатики. Некоторые методы считаются частью обоих направлений — и вы увидите, что зачастую эти слова определенным образом заменяют друг друга.

Некоторые методы из нашей книги, такие как линейная регрессия, вы найдете в книгах по машинному обучению и во введениях в книги по статистике. Спор о том, кто является правомочным обладателем этих методов, не обязательно полезен, однако стоит заметить, что действительные различия становятся несколько размытыми и неточными.

В общем и целом алгоритмы машинного обучения — базис искусственного интеллекта и используются, например, для распознавания изображений/речи, в рекомендательных системах, для ранжирования и персонализации контента (что зачастую лежит в основе результатов обработки данных). Но эти алгоритмы, как правило, не изучаются в рамках программы по основам статистики или на факультете статистики. Как правило, они не подразумевают лежащий в основе *генеративный процесс* (например, моделирование чего-либо), но скорее предназначены для прогнозирования или классификации чего-либо с высокой точностью.

Эти различия в методах отражаются в культурных различиях в подходах тех, кто занимается обучением машин и статистиков, которые Рэйчел наблюдала в Google и на отраслевых конференциях. Конечно, исследователи данных могут и должны использовать оба подхода.

Следует обратить внимание на несколько очень широких обобщений.

- *Интерпретация параметров.* По восприятию статистиков, параметры в своей линейной регрессии имеют интерпретации в реальном мире и, как правило, хотят иметь возможность найти смысл в поведении или описать феномен из реального мира в соответствии с этими параметрами. В то же время программные инженеры или специалисты в области компьютерных наук могут иметь потребность превратить алгоритм с линейной регрессией в программный код промышленного уровня. Для последних прогнозная модель известна как алгоритм типа «черный ящик»: они не концентрируются на интерпретации параметров, а если и делают это, то для того, чтобы вручную настроить эти алгоритмы для оптимизации *прогностической силы*.

- *Доверительные интервалы*. Статистики предоставляют доверительные интервалы и апостериорные распределения для параметров и статистических оценок, будучи заинтересованными в выявлении изменчивости или неопределенности параметров. Во многих алгоритмах машинного обучения, таких как метод k-средних или k-ближайших соседей (о которых мы поговорим чуть ниже в этой главе), не используются понятия «доверительные интервалы» или «неопределенность».
- *Роль эксплицитных допущений*. Статистические модели делают эксплицитные допущения о процессах генерации данных и распределении, а вы используете эти данные для оценки параметров. Непараметрические решения, как те, что мы увидим далее в этой главе, не подразумевают никаких допущений по поводу распределения вероятностей, то есть являются имплицитными.

Следующее мы говорим с любовью и уважением: статистики сделали выбор посвятить свою жизнь изучению неопределенности и никогда ни в чем не уверены на 100 %. Инженеры по программному обеспечению любят создавать. Они хотят разрабатывать модели, прогнозирующие так хорошо, насколько это возможно, но при этом не заботятся о неопределенности: просто проектируют! В компаниях, подобных Google и Facebook, философия такова: создавай и часто обновляй. Если что-то сломалось, то это можно починить. Исследователя данных, который сможет определить баланс между подходами из статистики и информатики, а также найти ценность в обеих формах существования, ждет успех. Исследователи данных — мультикультурный гибрид статистика и специалиста в области компьютерных наук, поэтому мы не привязаны ни к одному образу мышления, так как оба образа имеют свою ценность. Мы подведем итог часто цитируемым твитом нашего приглашенного докладчика Джоша Виллса (Josh Wills):

> «Исследователь данных (существительное): человек, лучше разбирающийся в статистике, чем любой инженер по программному обеспечению, и лучше разбирающийся в программном инжиниринге, чем любой статистик».

Три основных алгоритма

Многие проблемы из бизнеса или реального мира, которые можно решить с помощью данных, вполне относятся к проблемам *классификации* и *прогнозирования*, если выразить их математически. К счастью, для классификации и прогнозирования подходит целый диапазон моделей и алгоритмов.

После изучения применения моделей и алгоритмов реальная сложность для вас как исследователя данных будет состоять в понимании того, какие модели и алгоритмы следует задействовать в зависимости от контекста проблемы и основополагающих допущений. Частично это приходит с опытом: увидев достаточное количество проблем, вы начинаете понимать: «А, это проблема классификации с бинарным

результатом» или «Это проблема классификации, но, что странно, у меня нет никаких меток» — и вы знаете, что делать. (В первом случае вы можете использовать логистическую регрессию или наивный классификатор Байеса, а во втором могли бы начать с метода *k*-средних — вскоре мы все это обсудим подробнее!)

Однако изначально, когда вы слышите о таких методах изолированно, вам как студенту или изучающему достаточно сложно представить, откуда вы можете знать в реальном мире, что тот или иной алгоритм — решение поставленной перед вами задачи.

Быть человеком с молотком, слоняющимся в поисках гвоздя, который можно забить, — это ошибка; все равно что сказать: «Я знаю линейную регрессию и буду стараться решить этим способом все попадающиеся мне задачи». Не поступайте так. Вместо этого попытайтесь понять контекст задачи, а также ее атрибуты *как проблемы*. Подумайте об этих математических терминах, а затем об известных вам алгоритмах и о том, как они соотносятся с задачей данного типа.

Если вы не уверены, то обсудите это с кем-то, кто уверен. Спросите коллегу, сходите на собрание или организуйте встречу в своем отделе! Кроме того, относитесь к решению задачи как к чему-то *неочевидному*, делающему задачу задачей, благодаря чему подойдете к решению обдуманно и методично. Вам не нужно быть всезнайкой, который заявляет: «*Очевидно*, нам следует использовать линейную регрессию со штрафной функцией для регуляризации», даже если это кажется правильным подходом.

Мы говорим все это из-за нежелательных особенностей книг, в которых зачастую дается набор техник и задач и говорится, *какой* метод необходимо использовать для их решения (например, задействуйте линейную регрессию для прогнозирования роста по весу). Да, первые несколько раз применение и понимание линейной регрессии не является чем-то очевидным, так что нужно попрактиковаться в этом, но, набив руку в использовании данной техники, вы должны решить реально сложную задачу: *осознать, когда применять линейную регрессию в первую очередь*.

Мы не дадим вам всеобъемлющий обзор *всех* возможных алгоритмов машинного обучения, поскольку подобный подход сделал бы эту книгу книгой по машинному обучению, а таких множество.

Сказав это, три базовых алгоритма мы представим в этой главе, а другие — по тексту книги в контексте. К концу книги вы должны быть более уверены в вашей способности изучать новые алгоритмы, чтобы иметь возможность воспринимать их по мере необходимости для решения задач.

Мы также приложим максимум усилий, чтобы продемонстрировать мыслительный процесс исследователей данных, которым пришлось решить, какой алгоритм использовать в заданном контексте и почему, но ваша задача как студента и изучающего — *заставить себя* думать о том, какие атрибуты задачи сделали выбор указанного алгоритма правильным решением.

С учетом вышесказанного нам все же требуется представить несколько базовых инструментов для пользования, поэтому начнем с линейной регрессии, *k*-ближайших соседей (*k*-БС) и *k*-средних. Вдобавок к тому, что мы сказали о попытках понять атрибуты задач, для решения которых могут использоваться указанные решения, смотрите на эти три алгоритма следующим образом: какие закономерности мы как люди можем увидеть в данных, которые нам хотелось бы автоматизировать с помощью машины, особенно учитывая тот факт, что по мере усложнения данных мы теряем возможность видеть закономерности?

Линейная регрессия

Линейная регрессия — один из наиболее часто применяемых статистических методов. В самом простом случае она применяется, если необходимо выразить математическое отношение между двумя переменными или атрибутами. При использовании этого метода вы допускаете существование *линейного* отношения между выходной переменной (иногда еще именуемой переменной отклика, зависимой переменной или меткой) и показателем (предиктором, predictor, изредка также называемым независимой переменной, каузальной переменной или характеристикой) либо между одной переменной и несколькими. В этих случаях вы *моделируете* отношение как имеющее линейную структуру.

ТАК ЭТО АЛГОРИТМ ИЛИ МОДЕЛЬ?

Мы попытались провести различие между двумя понятиями ранее, но вынуждены признать, что разговорное употребление слов «модель» и «алгоритм» только запутывает, поскольку кажется, что оба слова употребляются как взаимозаменяемые, в то время как их определения различаются. В попытках найти здравый смысл скажем, что алгоритм — набор правил или шагов, которые необходимо сделать для выполнения задания, а модель — попытка изобразить мир. Эти понятия кажутся очевидно различными, потому разница между ними должна быть хорошо заметна. К сожалению, это не так. Например, регрессию можно описать и как статистическую модель, и как алгоритм машинного обучения. Пытаться заставить людей говорить об этих двух понятиях с какой-либо степенью точности — впустую тратить время.

В каком-то смысле это исторический артефакт того времени, когда статистики и сообщества компьютерных наук создавали методы и техники параллельно, но задействовали при этом разные слова для одних и тех же методов. Результатом выступает нечеткая граница между машинным обучением и статистическим моделированием. Некоторые методы (например, обсуждаемый в следующем подразделе метод *k*-средних) мы можем называть *алгоритмами*, поскольку они представляют собой последовательность шагов вычисления, используемых для кластеризации и классификации объектов. С другой стороны, *k*-средние можно интерпретировать как особый случай гауссовой *модели* распределения. Конечный результат заключается в следующем: в речи люди употребляют термины «алгоритм» и «модель» как взаимозаменяемые, когда речь заходит о большинстве этих методов, так что попытайтесь не переживать по данному поводу. (Хотя нас он тоже беспокоит.)

Предполагать наличие *линейной* зависимости между выходной переменной и показателем — большое допущение, но вместе с тем и наиболее простое допущение из тех, что вы *можете* сделать: линейные функции проще нелинейных в математическом смысле, ввиду чего в данном случае это хорошая точка для старта.

В некоторых случаях предположение, что изменения в одной переменной имеют линейную корреляцию с изменениями в другой, имеет смысл. Например, вполне разумно предположить, что, продав больше зонтов, вы заработаете больше денег. В таких случаях вы можете не переживать за любое линейное допущение. В других же случаях сложнее оправдать допущение линейности, за исключением локальных случаев: по принципам математического анализа все может быть упрощено по сегментам прямой до тех пор, пока функции непрерывны.

Отмотаем назад. Зачем бы вообще потребовалось создать линейную модель? Вам может понадобиться использовать такое отношение для *прогнозирования* будущих результатов либо если нужно понять или *описать* отношения, чтобы понять ситуацию. Представьте, что изучаете отношение между уровнем продаж некой компании и тем, сколько эта компания тратит на рекламу, или количеством друзей в социальной сети и количеством времени, который человек проводит на ее сайте ежедневно. Все результаты числовые, так что линейная регрессия будет разумным выбором как минимум для первого подхода к решению данной задачи.

Одна из стартовых точек, с которой можно начать обдумывание линейной регрессии, — сначала поразмыслить о детерминированных линиях. В школе мы изучали, что линию можно описать с помощью угла наклона и точки пересечения с осью координат, $y = f(x) = \beta_0 + \beta_1 \times x$. Но установка здесь всегда была детерминированной (однозначной).

Даже для самых искушенных в математике анализ стохастических функций будет новым мироощущением, особенно если вы не делали этого раньше. У нас по-прежнему те же самые компоненты: точки, перечисленные эксплицитно в таблице (или в виде пар чисел), и функции, представленные в виде уравнений или нанесенные на график. Поэтому дойдем до линейной регрессии, начав с детерминированной функции.

Пример 1. Для начала простейший пример

Предположим, вы владелец сайта социальной сети, который взимает плату за подписку в размере 25 долларов в месяц, и это ваш единственный источник дохода. Каждый месяц вы собираете данные и вычисляете общее количество пользователей и свой общий доход. На протяжении последних двух лет вы делали это ежедневно и заносили полученные данные в электронную таблицу. Вы могли бы выразить информацию в виде последовательности точек. Ниже приведены четыре первые точки:

$$S = \{(x, y) = (1, 25),(10, 250),(100, 2500),(200, 5000)\}.$$

Если бы вы показали это кому-то, кто не знает, сколько вы берете с подписчиков, или ничего не знает о вашей бизнес-модели (что это за друг, который ничего не знает о вашей бизнес-модели?!), то, возможно, данный человек заметил бы четкую взаимосвязь между всеми точками, а именно — $y = 25x$. Вероятнее всего, он сделал это в уме, в подобном случае он также выяснил, что:

- есть линейная закономерность;
- коэффициент, связывающий x и y, равен 25;
- функция кажется детерминированной.

Вы даже можете нанести эту функцию на график, как показано на рис. 3.1, чтобы убедиться в правоте данного человека (несмотря на то что знали о его правоте, так как это вы создали бизнес-модель). Это прямая!

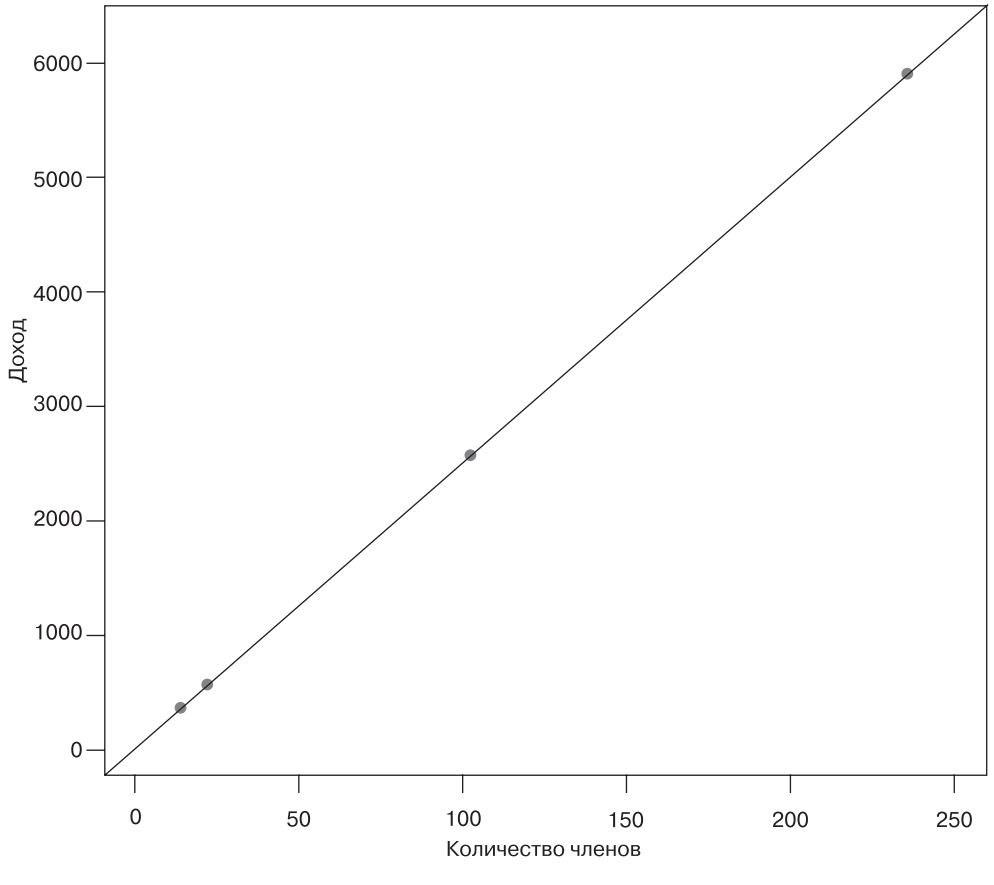


Рис. 3.1. Очевидная линейная закономерность

Пример 2. Смотрим на данные на уровне пользователя

Предположим, у вас есть набор данных, где *ключом* выступает пользователь (это значит, что каждая строка содержит информацию об одном пользователе), а столбцы представляют поведение этого человека на сайте социальной сети на протяжении недели. Предположим, вам удобно, что на текущем этапе данные чисты и у вас упорядочены сотни тысяч пользователей. У столбцов следующие имена: `total_num_friends`, `total_new_friends_this_week`, `num_visits`, `time_spent`, `number_apps_downloaded`, `number_ads_shown`, `gender`, `age` и т. д. Во время исследовательского анализа вы случайным образом выбрали 100 пользователей, чтобы не усложнять анализ, и нанесли на график пары этих переменных, например x = `total_new_friends`, а y = `time_spent` (в секундах). Может сложиться так, что на определенном этапе вы захотите пообещать рекламодателям, которые зарезервировали рекламное место на вашем сайте, конкретное количество пользователей, поэтому вам нужно иметь возможность спрогнозировать количество последних, которое будет через несколько дней или недель. Но сейчас вы просто пытаетесь натренировать интуицию и понять ваш набор данных.

Вы смотрите на первые несколько строк и видите:

```
7               276
3               43
4               82
6               136
10              417
9               269
```

Сейчас ваш мозг не может понять, что происходит, просто взглянув на цифры (мозг вашего друга, вероятно, также не способен это сделать). Цифры не находятся в каком-то очевидном конкретном порядке, кроме того, их много. Поэтому вы пытаетесь нанести их на график, как показано на рис. 3.2.

Похоже, здесь есть *что-то наподобие* линейной последовательности, и это логично: чем больше у вас новых друзей, тем больше времени вы проводите на сайте. Но как понять, с помощью чего описать это отношение? Обратим также внимание на отсутствие идеально *детерминированного* отношения между количеством новых друзей и временем, проведенным на сайте, но вполне логично, что между этими двумя переменными есть определенная *взаимосвязь*.

Начните с записывания

В модели требуется отразить два аспекта. Первый — это *тренд*, а второй — *отклонение*. Начнем с тренда.

Для начала начнем с предположения о том, что *имеется* некое отношение и оно является линейным. Это лучшее, что можно сделать на данном этапе.

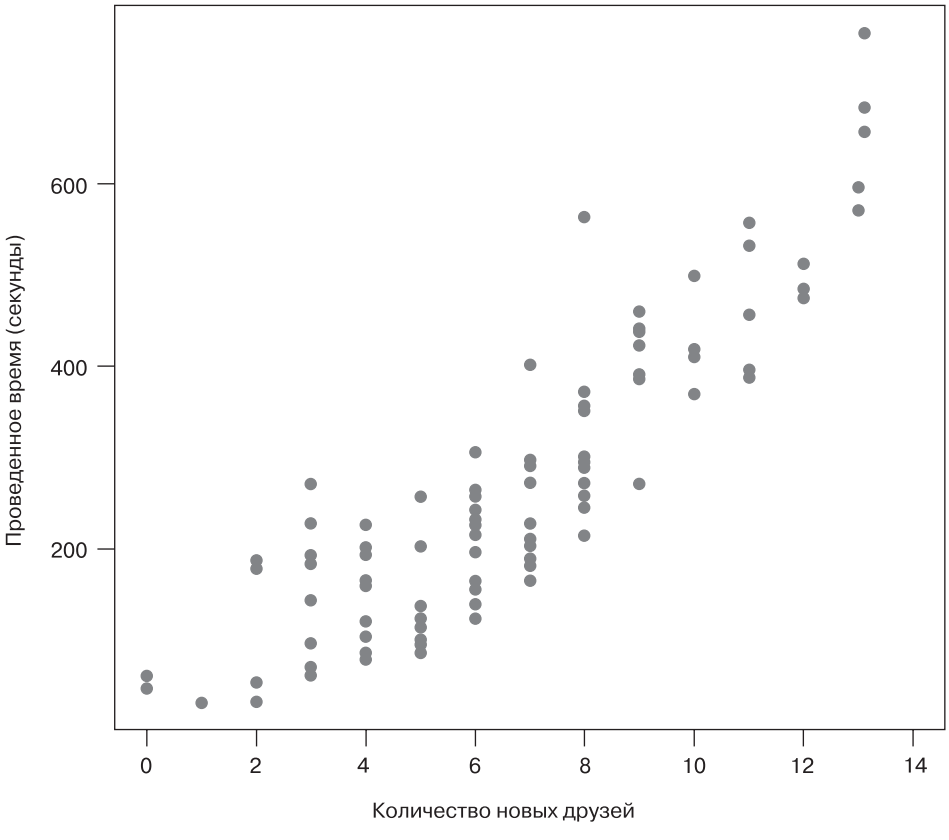


Рис. 3.2. Похоже на линейность

На рис. 3.3 множество на первый взгляд вполне подходящих прямых.

Итак, как же выбрать наиболее подходящую прямую?

Поскольку вы допускаете наличие линейного отношения, то начните создание модели с допущения следующей формы функции:

$$y = \beta_0 + \beta_1 x.$$

Теперь ваша задача — найти лучшие значения β_0 и β_1, используя при этом полученные в ходе наблюдения данные: (x_1, y_1), (x_2, y_2) ... (x_n, y_n).

В виде матрицы это будет выглядеть следующим образом:

$$y = x \times \beta.$$

Вот и все: вы записали модель, теперь осталось ее *обучить*.

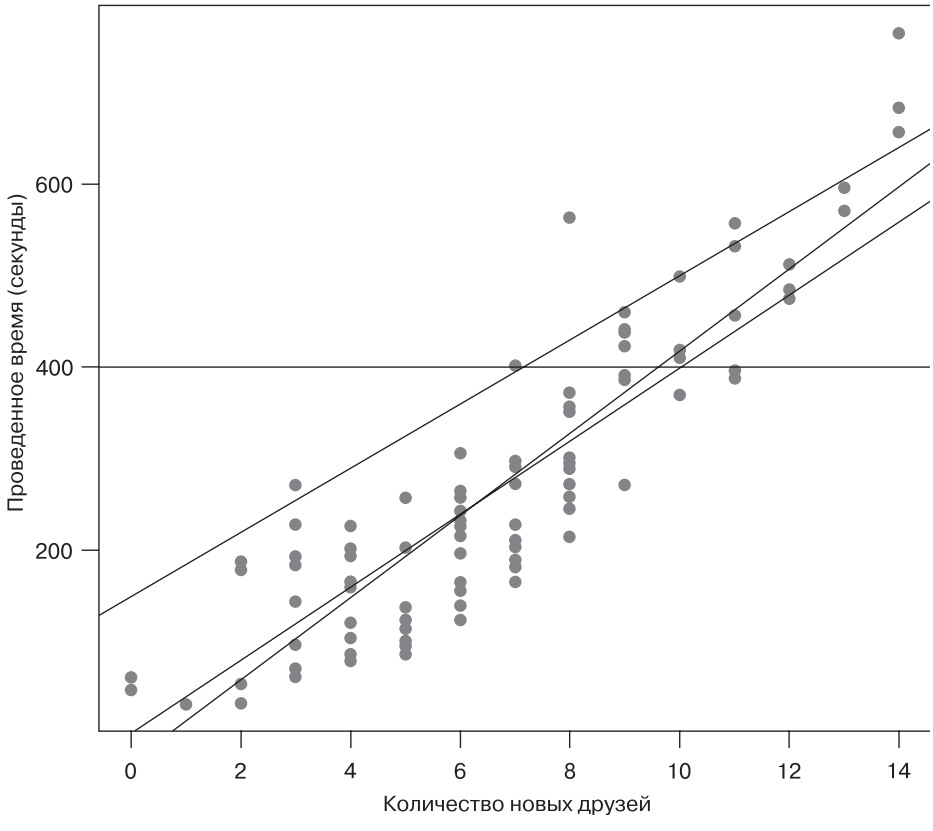


Рис. 3.3. Какая прямая подходит лучше?

Обучение модели

Как же вычислить β? Интуиция, лежащая в основе линейной регрессии, говорит, что нужно найти прямую, минимизирующую расстояние между всеми точками и самой прямой.

Многие прямые на первый взгляд выглядят правильными, но ваша цель — найти самую оптимальную. Слово «оптимальный» может иметь много разных значений, но начнем с того, что «оптимальная прямая» — это такая прямая, которая в среднем наименее удалена от всех точек. Но что значит *наименее удалена*?

Взгляните на рис. 3.4. Линейная регрессия пытается найти такую прямую, которая минимизировала бы сумму квадратов расстояний по вертикали между приближенным или спрогнозированным значением $\hat{y}_i s$ и наблюдаемым значением $y_i s$. Вы делаете это для минимизации погрешностей в прогнозах. Данный подход называется *оценкой методом наименьших квадратов*.

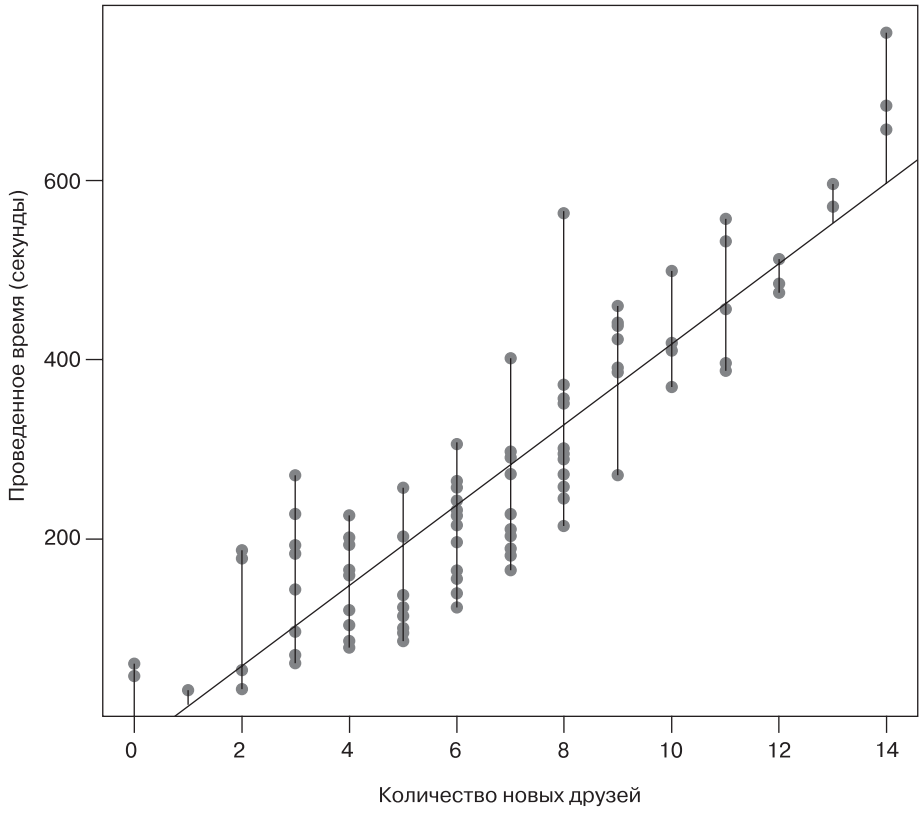


Рис. 3.4. Прямая, наименее удаленная от всех точек

Для того чтобы найти такую прямую, вам потребуется следующим образом определить остаточную сумму квадратов (RSS), обозначенную как $RSS(\beta)$:

$$RSS(\beta) = \sum_i (y_i - \beta x_i)^2,$$

где i принимает значение различных точек на графике. Это сумма всех квадратов расстояний по вертикали между наблюдаемыми точками и любой заданной прямой. Обратите внимание, что перед нами функция β — и для поиска оптимальной прямой нужно выполнить оптимизацию относительно этой функции.

Для минимизации $RSS(\beta) = (y - \beta x)^t(y - \beta x)$ выполните дифференциацию относительно β, приравняйте к нулю, затем решите для β. В результате вы получите:

$$\hat{\beta} = (x^t x)^{-1} x^t y.$$

Небольшой символ «шапка» над переменной β указывает на то, что перед вами *оценщик* для этой переменной. Вы не знаете истинное значение β, все, что у вас

есть, — полученные в результате наблюдения данные, которые вы подключаете к оценщику для получения оцененного значения.

Для обучения этого выражения и получения значений β потребуется единственная строка кода на языке R, где у вас есть столбец значений y и (единственный) столбец значений x:

```
model <- lm(y ~ x)
```

Для примера, где первые строки данных выглядели вот так:

```
x             y
7             276
3             43
4             82
6             136
10            417
9             269
```

код на языке R будет следующим:

```
> model <- lm (y~x)
> model

Call:
lm(formula = y ~ x)

Coefficients:
(Intercept)        x
-32.08          45.92

> coefs <- coef(model)
> plot(x, y, pch=20,col="red", xlab="Number new friends",
  ylab="Time spent (seconds)")
> abline(coefs[1],coefs[2])
```

Расчетная прямая: $\hat{y} = -32{,}08 + 45{,}92x$, которую вы вполне можете округлить до $\hat{y} = -32 + 46x$. Соответствующий график показан в левой части рис. 3.5.

Но от вас как исследователя данных зависит, станете ли вы на самом деле применять эту линейную модель для описания отношений и прогнозирования результатов. При появлении нового значения 5, означающего, что у пользователя пять новых друзей, насколько вы будете уверены, что выходящее значение составит $-32{,}08 + 45{,}92 \times 5 = 195{,}7$ секунды?

Чтобы с уверенностью ответить на этот вопрос, вам нужно расширить модель. Вы знаете о различии во времени, проведенном пользователями с пятью новыми друзьями на сайте, то есть точно не будете утверждать, будто все пользователи с пятью новыми друзьями гарантированно проводят на сайте 195,7 секунды. Таким образом, к настоящему моменту вы смоделировали *тренд*, но еще не *вариативность*.

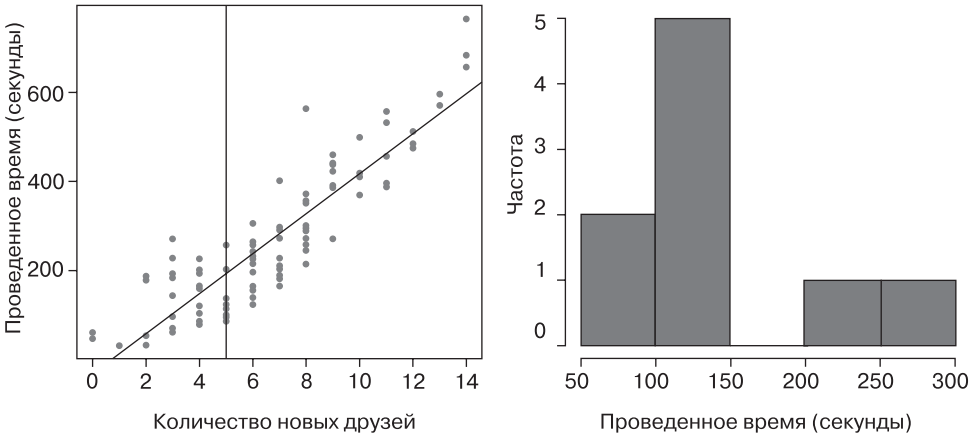


Рис. 3.5. Слева — обученная прямая. Мы видим, что для любого фиксированного значения, например 5, значения переменной y могут отличаться. Для людей, имеющих пять новых друзей, на графике справа мы показываем время, которое они провели на сайте

Расширение вне метода наименьших квадратов

Создав *модель простой линейной регрессии* (один результат, один показатель), в которой для оценки значений переменной β используется оценка методом наименьших квадратов, вы можете выполнить надстройку на эту модель тремя основными путями, описанными ниже:

- добавлением в модель допущений погрешностей;
- добавлением показателей;
- трансформацией показателей.

Добавление в модель допущений погрешностей. Если вы используете модель для прогнозирования значения y для заданного значения x, то ваш прогноз детерминистический и не отражает изменчивость наблюдаемых данных. В правой части рис. 3.5 вы можете видеть, что для фиксированного значения $x = 5$ наблюдается изменчивость количества времени, проведенного на сайте. Вам необходимо запечатлеть эту изменчивость в своей модели, поэтому вы расширяете свою модель следующим образом:

$$y = \beta_0 + \beta_1 x + \epsilon,$$

где новый член уравнения ϵ называется *шумом (помехой)* и представляет собой нечто еще не учтенное в уже обнаруженной зависимости. Эта переменная также называется *рассогласованием*: переменная ϵ представляет *действительную погрешность* — разницу между наблюдаемым и *истинной* прямой регрессии, что вы никогда не будете знать и сможете только оценить с помощью переменных $\hat{\beta}s$.

Зачастую в модели делается допущение, что шум распределен нормально; это записывается следующим образом:

$$\in \approx N(O, \sigma^2).$$

Обратите внимание: такое допущение не всегда резонно. Если заранее известно, что вы работаете с «толстохвостым» распределением, а ваша линейная модель захватывает только небольшую часть значений переменной y, то в подобном случае рассогласование, скорее всего, также является «толстохвостым». Это наиболее часто встречающаяся ситуация в финансовом моделировании.

Однако мы не хотим сказать, что линейная регрессия не используется в финансовом секторе. Просто в этой отрасли мы не вводим в модель допущение, что шум распределен нормально.

Учитывая предыдущее допущение, модель говорит, что для любого заданного значения x условное распределение значений y по заданному значению x выражается как $p(y \mid x) \sim N(\beta_0 + \beta_1 x, \sigma^2)$.

Таким образом, среди набора людей, имеющих пятерых новых друзей, количество проведенного ими на сайте времени характеризуется *нормальным распределением* со средним значением $\beta_0 + \beta_1 \times 5$ и отклонением σ^2, а вы оцените показатели β_0, β_1, σ, используя данные.

Как подогнать эту модель? Как получить показатели β_0, β_1, σ с помощью данных?

Оказывается, как бы ни были распределены ϵ, уже вычисленные вами оценки по методу наименьших квадратов — это оптимальные оценщики значений переменных β, поскольку обладают свойством объективности и минимальной вариативности. Если хотите получить более подробную информацию об этих свойствах и увидеть доказательство этой мысли, то рекомендуем прочесть любую хорошую книгу по статистическому анализу (например, Statistical Inference («Статистический анализ») авторства Каселлы (Casella) и Бергера (Berger)).

Что же вы можете сделать с вашими данными, полученными из наблюдения, для оценки вариативности погрешностей? Теперь, имея расчетную прямую, вы можете пронаблюдать, как далеко отклоняются от нее наблюдаемые точки, и обрабатывать эти отличия, также называемые *наблюдаемыми* или *остаточными погрешностями*, как наблюдения сами по себе или оценки действительных погрешностей, то есть ϵ.

Определим, что $e_i = y_i - \hat{y}_i = y_i - (\hat{\beta}_0 + \hat{\beta}_1 x_i)$ для $i = 1 \ldots n$.

Теперь вы можете оценить вариативность (σ^2) шума ϵ как:

$$\frac{\sum_i e_i^2}{n-2}.$$

Почему мы выполняем деление на n – 2? Вполне естественный вопрос. Деление на n – 2 вместо n создает неискаженную оценку. Число 2 соотносится с количеством параметров модели. В этом случае книга Каселлы и Бергера — отличный ресурс дополнительной информации.

Приведенная формула называется *среднеквадратичной погрешностью* и отражает, насколько предсказанное значение отличается от наблюдаемого. *Среднеквадратичная погрешность* — полезный количественный показатель для любой прогностической задачи. В частности, в случае с регрессией это *также* оценщик вариативности, однако данный показатель не всегда можно использовать или интерпретировать подобным образом. Мы увидим это далее.

Оценочные показатели

Ранее мы спрашивали, насколько уверенными вы будете в этих оценках и в своих моделях. В выводе R-функции есть два значения, которые могут показать, насколько вы можете быть уверены в своих оценках: *p*-значение и *R*-квадрат. Возвращаясь к нашей модели на языке R, если мы введем выражение **`summary(model)`** — имя, которое дали этой модели, то резюмированный вывод предстанет в следующем виде:

```
summary (model)
Call:
lm(formula = y ~ x)

Residuals:
    Min      1Q Median      3Q     Max
-121.17  -52.63  -9.72  41.54  356.27

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  -32.083     16.623   -1.93   0.0565 .
x             45.918      2.141   21.45   <2e-16 ***
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 77.47 on 98 degrees of freedom
Multiple R-squared: 0.8244,  Adjusted R-squared: 0.8226
F-statistic:  460 on 1 and 98 DF,  p-value: < 2.2e-16
```

- *R-квадрат*. $R^2 = 1 - \frac{\sum_i (y_i - \hat{y}_i)^2}{\sum_i (y_i - \bar{y})^2}$. Данное уравнение можно интерпретировать как пропорцию вариативности, объясненную нашей моделью. Обратите внимание: здесь среднеквадратичная погрешность делится на общую погрешность — пропорцию, *не объясненную* нашей моделью, но мы производим вычисления, вычитая все это из 1.
- *p-значения*. Взглянем на вывод; оценочные значения β представлены в столбце, обозначенном как `Estimate`. Чтобы увидеть *p*-значения, обратим внимание на $Pr(>|t|)$. Мы можем интерпретировать значения в этом столбце следующим

образом: создаем нуль-гипотезу о том, что переменные β равны нулю. Для любого заданного значения β *p*-значения отражают вероятность наблюдения данных и получения полученной тестовой статистики *по нуль-гипотезе*; при этом очень вероятно, что коэффициент не будет равен нулю, а следовательно, будет иметь значение.

- *Перекрестная проверка*. Еще один подход оценки модели выглядит следующим образом. Разделим наши данные на обучающий набор и тестовый набор: 80 % на обучение и 20 % на тестирование. Обучите модель на обучающем наборе, а затем посмотрим на *среднеквадратичную погрешность* на тестовом наборе и сравним это значение с полученным на обучающем наборе. Проведем это сравнение также в рамках выборки. Если среднеквадратичные погрешности примерно равны, то мы не рискуем совершить переобучение. На рис. 3.6 показано, как это может выглядеть. Мы очень рекомендуем пользоваться этим подходом.

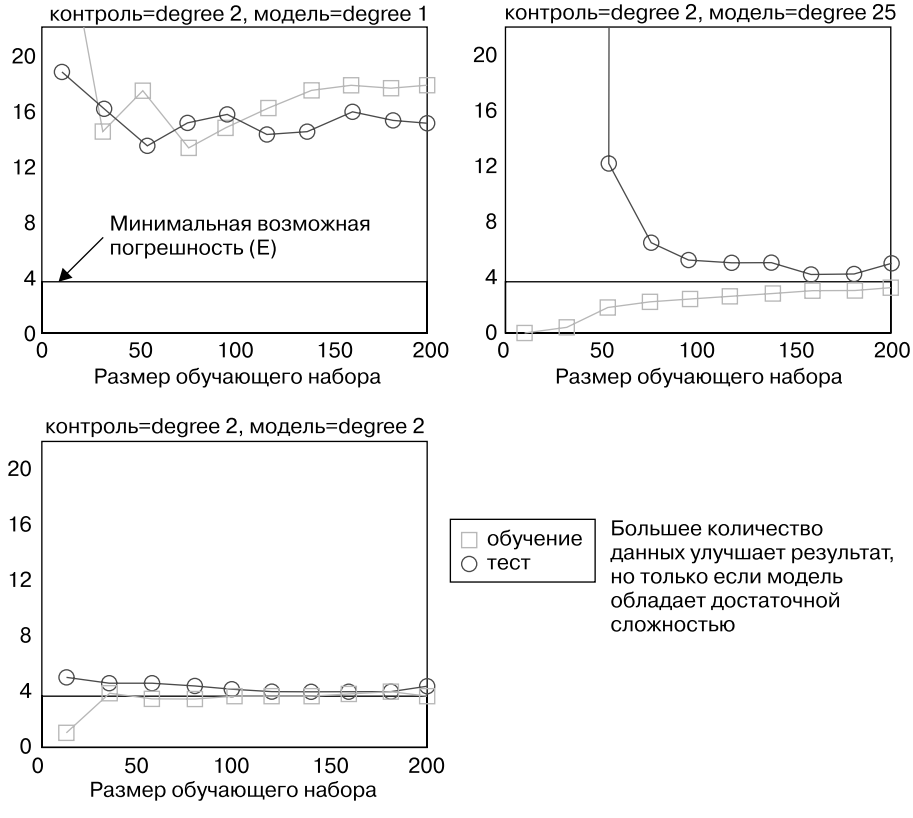


Рис. 3.6. Сравнение среднеквадратичной погрешности в обучающем и тестовом наборах, взятое со слайда профессора Нандо де Фрейтаса (Nando de Freitas). В этом случае контрольная информация известна, так как получена из набора, где данные симулированы из известного распределения

Другие модели величин погрешности

Среднеквадратичная погрешность — пример того, что принято называть *функцией потерь*. Эта разновидность — стандарт для использования в линейной регрессии, так как данная погрешность позволяет хорошо измерить точность обучения. Для среднеквадратичной погрешности есть дополнительное желательное свойство, когда мы допускаем, что значения ε распределены нормально, и можем полагаться на принцип максимального правдоподобия. Существуют также и другие функции потерь, которые полагаются на абсолютные значения, а не на квадратичные. Кроме того, есть возможность создания собственных функций потерь в зависимости от конкретной задачи или контекста, однако на данный момент вполне пригодна среднеквадратичная погрешность.

Добавление других показателей. Мы рассмотрели простую линейную регрессию: один результат или зависимая переменная и один показатель. Но мы можем расширить данную модель, добавив другие показатели, и это будет называться *множественной линейной регрессией*:

$$y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \beta_3 x_3 + \epsilon.$$

Все вычисления, произведенные нами ранее, остаются в силе, так как мы выразили их в матричном представлении, то есть оно уже было обобщено для выдачи подходящих оценок β. В примере, в котором мы прогнозировали проведенное на сайте время, мы могли бы использовать и другие показатели, например пол и возраст пользователя. Мы рассмотрим выбор признаков (то есть выяснение того, какие дополнительные показатели нужно включить в модель) в главе 7. Код на языке R должен выглядеть следующим образом:

```
model <- lm(y ~ x_1 + x_2 + x_3)
```

Или для добавления взаимодействий между переменными:

```
model <- lm(y ~ x_1 + x_2*x_3)
```

Ключевой аспект здесь — создать диаграмму рассеивания значений y относительно каждого из показателей, а также между показателями и гистограммой $y \mid x$ для различных значений каждого из показателей в целях развития интуиции. Как и в примере с линейной регрессией, вы можете использовать одинаковые методы для оценки модели, как было описано ранее: посмотрев на R^2, p-значения, использование обучающего и тестового наборов.

Трансформации. Возвращаясь к единичному x, предсказывающему единственный y, почему мы допустили наличие линейного отношения? Может быть, лучшей моделью было бы полиномиальное отношение, например такое:

$$y = \beta_0 + \beta_1 x_1 + \beta_2 x^2 + \beta_3 x^3.$$

Погодите-ка, разве это не *линейная* регрессия? При последней проверке полиномиальные отношения не были линейными. Думая об этом отношении как

о *линейном*, вы трансформируете или создаете переменные, например $z = x^2$, и создаете линейную регрессию, основанную на z. Другие часто используемые трансформации — логарифмирование или выбор предела и превращение его в бинарный показатель.

Если вы взглянете на график проведенного на сайте времени в сравнении с количеством друзей, то увидите, что его форма несколько искривлена. Вы могли бы исследовать этот феномен более детально и создать модель для проверки того, требует ли данный аспект улучшения.

То, с чем вы здесь сталкиваетесь, — одна из наиболее сложных задач для моделиста: вы никогда не знаете истины. Возможно, истинная модель квадратична, но вы допускаете линейность или наоборот. Вы прикладываете максимум усилий для оценки модели, как описано выше, но *на самом деле* никогда не знаете, правы или нет. Получение все большего количества данных может помочь и в этом.

Обзор

Еще раз взглянем на те допущения, которые мы сделали при создании и обучении модели:

- линейность;
- величины погрешностей распределены нормально со средним 0;
- величины погрешностей не зависят друг от друга;
- для величин погрешностей характерна постоянная вариативность относительно значений x;
- используемые нами показатели — *нужные*.

Когда и зачем мы выполняем линейную регрессию? Чаще всего по двум причинам:

- если хотим предсказать одну переменную, зная значения других;
- если хотим объяснить или понять отношения между двумя или несколькими сущностями.

Упражнение

Для того чтобы помочь вам понять и исследовать новые концепции, мы можем симулировать имитационные наборы данных на языке R. Преимущество такого подхода в том, что вы можете «поиграть в бога», поскольку на самом деле знаете лежащую в основе истину, и пронаблюдать, насколько хорошо работает ваша модель для выяснения этой истины.

Когда вы лучше поймете, что происходит при обработке вашего имитационного набора данных, можете перейти к работе с реальным набором данных. Здесь мы покажем, как симулировать имитационный набор данных, затем представим несколько идей о том, как это изучить более детально:

```
# Симуляция имитационных данных
x_1 <- rnorm(1000,5,7) # из нормального распределения симулировать
                       # 1000 значений со средним 5
                       # и стандартным отклонением 7
hist(x_1, col="grey")  # нанести на график p(x)
true_error <- rnorm(1000,0,2)
true_beta_0 <- 1.1
true_beta_1 <- -8.2

y <- true_beta_0 + true_beta_1*x_1 + true_error
hist(y) # нанести на график p(y)
plot(x_1,y, pch=20,col="red") # нанести на график p(x,y)
```

1. Создайте модель с регрессией и убедитесь, что она восстанавливает истинные значения β.
2. Симулируйте новую имитационную переменную x_2, для которой характерно гамма-распределение по выбранным вами параметрам. Теперь сделайте истинным утверждение, что y есть линейное сочетание переменных x_1 и x_2. Обучите модель, которая зависит только от x_1. Обучите модель, которая зависит только от x_2. Обучите модель, в которой используются обе переменных. Измените размер выборки и создайте график среднеквадратичной погрешности обучающего набора и тестового набора против размера выборки.
3. Создайте новую переменную z, равную x_1^2. Включите ее в свою модель в качестве одного из показателей. Пронаблюдайте, что произойдет, когда вы обучите модель, зависящую только от x_1 и затем также зависящую от z. Измените размер выборки и постройте график среднеквадратичной погрешности обучающего и тестового наборов против размера выборки.
4. Попрактикуйтесь еще: а) изменяя значения параметров (истинные β); б) изменяя распределение истинной погрешности; в) включая в модель дополнительные показатели с другими типами распределения вероятности. (Функция `rnorm()` генерирует случайные значения для нормального распределения, функция `rbinom()` выполняет то же самое, но для биномиального распределения. Посмотрите эти функции в Интернете и попытайтесь выяснить дополнительную информацию.)
5. Создайте диаграммы рассеивания для всех пар переменных и гистограммы для единичных переменных.

k-ближайшие соседи

k-БС — алгоритм, который может использоваться, когда перед вами набор ранее неким образом классифицированных и обозначенных объектов, а также набор ранее не классифицированных и не обозначенных объектов, и вы хотите добиться автоматического присвоения меток этим объектам.

Объектами могут выступать исследователи данных, которые были классифицированы как «привлекательные» и «непривлекательные», или люди с присвоенным

рангом «высокая кредитоспособность» либо «низкая кредитоспособность», или рестораны, обозначенные как «пять звезд», «четыре звезды», «три звезды», «две звезды», «одна звезда» либо, если жизнь не всегда потакает, «ноль звезд». В более серьезных случаях это могут быть пациенты, классифицированные как «высокий риск рака» и «низкий риск рака».

Задумайтесь на секунду, сработает ли линейная регрессия при решении проблем этого типа.

Ответ: не всегда. Когда вы используете линейную регрессию, результат — всегда непрерывная переменная. Здесь же результатом вашего алгоритма будет метка категории, поэтому линейная регрессия не решит задачу так, как данная задача была описана.

Впрочем, эту задачу можно решить и с помощью линейной регрессии с привлечением понятия «предел». Например, если вы пытаетесь предсказать очки кредитоспособности людей, исходя из их возраста и уровней дохода, а затем выбираете предел 700 (это значит, что для данного человека возраст и уровень дохода превышает 700), то пометили бы их кредитоспособность как «высокую», в противном случае отправили бы их в корзину с меткой «низкая кредитоспособность». С бóльшим количеством пределов вы могли бы задать более мелкие категории, такие как «очень низкий», «низкий», «средний», «высокий» и «очень высокий».

Чтобы сделать это, используя линейную регрессию, вам потребовалось бы установить корзины и диапазоны для непрерывного вывода. Но не все находится в непрерывной системе координат, как очки кредитоспособности. А если ваши метки гласят: «скорее всего, демократ», «скорее всего, республиканец» и «скорее всего, беспартийный»? Что вы будете делать в таком случае?

Интуиция, лежащая в основе метода *k*-БС, говорит о необходимости рассмотреть другие элементы, *наиболее похожие* по их атрибутам, внешнему виду и меткам, а затем дать неподписанному элементу большинство голосов. В случае равного исхода вы случайным образом выбираете метки, для которых наблюдался равный исход.

Например, если у вас есть определенное количество фильмов, подписанных «хорошо» или «плохо», а затем вам дают фильм «Сумасшествие данных», который еще не был оценен, то вам нужно посмотреть на атрибуты этого фильма: длину, жанр, количество сексуальных сцен, задействованных актеров — обладателей премии «Оскар» и бюджет. Далее вы должны найти другие фильмы с похожими атрибутами, посмотреть на *их* рейтинги, а затем присвоить фильму «Сумасшествие данных» рейтинг, даже не смотря его.

Для автоматизации описанного процесса вы должны принять два решения: во-первых, как вы устанавливаете *похожесть*, или подобие? Определив это понятие, для каждого неоцененного элемента вы сможете сказать, насколько похожи на него *все* подписанные элементы, затем выбрать *наиболее похожие* элементы и назвать их *соседями*, у каждого из которых есть «голос».

Это приводит вас ко второму решению: на какое количество соседей вы должны опираться и скольким «давать право голоса»? Данное значение и есть переменная *k*, которую выберете вы, будучи исследователем данных, а мы расскажем вам как.

Логично? Опробуем на более реалистичном примере.

Пример с очками кредитоспособности

Предположим, у вас есть данные о возрасте, уровне дохода группы людей и две категории кредитоспособности: высокая и низкая — и вам нужно предсказать категорию кредитоспособности нового человека, используя его возраст и уровень дохода.

Например, ниже приведены строки набора данных, доход указан в тысячах:

```
age income  credit
69      3      low
66     57      low
49     79      low
49     17      low
58     26     high
44     71     high
```

Вы можете нанести людей на график в виде точек, при этом использовать пустые точки для людей с низким кредитным рейтингом, как показано на рис. 3.7.

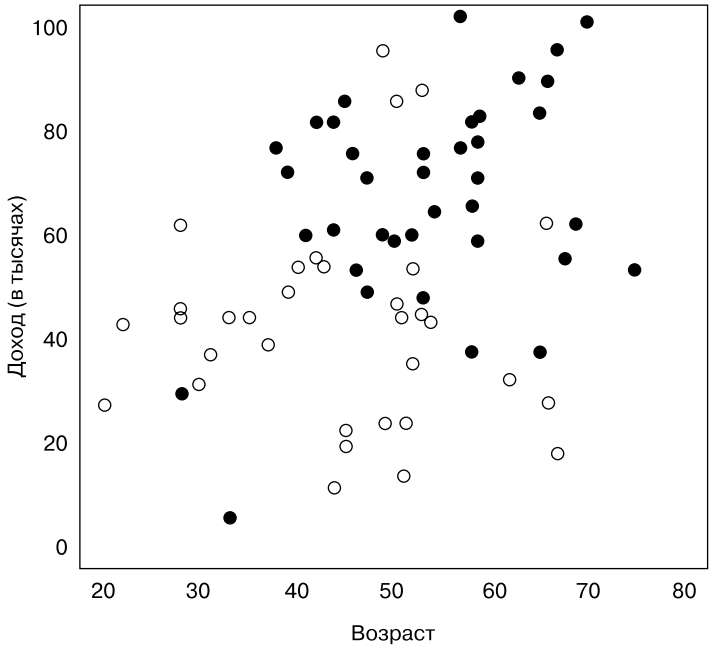


Рис. 3.7. Кредитный рейтинг как функция возраста и дохода

Что, если новому мужчине 57 лет и его доход равен 37 000 долларам? Какая, вероятнее всего, у него кредитная категория? Взгляните на рис. 3.8. Основываясь на других людях по соседству с ним, какую кредитную категорию, по вашему мнению, мы должны присвоить ему? Воспользуемся методом *k*-БС для автоматического выполнения данной задачи.

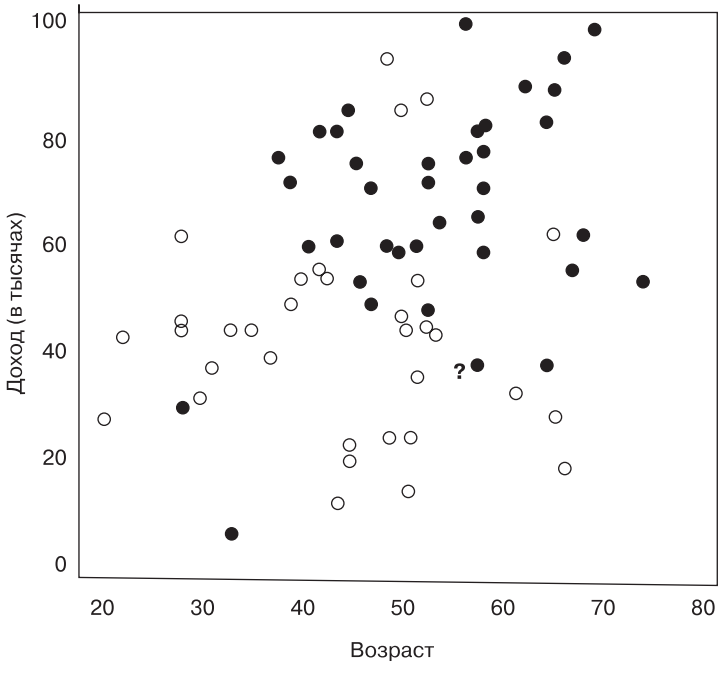


Рис. 3.8. Как насчет этого мужчины?

Ниже приведено резюме данного процесса.

1. Выберите меру *подобия* и *расстояния*.
2. Разделите исходный подписанный набор данных на обучающий и тестовый.
3. Выберите меру оценки. (Хорошая мера — уровень неправильной классификации: мы объясним ее чуть позже.)
4. Запустите *k*-БС несколько раз, изменяя значение *k* и проверяя меру оценки.
5. Оптимизируйте значение *k*, выбрав наилучшую меру оценки.
6. Выбрав значение *k*, используйте тот же самый обучающий набор и создайте новый тестовый набор с возрастом и уровнем дохода людей, для которых у вас *нет меток*, но нуждающихся в прогнозировании. В данном случае ваш тестовый набор содержит только одну строку: мужчина 57 лет.

Меры подобия и расстояния

Определения «близости» и подобия могут отличаться в зависимости от контекста: так, близость в социальных сетях можно определить как количество общих друзей.

Для целей нашей задачи и определения соседей мы можем воспользоваться евклидовым расстоянием на графике, если переменные находятся на одной шкале. Но иногда это будет сплошным большим ЕСЛИ.

ОСТОРОЖНО: ОПАСНОСТЬ МОДЕЛИРОВАНИЯ!

Вопрос масштабирования очень важен, и если вы допустите ошибку, то ваша модель может просто работать неправильно.

Рассмотрим пример: вы измеряете возраст в годах, доход — в долларах, а кредитный рейтинг — в кредитных очках, что-то вроде очков SAT. Два человека будут представлены в виде двух триплетов, например (25, 45 000, 700) и (35, 76 000, 730). В частности, «расстояние» этих двух человек будет по большей части определяться уровнем их дохода.

С другой стороны, если бы вы измеряли доход в *тысячах долларов*, то они были бы представлены как (25, 45, 700) и (35, 76, 730), что придало бы трем переменным подобный уровень влияния.

Таким образом, то, как вы масштабируете свои переменные, эквивалентно тому, как в данной ситуации вы определяете понятие расстояния, и потенциально оказывает огромное воздействие на результат. В статистике это называют приоритетом.

Евклидово расстояние хорошо подходит для измерения расстояния перехода в случае атрибутов с вещественными значениями, которые могут быть нанесены на график с простым или многомерным пространством. Есть и другие меры.

- *Косинусный коэффициент.* Может применяться между двумя вещественными векторами $\vec{x}$ и $\vec{y}$ и принимать значение в диапазоне от −1 (абсолютная противоположность) до 1 (абсолютное совпадение) с промежуточным значением 0, говорящим об отсутствии зависимости. Вспомните определение $(\vec{x}, \vec{y}) = \frac{\vec{x} \cdot \vec{y}}{\|\vec{x}\| \|\vec{y}\|}$.
- *Расстояние Жаккара, или подобие.* Это уравнение вычисляет расстояние между наборами объектов, например списком друзей Кэти — A = {*Kahn*, *Mark*, *Laura*...} и Рэйчел — B = {*Mladen*, *Kahn*, *Mark*...}, и говорит, насколько близки эти два набора: $J(A,B) = \frac{|A \cap B|}{|A \cup B|}$.
- *Расстояние Махаланобиса.* Может использоваться между двумя вещественными векторами и обладает преимуществом евклидова расстояния — рассмотрение

корреляции инвариантно к шкале, $d(\vec{x},\vec{y}) = \sqrt{(\vec{x}-\vec{y})^T S^{-1}(\vec{x}-\vec{y})}$, где S — ковариационная матрица.

- *Расстояние Хемминга*. Может служить для поиска расстояния между двумя строками либо парами слов или последовательностями ДНК одинаковой длины. Расстояние между словами olive и ocean равно 4, так как все буквы, за исключением «o», отличаются. Расстояние между словами shoe и hose равно 3, поскольку все буквы, кроме «e», отличаются. Вы выполняете проход по всем позициям для сличения букв, и если в проходимой позиции они отличаются, то увеличиваете получившееся значение на 1.
- *Манхэттен*. Это также расстояние между двумя вещественными k-размерными векторами. Чтобы запомнить, представьте такси, которое ездит по улицам Манхэттена, распланированным в виде прямоугольной сетки (вы не можете двигаться по диагонали через здания). Таким образом, расстояние определяется как $d(\vec{x},\vec{y}) = \sum_i^k \left|x_i - y_i\right|$, где i — i-й элемент каждого вектора.

Существует и много других мер расстояния, которые вы можете использовать в зависимости от того, с данными какого типа работаете. Если не знаете, с чего начать, то пусть это будет поиск в Google.

Что, если ваши атрибуты представлены смесью типов данных? Это происходит в случае с рейтингом фильмов, например: одни атрибуты были числовыми (бюджет и количество актеров), а другие — категорийными (жанр). Но вы всегда можете определить собственную меру расстояния.

Например, вы можете сказать: если фильмы относятся к одному жанру, то это добавит 0 к их расстоянию. Но отношение их к разным жанрам добавит 10, где выбранное вами значение 10 находится на той же шкале, что и бюджет (миллионы долларов), то есть в диапазоне от 0 до 100. Вы можете сделать то же самое и с количеством актеров.

Вам потребуется обосновать свой выбор. Вы можете сказать, что попробовали разные значения и при тестировании алгоритма тот или иной выбор дал наилучшую меру оценки. В частности, 10 — это либо второй настроечный параметр, введенный в алгоритм поверх переменной k, либо приоритет, введенный в модель, — все зависит от вашей точки зрения и того, как данное значение будет применяться впоследствии.

Учебные и тестовые наборы

Общий подход для любого алгоритма машинного обучения заключается в наличии фазы обучения, на протяжении которой вы создаете и обучаете модель, а затем начинается фаза тестирования, когда вы используете новые данные для проверки того, насколько хороша ваша модель.

Для метода *k*-БС фаза обучения очень прямолинейна: это просто считывание ваших данных с отметками «высокой» и «низкой» кредитоспособности. При тестировании вы делаете вид, что не знаете истинного значения метки, и проверяете, насколько хорошо ваша модель угадывает, используя алгоритм *k*-БС.

Чтобы сделать это, вам потребуется сохранить некоторое количество чистых данных из общего набора для использования в рамках фазы тестирования. Обычно вам нужно сохранить случайным образом отобранные данные, скажем, 20 %.

Вывод консоли R может выглядеть следующим образом:

```
> head(data)
  age  income credit
1  69       3    low
2  66      57    low
3  49      79    low
4  49      17    low
5  58      26   high
6  44      71   high

n.points <- 1000 # количество строк в наборе данных
sampling.rate <- 0.8

# для вычисления уровня ошибочной классификации
# нам потребуется количество точек данных в наборе
num.test.set.labels <- n.points * (1 - sampling.rate)

# случайная выборка строк для использования в обучающем наборе
training <- sample(1:n.points, sampling.rate * n.points,
replace=FALSE)

# определить, что обучающий набор содержит эти строки
train <- subset(data[training, ], select = c(Age, Income))

# остальные строки переходят в тестовый набор
testing <- setdiff(1:n.points, training)

# определить, что тестовый набор содержит другие строки
test <- subset(data[testing, ], select = c(Age, Income))

# подмножество меток для обучающего набора
cl <- data$Credit[training]
# подмножество меток для тестового набора, их мы удерживаем
true.labels <- data$Credit[testing]
```

Выбор меры оценки

Как оценить, хорошо ли сработала модель?

Легкого или универсального ответа на этот вопрос нет: вы можете захотеть отбраковать ошибки классификации одного типа с большей тщательностью, чем другого. Ложноотрицательные результаты могут быть значительно хуже

ложноположительных. Разработка меры оценки может потребовать участия эксперта в предметной области.

Например, если бы вы использовали алгоритм классификации для прогнозирования того, болен ли человек раком, то вам потребовалось бы минимизировать ложноотрицательные результаты (ошибочная постановка диагноза «рак не обнаружен» при его фактическом наличии), для этого вам пришлось бы работать с врачом для настройки вашей меры оценки.

Обратите внимание: вам следует быть осторожными. Если бы вам нужно было обеспечить *отсутствие* ложноотрицательных результатов, то вы могли бы просто сказать *всем*, что у них обнаружен рак. Как следствие, это некий компромисс между *чувствительностью* и *специфичностью*, где первая определяется как возможность корректного диагностирования больного пациента, тогда как вторая — вероятность корректного диагностирования здорового пациента.

Другие термины для чувствительности и специфичности

Чувствительность также принято называть долей истинно положительных результатов или обратным вызовом, терминология зависит от того, из какой научной сферы вы попали в машинное обучение, но все эти термины означают одно и то же. Специфичность еще называют долей истинно отрицательных результатов. Кроме того, существует доля ложноположительных результатов и доля ложноотрицательных результатов, но у этих терминов нет синонимов.

Другая мера оценки, которую вы можете использовать, — *точность*, определяемая нами в главе 5. Факт, что одинаковые формулы носят разные названия, можно объяснить так: разные научные дисциплины пришли к ним независимо друг от друга. Поэтому *точность* и *обратный вызов* — количества, применяемые в области получения информации. Обратите внимание, что *точность* — не то же самое, что *специфичность*.

Наконец, у нас также есть *безошибочность* — отношение количества корректных меток к общему количеству последних. Кроме того, есть ошибочная классификация, равная просто 1, — *безошибочность*. Таким образом, минимизация доли ошибок классификации максимизирует *безошибочность*.

Соберем все вместе

Теперь, имея все меры расстояния и оценки, вы готовы идти в бой.

Для каждого человека в наборе вы делаете вид, что не знаете его метку. Взгляните на метки, скажем, трех ближайших соседей и воспользуйтесь меткой большинства голосов для обозначения этого человека. Обозначьте всех членов тестового набора, а затем воспользуйтесь значением доли ошибочной классификации для оценки

того, насколько хорошо вы сработали. На языке R все это выполняется автоматически, потребуется всего одна строка кода:

```
knn (train, test, cl, k=3)
```

Выбор k

Как выбрать значение *k*? Это как раз тот параметр, над которым у вас есть контроль. Для хорошего подбора значения вам может потребоваться довольно хорошо понимать ваши данные. Вы также можете испробовать несколько значений *k* с тем, чтобы пронаблюдать изменение оценки. Так вы запустите *k*-БС несколько раз, изменяя значение *k* и всякий раз проверяя меру оценки.

Бинарные классы

Когда у вас есть такие бинарные классы, как «высокая кредитоспособность» и «низкая кредитоспособность», то выбор нечетного значения k может быть хорошей идеей, поскольку таким образом всегда будет большинство голосов без равных исходов. При равном исходе алгоритм осуществляет просто случайный выбор.

```
# мы выполним циклический проход и посмотрим,
# какова доля ошибок классификации для разных значений k
for (k in 1:20) {
   print(k)
   predicted.labels <- knn(train, test, cl, k)
   # мы используем функцию knn() языка R
   num.incorrect.labels <- sum(predicted.labels != true.labels)
   misclassification.rate <- num.incorrect.labels /
                             num.test.set.labels
   print(misclassification.rate)
}
```

Код, приведенный выше, дает следующий вывод (*k*, доля ошибочной классификации):

```
k  misclassification.rate
1, 0.28
2, 0.315
3, 0.26
4, 0.255
5, 0.23
6, 0.26
7, 0.25
8, 0.25
9, 0.235
10, 0.24
```

Итак, остановимся на $k = 5$, так как данное значение дает наименьшую долю ошибочной классификации, и теперь вы можете применить его к новому человеку 57 лет с зарплатой 37 000 долларов. В консоли R это будет выглядеть следующим образом:

```
> test <- c(57,37)
> knn(train,test,cl, k = 5)
[1] low
```

Результат по большинству голосов — низкий кредитный рейтинг при $k = 5$.

Тестовый набор в k-БС

Обратите внимание: мы использовали функцию knn() дважды, причем каждый раз иначе. Сначала в тестовом наборе были некие данные, которые мы применили для оценки того, насколько хороша наша модель. Затем «тестовым» набором на самом деле была новая точка данных, для которой мы хотели сделать прогноз. Мы могли бы передать в функцию несколько строк с данными людей, для которых нужно было сделать прогноз. Однако заметьте: язык R не знает того, являются ли вводимые в тестовый набор данные на самом деле истинным «тестовым» набором, для которого вы знаете действительные метки, или же вы не знаете меток и хотите прогноз.

Что такое допущения при моделировании

В предыдущей главе мы обсуждали моделирование и допущения при моделировании. Что же такое допущения при моделировании здесь?

Алгоритм k-БС — пример непараметрического подхода. При моделировании вы не делали никаких допущений о лежащих в основе распределениях, генерирующих данные, также не пытались оценить какие-либо параметры. Однако вы все равно сделали *некоторые* допущения:

- данные — это некое пространство признаков, где понятие «расстояние» имеет значение;
- обучающие данные были помечены или классифицированы как два или более класса;
- вы сами выбираете используемое количество соседей, k;
- вы допускаете, что *наблюдаемые признаки* и *метки* каким-то образом связаны. Связи может и не быть, но ваша мера оценки позволит определить, как хорошо ваш алгоритм справляется с присвоением меток. Вы можете добавить больше признаков, чтобы проверить, как это повлияет на меру оценки. Впоследствии вам потребуется настроить *оба* используемых признака и k. Впрочем, здесь вы, как всегда, сталкиваетесь с опасностью переобучения.

Линейная регрессия и k-БС — примеры «обучения под наблюдением», где вы наблюдаете обе переменные x и y и хотите узнать, какая функция приводит x к y. Далее посмотрите на алгоритм, который можете использовать, если не знаете правильный ответ.

k-средние

До сих пор мы только видели примеры обучения под наблюдением, когда заранее знаем метку (то есть «правильный ответ) и пытаемся добиться, чтобы наша модель была как можно более точной, что определяется выбранной нами мерой оценки.

k-средние — это первый из методов *неконтролируемого* обучения, которые мы рассмотрим. Цель алгоритма — определить понятие правильного ответа, выполнив для вас поиск кластеров данных.

Предположим, у вас есть некие данные уровня пользователя, например данные Google+, опроса, медицинские данные или очки SAT.

Начнем с добавления структуры ваших данных, а именно допустим, что каждая строка набора соответствует одному пользователю следующим образом:

```
age gender income state household size
```

Ваша цель — *сегментировать* пользователей. Этот процесс называют по-разному: кроме сегментирования, вы также можете сказать, что собираетесь *стратифицировать*, *разбить на группы* или *кластеры* данные. Все эти термины означают поиск и группировку пользователей подобных типов.

Зачем это может потребоваться? Приведем несколько примеров.

- Вам может понадобиться дать разным пользователям различные возможности. Такое часто встречается в маркетинге: например, тонер предлагают людям, о которых известно, что у них есть принтер.
- Вы можете иметь модель, лучше работающую для определенных групп, или создать разные модели для различных групп.
- Иерархическое моделирование (http://www.stat.columbia.edu/~gelman/research/published/multi2.pdf) в статистике делает что-то подобное и используется, например, для раздельного моделирования географических эффектов и эффектов домохозяйств в результатах опросов.

Чтобы понять, как такой алгоритм может быть полезен, сначала сконструируем некий элемент вручную, то есть распределим пользователей по группам, используя созданные вручную пределы.

Для таких атрибутов, как возраст, вы создадите группы: 20–24, 25–30 и т. д. Аналогичная техника может использоваться и для других атрибутов, например для уровня доходов. Штаты и города в каком-то смысле являются группами сами по себе, но вам может потребоваться меньшее количество групп, в зависимости от вашей модели и количества точек данных. В подобном случае вы можете создать группы, например, «Восточное побережье», «Средний запад» и т. п.

Предположим, вы проделали это для каждого атрибута. У вас может быть десять возрастных групп, две половые группы и т. д., что приведет к созданию $10 \times 2 \times 50 \times 10 \times 3 = 30\,000$ возможных групп, а это немало.

Представьте упомянутые данные в пятимерном пространстве, где каждая ось соответствует одному из атрибутов. То есть у нас есть ось пола, дохода и т. д. Вы также можете обозначить всевозможные группы вдоль соответствующих осей — и если вы выполните данное действие, то ваша координатная сетка будет содержать все вероятные группы: по группе для каждого потенциального сочетания атрибутов.

Таким образом, каждый пользователь будет находиться в одной из 30 000 пятимерных ячеек. Но погодите, вряд ли вам может потребоваться создавать отдельную маркетинговую кампанию для каждой группы. Следовательно, понадобится объединить группы...

Теперь вы, наверное, уже осознаете полезность алгоритма, который сделает это за вас, особенно если вы заранее можете установить, сколько групп вам нужно. Как раз именно это и есть метод k-средних: алгоритм *кластеризации*, где k — количество групп.

Двухмерная версия

Вернемся к более простому примеру, чем тот, что мы обсудили чуть выше, с пятимерным пространством. Предположим, у вас есть пользователи и вы знаете, сколько рекламы было им показано (количество показов) и сколько раз каждый из них щелкнул кнопкой мыши на рекламе (количество переходов).

На рис. 3.9 показано упрощенное представление того, как это может выглядеть.

Визуально вы можете видеть на верхнем левом рисунке, что данные естественным образом распадаются на кластеры. Это легко можно отследить невооруженным глазом при наличии только двух измерений и небольшого количества точек данных, но при увеличении количества измерений и данных вам потребуется алгоритм, помогающий найти закономерности. Алгоритм k-средних ищет кластеры в d-измерениях, где d — количество признаков для каждой точки данных.

Вот как работает алгоритм, показанный на рис. 3.9.

1. Изначально вы случайным образом выбираете k центроидов (или точек, которые будут находиться в центе ваших кластеров) в d-измерении. Постарайтесь установить центроиды рядом с данными, но так, чтобы они отличались друг от друга.
2. Затем назначьте точки данных ближайшей центроиде.
3. Переместите центроиды в среднее положение точек данных, назначенных им (что в текущем примере соответствует пользователям).

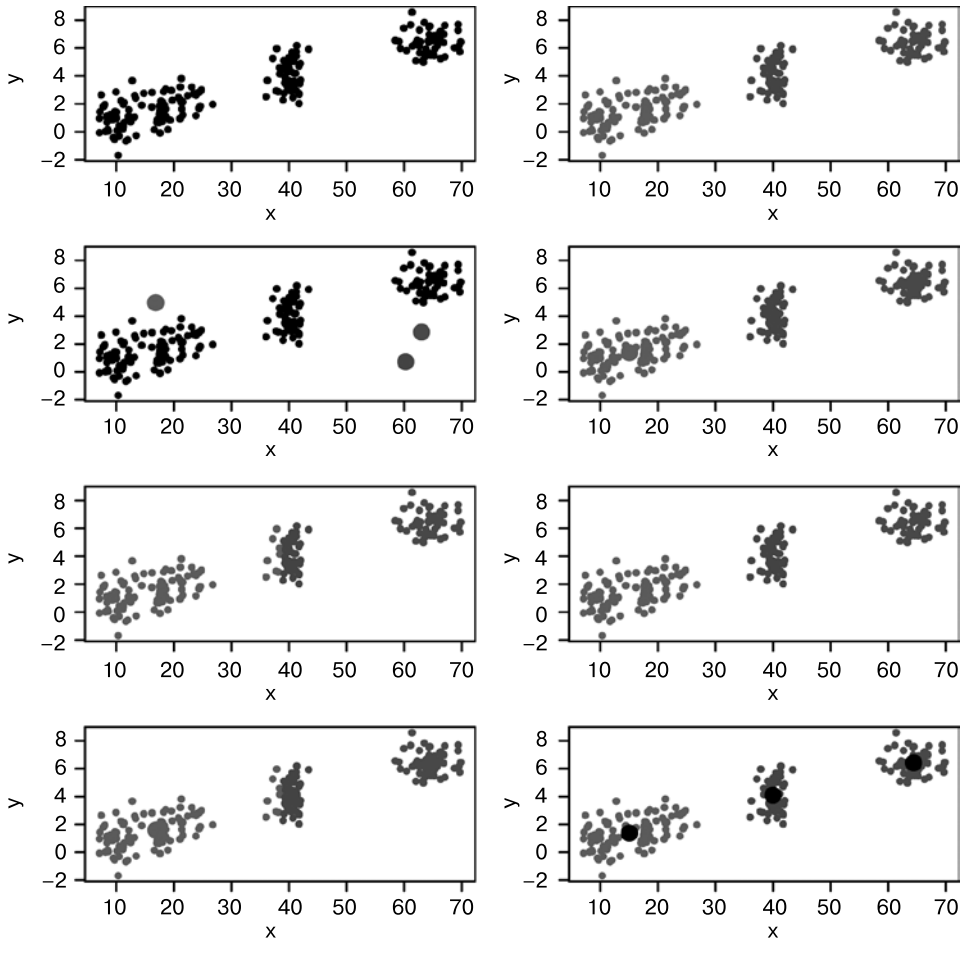


Рис. 3.9. Кластеризация в двух измерениях. Просмотрите сверху вниз графики в левом столбце, а затем сверху вниз — в правом

4. Повторяйте два предыдущих шага до тех пор, пока назначения не изменятся ощутимо или весьма незначительно.

От вас зависит, существует ли естественный способ интерпретации этих групп по окончании работы алгоритма. Иногда вам придется несколько раз изменять значения k, пока вы получите естественные группировки.

Это пример *неконтролируемого* обучения, так как метки неизвестны и обнаруживаются алгоритмом.

У метода k-средних есть несколько известных проблем.

- Выбор значения k больше искусство, чем наука, несмотря на существование рамок: $1 \leq k \leq n$, где n — количество точек данных.
- Существуют проблемы конвергенции: решения может не быть, если алгоритм, например, попадает в бесконечный цикл и продолжает переходить от одного решения к другому, или, иными словами, не существует единого уникального решения.
- Интерпретируемость может быть проблемой: иногда ответ бесполезен в принципе. В реальности зачастую это самая большая проблема.

Несмотря на все проблемы, данный алгоритм работает достаточно быстро (в сравнении с другими алгоритмами кластеризации) и широко применяется в маркетинге, машинном зрении (разбиение изображения) или в качестве начальной точки других методов.

На деле это всего лишь одна строка кода на языке R:

```
kmeans(x, centers, iter.max = 10, nstart = 1,
algorithm = c("Hartigan-Wong", "Lloyd", "Forgy", "MacQueen"))
```

Ваш набор данных должен быть матрицей, *x*, каждый столбец которой — один из ваших признаков. Вы указываете *k*, выбирая центры. По умолчанию установлено определенное количество итераций, но вы можете изменить значение этого аргумента, как и выбрать конкретный алгоритм, который будет использоваться для обнаружения кластеров.

ИСТОРИЧЕСКАЯ СПРАВКА: K-СРЕДНИЕ

Постойте, разве мы только что не описали алгоритм? Оказывается, существует несколько способов кластеризации методом *k*-средних.

Стандартный алгоритм *k*-средних относят к отдельной работе Хьюго Стайнхауса (Hugo Steinhaus) и Стюарта Ллойда (Stuart Lloyd) в 1957-м, но тогда данный алгоритм назывался иначе. Первым использовал термин «*k*-средние» Джеймс Маккуин (James MacQueen) в 1967-м, однако обозначение не было опубликовано вне лаборатории Bell Labs до 1982 года.

Более новые версии алгоритма: методы Хартигана — Вонга, Ллойда и Форги, носящие имена своих изобретателей, были разработаны в 60-х и 70-х годах прошлого века. Алгоритм, который мы обсудили, является опцией по умолчанию и построен по методу Хартигана — Вонга.

Но история не стоит на месте, поэтому стоит также взглянуть на более новый метод «*k*-средние++», разработанный в 2007 году Дэвидом Артуром (David Arthur) и Сергеем Васильвитским (Sergei Vassilvitskii) (сейчас в Google). Метод позволяет избежать проблем конвергенции с помощью оптимизации изначальных затравок.

Упражнение. Основные алгоритмы машинного обучения

Продолжайте работать с набором данных NYC (Manhattan) Housing из предыдущей главы: https://www.tripsavvy.com/sales-data-for-nyc-real-estate-2819364.

- Проанализируйте продажи с помощью регрессии с любыми, по вашему мнению, релевантными показателями. Докажите, почему регрессия подходит для использования.
- Визуализируйте коэффициенты и обученную модель.
- Спрогнозируйте район с помощью классификатора *k*-БС. Не забудьте вычленить поднабор данных для тестирования. Найдите значения переменных и *k*, при которых погрешность прогнозирования будет минимальной.
- Опишите и визуализируйте свои находки.
- Опишите любые решения, которые можно принять, и действия, которые могут быть предприняты из проведенного вами анализа.

Решения

В предыдущей главе мы показали, как обследовать и очистить набор данных, так что вам потребуется сделать это перед созданием модели регрессии. Далее приведено два кода на языке R. Первый демонстрирует вариант построения моделей регрессии, а второй — способ очистки и подготовки данных с дальнейшим созданием классификатора *k*-БС.

Пример кода на языке R: линейная регрессия на наборе данных о жилищных условиях

```
# Автор: Бен Редди (Ben Reddy)

model1 <- lm(log(sale.price.n) ~ log(gross.sqft),data=bk.homes)
## что здесь происходит?

bk.homes[which(bk.homes$gross.sqft==0),]

bk.homes <- bk.homes[which(bk.homes$gross.sqft>0 &
            bk.homes$land.sqft>0),]
model1 <- lm(log(sale.price.n) ~ log(gross.sqft),data=bk.homes)
summary(model1)

plot(log(bk.homes$gross.sqft),log(bk.homes$sale.price.n))
abline(model1,col="red",lwd=2)
plot(resid(model1))

model2 <- lm(log(sale.price.n) ~ log(gross.sqft) +
```

```
  log(land.sqft) + factor(neighborhood),data=bk.homes)
summary(model2)
plot(resid(model2))

## пропустить пересечение для упрощения интерпретации
model2a <- lm(log(sale.price.n) ~ 0 + log(gross.sqft) +
  log(land.sqft) + factor(neighborhood),data=bk.homes)
summary(model2a)
plot(resid(model2a))

## добавить тип строения
model3 <- lm(log(sale.price.n) ~ log(gross.sqft) +
  log(land.sqft) + factor(neighborhood) +
  factor(building.class.category),data=bk.homes)
summary(model3)
plot(resid(model3))

## связать микрорайон и тип строения
model4 <- lm(log(sale.price.n) ~ log(gross.sqft) +
  log(land.sqft) + factor(neighborhood)*
  factor(building.class.category),data=bk.homes)
summary(model4)
plot(resid(model4))
```

Пример кода на языке R: k-БС на наборе данных о жилищных условиях

```
# Автор: Бен Редди (Ben Reddy)

require(gdata)
require(geoPlot)
require(class)

mt <- read.xls("rollingsales_manhattan.xls",
  pattern="BOROUGH",stringsAsFactors=FALSE)
head(mt)
summary(mt)

names(mt) <- tolower(names(mt))

mt$sale.price.n <- as.numeric(gsub("[^[:digit:]]","",
                              mt$sale.price))
sum(is.na(mt$sale.price.n))
sum(mt$sale.price.n==0)

names(mt) <- tolower(names(mt))

## очистка/форматирование данных с помощью регулярных выражений
mt$gross.sqft <- as.numeric(gsub("[^[:digit:]]","",
                            mt$gross.square.feet))
mt$land.sqft <- as.numeric(gsub("[^[:digit:]]","",
                            mt$land.square.feet))

mt$sale.date <- as.Date(mt$sale.date)
```

```
mt$year.built <- as.numeric(as.character(mt$year.built))
mt$zip.code <- as.character(mt$zip.code)

## стандартизация данных (установить год начала строительства в 0; площадь
земельного участка и общую площадь в кв. м; реализационную цену (исключить $0
и, возможно, другие); может быть, налоговый блок; внешний набор данных
для координации налогового блока/парковка?)
min_price <- 10000
mt <- mt[which(mt$sale.price.n>=min_price),]

n_obs <- dim(mt)[1]

mt$address.noapt <- gsub("[,][[:print:]]*","",
                    gsub("[ ]+"," ",trim(mt$address)))

mt_add <- unique(data.frame(mt$address.noapt,mt$zip.code,
                 stringsAsFactors=FALSE))
names(mt_add) <- c("address.noapt","zip.code")
mt_add <- mt_add[order(mt_add$address.noapt),]

# найти дублирующиеся адреса с разными почтовыми индексами
dup <- duplicated(mt_add$address.noapt)
# удалить их
dup_add <- mt_add[dup,1]
mt_add <- mt_add[(mt_add$address.noapt != dup_add[1] &
          mt_add$address.noapt != dup_add[2]),]

n_add <- dim(mt_add)[1]

# отобрать 500 адресов, чтобы не превысить дневной лимит API системы Google Maps
# (и не ждать вечность)
n_sample <- 500
add_sample <- mt_add[sample.int(n_add,size=n_sample),]

# сначала попробовать создать запросы с имеющимися адресами
query_list <- addrListLookup(data.frame(1:n_sample,
 add_sample$address.noapt,rep("NEW YORK",times=n_sample),
 rep("NY",times=n_sample),add_sample$zip.code,
 rep("US",times=n_sample)))[,1:4]

query_list$matched <- (query_list$latitude != 0)

unmatched_inds <- which(!query_list$matched)
unmatched <- length(unmatched_inds)

# попытаться изменить EAST/WEST на E/W
query_list[unmatched_inds,1:4] <- addrListLookup
  (data.frame(1:unmatched,gsub(" WEST "," W ",
  gsub(" EAST "," E ",add_sample[unmatched_inds,1])),
  rep("NEW YORK",times=unmatched), rep("NY",times=unmatched),
    add_sample[unmatched_inds,2],rep("US",times=unmatched)))[,1:4]

query_list$matched <- (query_list$latitude != 0)
unmatched_inds <- which(!query_list$matched)
unmatched <- length(unmatched_inds)
```

```
# попытаться изменить STREET/AVENUE на ST/AVE
query_list[unmatched_inds,1:4] <- addrListLookup
  (data.frame(1:unmatched,gsub(" WEST "," W ",
  gsub(" EAST "," E ",gsub(" STREET"," ST",
  gsub(" AVENUE"," AVE",add_sample[unmatched_inds,1])))),
  rep("NEW YORK",times=unmatched), rep("NY",times=unmatched),
    add_sample[unmatched_inds,2],rep("US",times=unmatched)))[,1:4]

query_list$matched <- (query_list$latitude != 0)
unmatched_inds <- which(!query_list$matched)
unmatched <- length(unmatched_inds)

## быть пока довольными
add_sample <- cbind(add_sample,query_list$latitude,
  query_list$longitude)
names(add_sample)[3:4] <- c("latitude","longitude")

add_sample <- add_sample[add_sample$latitude!=0,]

add_use <- merge(mt,add_sample)
add_use <- add_use[!is.na(add_use$latitude),]

# координаты на карте
map_coords <- add_use[,c(2,4,26,27)]
table(map_coords$neighborhood)
map_coords$neighborhood <- as.factor(map_coords$neighborhood)

geoPlot(map_coords,zoom=12,color=map_coords$neighborhood)

## функция knn
## существуют более эффективные способы, ну да ладно...

map_coords$class <- as.numeric(map_coords$neighborhood)

n_cases <- dim(map_coords)[1]
split <- 0.8

train_inds <- sample.int(n_cases,floor(split*n_cases))
test_inds <- (1:n_cases)[-train_inds]

k_max <- 10
knn_pred <- matrix(NA,ncol=k_max,nrow=length(test_inds))
knn_test_error <- rep(NA,times=k_max)

for (i in 1:k_max) {
    knn_pred[,i] <- knn(map_coords[train_inds,3:4],
  map_coords[test_inds,3:4],cl=map_coords[train_inds,5],k=i)
    knn_test_error[i] <- sum(knn_pred[,i]!=
      map_coords[test_inds,5])/length(test_inds)
}

plot(1:k_max,knn_test_error)
```

МОДЕЛИРОВАНИЕ И АЛГОРИТМЫ В МАСШТАБЕ

Данные, с которыми вы работали в этой главе до сих пор, в масштабах больших данных считаются довольно маленькими. Что происходит с этими моделями при масштабировании их и алгоритмов на массивные наборы данных?

В некоторых случаях абсолютно правильно работать с небольшими наборами или запускать одну и ту же модель на нескольких *сегментированных* наборах. (Сегментирование — это когда данные разделены на части и хранятся на нескольких машинах, а затем вы смотрите на эмпирическое распределение оценщиков в моделях.) Иными словами, существуют статистические решения этих инженерных задач.

Однако иногда нам требуется обучить эти модели в масштабе, а задача масштабирования моделей, по сути, переходит в задачу создания параллельных версий или приближений методов оптимизации. Например, линейная регрессия в масштабе нуждается в инверсиях матрицы или приближениях инверсий матрицы.

Оптимизация при работе с большими данными требует новых подходов и теорий: и это передовая! Из речи Питера Риктарика (Peter Richtarick) из Университета Эдинбурга: «В предметной области больших данных неприменимы классические подходы, в которых используются методы оптимизации с несколькими итерациями, поскольку вычислительные затраты даже одной подобной итерации зачастую слишком высоки. Эти методы разработаны в прошлом, когда проблемы огромных размеров были редкостью. Таким образом, мы нуждаемся в более простых методах, осторожных в работе с данными, нетребовательных к ресурсам памяти и масштабируемых. Наша способность решать действительно крупномасштабные проблемы идет рука об руку с нашей способностью использовать современные архитектуры параллельных вычислительных систем, такие как многоядерные процессоры, графические процессорные устройства и компьютерные кластеры».

Многое из приведенного выше находится вне поля зрения данной книги, но как исследователь данных вы должны быть предупреждены об этих проблемах, некоторые из них обсуждаются в главе 14.

Резюмируя вышесказанное

Мы представили три алгоритма, являющиеся базисом решения многих проблем реального мира. Если вы понимаете эти алгоритмы, то это хорошее начало. В противном случае не переживайте: погружение в тему требует некоторого времени.

Регрессия — основа многих моделей, классификации или прогнозирования в целом наборе контекстов. Мы показали, как спрогнозировать непрерывную выходную переменную с помощью одного или нескольких показателей. Мы вернемся к этому в главе 5, где изучим *логистическую* регрессию, которая может использоваться для классификации бинарных результатов, и в главе 6, где увидим указанный механизм

в контексте моделирования временны*х* рядов. В главе 7 потренируем ваши навыки выбора признаков.

k-БС и *k*-средние — примеры алгоритмов кластеризации, в которых необходимо сгруппировать подобные объекты. Здесь понятия *расстояния* и *мер оценки* становятся важными, мы также видели, что в их выборе есть определенная субъективность. Мы изучим алгоритмы кластеризации, в том числе наивный классификатор Байеса в главах 4 и 10. Как мы увидим, *кластеризация графов* — интересная область для исследований. Другие примеры алгоритмов кластеризации, не затронутых в этой книге: *иерархическая кластеризация* и *кластеризация на основе модели*.

Для дополнительного чтения и более глубокой проработки этого материала мы рекомендуем классическую книгу Хасти (Hastie) и Тибширани (Tibshirani) *Elements of Statistical Learning* («Элементы статистического обучения») (Springer). Для глубинного исследования вопросов создания моделей регрессии в байесовском контексте очень рекомендуем работу Эндрю Гельмана (Andrew Gelman) и Дженнифер Хил (Jennifer Hill) *Data Analysis using Regression and Multilevel/Hierarchical Models* («Анализ данных с помощью регрессии и многоуровневых/иерархических моделей»).

Мысленный эксперимент: автоматизированный статистик

Рэйчел посещала семинар по интеллектуальному анализу больших данных в Имперском колледже Лондона в мае 2013 года. Один из докладчиков, профессор Зубин Гхахрамани (Zoubin Ghahramani) из Кембриджского университета, сказал, что один из его долгосрочных исследовательских проектов — создание «автоматизированного статистика». Как вы думаете, что это значит? По вашему мнению, что потребуется для его создания?

Пугает ли вас эта идея? Должна ли?

4 Фильтры спама, наивный классификатор Байеса и перебор данных

В написание этой главы внес вклад Джейк Хофман (Jake Hofman) (http://www.jakehofman.com/). После ухода из Yahoo! Research он работает в Microsoft Research (https://www.microsoft.com/en-us/research/?from=http%3A%2F%2Fresearch.microsoft.com%2F). Джейк получил ученую степень PhD в Университете округа Колумбия, где регулярно преподает фантастический курс по моделированию, основанному на данных (http://jakehofman.com/ddm/), а также более новый курс вычислительной социологии.

Как и с другими нашими докладчиками, сначала мы взглянули на профиль Джейка в науке о данных. Оказалось, он эксперт в категории, добавленной им в свое портфолио исследователя данных, под названием «выпас данных». Джейк признался: он не знает, из-за чего посвящает этому направлению столько времени: из-за того, что хорошо в нем разбирается, или из-за того, что плохо (он хорошо в нем разбирается).

Мысленный эксперимент: обучение на примере

Начнем со взгляда на текст, приведенный на рис. 4.1. Кажется, что строки содержат тему и первую строку входящего сообщения электронной почты.

Вы можете заметить, что некоторые строки текста похожи на спам.

Как вы это поняли? Можете ли вы написать код для автоматизированной фильтрации спама, которая есть у вас в голове?

У учеников Рэйчел было несколько идей относительно того, что может считаться четким признаком спама.

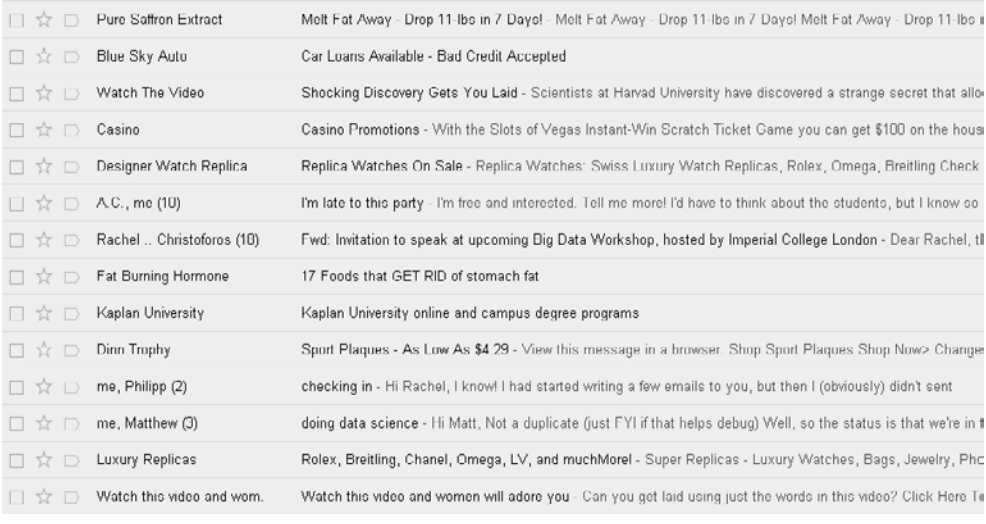


Рис. 4.1. Подозрительно много спама

- Любое электронное письмо — спам, если содержит ссылки на виагру. Это хорошее правило для начала работы, но, как вы, скорее всего, видели в своей почте, люди поняли данное правило и нашли способ, как его обойти, изменив написание. (Очень жаль, что спамеры настолько умны, но при этом не работают на развитие более важных проектов, чем продажа огромного количества виагры...)
- Вероятно, какие-то аспекты длины темы сообщения выдают в нем спам, или это может быть чрезмерное использование восклицательных знаков или другой пунктуации. Но некоторые слова, например Yahoo!, являются аутентичными, поэтому вам не нужно чрезмерно упрощать ваше правило.

Вот несколько предложений по поводу кода, который вы могли бы написать для идентификации спама.

- Попробуйте вероятностную модель. Иными словами, у вас не должно быть простых правил, вместо этого создайте много практических правил, соберите их вместе и представьте вероятность того, что любое заданное электронное письмо — спам. Это замечательная идея.
- Как насчет *k*-БС и линейной регрессии? Вы изучили эти техники в предыдущей главе, но применимы ли они к задачам такого типа? (Подсказка: нет.)

В этой главе для решения данной задачи мы воспользуемся наивным классификатором Байеса, который находится примерно между двумя вышеуказанными техниками. Но сначала...

Почему линейная регрессия не сработает для фильтрации спама

Так как мы уже знакомы с линейной регрессией и она входит в наш набор инструментов, начнем с разговора о действиях, которые пришлось бы сделать, если бы потребовалось использовать линейную регрессию. Мы уже знаем, что *не* будем ее применять, но все же обсудим и придем к пониманию почему. Представьте набор данных или матрицу, где каждая строка представляет одно сообщение электронной почты (ключ может быть задан как `email_id`). Теперь сделаем каждое слово в сообщении *признаком*: это значит, что мы создаем столбец с названием, например, Viagra, а затем для каждого сообщения, в котором это слово встречается хотя бы раз, вводим в этот столбец 1, в противном случае — 0. Альтернативный подход — указывать количество раз, которое это слово встречается в сообщении. Таким образом, каждый столбец представляет одно слово.

Возвращаясь к предыдущей главе: мы знаем, что для использования линейной регрессии нам необходимо иметь обучающий набор электронных писем, в котором сообщения *уже* были подписаны некой выходной переменной. В данном случае на выходе мы получаем результат, является ли сообщение спамом. Мы могли бы реализовать это с помощью людей, которые оценивали и обозначали бы сообщения как «спам», что вполне резонно, хотя и занимает большое количество времени. Другой подход заключался бы в использовании существующего фильтра спама, например, из почты Gmail и применении этих меток. (Конечно, если у вас уже есть фильтр спама Gmail то сложно сказать, зачем вам могло бы потребоваться создавать новый фильтр, но предположим, что такая необходимость есть.) После того как вы создадите модель, сообщения электронной почты будут приходить к вам без меток, а вы станете использовать модель для их прогнозирования.

Первое, что нужно рассмотреть, — вашей целью является бинарный результат (0, если не спам, и 1, если спам), но вы не получите 0 или 1, используя линейную регрессию: вы получите число. Строго говоря, эта опция действительно неидеальна: линейная регрессия предназначена для моделирования непрерывного вывода, а это — бинарный.

Проблема, в общем-то, в бесполезности. Нам следует использовать модель, подходящую для данных. Но если бы мы захотели обучить такую модель на языке R, то в теории такое могло бы сработать. Язык R не проверяет, подходит ли модель данным. Можно было бы применить этот метод, обучить линейную модель, а затем задействовать ее для прогнозирования, чтобы впоследствии выбрать критическое значение, при этом предсказанные значения больше критического мы называли бы 1, а меньше — 0.

Но если бы мы пошли дальше и попробовали, то все равно бы ничего не получилось, поскольку переменных больше, чем наблюдаемых данных. На 10 000 сообщений у нас приходилось бы 100 000 слов. Так не пойдет. Технически это соотносится с фактом, что матрица в уравнении для линейной регрессии неинвертируема, а на самом деле даже не близка к этому. Более того, возможно, у нас даже не получится ее сохранить из-за слишком большого размера.

Вероятно, мы могли бы ограничиться 10 000 слов? В этом случае у нас хотя бы была инвертируемая матрица. Даже если так, то переменных все равно намного больше, чем наблюдаемых данных. Имея тщательно отобранный набор признаков и знания предметной области, мы могли бы ограничиться 100 словами — и этого было бы достаточно! Но мы все равно столкнулись бы с проблемой, что линейная регрессия — модель, непригодная для бинарного результата.

Отвлечемся: ультрасовременные фильтры спама

В последние пять лет люди стали использовать методы стохастического градиента, чтобы избежать проблемы неинвертируемой (переобученной) матрицы. Переход к логистической регрессии со стохастическими методами градиента очень помог и может объяснить корреляции между словами. Несмотря на это, наивный классификатор Байеса довольно впечатляюще хорош, учитывая, насколько он прост.

Что насчет k-ближайших соседей

Скоро мы перейдем к наивному классификатору Байеса, но пока поразмыслим о попытке использования метода *k*-ближайших соседей (*k*-БС) для создания фильтра спама. Нам все равно нужно выбирать признаки, возможно соответствующие словам, и мы, вероятно, определим значение этих признаков как 0 или 1 в зависимости от того, присутствует ли в сообщении некое слово. Затем нужно определить, когда два письма находятся «рядом» друг с другом, основываясь на том, какие слова содержат оба этих письма.

Имея 10 000 писем и 100 000 слов, мы столкнемся с проблемой, отличной от проблемы неинвертируемой матрицы. А именно: проблема в том, что пространство, в котором мы работаем, имеет слишком *много измерений*. Да, подсчет расстояний в пространстве со 100 000 измерений требует много вычислительной работы. Но это не настоящая проблема.

Реальная проблема еще более фундаментальна: даже наши самые ближайшие соседи в действительности далеки. Это называется проклятием размерности, и оно делает метод *k*-БС алгоритмом, непригодным для данного случая.

ОТВЛЕЧЕМСЯ: РАСПОЗНАВАНИЕ ЦИФР

Скажем, вам нужен алгоритм для распознавания изображения рукописных цифр, показанных на рис. 4.2. В этом случае метод k-БС работает хорошо.

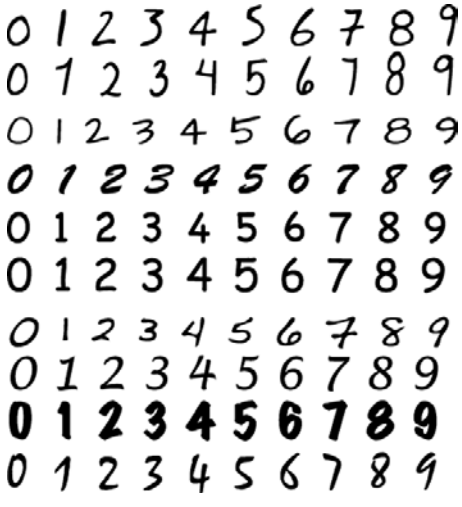


Рис. 4.2. Рукописные цифры

Чтобы настроить алгоритм, вам следует разобрать лежащее в основе графическое представление по пикселам, например, в сетке 16 × 16 пикселов и измерить яркость каждого из них. Разверните сетку 16 × 16 и поместите ее в 256-мерное пространство, что представляет естественную архимедову метрику. То есть расстояние между двумя разными точками в этом пространстве является квадратным корнем из суммы квадратов различий между их элементами. Другими словами, это длина вектора, идущего от одной точки к другой, или наоборот. Затем вы применяете алгоритм k-БС.

Изменение количества соседей преобразует форму границы, и вы можете подобрать k так, чтобы предотвратить переобучение. При должной внимательности можно получить точность 97 % с достаточно большим набором данных.

Более того, результат можно просмотреть в *матрице различий*. Она используется, когда вы пытаетесь классифицировать объекты в k групп, и представляет собой матрицу $k \times k$, соответствующую фактической метке по сравнению с предсказанной, а (i, j)-й элемент матрицы — это количество элементов, фактически помеченных как i, но для которых прогнозировалась метка j. Из матрицы различий вы можете получить *безошибочность* — общую долю сбывшихся предсказаний. В предыдущей главе мы обсудили долю ошибочной классификации. Обратите внимание: безошибочность = 1 — доля ошибочной классификации.

Наивный классификатор Байеса

Итак, теперь мы несем потери: два знакомых нам метода — линейная регрессия и *k*-БС — не сработают для решения задачи фильтра спама? Нет! Наивный классификатор Байеса — еще один метод классификации, находящийся в нашем распоряжении. Он хорошо масштабируется и интуитивно понятен.

Закон Байеса

Начнем с еще более простого примера, чем фильтр спама, чтобы понять, как работает наивный классификатор Байеса. Предположим, мы тестируем на редкое заболевание, которым страдает 1 % населения. Наш тест очень чувствительный, специфический и не совсем совершенный.

- У 99 % больных пациентов положительный результат.
- У 99 % здоровых пациентов отрицательный результат.

Учитывая положительный анализ пациента, какова вероятность того, что этот пациент действительно болен?

Наивный подход к ответу на данный вопрос таков: представьте, что есть $100 \times 100 = 10\,000$ среднестатистических человек. Это будет означать следующее: 100 человек больны, а 9900 — здоровы. Более того, после проведения анализа для 99 больных мы получили бы результат, что они больны, но и для 99 здоровых мы получили бы такой же результат. Если у вас положительный результат, то, другими словами, вы в равной степени можете быть и здоровыми, и больными; ответ составляет 50 %. Древовидная схема этого подхода показана на рис. 4.3.

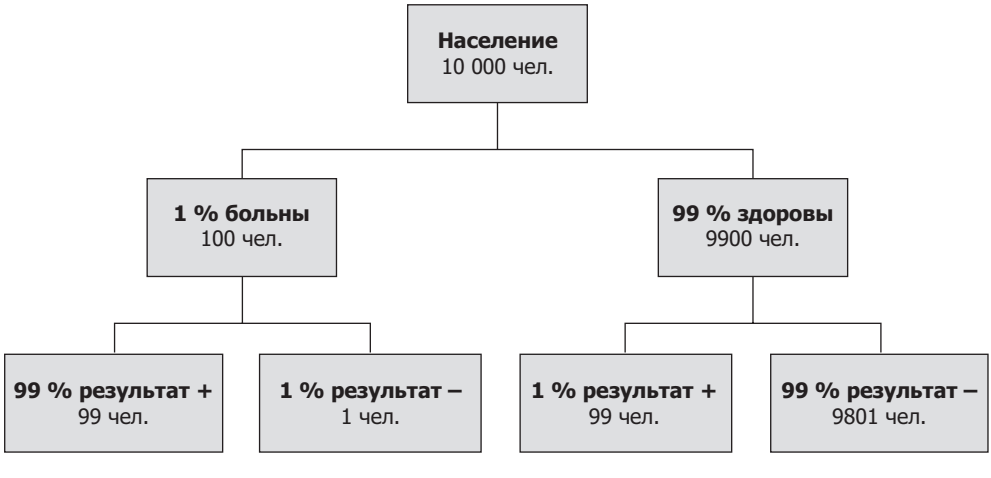


Рис. 4.3. Древовидная схема для развития интуиции

Сделаем это снова, используя вычурные обозначения, чтобы почувствовать себя умными.

Вспомните из базового курса статистики, что для заданных событий x и y существует связь между вероятностями наступления одного из событий (обозначается как $p(x)$ и $p(y)$), совместная вероятность (происходят оба события, обозначается как $p(x, y)$) и условная вероятность (событие x наступает при наступлении события y, обозначается как $p(x \mid y)$):

$$p(y \mid x)p(x) = p(x, y) = p(x \mid y)p(y).$$

Используя данное тождество, мы решаем уравнение для $p(y \mid x)$ (при $p(x) \neq 0$), чтобы прийти к тому, что называется *законом Байеса*:

$$p(y \mid x) = \frac{p(x \mid y)p(y)}{p(x)}.$$

Значение знаменателя $p(x)$ часто вычисляется неявно и поэтому может рассматриваться как «константа нормализации». В нашей текущей ситуации установите y для обозначения события «Я болен» (I am sick) или «болен» (sick) для краткости, установите x для обозначения события «анализ положительный» или «+» для краткости. Тогда мы действительно знаем или по крайней мере можем вычислить каждое значение:

$$p(sick \mid +) = \frac{p(x \mid sick)p(sick)}{p(+)} = \frac{0,99 \cdot 0,01}{0,99 \cdot 0,01 + 0,01 \cdot 0,99} = 0,50 = 50\,\%.$$

Спам-фильтр для отдельных слов

Как же использовать закон Байеса для создания хорошего фильтра спама? Подумайте об этом следующим образом: появление слова viagra увеличит вероятность того, что письмо — спам. Но это еще не окончательно. Нам нужно узнать остальное содержание данного письма.

Для начала будем сосредотачиваться только на одном слове за раз, его мы обозначим как word. Затем, применив закон Байеса, мы получим:

$$p(spam \mid word) = \frac{p(word \mid spam)p(spam)}{p(word)}.$$

Правая часть этого уравнения вычисляется с помощью уже достаточным образом подписанных данных. Если мы будем относиться к слову ham как к не спаму (spam)[1], то нам останется лишь вычислить $p(word \mid spam)$, $p(word \mid ham)$, $p(spam)$ и $p(ham) = 1 - p(spam)$. Это обусловлено тем, что мы можем вычислить знаменатель с помощью формулы, которую использовали ранее в примере с медицинским анализом, а именно:

[1] В англ. языке оба слова: и ham, и spam — обозначают ветчину (а spam еще обозначает спам). — *Примеч. пер.*

$$p(word) = p(word \mid spam)p(spam) + p(word \mid ham)p(ham).$$

Другими словами, мы упростили все до алгебраического упражнения: $p(spam)$ подсчитывает сообщения со спамом в сравнении со всеми сообщениями, $p(word \mid spam)$ — преобладание сообщений со спамом, включающих word, а $p(word \mid ham)$ считает преобладание сообщений о ветчине, содержащих word.

Чтобы проделать указанные действия самостоятельно, зайдите в Интернет и загрузите электронные письма компании Enron (https://www.cs.cmu.edu/~enron/). Построим фильтр спама в этом наборе данных. Это действительно значит, что мы создаем новый фильтр поверх того, который существовал для сотрудников компании Enron. Мы будем использовать их определение спама для обучения нашего фильтра. (Это, однако, означает следующее: если спамеры научились чему-либо с 2001 года, то нам не повезло.)

Мы могли бы быстро написать простой сценарий оболочки в bash, который запускает этот процесс, что и сделал Джейк. Фильтр загружает и распаковывает файл, создает папку, при этом каждый текстовый файл — это электронное письмо, спам и ветчина попадают в отдельные папки.

Рассмотрим некоторые основные статистические данные об электронном письме случайного сотрудника Enron. Мы можем насчитать 1500 сообщений со спамом и 3672 сообщения о ветчине, поэтому уже знаем $p(spam)$ и $p(ham)$. Используя инструменты командной строки, мы также можем подсчитать количество экземпляров слова meeting в папке со спамом:

```
grep -il meeting enron1/spam/*.txt | wc -l
```

Результат выполнения этой команды — 16. Проделав то же самое для папки `ham`, мы получим 153. Теперь можно вычислить вероятность того, что письмо является спамом, зная только о наличии в его содержании слова meeting:

$$\hat{p}(spam) = 1500 / (1500 + 3672) = 0,29,$$

$$\hat{p}(ham) = 0,71,$$

$$\hat{p}(meeting \mid spam) = 16 / 1500 = 0,0106,$$

$$\hat{p}(meeting \mid ham) = 153 / 3672 = 0,0416,$$

$$\hat{p}(meeting \mid spam) = \hat{p}(meeting \mid spam) \cdot \hat{p}(spam) / \hat{p}(meeting) =$$
$$= (0,0106 \cdot 0,29) / (0,0106 \cdot 0,29 + 0,0416 \cdot 0,071) = 0,09 = 9\%.$$

Обратите внимание: для выполнения этих действий нам не потребовалась воображаемая среда для программирования.

Далее мы можем попробовать:

- money: вероятность спама — 80 %;
- viagra: вероятность спама — 100 %;
- Enron: вероятность спама — 0 %.

Это показывает, что модель в ее нынешнем виде является примером переобучения; мы становимся слишком самоуверенными из-за необъективных данных. На самом ли деле любое письмо, содержащее слово viagra, — спам? Разумеется, можно написать обычное письмо с этим словом, а также мусорное со словом Enron.

Спам-фильтр, комбинирующий слова: наивный классификатор Байеса

Теперь сделаем это для всех слов. Каждое электронное письмо может быть представлено двоичным вектором, чей *j*-й экземпляр равен 1 или 0 в зависимости от того, появится ли *j*-е слово. Обратите внимание: это огромный вектор, учитывая общее количество слов, и мы, вероятно, захотим представить его с индексами слов, которые действительно появляются.

Вывод модели — вероятность того, что мы увидим данный вектор слов, учитывая сведения о том, что перед нами spam (или это ham). Обозначим вектор электронного письма как x и различные элементы xj, где j индексирует слова. В настоящее время мы можем обозначить spam переменной c — и получим следующую модель для $p(x|c)$, то есть вероятность того, что вектор письма будет выглядеть так, учитывая, что письмо является спамом:

$$p(x \mid c) = \prod_j \theta_{jc}^{xj}(1-\theta_{jc})^{(1-x_j)}.$$

Здесь θ — вероятность того, что отдельное слово присутствует в мусорном письме. В предыдущем подразделе мы видели, как алгебраически вычислить эту вероятность, и поэтому можем предположить, что отдельно и параллельно вычислили такую вероятность для каждого слова.

Мы моделируем слова *независимо* от других слов (это также называется «независимые испытания»), так что берем результат в правой части предыдущей формулы и не подсчитываем, сколько раз встречаются эти слова. Вот почему данный метод называется наивным: мы знаем, что на самом деле есть определенные слова, которые, как правило, появляются вместе, но мы игнорируем данный факт.

Итак, вернемся к уравнению. Стандартный трюк при работе с произведением вероятностей — взять логарифм от обеих части уравнения, чтобы суммировать вместо умножения:

$$\log(p(x \mid c)) = \sum_j x_j log(\theta_j / (1-\theta_j)) + \sum_j \log(1-\theta_j).$$

Логарифмирование полезно, так как перемножение малых величин может повлечь проблемы с числами.

Выражение $log(\theta_j / (1-\theta_j))$ не зависит от какого-то конкретного письма, только от слова, поэтому переименуем его в w_j и предположим, что вычислили и сохранили значение данного выражения. То же и с количеством $\sum_j \log(1-\theta_j) = w_0$. Теперь наше уравнение выглядит следующим образом:

$$\log(p(x \mid c)) = \sum_j x_j w_j + w_0.$$

Веса, отличающиеся от письма к письму, — x_j. Нам потребуется вычислять указанные значения отдельно для каждого письма, но это не очень сложно.

Мы можем собрать все известные сведения для вычисления $p(x \mid c)$, а затем использовать закон Байеса в целях получения оценки $p(c \mid x)$, а это именно то, что мы хотим. Другие выражения в законе Байеса легче, чем это, и не требуют отдельных вычислений для каждого письма. Мы также можем обойтись без вычисления всех выражений, если хотим лишь знать, является ли письмо скорее спамом (spam) или содержит информацию о ветчине (ham). Затем нужно вычислить только изменяющееся выражение.

Вы можете заметить, что это выглядит как линейная регрессия, но вместо вычисления коэффициентов w_j путем инвертирования огромной матрицы мы получаем веса из алгоритма наивного классификатора Байеса.

Этот алгоритм работает очень хорошо и «дешев» в обучении при наличии предварительно подписанного набора данных для тренировки. Получив в работу тонну электронных писем, мы всего лишь подсчитываем слова в сообщениях со спамом и без него. Если получим больше обучающих данных, то сможем легко увеличить наши подсчеты — и тем самым улучшить фильтр. На практике существует глобальная модель, которую мы персонализируем для отдельных лиц. Более того, есть много жестко закодированных дешевых правил, используемых прежде, чем письмо будет запущено в вычурную и медленную модель.

Мы предлагаем прочитать следующие работы, посвященные наивному классификатору Байеса:

- *Idiot's Bayes — not so stupid after all?* («Байес идиотов: в конце концов, не так уж и глупо?») (https://www.jstor.org/stable/1403452) — работа посвящена тому, почему этот алгоритм неплох, что связано с избыточностью языка;
- *Naive Bayes at Forty: The Independence Assumption in Information* («Наивный классификатор Байеса в сорок: допущение независимости в информации») (http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.11.8264);
- *Spam Filtering with Naive Bayes — Which Naive Bayes?* («Фильтрация спама с помощью наивного классификатора Байеса — какого наивного классификатора Байеса?») (http://www2.aueb.gr/users/ion/docs/ceas2006_paper.pdf).

Пофантазируем: сглаживание Лапласа

Помните переменную θ_j из предыдущего подраздела? Она означала вероятность увидеть заданное слово (индексированное *j*) в мусорном письме. Если об этом поразмыслить, то мы увидим, что данное значение — просто отношение двух чисел: $\theta_j = n_{js} / n_{jc}$, где n_{js} обозначает количество раз, когда это слово появляется в спам-письме, а n_{jc} — количество появлений данного слова в любом письме.

Сглаживанием Лапласа называют идею замены нашей прямолинейной оценки θ_j чем-то немного вычурным:

$$\theta_{jc} = (n_{jc} + \alpha) / (n_{jc} + \beta).$$

Мы можем зафиксировать значения переменных, скажем, $\alpha = 1$ и $\beta = 10$, с целью избежать возможности получения 0 или 1 для вероятности; ранее мы видели, что такое бывает, на примере слова viagra. Это кажется совершенно бессистемным, не правда ли? Ну если хотим пофантазировать, то можем воспринимать этот подход как эквивалент вычисления априорной вероятности и выполнения при этом оценки по принципу максимального правдоподобия. Проявим фантазию! Обозначив оценку по принципу максимального правдоподобия как *ML*, а набор данных — *D*, получим:

$$\theta_{ML} = \arg\max_{\theta} p(D \mid \theta).$$

Иными словами, вектор значений θ_j является ответом на вопрос: для какого значения θ были наиболее вероятными данные *D*? Если мы снова примем независимые испытания, как при нашей первой попытке использования наивного классификатора Байеса, то нужно выбрать θ_j, чтобы раздельно максимизировать следующую величину для каждого *j*:

$$\log(\theta_j^{n_{js}} (1 - \theta_j))^{n_{jc} - n_{js}}.$$

Если взять производную и установить ее в нуль, то получим:

$$\hat{\theta}_j = n_{js} / n_{jc}.$$

Другими словами, имеем то же самое, что было раньше. Так мы обнаружили следующее: максимальная оценка правдоподобия возвращает наш результат до тех пор, пока мы допускаем независимость.

Теперь добавим приоритет. Для текущего обсуждения можно убрать *j* из обозначений для ясности, но имейте в виду, что мы фиксируем *j*-е слово для обработки. Обозначим как *MAP* максимум апостериорной вероятности (https://ru.wikipedia.org/wiki/Оценка_апостериорного_максимума):

$$\theta_{MAP} = \arg\max_{\theta} p(\theta \mid D).$$

Это выражение также отвечает на вопрос: учитывая представленные данные, какой параметр θ является наиболее вероятным?

Здесь мы применим дух закона Байеса с целью преобразовать θMAP в нечто, в зависимости от константы, эквивалентное $p(D \mid \theta) \times p(\theta)$. Выражение $p(\theta)$ называется приоритетом, и, чтобы оно стало полезным, мы должны сделать предположение о его форме. Если предположим, что распределение вероятностей θ имеет вид $\theta^{\alpha}(1 - \theta)^{\beta}$ для неких α и β, то восстановим результат, сглаженный по методу Лапласа.

РЕЗОННО ЛИ ЭТО ДОПУЩЕНИЕ?

Напомним: θ — вероятность того, что слово находится в спаме, если оно пребывает в некоем электронном письме. С одной стороны, до тех пор, пока $\alpha > 0$ и $\beta > 0$, данное распределение обращается в нуль при 0 и 1. И это резонно: вы хотите, чтобы очень мало слов *никогда* не появлялось в спаме или появлялось в нем *всегда*.

С другой стороны, когда значения α и β велики, форма распределения сгруппирована посередине; это отображает приоритет, что большинство слов одинаково вероятны и для спама, и для нормальных сообщений. Данное утверждение тоже не похоже на правду.

Компромисс заключался в положительных, но небольших значениях α и β, например 1/5. Это не позволит вашему фильтру спама переусердствовать, имея неправильное представление. Конечно, вы могли бы облегчить приоритет, если бы обладали бо́льшим количеством более качественных данных. В целом сильные приоритеты нужны, только когда у вас недостаточно данных.

Сравнение наивного классификатора Байеса с k-БС

Иногда α и β называют псевдоотсчетом. Другое распространенное название — гиперпараметры. Они причудливы, но и просты. Вы, исследователь данных, должны установить значения двух указанных гиперпараметров в числителе и знаменателе для сглаживания — и это даст вам две ручки для настройки. Однако *k*-БС имеет лишь одну ручку, а именно *k* — количество соседей. Наивный классификатор Байеса — линейный, а *k*-БС — нет. Проклятие размерности и больших наборов признаков является проблемой для *k*-БС, тогда как наивный классификатор Байеса работает хорошо. Алгоритм *k*-БС не требует обучения (просто загрузите набор данных), в отличие от наивного классификатора Байеса. Оба алгоритма являются примерами обучения под наблюдением (данные помечаются предварительно).

Пример кода в оболочке bash

```
#!/bin/bash
#
# файл: enron_naive_bayes.sh
#
# описание: обучение простого фильтра спама на основе
# наивного классификатора Байеса и данных enron
#
# использование: ./enron_naive_bayes.sh <word>
#
# требования:
#   wget
#
# автор: jake hofman (gmail: jhofman)
#
# как использовать этот код
if [ $# -eq 1 ]
then
word=$1
else
echo "usage: enron_naive_bayes.sh <word>"
exit
fi

# если файл не существует, то загрузить из сети
if ! [ -e enron1.tar.gz ]
then
wget 'http://www.aueb.gr/users/ion/data/enron-spam/preprocessed/enron1.tar.gz'
fi

# если каталог не существует, то распаковать .tar.gz
if ! [ -d enron1 ]
then
tar zxvf enron1.tar.gz
fi

# перейти в enron1
cd enron1

# получить общее количество spam, ham и всех сообщений
Nspam='ls -l spam/*.txt | wc -l'
Nham='ls -l ham/*.txt | wc -l'
Ntot=$Nspam+$Nham

echo $Nspam spam examples
echo $Nham ham examples

# получить количество писем, содержащих слово в классах spam и ham
Nword_spam='grep -il $word spam/*.txt | wc -l'
Nword_ham='grep -il $word ham/*.txt | wc -l'

echo $Nword_spam "spam examples containing $word"
echo $Nword_ham "ham examples containing $word"
```

```
# вычислить вероятность с помощью bash-калькулятора "bc"
Pspam='echo "scale=4; $Nspam / ($Nspam+$Nham)" | bc'
Pham='echo "scale=4; 1-$Pspam" | bc'
echo
echo "estimated P(spam) =" $Pspam
echo "estimated P(ham) =" $Pham

Pword_spam='echo "scale=4; $Nword_spam / $Nspam" | bc'
Pword_ham='echo "scale=4; $Nword_ham / $Nham" | bc'
echo "estimated P($word|spam) =" $Pword_spam
echo "estimated P($word|ham) =" $Pword_ham

Pspam_word='echo "scale=4; $Pword_spam*$Pspam" | bc'
Pham_word='echo "scale=4; $Pword_ham*$Pham" | bc'
Pword='echo "scale=4; $Pspam_word+$Pham_word" | bc'
Pspam_word='echo "scale=4; $Pspam_word / $Pword" | bc'
echo
echo "P(spam|$word) =" $Pspam_word

# вернуться в исходный каталог
cd ..
```

Веб-агрегация: API и другие инструменты

Как исследователю данных вам не всегда просто передают некоторые данные и просят что-то выяснить на их основе. Часто вам нужно выяснить, как получить данные, необходимые, чтобы задать вопрос, решить проблему, провести некое исследование и т. д. Один из способов сделать это — применить API. В контексте данного обсуждения API (интерфейс прикладного программирования) — то, что сайты предоставляют разработчикам, чтобы те могли легко и просто загружать данные с сайта в стандартном формате. (API используются и по-другому, но для наших целей мы взаимодействуем с этими интерфейсами именно таким образом.) Обычно разработчик должен зарегистрироваться и получить «ключ», который является чем-то вроде пароля. Например, API издания The New York Times находятся здесь (http://developer.nytimes.com/).

Предупреждение об API

Всегда читайте условия использования API сайтов. Кроме того, некоторые сайты ограничивают то, к каким данным вы имеете доступ через интерфейсы, или то, как часто вы можете запрашивать данные бесплатно.

Идя этим путем, вы часто получаете данные в странных форматах, иногда в JSON, разные сайты предоставляют разные «стандартные» форматы.

Один из способов выйти за рамки этого — использовать язык YQL от Yahoo! (https://developer.yahoo.com/yql/?guccounter=1), позволяющий перейти на сайт плат-

формы Yahoo! Developer Network и писать SQL-подобные запросы, которые взаимодействуют со многими API на популярных сайтах:

```
select * from flickr.photos.search where text="Cat"
and api_key="lksdjflskjdfsldkfj" limit 10
```

Вывод является стандартным, и вам нужно только однажды разобрать его на языке Python.

Но что, если вы хотите получить данные, но подходящие API отсутствуют?

В этом случае будет полезным нечто наподобие расширения Firebug для браузера Firefox. Вы можете «проверять элемент» на любой веб-странице, а Firebug позволяет захватить поле внутри HTML. Фактически он дает доступ к полному HTML-документу, чтобы вы могли взаимодействовать с ним и редактировать. Таким образом, можно воспринимать HTML как карту страницы, а расширение Firebug — как своего рода путеводитель.

После того как вы нашли нужный материал в HTML, можете использовать `curl`, `wget`, `grep`, `awk`, `perl` и пр., чтобы на скорую руку написать сценарий оболочки для захвата необходимых элементов; это особенно подходит для одноразового захвата. Если хотите быть более систематичными, то можете написать сценарий и с помощью языков Python или R.

МЫСЛЕННЫЙ ЭКСПЕРИМЕНТ: РАСПОЗНАВАНИЕ ИЗОБРАЖЕНИЙ

Как определить, является ли изображение ландшафтом или это фото человека анфас?

Начните со сбора данных. Либо кому-то придется отметить нужные объекты, что подразумевает немало работы, либо потребуется скачать много фотографий с flickr (https://www.flickr.com/) и попросить фотографии, которые уже были отмечены.

Представляйте каждое изображение группированной гистограммой интенсивности RGB (красный, зеленый, синий). Иными словами, для каждого пиксела и каждого из красного, зеленого и синего цветов, которые являются основными цветами в пикселах, вы измеряете интенсивность, являющуюся числом в диапазоне от 0 до 255.

Затем нарисуйте три гистограммы, по одной для каждого базового цвета, показывая, сколько пикселов имеет какую интенсивность. Лучше делать группированную гистограмму, так у вас будет количество пикселов с заданной интенсивностью, например 0–51 и т. д. В результате для каждого изображения у вас есть 15 чисел, соответствующих трем цветам и пяти группам на цвет. Мы предполагаем, что каждое изображение имеет одинаковое количество пикселов.

Наконец, используйте алгоритм k-БС, чтобы определить, из скольких «синих» пикселов состоит пейзаж, а из скольких — портрет. Можете настроить гиперпараметры, которые в этом случае являются количеством групп, а также k.

Другие инструменты анализа, которые вы, возможно, захотите изучить:

- *lynx и lynx --dump* (http://lynx.browser.org/): хороши, если вы скучаете по 1970 годам. О, подождите, по 1992-му. В общем, неважно;
- *Beautiful Soup* (https://www.crummy.com/software/BeautifulSoup/): надежный, но медленный;
- *Mechanize* (http://mechanize.rubyforge.org/Mechanize.html) (или https://pypi.org/project/mechanize/): суперклассный инструмент, но не разбирает JavaScript;
- *PostScript* (https://ru.wikipedia.org/wiki/PostScript): классификация изображений.

Упражнение от Джейка: использование наивного классификатора Байеса для классификации статей

В этой задаче рассматривается применение наивного классификатора Байеса для классификации многоклассовых текстов. Во-первых, вы будете использовать New York Times Developer API для получения последних статей из нескольких разделов. Затем, применяя простую модель Бернулли для вхождений слов, станете внедрять классификатор, который, учитывая текст статьи из The New York Times, предскажет раздел, к которому та принадлежит.

Сначала зарегистрируйтесь для получения ключа к New York Times Developer API и запросите доступ к API поиска статей (Article Search API). Просмотрев документацию по API, напишите код, чтобы загрузить 2000 самых последних статей для каждого раздела: Arts («Искусство»), Business («Бизнес»), Obituaries («Некрологи»), Sports («Спорт») и World («В мире»). (Подсказка: используйте аспект `nytd_section_facet`, чтобы указать разделы статьи.) Консоль разработчика может быть полезна для быстрого изучения API. Ваш код должен сохранять статьи из каждого раздела в отдельный файл в формате с разделителями табуляции, где первый столбец — URL статьи, второй — ее заголовок, а третий — тело, возвращаемое API.

Затем реализуйте код для обучения простой модели Бернулли с применением наивного классификатора Байеса, используя эти статьи. Вы можете рассматривать документы в одной из категорий C, где метка i-го документа кодируется как $y_i \in 0, 1, 2... C$, например Arts = 0, Business = 1 и т. д., и документы представлены разреженной двоичной матрицей X, где $X_{ij} = 1$ указывает, что i-й документ содержит j-е слово в нашем словаре.

Вы обучаете модель путем подсчета слов и документов внутри классов для оценки θ_{jc} и θ_c:

$$\hat{\theta}_{jc} = \frac{n_{jc} + \alpha - 1}{n_c + \alpha + \beta - 2},$$

$$\hat{\theta}_c = \frac{n_c}{n},$$

где n_{jc} — количество документов класса c, содержащих j-е слово, n_c — количество документов класса c, n — общее количество документов, а выбранные пользователем гиперпараметры α и β — псевдоотсчеты, «сглаживающие» оценки параметров. Учитывая эти оценки и слова в документе x, вы вычисляете логарифмические коэффициенты вероятности для каждого класса (относительно базового класса $c = 0$), просто добавляя весовые значения слов в зависимости от класса, которые отображаются соответствующему члену смещения:

$$\log\left(\frac{p(y=c \mid x)}{p(y=0 \mid x)}\right) = \sum_j \hat{w}_{jc} x_j + \hat{w}_{0c},$$

где

$$\hat{w}_{jc} = \log \frac{\hat{\theta}_{jc}(1-\hat{\theta}_{j0})}{\hat{\theta}_{j0}(1-\hat{\theta}_{jc})},$$

$$\hat{w}_{0c} = \sum_j \log \frac{1-\hat{\theta}_{jc}}{1-\theta_{j0}} + \log \frac{\hat{\theta}_c}{\hat{\theta}_0}.$$

Ваш код должен считывать заголовок и тело каждой статьи, удалять ненужные символы (например, пунктуацию) и разбивать содержимое статьи на слова, отфильтровывая стоп-слова (приводятся в файле стоп-слов). На фазе обучения кода нужно использовать эти анализируемые признаки документа для оценки весов $\hat{w}$, где гиперпараметры α и β применяются в качестве входных данных. На фазе предсказания эти веса следует принимать как входящие параметры вместе с признаками для новых примеров, также должны выводиться апостериорные вероятности для каждого класса.

Оцените работу алгоритма при рандомизированном разбиении данных между обучающей и контрольной последовательностью в пропорции 50/50, включая точность и время выполнения. Прокомментируйте эффекты изменения значений α и β. Представьте свои результаты в таблице несоответствий (5×5), показывающей подсчеты для фактических и прогнозируемых разделов, где каждому документу присваивается наиболее вероятный раздел. Для каждого раздела сообщите десять самых информативных слов. Кроме того, представьте и прокомментируйте топ-10 «наиболее трудноклассифицируемых» статей в тестовом наборе.

Кратко обсудите ваши ожидания по обобщению изученного вами классификатора для других контекстов, например, в статьях из других источников или периодов времени.

Пример кода на языке R для работы с NYT API

```
# автор: Джаред Ландер (Jared Lander)
#
# жесткозакодированный вызов API
theCall <- "http://api.nytimes.com/svc/search/v1/
article?format=json&query=nytd_section_facet:
```

```
[Sports]&fields=url,title,body&rank=newest&offset=0
&api-key=Your_Key_Here"

# нам нужны пакеты rjson, plyr и RTextTools
require(plyr)
require(rjson)
require(RTextTools)

## сначала взглянем на отдельный вызов
res1 <- fromJSON(file=theCall)
# какова длина результата
length(res1$results)
# взглянем на первый элемент
res1$results[[1]]
# заголовок первого элемента
res1$results[[1]]$title
# первый элемент, конвертированный в data.frame, просматриваемый
# в data viewer
View(as.data.frame(res1$results[[1]]))

# конвертировать результаты вызова в data.frame,
# должно получиться десять строк и три столбца
resList1 <- ldply(res1$results, as.data.frame)
View(resList1)

## теперь создадим это для нескольких вызовов
# создать строку, где мы заменим раздел для первой %s
# и отодвинем для второй %s
theCall <- "http://api.nytimes.com/svc/search/v1/
article?format=json&query=nytd_section_facet:
[%s]&fields=url,title,body&rank=newest&offset=%s
&api-key=Your_Key_Here"
# создать пустой список для хранения трех наборов результатов
resultsSports <- vector("list", 3)
## пройти циклически по 0, 1 и 2 с вызовом API для каждого значения
for(i in 0:2)
{
    # сначала создать строку запроса, заменяя первую %s на Sports
    # и вторую %s на текущее значение i
    tempCall <- sprintf(theCall, "Sports", i)
    # создать запрос и получить результат json
    tempJson <- fromJSON(file=tempCall)
    # конвертировать json в данные 10 x 3 data.frame и сохранить данные
    # в список
    resultsSports[[i + 1]] <- ldply(tempJson$results,
    as.data.frame)
}
# конвертировать список в data.frame
resultsDFSports <- ldply(resultsSports)
# создать новую колонку, обозначить источник Sports
resultsDFSports$Section <- "Sports"

## повторить эти действия для arts
## в идеале вы должны делать это в более функциональном виде,
## но здесь мы приводим код для иллюстрации
```

```
resultsArts <- vector("list", 3)
for(i in 0:2)
{
    tempCall <- sprintf(theCall, "Arts", i)
    tempJson <- fromJSON(file=tempCall)
    resultsArts[[i + 1]] <- ldply(tempJson$results,
    as.data.frame)
}
resultsDFArts <- ldply(resultsArts)
resultsDFArts$Section <- "Arts"

# объединить оба набора в data.frame
resultBig <- rbind(resultsDFArts, resultsDFSports)
dim(resultBig)
View(resultBig)

## настало время для разбиения на слова
# создайте матрицу документ-термин на английском языке, удалив числа,
# стоп-слова и составные слова
doc_matrix <- create_matrix(resultBig$body, language="english",
removeNumbers=TRUE, removeStopwords=TRUE, stemWords=TRUE)
doc_matrix
View(as.matrix(doc_matrix))

# создайте наборы для обучения и тестирования
theOrder <- sample(60)
container <- create_container(matrix=doc_matrix,
labels=resultBig$Section, trainSize=theOrder[1:40],
testSize=theOrder[41:60], virgin=FALSE)
```

Исторический контекст: обработка естественного языка

Пример в этой главе, где в качестве исходных данных используется текст, является лишь верхушкой айсберга целой области исследований в информатике, называемой обработкой естественного языка. Типы проблем, которые можно решить с помощью обработки естественного языка, включают машинный перевод, где текст дается на одном языке, а алгоритм может перевести на другой, семантический анализ, тегирование частей речи и классификацию документов (примером которых является фильтрация спама). Исследования в этих областях относятся к 1950-м годам.

5 Логистическая регрессия

Автор этой главы — Брайан Далессандро. Он работает в Media6Degrees в качестве вице-президента по науке о данных, активно участвует в жизни исследовательского сообщества. Является также сопредседателем конкурса KDD (http://kdd2012.sigkdd.org/). M6D (также известный под названием Media 6 Degrees) — это стартап в Нью-Йорке в области онлайн-рекламы. На рис. 5.1 показан научный профиль Брайана — его ось *Y* масштабируется от клоуна до рок-звезды.

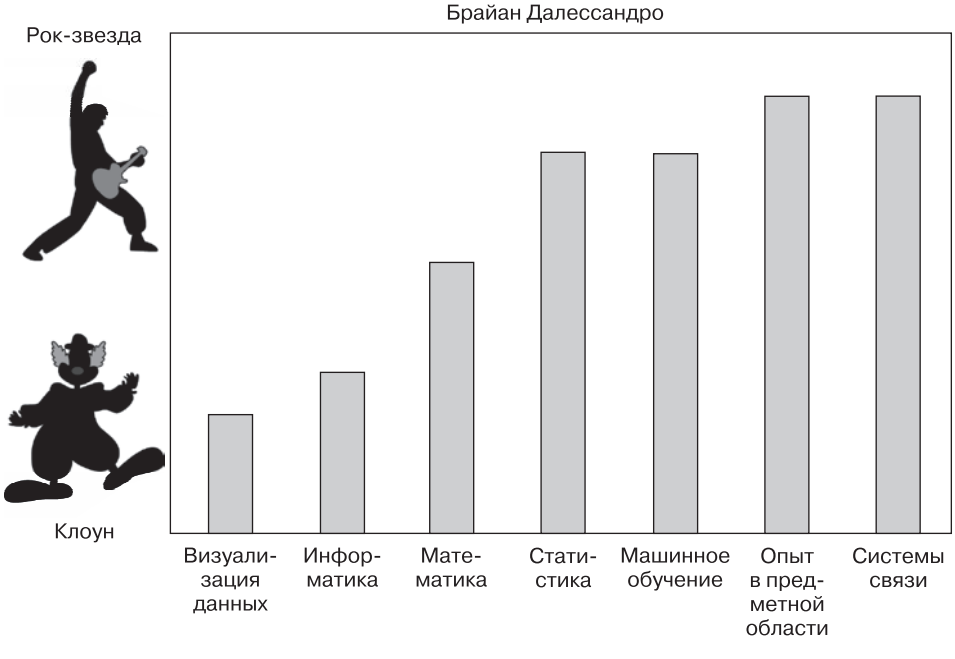


Рис. 5.1. Научный профиль Брайана

Брайан пришел поговорить с классом о логистической регрессии и оценке, но начал с двух мысленных экспериментов.

Мысленные эксперименты

1. Насколько иной была бы наука о данных, имей мы «великую объединенную теорию всего»? Эта теория была бы символическим объяснением того, как работает мир. Данный вопрос поднимает множество других вопросов.
   - Будь у нас такая теория, была бы нам нужна наука о данных в принципе?
   - Возможно ли теоретически существование такой теории? Лежат ли подобные теории только в области, скажем, физики, где мы можем ожидать точного возвращения кометы, которую видим раз в столетие?
   - В чем критическое различие между физикой и наукой о данных, которое делает такую теорию неправдоподобной?
   - Только ли в точности дело? Или, в общем, насколько мы можем себе представить то, что *можно* объяснить? Потому ли это, что мы прогнозируем поведение человека, *на которое* могут повлиять наши прогнозы, создавая цикл обратной связи?
   - Возможно, было бы полезно воспринимать науки как непрерывный спектр, где физика находится в правом его конце, а по мере перемещения влево хаотичность растет, поскольку повышается степень случайности (и зарплата). И где в этом спектре экономика? Маркетинг? Финансы?

     Если бы мы могли моделировать науку о данных так же хорошо, как уже умеем моделировать физику, мы бы действительно *знали*, когда люди щелкнут кнопкой мыши на том или ином объявлении, настолько же точно, как мы знаем, куда приземлится марсоход Mars Rover. Тем не менее существует общее мнение о том, что реальный мир не настолько хорошо осознан, и мы не ожидаем, что в будущем его изучат досконально.
2. В каком смысле наука о данных заслуживает слова «наука» в своем названии?

   Никогда не недооценивайте силу творчества — у людей часто есть видение, которое они могут описать, но нет метода, способного помочь осуществить увиденное. Как исследователь данных вы должны превратить это видение в математическую модель с определенными операционными ограничениями. Вам нужно поставить четко сформулированную задачу, утвердить систему мер и оптимизировать ее. Вы также должны убедиться, что действительно ответили на изначальный вопрос.

   В науке о данных есть *искусство*: оно проявляется в переводе человеческих проблем в математический контекст Data Science и обратно.

   Но всегда есть несколько способов осуществить такой перевод: более одной вероятной модели, более одной применимой системы мер и, возможно, более одной оптимизации. Таким образом, *наука* в даталогии — изначальные необработанные данные, ограничения и формулировка задачи; все перечисленное показывает, как перемещаться по этому лабиринту и делать лучшие выборы. Каждый предлагаемый вами вариант конструкции может быть сформулирован

как гипотеза, против которой вы будете использовать тщательное тестирование и экспериментирование, чтобы проверить или опровергнуть ее.

Этот процесс, с помощью которого человек формулирует четко определенную гипотезу, а затем проверяет ее, может в определенных случаях подняться до уровня науки. В частности, в Data Science используется следующий научный метод:

- придерживайтесь текущей наилучшей модели;
- при появлении новой идеи прототипа поставьте эксперимент, в котором бы соревновались две лучшие модели;
- «промойте и повторите»[1] (избегая переобучения).

Классификаторы

В этом разделе основное внимание уделяется процессу выбора *классификатора*. Классификация включает в себя сопоставление точек данных с конечным набором меток или с вероятностью конкретной метки или меток. В предыдущих главах мы уже видели некоторые образцы алгоритмов классификации, таких как наивный классификатор Байеса и *k*-ближайшие соседи (*k*-БС). В табл. 5.1 приведены несколько примеров возможного использования классификации.

Таблица 5.1. Пример классификации: вопросы и ответы

Кто-нибудь щелкнет кнопкой мыши на этой рекламе?	0 или 1 (нет или да)
Какое это число (распознавание изображений)?	0, 1, 2 и т. д.
О чем эта новостная статья?	«Спорт»
Это спам?	0 или 1
Это лекарство помогает при головных болях?	0 или 1

Отныне мы будем говорить только о бинарной классификации (0 или 1).

В этой главе мы говорим о логистической регрессии, но есть и другие доступные алгоритмы классификации, включая деревья принятия решений, случайные леса (которые мы рассмотрим в главе 7) и машины опорных векторов и нейронные сети (в данной книге не рассматриваются).

В общем и целом, получив данные, задачу классификации из реального мира и ограничения, вам необходимо определить следующее.

1. Какой классификатор использовать.
2. Какой метод оптимизации применять.

[1] Автор обыгрывает часто встречающуюся на шампунях фразу. — *Примеч. пер.*

3. Какую функцию потерь минимизировать.
4. Какие признаки взять из данных.
5. Какую меру оценки задействовать.

Обсудим первый пункт: как узнать, какой классификатор выбрать? Одна из возможностей — попробовать все и выбрать наиболее производительный. Это нормально, если у вас нет ограничений или если вы их игнорируете. Но обычно ограничения очень важны: у вас может быть множество данных либо мало времени или и то и другое. Это то, о чем люди говорят мало. Рассмотрим ряд ограничений, которые являются общими для большинства алгоритмов.

Время выполнения

Скажем, вам нужно обновить 500 моделей в день. Так обстоит дело в M6D, где их модели в конечном итоге являются решениями о торгах. В этом случае различные проблемы скорости становятся существенными. Во-первых, сколько времени требуется для обновления модели, а во-вторых, сколько его нужно, чтобы использовать модель для принятия решения при ее наличии. Этот второй вид проблем обычно более важен и называется *временем выполнения*.

Некоторые алгоритмы работают медленно во время выполнения. Например, рассмотрим метод k-БС: получив новую точку данных в каком-то крупномерном пространстве, вам действительно нужно найти k ближайших точек. В частности, следует загрузить все ваши данные в память.

Линейные модели, напротив, очень быстры как для обновления, так и для использования во время работы. В главе 6 вы увидите, что можете продолжать оценивать составные части и просто обновлять модель, добавляя новые данные; это выполняется быстро и не требует хранения старых данных в памяти. Когда у вас есть линейная модель, получение ответа — всего лишь вопрос сохранения вектора коэффициентов в машине времени выполнения и вычисления продукта одной точки против пользовательского вектора признаков.

Одно недооцененное ограничение того, что вы исследователь данных, — ваше собственное понимание алгоритма. Строго спросите себя, понимаете ли алгоритм по-настоящему? *В самом деле*? Признаться себе в непонимании нормально.

Можно быть хорошим исследователем данных, не обязательно являясь гуру каждого алгоритма. По правде говоря, получение наилучших результатов обучения алгоритма часто требует его глубокого знания. Иногда вам нужно дополнительно настроить алгоритм, чтобы он соответствовал вашим данным. Общая ошибка для людей, не полностью знакомых с алгоритмом, — переобучение, которое они воспринимают как дополнительную настройку.

Интерпретируемость

Вам часто может требоваться умение интерпретировать свою модель для целей бизнеса. Деревья принятия решений очень легко интерпретировать, тогда как случайные леса — нет, хотя это почти одно и то же. Полное объяснение последних может занять экспоненциально больше времени, чем объяснение первых. Если у вас нет 15 лет для того, чтобы лучше понять результат, вы можете пожертвовать некоторой точностью ради облегчения понимания.

Например, по закону некоторых стран компании-эмитенты кредитных карт должны быть в состоянии объяснить свои решения об отказе в кредите (https://www.consumer.ftc.gov/articles/0347-your-equal-credit-opportunity-rights#right), поэтому деревья решений имеют больше смысла, чем случайные леса. Там, где вы работаете, может не быть такого закона, но тем не менее иметь под рукой более простой способ объяснить решение модели может быть существенно для вашего бизнеса.

Масштабируемость

Как насчет масштабируемости? Есть три аспекта, которые вы должны учитывать при изучении масштабируемости.

1. Время обучения: сколько времени требуется для обучения модели?
2. Время подсчета баллов: сколько времени уйдет на присвоение новому пользователю оценки в баллах после запуска модели в работу?
3. Хранение модели: сколько памяти задействует промышленная модель?

Вот полезная статья, на которую следует обратить внимание при сравнении моделей: *An Empirical Comparison of Supervised Learning Algorithms* («Эмпирическое сравнение алгоритмов обучения под наблюдением») (http://www.niculescu-mizil.org/papers/empirical.icml06.pdf). Из нее мы узнали:

- что более простые модели более интерпретируемы, но не настолько хороши;
- ответ на вопрос о том, какой алгоритм работает лучше всего, зависит от проблемы;
- он также зависит от ограничений.

Тематическое исследование логистической регрессии M6D

Будучи исследователями данных, Брайан и его команда работают в M6D с тремя основными проблемами.

1. Конструирование признаков: определение того, какие признаки и как использовать.

2. Прогноз конверсии на уровне пользователя: прогнозирование времени перехода, осуществленного кем-то.
3. Ставки: сколько стоит показ этого объявления данному пользователю?

Это тематическое исследование фокусируется на второй проблеме. В M6D используется логическая регрессия для решения проблемы, поскольку подобная модель очень масштабируема и отлично подходит для двоичных результатов, таких как щелчки кнопкой мыши на рекламном баннере.

Модели переходов

Задача M6D — сводить своих клиентов, представляющих собой рекламные компании, с отдельными пользователями. Если не вдаваться в подробности, то рекламные компании хотят настроить таргетинг объявлений на пользователей, исходя из вероятности того, что пользователь щелкнет на баннере. В первую очередь обсудим, какие данные есть у исследователей, а затем то, как бы вы построили модель с помощью этих данных.

M6D отслеживает посещаемые сайты, но исследователи данных не смотрят на содержимое этих страниц. Вместо этого они берут URL страницы и хешируют его в некую случайную строку. Таким образом они накапливают информацию о пользователях, и фиксируют ее в векторе. В качестве примера рассмотрим пользователя `u` в течение выбранного периода времени:

```
u = < &ltfxyz, 123, sdqwe, 13ms&gtg >
```

Это значит, что `u` посетил четыре сайта, а URL, которые он посетил, были хешированы до вышеприведенных строк. После сбора такой информации для всех пользователей исследователи создают гигантскую матрицу, столбцы которой соответствуют сайтам, а строки — пользователям, а каждой записи задается значение 1, если данный пользователь перешел на данный сайт. Обратите внимание: это разреженная матрица, поскольку большинство людей посещает не так уж много сайтов.

Чтобы сделать это задачей классификации, исследователи должны располагать *классами*, которые они будут пытаться предсказать. Скажем, в этом случае они хотят предсказать, перейдет ли данный пользователь по объявлению о продаже обуви. Итак, есть два класса: «пользователи, которые перешли по объявлению о продаже обуви» и «пользователи, которые не перешли по объявлению о продаже обуви». В обучающем наборе данных, кроме описанной разреженной матрицы, исследователи также создадут переменную или столбец для записи меток. Например, они обозначают поведение «перешел по объявлению о продаже обуви» как 1 и «не перешел» как 0. Сразу после того, как исследователи обучат классификатор на наборе данных, для любого нового пользователя классификатор будет прогнозировать на основе показателей (история браузера пользователя, отраженная в матрице URL), перейдет ли этот новый пользователь по объявлению (метка).

Теперь ваша очередь: цель — из обучающего набора построить и обучить модель. Напомним, что из главы 4 вы узнали о *классификаторах спама*, где слова — характеристики. Но вы не особенно заботились о значении этих слов. Признаки также могут быть и строками. Как только вы присвоите метки, что было описано ранее, задача будет выглядеть аналогичной классификации спама, поскольку у вас есть двоичный результат с большой разреженной бинарной матрицей, отражающей некие показатели. Теперь вы можете опираться на хорошо разработанные алгоритмы, созданные для обнаружения спама.

Вы уменьшили свою текущую проблему до ранее решенной проблемы! В предыдущей главе мы показали, как решить эту проблему с помощью наивного классификатора Байеса, но здесь сосредоточимся на использовании логистической регрессии в качестве модели.

Результат модели логистической регрессии в данном контексте — *вероятность* осуществления заданного перехода. Аналогично фильтры спама действительно определяют *вероятность* того, что заданное письмо является спамом. Вы можете использовать эти вероятности напрямую или же установить порог. Если вероятность выше его (скажем, 0,75), то прогнозируете переход (то есть показываете объявление); в противном случае решаете, что объявление показывать не стоит. Суть здесь заключается в следующем: в отличие от линейной регрессии, которая делает все возможное для прогнозирования фактического значения, задачей логистической регрессии является не прогнозирование фактического значения (0 или 1), а показ вероятности.

На таком наборе данных технически возможно реализовать линейную модель типа «линейная регрессия» (то есть язык R позволит сделать это и не сломается и не сообщит, что вы не должны этого делать). Однако одна из проблем с линейной моделью, подобной линейной регрессии, заключается в том, что она будет давать прогнозы ниже 0 и выше 1, поэтому их нельзя непосредственно интерпретировать как вероятности.

Математическая основа

До сих пор мы видели, что красота логистической регрессии заключается в выводе значений, ограниченных 0 и 1, благодаря чему эти значения можно непосредственно интерпретировать как вероятности. Немного поразмышляем над математикой, лежащей в основе всего этого. Вам нужна функция, которая берет данные и преобразует их в единственное значение, ограниченное внутри замкнутого интервала [0; 1]. В качестве примера функции, ограниченной между 0 и 1, рассмотрим инвертированную логит-функцию, показанную на рис. 5.2.

$$P(t) = \text{logit}^{-1}(t) \equiv \frac{1}{(1+e^{-t})} = \frac{e^t}{1+e^t}.$$

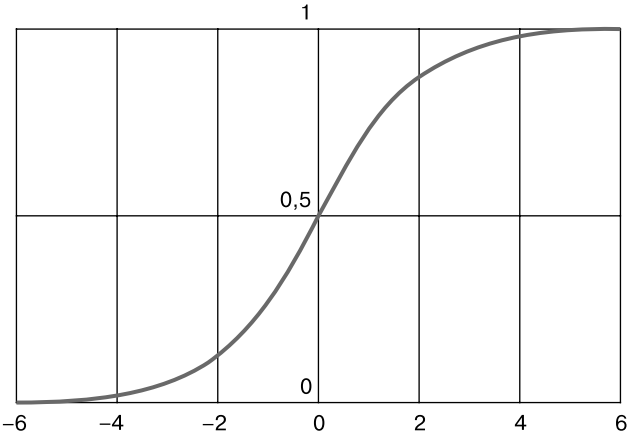


Рис. 5.2. Инвертированная логит-функция

ЛОГИТ-ФУНКЦИЯ И ИНВЕРТИРОВАННАЯ ЛОГИТ-ФУНКЦИЯ

Логит-функция принимает значения x в диапазоне [0; 1] и преобразует их в значения y вдоль всей реальной линии:

$$\text{logit}(p) = \log(\frac{p}{1-p}) = \log(p) - \log(1-p).$$

Инвертированная логит-функция выполняет обратное действие: принимает значения x вдоль всей реальной линии и преобразует их в значения y в диапазоне [0; 1].

Обратите внимание: если значение t велико, то e^{-t} мизерно, поэтому знаменатель близок к 1 и общее значение близко к 1. Аналогично, когда t мизерно, e^{-t} велико, поэтому знаменатель большой, что делает функцию близкой к нулю. Таким образом, перед нами инвертированная логит-функция, которую вы будете использовать, чтобы начать выводить модель логистической регрессии. Для моделирования данных вам нужно работать с чуть более общей функцией, выражающей связь между данными и вероятностью перехода. Начните с определения:

$$P(c_j \mid x_j) = \left[\text{logit}^{-1}(\alpha + \beta^\tau x_i)\right]^{c_i} \cdot \left[1 - \text{logit}^{-1}(\alpha + \beta^\tau x_i)\right]^{1-c_i}.$$

Здесь c_i — метки или классы (был переход или нет), а x_i — вектор признаков для пользователя i. Заметим: c_i может равняться только 1 или 0, а это значит, что если $c_i = 1$, то второй член отменяется, и получается:

$$P(c_j \equiv 1 \mid x_j) = \frac{1}{1 + e^{-(\alpha + \beta^\tau x_i)}} = \text{logit}^{-1}(\alpha + \beta^\tau x_i).$$

Точно так же, если $c_i = 0$, то первый член отменяется и появляется:

$$P(c_j = 0 \mid x_j) = 1 - \text{logit}^{-1}(\alpha + \beta^{\tau} x_j).$$

Чтобы сделать из этого линейную модель в результатах c_i, возьмите логарифм коэффициентов вероятности (https://ru.wikipedia.org/wiki/Отношение_шансов):

$$\log(P(c_j = 1 \mid x_i) / (1 - P(c_i = 1 \mid x_i))) = \alpha + \beta^{\tau} x_i.$$

Данное выражение также можно записать следующим образом:

$$\log it(P(c_i = 1 \mid x_i)) = \alpha + \beta^{\tau} x_i.$$

Если вам кажется, будто мы здесь немного ходим по кругу (последнее уравнение также подразумевалось предыдущими), то это потому, что в действительности так и есть. Цель наших примеров было показать, как переходить между вероятностями и линейностью.

Таким образом, логит вероятности того, что пользователь перейдет по объявлению о продаже обуви, моделируется как линейная функция признаков, представлявших собой URL, которые посетил i. Эта модель называется моделью *логистической регрессии*.

Параметр α есть то, что мы называем *базовой ставкой* (еще безусловной вероятностью 1 или переходом), ничего не знающей о векторе признаков x_i заданного пользователя. В случае измерения вероятности перехода среднего пользователя по объявлению базовая ставка будет соответствовать коэффициенту откликов, то есть тенденции всех пользователей переходить по объявлениям. Обычно это порядка 1 %.

Если у вас не было информации о вашей конкретной ситуации за исключением базовой ставки, то среднее предсказание было бы задано просто α:

$$P(c_j = 1) = \frac{1}{1 + e^{-\varepsilon}}.$$

Переменная β определяет наклон логит-функции. Обратите внимание: в целом это вектор, длина которого определяется количеством признаков, используемых вами для каждой точки данных. Вектор β определяет степень, в которой определенные признаки являются маркерами для увеличения или уменьшения вероятности перехода по объявлению.

Оценка α и β

Ваша непосредственная цель моделирования — применить обучающие данные, найти оптимальные значения переменных α и β. В общем и целом вам необходимо решить эту проблему с помощью оценки максимального правдоподобия и алгоритма выпуклой оптимизации, поскольку функция правдоподобия является выпуклой. Вы не сможете просто использовать производные и векторные вычисления, как

было с линейной регрессией, — это сложная функция ваших данных, и, в частности, не существует решения закрытой формы.

Обозначим через θ пару {α, β}. *Функция правдоподобия L* определяется следующим образом:

$$L(\Theta \mid X_1, X_2, ..., X_n) = P(X \mid \Theta) = P(X_1 \mid \Theta) \cdot .... \cdot P(X_n \mid \Theta),$$

где вы предполагаете, что точки данных X_i независимы, где $i = 1... n$ представляют ваши n пользователей. Это предположение о независимости соответствует следующему утверждению: поведение нажатия для любого данного пользователя не влияет на поведение нажатий всех других пользователей: в таком случае «поведение нажатий» означает «вероятность перехода». Это относительно безопасное предположение в заданный момент времени, но не навсегда. (Помните: предположение о независимости есть то, что позволяет выразить функцию правдоподобия как произведение плотностей для каждого из n наблюдений.)

Затем вы ищете параметры, которые максимизируют вероятность, наблюдая при этом ваши данные:

$$\Theta_{MLE} = \arg\max_{\Theta} \prod_1^n P(X_i \mid \Theta).$$

После установки $p_i = 1/(1+e^{-(\alpha+\beta^T x_i)})$ вероятность единичного наблюдения $P(X_i \mid \Theta)$ будет:

$$p_i^{c_i} \times (1-p_i)^{1-c_i}.$$

Таким образом, совместив все, мы получим:

$$\Theta_{MLE} = \arg\max_{\Theta} \prod_1^n p_i^{c_i} \cdot (1-p_i)^{1-c_i}.$$

Как мы теперь максимизируем вероятность?

Столкнувшись с этой ситуацией с линейной регрессией, вы взяли производную от вероятности по отношению к α и β, установили ее равной нулю — и решили задачу. Но если вы попробуете подобные действия и в данной ситуации, то будет невозможно получить решение с закрытой формой. Ключ состоит в том, что значения, которые максимизируют вероятность, максимизируют и логарифмическую вероятность, что эквивалентно минимизации *отрицательной* логарифмической вероятности. Таким образом, вы преобразуете задачу, чтобы найти минимум отрицательной логарифмической вероятности.

Теперь нужно решить, какой метод оптимизации использовать. Как оказалось, в разумных условиях оба описанных ниже метода оптимизации сходятся к *глобальному максимуму*, когда вообще сходятся. «Разумное условие» состоит в том, что переменные не зависят от линейности и, в частности, гарантируют положительную определенность матрицы Гессиана (https://ru.wikipedia.org/wiki/Гессиан_функции).

Подробнее об оценке максимального правдоподобия

Мы понимаем, что представили данную тему немного сжато. При желании получить более подробную информацию относительно оценки максимального правдоподобия (вероятности) предлагаем посмотреть информацию в работе Statistical Inference («Статистический анализ») Каселлы и Бергера. Или если вы хотите получить более подробные сведения о линейной алгебре в целом, то прочтите Linear Algebra and Its Applications («Линейная алгебра и ее применение») Гильберта Стрэнга (Gilbert Strang).

Метод Ньютона

Вы можете использовать численные методы, чтобы найти глобальный максимум, следуя рассуждениям, которые лежат в основе метода Ньютона (https://ru.wikipedia.org/wiki/Метод_Ньютона): довольно хорошо аппроксимировать функцию с помощью первых нескольких членов ряда Тейлора (https://ru.wikipedia.org/wiki/Ряд_Тейлора).

В частности, при заданном размере шага γ вы вычисляете локальный градиент $\nabla\Theta$, соответствующий первой производной, и матрицу Гессе H, соответствующую второй. Вы складываете их — и каждый шаг алгоритма выглядит примерно так:

$$\theta_{n+1} = \theta_n - \gamma H^{-1} \times \nabla\theta.$$

В методе Ньютона используется кривизна логарифма вероятности для выбора подходящего направления шага. Обратите внимание: этот расчет включает инверсию матрицы Гессе $k \times k$, что плохо при большом количестве признаков, например, 10 000 или т. п. Обычно у вас нет такого множества признаков, но это вполне реально.

На практике вы никогда бы не инвертировали матрицу Гессе, вместо этого решили бы уравнение вида $Ax = y$, которое намного более устойчиво по сравнению с поиском A^{-1}.

Стохастический градиентный спуск

Еще один возможный метод максимизации вероятности (или минимизации отрицательной логарифмической вероятности) — стохастический градиентный спуск (https://en.wikipedia.org/wiki/Stochastic_gradient_descent и https://ru.wikipedia.org/wiki/Градиентный_спуск). Он приближает градиент, используя одно наблюдение за раз. Алгоритм обновляет текущие наилучшие параметры всякий раз, когда видит новую точку данных. Хорошая новость в том, что нет большой инверсии матрицы и метод хорошо работает как с огромными данными, так и с рассеянными признаками; это большой успех Mahout (http://mahout.apache.org/) и Vowpal Wabbit (http://hunch.net/~vw/) — двух проектов с открытым исходным кодом, позволяющих широкомасштабное машинное обучение по различным алгоритмам. Плохая новость заключается в том, что это не такой уж хороший оптимизатор и он очень зависит от размера шага.

Реализация

На практике вам не нужно самостоятельно писать код итеративного метода взвешивания с применением методов наименьших квадратов или стохастического градиента; это реализуется в языке R или любом пакете, реализующем логистическую регрессию. Предположим, у вас есть набор данных, в котором были следующие первые пять строк:

```
click url_1  url_2  url_3  url_4  url_5
1     0      0      0      1       0
1     0      1      1      0       1
0     1      0      0      1       0
1     0      0      0      0       0
1     1      0      1      0       1
```

Назовите эту матрицу обучающей, а введите команду в R:

```
fit <- glm(click ~ url_1 + url_2 + url_3 + url_4 + url_5,
           data = train, family = binomial(logit))
```

Оценка

Вернемся к общей картине: ранее в этой главе мы сказали, что при столкновении с проблемой классификации вам нужно сделать много выборов. Один из таких выборов — то, как вы собираетесь оценивать свою модель. Мы обсуждали это уже в главе 3 в контексте линейной регрессии и *k*-БС, а также в предыдущей главе, когда говорили о наивном классификаторе Байеса. Обычно мы используем разные оценочные показатели для отличных моделей в несходных контекстах. Даже логистическая регрессия применима в нескольких контекстах, и, в зависимости от контекста, ее можно оценить по-разному.

Во-первых, рассмотрим контекст применения логистической регрессии как модели ранжирования. Речь о следующем: вы пытаетесь определить *порядок*, в котором показываете рекламу или элементы пользователю, на основе вероятности того, что этот пользователь перейдет по рекламе. Вы можете задействовать логистическую регрессию для оценки вероятностей, а затем ранжировать объявления или элементы в порядке убывания вероятности перехода на основе вашей модели. Если хотите знать, насколько хороша ваша модель в обнаружении относительного рейтинга (обратите внимание: в этом случае вы можете не беспокоиться об абсолютных оценках), то можете посмотреть на следующие аспекты.

- *Площадь под кривой рабочей характеристики приемника (AUC)* (https://ru.wikipedia.org/wiki/ROC-кривая). В теории обнаружения сигнала кривая рабочей характеристики приемника, или кривая ROC, определяется как график уровней истинно положительных результатов против уровней ложноположительных результатов для задач двоичной классификации при изменении порогового значения. В частности, если вы возьмете свой обучающий набор, оцените элементы в соответствии с их вероятностями, измените пороговое значение

(от ∞ до −∞), определяющее, следует ли классифицировать элемент как 1 или 0, и продолжите строить график уровней истинно положительных результатов против уровней ложноположительных результатов, то получите кривую ROC. Область под данной кривой, называемая AUC, является способом измерения успеха классификатора или сравнения двух классификаторов.

- *Площадь под кумулятивной кривой подъема* (http://mrvar.fdv.uni-lj.si/pub/mz/mz3.1/vuk.pdf). Альтернатива — область под кумулятивной кривой подъема, которая часто применяется в адресном маркетинге и фиксирует, во сколько раз лучше задействовать модель, чем не использовать ее (то есть просто выбирать наугад).

Мы вернемся к обеим кривым в главе 13.

ВНИМАНИЕ: ПЕТЛЯ ОБРАТНОЙ СВЯЗИ!

Допустим, вам нужно создать логистическую регрессию для ранжирования объявлений или элементов на основе переходов и показов. Подумаем о том, что это значит с точки зрения генерирования данных. Итак, скажем, вы поместили объявление о геле для волос выше объявления о дезодоранте, а затем еще больше людей щелкнули на вашем объявлении о геле. Это потому, что вы поместили его сверху, или потому, что больше людей хотят гель для волос? Как вы можете передавать эти данные в будущие итерации вашего алгоритма, учитывая, что потенциально именно вы самостоятельно и вызвали переходы и это не имеет никакого отношения к качеству объявления? Одно из решений — всегда регистрировать *позицию* или *рейтинг*, который вы показывали в объявлениях, а затем использовать *их* как один из показателей в вашем алгоритме. Таким образом, вы бы затем моделировали вероятность перехода как функцию позиции, вертикаль, бренд или любую другую характеристику по желанию. Затем можете использовать параметр, оцененный для позиции, и применить его в качестве «нормализатора позиции». В Google есть целое подразделение под названием «Качество рекламы», которое занимается такими проблемами, поэтому один маленький абзац не может осветить все нюансы этого процесса.

Во-вторых, предположим, что вы используете логистическую регрессию для целей классификации. Помните: хотя наблюдаемый результат был двоичным (1,0), результат логистической модели — это вероятность. Чтобы применить такую модель для целей классификации, для любого данного непомеченного элемента вы получите прогнозируемую вероятность перехода. Далее, чтобы свести к минимуму уровень ошибочной классификации: если прогнозируемая вероятность того, что метка должна быть 1 (переход), > 0,5, то элемент помечается 1 (переход), а в противном случае — 0. У вас есть несколько вариантов оценки качества модели, некоторые из вариантов мы уже обсуждали в главах 3 и 4, но расскажем о них здесь снова, чтобы вы увидели их повсеместность.

- *Подъем*. Насколько больше людей совершают покупки или переходят по объявлению благодаря модели (как только мы запустим ее в промышленное использование).

- *Безошибочность* (https://en.wikipedia.org/wiki/Accuracy_and_precision). Насколько часто прогнозируется правильный результат, как обсуждалось в главах 3 и 4.
- *Точность* (https://en.wikipedia.org/wiki/Precision_and_recall и https://ru.wikipedia.org/wiki/Информационный поиск#Точность_(precision)). Это (количество истинно положительных результатов) / (количество истинно положительных результатов + + количество ложноположительных результатов).
- *Полнота* (https://en.wikipedia.org/wiki/Precision_and_recall). Это (количество истинно положительных результатов) / (количество истинно положительных результатов + количество ложноотрицательных результатов).
- *F-мера* (https://en.wikipedia.org/wiki/F1_score). Мы еще не рассказывали вам об этом показателе. По сути, он сочетает точность и полноту в единое значение. Это гармоническое среднее точности и полноты, вследствие чего (2 × точность × × полнота) / (точность + полнота). Есть обобщения этого, которые существенно меняют то, насколько вы оцениваете вес одного или другого показателя.

Наконец, для оценки плотности в случае необходимости знать реальную вероятность, а не относительную оценку, мы бы посмотрели на следующие показатели.

- *Среднеквадратичная погрешность* (https://en.wikipedia.org/wiki/Mean_squared_error и https://ru.wikipedia.org/wiki/Среднеквадратическое_отклонение). Мы обсудили данный показатель, когда говорили о линейной регрессии. Напомним, что это среднеквадратичное расстояние между прогнозируемыми и фактическими значениями.
- *Квадратичная погрешность*. Квадратный корень среднеквадратичной погрешности.
- *Средняя абсолютная погрешность* (https://en.wikipedia.org/wiki/Mean_absolute_error). Это вариация среднеквадратичной погрешности, является просто средним значением абсолютной величины разницы между прогнозируемыми и фактическими значениями.

В общем и целом трудно сравнить кривые подъема, но вы можете сравнить AUC (площади под кривой рабочей характеристики приемника) — «инварианты базового уровня». Другими словами, если вы используете показатель кликабельности (click-through rate, CTR) от 1 до 2 % , то это 100%-ный подъем; однако увеличение с 4 до 7 % покажет меньший подъем, но большую эффективность. Показатель AUC работает лучше при необходимости провести сравнение.

Тесты оценки плотности говорят, насколько хорошо вы обучаете условной вероятности. В рекламе это может возникнуть в ситуации, когда каждый показ рекламы стоит *c* долларов, а за каждую конверсию вы получаете *q* долларов. Вам потребуется настроить таргетинг на каждую конверсию с положительным ожидаемым значением, то есть всегда, когда:

$$P(\text{Конверсия} \mid X) \times \$q > \$c.$$

Но для этого нужно убедиться в *безошибочности* вероятностной оценки слева, в данном случае речь о том, что такие значения, как среднеквадратичная погрешность оценочного значения, малы. Обратите внимание: модель может дать хорошее относительное ранжирование — она получает правильный порядок, но плохие оценки вероятности.

Подумайте об этом следующим образом: модель может указать на необходимость ранжировать позиции в порядке 1, 2 и 3 и оценивать вероятности как 0,7, 0,5 и 0,3. Возможно, *истинные* вероятности равны 0,03, 0,02 и 0,01, поэтому наши оценки полностью неверны, однако ранжирование было верным.

ИСПОЛЬЗОВАНИЕ A/B-ТЕСТИРОВАНИЯ ДЛЯ ОЦЕНКИ

Когда мы строим модели и оптимизируем их по отношению к некой оценочной метрике, такой как безошибочность или среднеквадратичная погрешность, сам метод оценки строится для оптимизации параметров *относительно* этих метрик. В отдельных контекстах показатели, которые мы, возможно, захотим оптимизировать, — нечто совсем иное, например доход. Поэтому мы можем попытаться построить алгоритм, оптимизируемый под безошибочность, когда наша реальная цель — зарабатывать деньги. Сама модель не способна непосредственно зафиксировать это. Таким образом, способ отразить нашу цель — запустить A/B-тесты (или статистические эксперименты), в которых мы даем некоему набору пользователей указание применять одну версию алгоритма, а другому набору пользователей — другую версию и проверяем разницу в эффективности интересующих нас метрик, например, «доход», или «доход на каждого пользователя», или нечто подобное. Мы обсудим A/B-тестирование в главе 11.

Упражнение от компании Media 6 Degrees

Компания Media 6 Degrees любезно предоставила набор данных, который идеально подходит для изучения моделей логистической регрессии и оценки того, насколько хороши модели. Проследуйте за нами, реализовав следующий код на языке R. Набор данных можно найти на сайте https://github.com/oreillymedia/doing_data_science.

Пример кода на R

```
# Автор: Брайан Далессандро (Brian Dalessandro)
# Считывание данных, просмотр переменных и создание наборов для обучения
# и тестирования
file <- "binary_class_dataset.txt"
set <- read.table(file, header = TRUE, sep = "\t", row.names = "client_id")
names(set)

split <- .65
```

```
set["rand"] <- runif(nrow(set))
train <- set[(set$rand <= split), ]
test <- set[(set$rand > split), ]
set$Y <- set$Y_BUY


############################################################
###########              Функции R              ##########
############################################################

library(mgcv)

# График сглаживания обобщенной аддитивной модели (ОАМ)
plotrel <- function(x, y, b, title) {
    # Произвести представление данных, сглаженных ОАМ
    g <- gam(as.formula("y ~ x"), family = "binomial", data = set)
    xs <- seq(min(x), max(x), length = 200)
    p <- predict(g, newdata = data.frame(x = xs), type = "response")

    # Получить эмпирические оценки (и дискретизировать,
    # если не дискретизировано)
    if (length(unique(x)) > b) {
        div <- floor(max(x) / b)
        x_b <- floor(x / div) * div
        c <- table(x_b, y)
    }
    else { c <- table(x, y) }
    pact <- c[ , 2]/(c[ , 1]+c[ , 2])
    cnt <- c[ , 1]+c[ , 2]
    xd <- as.integer(rownames(c))
    plot(xs, p, type="l", main=title,
         ylab = "P(Conversion | Ad, X)", xlab="X")
         points(xd, pact, type="p", col="red")
    rug(x+runif(length(x)))
}

library(plyr)
# График взвешенной среднеквадратичной погрешности (вСКП) и вычисление
getmae <- function(p, y, b, title, doplot) {
    # Нормализовать интервал [0,1]
    max_p <- max(p)
    p_norm <- p / max_p
    # разбиение на группы и масштабирование
    bin <- max_p * floor(p_norm * b) / b
    d <- data.frame(bin, p, y)
    t <- table(bin)
    summ <- ddply(d, .(bin), summarise, mean_p = mean(p), mean_y = mean(y))
    fin <- data.frame(bin = summ$bin, mean_p = summ$mean_p, mean_y = summ$mean_y, t)
    # получение вСКП
    num = 0
    den = 0
    for (i in c(1:nrow(fin))) {
    num <- num + fin$Freq[i] * abs(fin$mean_p[i] - fin$mean_y[i])
    den <- den + fin$Freq[i]
    }
```

```
    wmae <- num / den
    if (doplot == 1) {
        plot(summ$bin, summ$mean_p, type = "p",
             main = paste(title," MAE =", wmae),
             col = "blue", ylab = "P(C | AD, X)",
             xlab = "P(C | AD, X)")
        points(summ$bin, summ$mean_y, type = "p", col = "red")
        rug(p)
    }
    return(wmae)
}

library(ROCR)
get_auc <- function(ind, y) {
    pred <- prediction(ind, y)
    perf <- performance(pred, 'auc', fpr.stop = 1)
    auc <- as.numeric(substr(slot(perf, "y.values"), 1, 8), double)
    return(auc)
}

# Получение метрик эффективности с перекрестной
# валидацией для заданного набора признаков

getxval <- function(vars, data, folds, mae_bins) {
    # Назначить каждое наблюдение сборке
    data["fold"] <- floor(runif(nrow(data)) * folds) + 1
    auc <- c()
    wmae <- c()

    fold <- c()
    # создать объект формулы
    f = as.formula(paste("Y", "~", paste(vars, collapse = "+")))
    for (i in c(1:folds)) {
        train <- data[(data$fold != i), ]
        test <- data[(data$fold == i), ]
        mod_x <- glm(f, data=train, family = binomial(logit))
        p <- predict(mod_x, newdata = test, type = "response")
        # Get wMAE
        wmae <- c(wmae, getmae(p, test$Y, mae_bins, "dummy", 0))
        fold <- c(fold, i)
        auc <- c(auc, get_auc(p, test$Y))
    }
    return(data.frame(fold, wmae, auc))
}

#################################################################
##########         ГЛАВНОЕ: МОДЕЛИ И ГРАФИКИ         ##########
#################################################################

# Теперь создадим модель для всех переменных и посмотрим
# на коэффициенты и обучение модели
vlist      <-      c("AT_BUY_BOOLEAN",     "AT_FREQ_BUY",
"AT_FREQ_LAST24_BUY",
    "AT_FREQ_LAST24_SV", "AT_FREQ_SV", "EXPECTED_TIME_BUY",
    "EXPECTED_TIME_SV", "LAST_BUY", "LAST_SV", "num_checkins")
```

```
f = as.formula(paste("Y_BUY", "~" , paste(vlist, collapse = "+")))
fit <- glm(f, data = train, family = binomial(logit))
summary(fit)

# Получить метрики эффективности для каждой переменной

vlist      <-      c("AT_BUY_BOOLEAN",     "AT_FREQ_BUY",
"AT_FREQ_LAST24_BUY",
    "AT_FREQ_LAST24_SV", "AT_FREQ_SV", "EXPECTED_TIME_BUY",
    "EXPECTED_TIME_SV", "LAST_BUY", "LAST_SV", "num_checkins")

# Создать пустые векторы для хранения показателей эффективности/оценки
auc_mu <- c()
auc_sig <- c()
mae_mu <- c()
mae_sig <- c()

for (i in c(1:length(vlist))) {
    a <- getxval(c(vlist[i]), set, 10, 100)
    auc_mu <- c(auc_mu, mean(a$auc))
    auc_sig <- c(auc_sig, sd(a$auc))
    mae_mu <- c(mae_mu, mean(a$wmae))
    mae_sig <- c(mae_sig, sd(a$wmae))
}

univar <- data.frame(vlist, auc_mu, auc_sig, mae_mu, mae_sig)

# Получить график OAM для одной переменной — использовать группу проверок
# для оценки
set <- read.table(file, header = TRUE, sep = "\t", row.names="client_id")
names(set)

split<-.65
set["rand"] <- runif(nrow(set))
train <- set[(set$rand <= split), ]
test <- set[(set$rand > split), ]
set$Y <- set$Y_BUY

fit <- glm(Y_BUY ~ num_checkins, data = train, family = binomial(logit))
y <- test$Y_BUY
p <- predict(fit, newdata = test, type = "response")

getmae(p,y,50,"num_checkins",1)

# Жадный выбор вперед
rvars <- c("LAST_SV", "AT_FREQ_SV", "AT_FREQ_BUY",
    "AT_BUY_BOOLEAN", "LAST_BUY", "AT_FREQ_LAST24_SV",
    "EXPECTED_TIME_SV", "num_checkins",
    "EXPECTED_TIME_BUY", "AT_FREQ_LAST24_BUY")

# Создать пустые векторы
auc_mu <- c()
auc_sig <- c()
mae_mu <- c()
mae_sig <- c()
```

```
for (i in c(1:length(rvars))) {
    vars <- rvars[1:i]
    vars
    a <- getxval(vars, set, 10, 100)
    auc_mu <- c(auc_mu, mean(a$auc))
    auc_sig <- c(auc_sig, sd(a$auc))
    mae_mu <- c(mae_mu, mean(a$wmae))
    mae_sig <- c(mae_sig, sd(a$wmae))
}
kvar<-data.frame(auc_mu, auc_sig, mae_mu, mae_sig)

# Создать график трех кривых AUC
y <- test$Y_BUY

fit <- glm(Y_BUY~LAST_SV, data=train, family = binomial(logit))
p1 <- predict(fit, newdata=test, type="response")
fit <- glm(Y_BUY~LAST_BUY, data=train, family = binomial(logit))
p2 <- predict(fit, newdata=test, type="response")
fit <- glm(Y_BUY~num_checkins, data=train, family = binomial(logit))
p3 <- predict(fit, newdata=test,type="response")

pred <- prediction(p1,y)
perf1 <- performance(pred,'tpr','fpr')
pred <- prediction(p2,y)
perf2 <- performance(pred,'tpr','fpr')
pred <- prediction(p3,y)
perf3 <- performance(pred,'tpr','fpr')


plot(perf1, color="blue", main="LAST_SV (blue), LAST_BUY (red), num_checkins
(green)")
plot(perf2, col="red", add=TRUE)
plot(perf3, col="green", add=TRUE)
```

6 Метки времени и финансовое моделирование

В данной главе у нас есть два докладчика — Кайл Тиг (Kyle Teague) из GetGlue и тот, с кем вы уже знакомы, — Кэти О'Нил. Прежде чем она погрузится в рассказ о главных темах главы — временных рядах, финансовом моделировании и регрессии, — мы узнаем от Кайла Тига, что он и его компания думают о создании рекомендательной системы. (Подробнее об этом мы поговорим в главе 7.) Затем заложим основу для размышлений о данных с присвоенной меткой времени, разговор о которых продолжит Кэти.

Кайл Тиг и GetGlue

Нам нужно было услышать информацию от Кайла Тига (https://www.linkedin.com/in/kyleteague), вице-президента по анализу и обработке данных и проектированию в GetGlue (https://telfie.com/?ad=getglue.com&utm_source=getglue.com&utm_campaign=getglue%2520404%2520redirect&utm_medium=redirect&utm_content=getglue%2520404%2520redirect). Кайл имеет опыт в электротехнике. Он считает время, потраченное на обработку сигналов в исследовательской лаборатории, сверхполезным и программирует с детства. Он разрабатывает на Python.

GetGlue — нью-йоркский стартап, основной целью которого является решение проблем, связанных с изучением контента в области кино и телевидения. Обычная модель поиска того, что показывают по ТВ, — это программа *TV Guide* («Телегид») 1950-х годов, именно таким способом многие из нас до сих пор ищут что посмотреть. С учетом существования тысячи каналов становится все сложнее найти нечто стоящее на ТВ.

GetGlue хочет изменить эту модель, предложив людям персональные телепутеводители и рекомендации. В частности, пользователи «регистрируются» на телешоу; это значит следующее: они могут сообщить другим людям о том, что смотрят шоу, с помощью создания точки данных с меткой даты/времени.

Они также могут выполнять другие действия, такие как «поставить лайк» или «комментировать шоу».

Информация хранится в триплетах вида `{user, action, item}` (пользователь, действие, элемент), где элемент представляет собой телешоу (или фильм). Одним из способов визуализации хранимых данных является изображение *двудольного графа*, представленного на рис. 6.1.

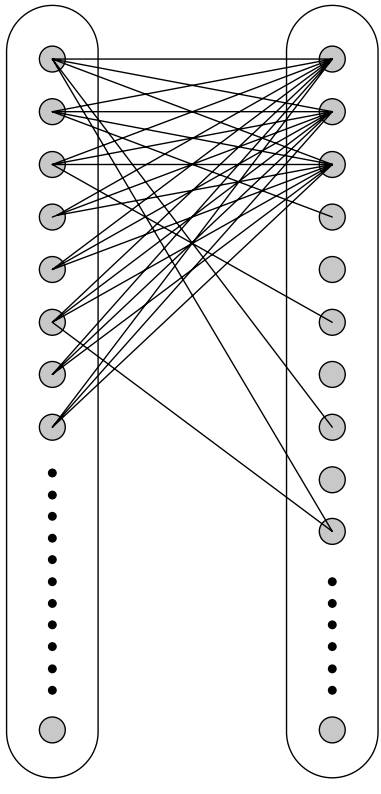


Рис. 6.1. Двудольный граф с пользователями и элементами (шоу) в качестве узлов

Мы будем рассматривать графы в следующих главах, но на данный момент вам нужно знать, что точки называются *узлами*, а линии — *ребрами*. Для этого специфического вида графов, носящего название двудольного, характерно наличие узлов двух типов, в данном случае соответствующих пользователям и элементам. Все ребра проходят между пользователем и элементом, а именно, если данный пользователь произвел какое-либо действие относительно данного шоу. Никогда не бывает ребер между разными пользователями или шоу. Граф на рис. 6.1 может отображать ситуацию, когда определенные пользователи положительно оценили определенные телешоу.

GetGlue усовершенствовали этот граф следующим образом: он ищет пути создания связей между пользователями и шоу, хотя и с другими типами ребер. Например, пользователи могут подписаться друг на друга или быть друзьями на GetGlue, который создает *направленные ребра* — идущие от одного узла к другому в направлении, обычно обозначаемом стрелкой. Аналогично с помощью набора предпочтений GetGlue можно обнаружить, что два человека имеют схожие вкусы и также могут быть связаны между собой этим способом, но, по всей видимости, без направленности.

Кроме того, GetGlue нанимают экспертов для создания соединений или направленных ребер между шоу. Так, *True Blood* («Настоящая кровь») и *Buffy the Vampire Slayer* («Баффи — истребительница вампиров») могут быть похожи по некоторым причинам, поэтому люди создают между ними ребро «схожести» на графе. У направленных ребер есть своя специфика; к примеру, можно нарисовать стрелку от «Баффи» к «Настоящей крови», но не наоборот, то есть понятие «схожести» или «близости» охватывает как содержание, так и популярность. Pandora (http://www.pandora.com/restricted) тоже намеревается сделать нечто подобное.

Еще один важный аспект, особенно в стремительно меняющемся телевизионном пространстве, — время. Пользователь зарегистрировался или лайкнул шоу в определенный момент времени, в связи с чем нужно, чтобы у данных для регистрации действия была метка времени: `{user, action, item, timestamp}` (пользователь, действие, элемент, метка времени). Метки полезны для того, чтобы видеть, как действие распространяется в построенном графе или граф развивается во времени.

Метки времени

Данные о событиях с метками даты/времени — общий тип данных в век больших данных. На самом деле это одно из оснований больших данных. Тот факт, что компьютеры могут записывать все действия, предпринимаемые пользователем, означает следующее: один пользователь может генерировать тысячи точек данных в течение дня. Когда люди посещают сайт, применяют приложение или взаимодействуют с компьютерами и телефонами, их действия могут регистрироваться, а точное время и характер их взаимодействия записываются. Когда создается новый продукт или признак, инженеры, работающие над этим, пишут код для фиксации событий, которые происходят во время перемещения и использования продукта, — данный захват является частью продукта.

Например, представьте, что пользователь посещает главную страницу The New York Times. Сайт фиксирует, какие новостные сюжеты отображаются для этого пользователя и на каких из них он щелкал. Это создает журналы событий. Каждая запись — событие, которое произошло между пользователем и приложением или сайтом.

Ниже приведен пример исходной точки данных из GetGlue:

```
{"userid": "rachelschutt", "numCheckins": "1",
"modelName": "movies", "title": "Collaborator",
"source": "http://getglue.com/stickers/tribeca_film/
collaborator_coming_soon", "numReplies": "0",
"app": "GetGlue", "lastCheckin": "true",
"timestamp": "2012-05-18T14:15:40Z",
"director": "martin donovan", "verb": "watching",
"key": "rachelschutt/2012-05-18T14:15:40Z",
"others": "97", "displayName": "Rachel Schutt",
"lastModified": "2012-05-18T14:15:43Z",
"objectKey": "movies/collaborator/martin_donovan",
"action": "watching"}
```

Если мы извлекаем четыре поля: `{"userid":" rachelschutt", "action": "watching", "title":"Collaborator", timestamp:"2012-05-18T14:15:40Z"}`, то можем представить их в порядке, который только что обсуждали, а именно: `{user, verb, object, timestamp}` (пользователь, операция, объект, метка времени).

Разведочный анализ данных (РАД)

Как мы говорили в главе 2, лучше всего начинать свой анализ с РАД, так вы сможете усилить свое интуитивное восприятие данных перед построением модели на их основе. Углубимся в пример РАД, который вы можете сделать с пользовательскими данными в стиле потока сознания. Это иллюстрация более крупного метода, и все, что мы здесь делаем, можно модифицировать для других типов данных, но вам также может потребоваться сделать дополнительно что-то еще — все зависит от обстоятельств.

Самое первое, с чем нужно разобраться при работе с пользовательскими данными, — это *индивидуальные графики пользователя во времени*. Убедитесь, что данные имеют смысл для вас при исследовании концепции данных, которая отражена с точки зрения *одного человека*.

Чтобы сделать это, возьмите случайную выборку пользователей: начните с чего-то малого, например со 100 пользователей. Да, возможно, ваш набор данных содержит миллионы пользователей, но для начала нужно получить ощущение данных. Рассматривать миллионы точек данных слишком тяжело для вас как человека. Но, просто взглянув на сотню точек, вы начнете понимать данные и увидите, чистые ли они. Конечно, в какой-то мере подобный размер выборки недостаточно велик, если вы начнете делать выводы обо всем наборе.

Получить такую выборку можно путем поиска имен пользователей и обработки с помощью команды `grep` или поиска 100 случайных выборок, по одной за раз. Для каждого пользователя создайте график, как показано на рис. 6.2.

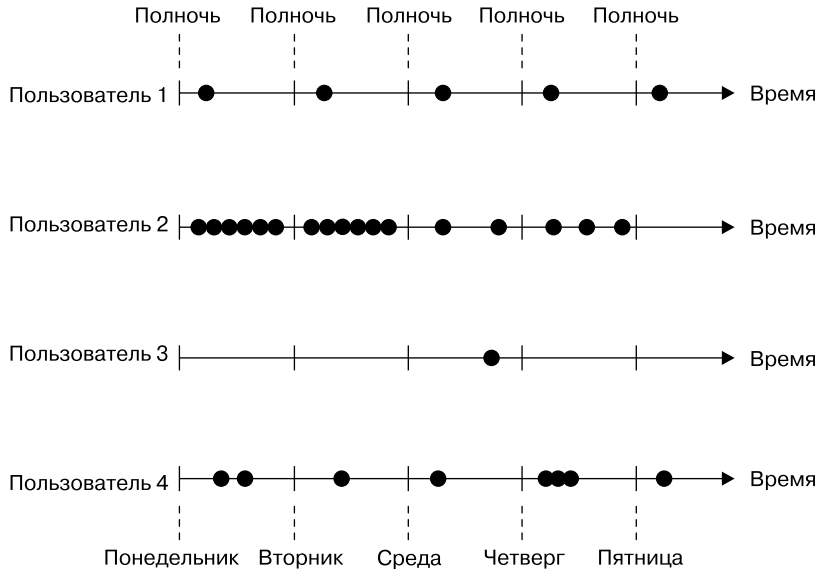


Рис. 6.2. Пример способа визуального отображения данных пользовательского уровня во времени

Теперь попробуем построить интерпретацию из данного графика. Так, мы можем сказать, что пользователь 1 приходит в одно и то же время каждый день, тогда как пользователь 2 начал активную работу в этот период времени, но затем приходил все реже и реже. Для пользователя 3 требуется более продолжительный период времени, чтобы понять его или ее поведение, тогда как пользователь 4 выглядит «нормальным», что бы это ни значило.

Сформулируем вопросы из этой интерпретации.

- Что *делает* типичный или средний пользователь?
- На что похожи эти вариации?
- Как мы будем классифицировать пользователей по различным сегментам на основе их поведения во времени?
- Как мы станем количественно определять различия между этими пользователями?

Взглянем абстрактно на один из типичных вопросов из практики очистки данных. Скажем, у нас есть исходные данные, где каждая точка данных является событием, но мы хотим иметь данные, хранящиеся в строках, где каждая строка состоит из пользователя, затем идет множество меток времени, соответствующих действиям, которые выполнял пользователь. Как мы получим данные с этой точки зрения?

Обратите внимание, что разные пользователи будут иметь разное количество меток времени.

Давайте опишем цепь рассуждений явным образом: как написать код для создания вышеприведенного графика? Как подступиться к очистке данных?

Предположим, пользователь может предпринять несколько действий: thumbs_up («палец вверх») или thumbs_down («палец вниз»), like («нравится») и comment («комментировать»). Как мы можем отобразить эти события? Изменять наши метрики? Закодировать пользовательские данные с помощью этих разных действий? Пример для первого вопроса приведен на рис. 6.3, на котором мы раскрашиваем действия «палец вверх» и «палец вниз», обозначенные как thumbs_up и thumbs_down.

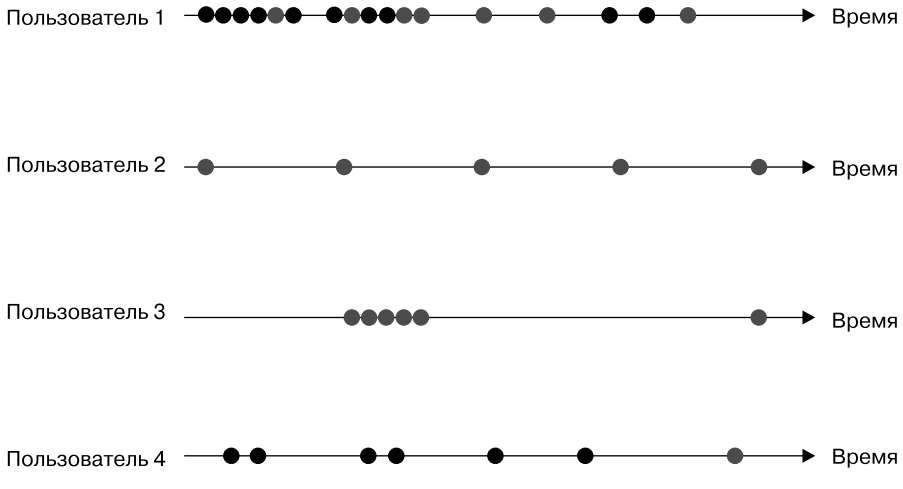


Рис. 6.3. Используйте цвет, чтобы добавить на экран больше информации о действиях пользователя

На этом примере мы видим, что все пользователи одновременно сделали одну и ту же вещь в одно и то же время в правом конце графиков. Минутку, это настоящее событие или сбой в системе? Как мы это проверяем? Существует ли большое одновременное присутствие некоторых действий у всех пользователей? Является ли «черное» более распространенным, чем «красное»? Может быть, одни пользователи всегда выбирают «палец вверх», другим нравится «палец вниз», а пользователи из третьей группы являются смесью первой и второй. Что такое смесь?

Теперь, когда мы начали получать некоторое представление о различиях между пользователями, можем подумать, как объединить пользователей. Можно сделать так, чтобы на оси *X* отображалось время, а на оси *Y* — количество, как показано на рис. 6.4.

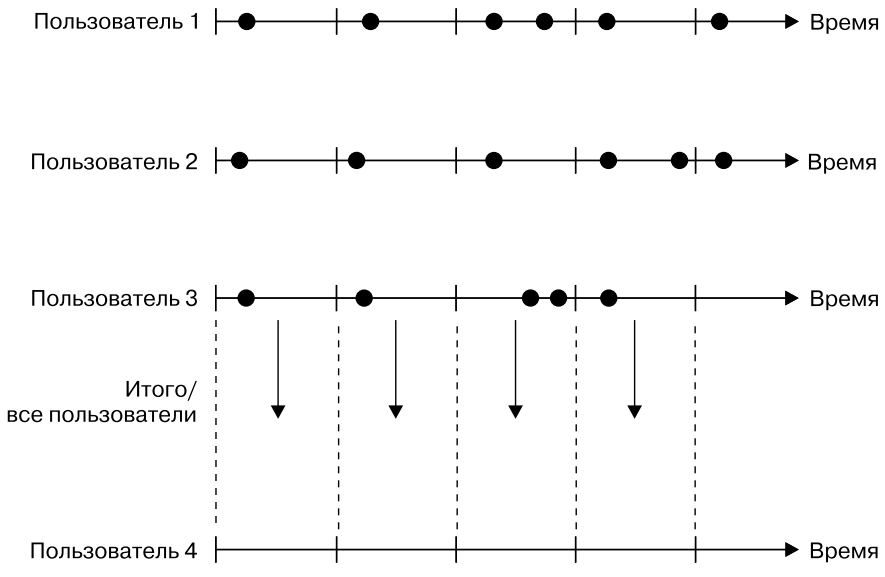


Рис. 6.4. Объединение действий пользователя в подсчеты

Мы больше не работаем со 100 отдельными пользователями, но по-прежнему делаем выборы, и последние могут повлиять на наше восприятие и понимание набора данных.

Например, подсчитываем количество уникальных пользователей или общее количество учетных записей пользователей? Ввиду того что некоторые пользователи регистрируются несколько раз, это может иметь огромное влияние. Подсчитываем ли мы количество действий или пользователей, которые выполняли данное действие по крайней мере один раз в течение определенного отрезка времени?

Каков наш временной интервал? Мы считаем в секунду, минуту, час, восьмичасовые сегменты, день или неделю? Почему мы это выбрали? Сигнал перегружен сезонностью или мы *ищем* сезонность?

Находятся ли наши пользователи в разных часовых поясах? Если пользователи 1 и 2 находятся в Нью-Йорке и Лондоне соответственно, то это семь часов утра в Нью-Йорке и полдень в Лондоне. Если мы считаем это как семь утра и говорим, что «30 000 пользователей совершили данное действие утром», то это вводит в заблуждение, поскольку в Лондоне не утро. Как мы собираемся обрабатывать подобные случаи? Мы можем перенести данные в сегмент, так что для *пользователя* будет семь часов утра, а не семь часов утра в Нью-Йорке, но тогда мы сталкиваемся с другими проблемами. Это решение, которое мы должны сделать и обосновать, и оно может идти по другому пути в зависимости от обстоятельств.

Метки времени коварны

Мы не собираемся обманывать, метки времени — одна из самых сложных вещей для понимания в моделировании, особенно в отношении изменений во времени. Вот почему иногда проще всего преобразовать все метки времени в секунды с начала отсчета времени (https://www.epochconverter.com/).

Возможно, мы хотим сделать разные графики для различных типов действий или, вероятно, будем группировать действия в более широкие категории, чтобы всмотреться в них и достичь понимания.

Метрики и новые переменные или признаки

Полученные из РАД интуитивные представления помогут нам теперь сформировать метрики. Скажем, мы можем оценивать пользователей с помощью меток (или простых булевых переменных, например, «выполнил действие как минимум однократно») для частот или количеств их действий, времени до первого события. Если хотим сравнить пользователей, то можем создать метрики сходства или различия. Нам может пригодиться агрегирование по пользователю (найдем кумулятивную сумму выполняемых им действий либо потраченных денег) или агрегирование по действиям для получения среднего количества пользователей, совершающих конкретное действие один раз либо несколько.

Это далеко не полный список. Метрики могут быть любой функцией данных при условии наличия цели и причины их создания и интерпретации.

Что дальше?

Мы хотим приступить к моделированию, алгоритмам и анализу, использовав интуитивные представления, полученные из РАД, в наших моделях и алгоритмах. Вот несколько примеров того, что мы могли бы сделать.

Возможно, нам будет интересно моделирование временных рядов, которое включает авторегрессию. Мы поговорим об этом подробнее в разделе, посвященном финансовому моделированию, но в общих чертах — мы работаем с временными рядами, когда пытаемся предсказать события, которые являются сверхчувствительными во времени, скажем рынки, или в некоторой степени предсказуемыми, основанными на том, что уже произошло, например, сколько денег вкладывается в пенсионные фонды в месяц.

Мы можем начать кластеризацию, которую обсуждали в главе 3. Для этого нужно определить *близость* пользователей друг к другу.

Возможно, нам захочется создать монитор, который мог бы автоматически обнаруживать закономерности общего поведения. Конечно, сначала нужно определить,

что такое закономерность общего поведения и что может выходить за рамки обычного поведения.

Мы можем попробовать свои силы в обнаружении точки изменения, которое, к слову, является способностью идентифицировать момент, когда произошло какое-то большое событие. Какое поведение в нашей системе должно вызывать тревогу? Или попытаемся установить причинность. Это может быть очень сложно, если мы не создали эксперимент. Наконец, нам может понадобиться обучить рекомендательную систему.

ИСТОРИЧЕСКАЯ ПЕРСПЕКТИВА: ЧТО НОВОГО?

Сами по себе данные с меткой времени не новое явление, и анализ временных рядов — устоявшаяся сфера (см., например, книгу Time Series Analysis («Анализ на основе временных рядов») Джеймса Д. Хамильтона (James D. Hamilton) (https://www.amazon.com/Series-Analysis-James-Douglas-Hamilton/dp/0691042896/ref=sr_1_1?ie=UTF8&qid=1368648857&sr=8-1&keywords=time+series+analysis)). Исторически имеющиеся доступные наборы данных были довольно небольшими, и события регистрировались один раз в день или даже сообщались на агрегированных уровнях. В качестве примеров наборов данных с меткой даты/времени, существовавших некоторое время даже на детализированном уровне, можно назвать: котировки акций в финансах, транзакции с кредитными картами, записи телефонных звонков или книги, выписанные из библиотеки.

Тем не менее есть несколько аспектов, делающих эту сферу новой или по крайней мере заново расширяющих ее. Во-первых, теперь легко измерить поведение человека в течение дня, поскольку многие из нас носят с собой устройства, которые могут быть использованы и используются для измерения и записи действий. Во-вторых, метки времени точны, поэтому мы не зависим от отчетов пользователя, как известно, являющихся ненадежными. Наконец, вычислительная мощность позволяет хранить большие объемы данных и обрабатывать их достаточно быстро.

Кэти О'Нил

Теперь слово Кэти О'Нил. Вы уже знакомы с ней, но взглянем на ее профиль в науке о данных, представленный на рис. 6.5.

Самое слабое ее место — теория вычислительной техники. Хотя она отлично программирует на Python, владеет скрапингом и разбором данных, умеет создавать прототипы моделей и использовать matplotlib для рисования красивых картинок, отображение/свертка на языке Java остаются для нее «темным лесом», и она преклоняется перед теми, кто может это делать. У Кэти также совершенно нет опыта визуализации данных, хотя она знает достаточно для того, чтобы делать свое дело и проводить понятные слушателям презентации.

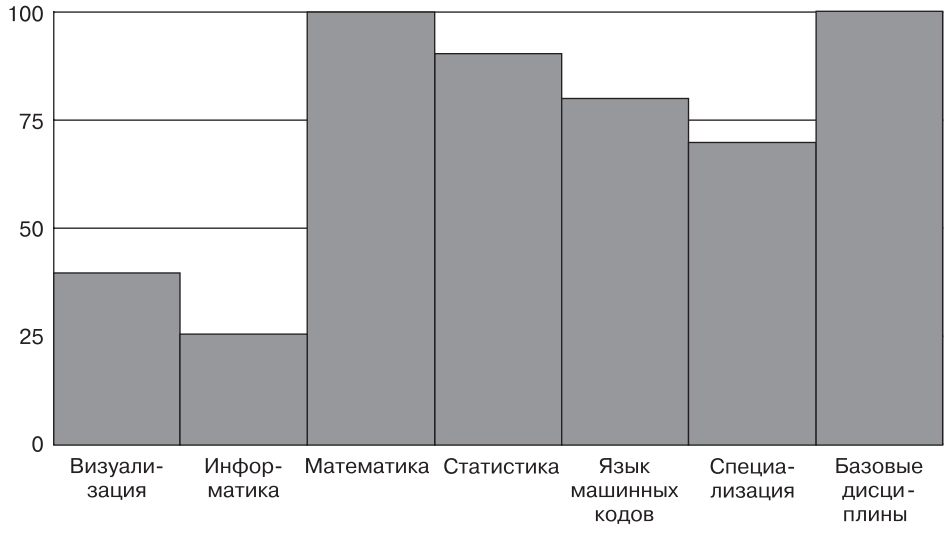


Рис. 6.5. Профиль Кэти в науке о данных

Мысленный эксперимент

Что вы теряете, когда думаете об обучающем наборе как о большой куче данных и игнорируете метки времени?

Главная мысль здесь заключается в том, что вы не можете выделять причину и следствие, если не имеете чувства времени.

Что, если мы исправим вопрос так, чтобы можно было собирать *относительные разности времени*, типа «время с момента последнего входа пользователя», или «время последнего щелчка кнопкой мыши», или «время последней инъекции инсулина», но не *абсолютные метки времени*?

В данном случае у вас все еще серьезные проблемы. Скажем, вы игнорируете тенденции вместе с сезонностью, если не упорядочиваете данные по времени. Возьмем пример с инсулином. Вы можете заметить, что в течение 15 минут после инъекции инсулина уровень сахара в крови неизменно снижается, но *не* заметить общей тенденции повышения сахара в крови в течение последних нескольких месяцев, если ваш набор данных за этот период не имеет абсолютных меток времени. Не приводя данные в порядок, вы можете пропустить закономерность, показанную на рис. 6.6.

Эта идея отслеживания тенденций и сезонности очень важна в финансовых данных и необходима для отслеживания, если вы хотите заработать деньги, с учетом того, насколько малы фиксируемые сигналы.

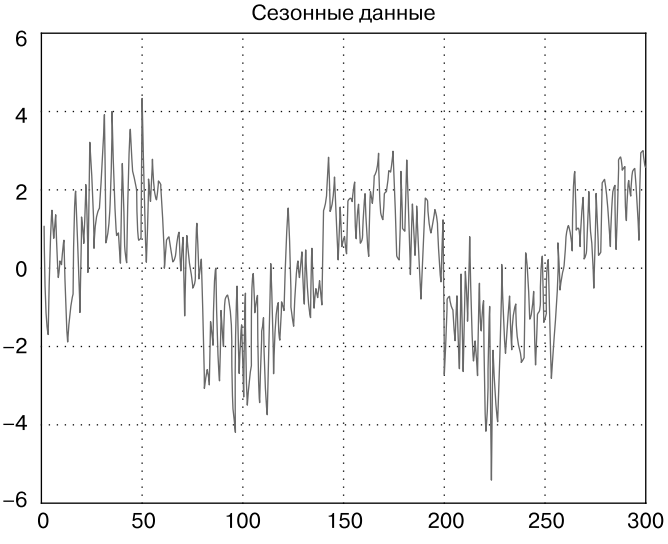


Рис. 6.6. Без отслеживания меток времени мы не можем увидеть закономерности во времени; здесь мы видим сезонную закономерность во временном ряду

Финансовое моделирование

До появления термина «исследователь данных» (data scientist) в финансах работал специалист по количественному анализу (quant). В работе этих специалистов есть много совпадающих аспектов, а также некоторые различия. Например, как мы увидим в данной главе, специалисты по количественному анализу одержимы метками времени и особо не заботятся о том, *почему* все работает, если оно просто работает.

Конечно, есть предел тому, что можно рассмотреть в одной главе, но материал предназначен для того, чтобы дать общее представление о таком подходе в финансовом моделировании.

В пределах выборки, за пределами выборки, причинная зависимость

Необходимо дать точное определение понятиям «данные в пределах выборки» и «за пределами выборки». Обратите внимание: данные вне выборки *не* являются данными для тестирования — оно происходит внутри «данных в пределах выборки». Скорее, данные вне выборки предназначены для использования после *финализации вашей модели*, чтобы вы имели представление о том, как она будет работать в процессе эксплуатации.

Мы даже должны ограничить количество проведения анализа данных вне выборки для этого набора данных, поскольку волей-неволей, каждый раз изучая материал об этих данных, будем подсознательно подчиняться им даже в разных ситуациях, с различными моделями.

Далее, будьте осторожны с постоянным выполнением *причинно-следственного моделирования* (обратите внимание: оно отличается от того, что специалисты по статистике подразумевают под причинностью). То есть *никогда не используйте информацию в будущем, чтобы предсказать что-то сейчас*. Или, по-другому, применяйте только информацию из прошлого и до настоящего момента для предсказания будущего. Это невероятно важно для финансового моделирования. Обратите внимание: *недостаточно* использовать данные о настоящем, если они на самом деле несвободны и недоступны в настоящий момент. Таким образом, необходимо быть очень осторожным с метками времени доступности, а также с эталонными метками времени.

Аналогичным образом, имея обучающий набор, мы не знаем «оптимальные коэффициенты» для них до момента последней метки времени для всех этих данных. При нашем движении вперед по времени от первой метки до последней мы ожидаем получить разные наборы коэффициентов по мере того, как происходит больше событий.

Одним из следствий этого является то, что вместо получения *одного* набора «оптимальных» коэффициентов мы фактически получаем *эволюцию* каждого коэффициента. Это полезно, поскольку дает представление о том, насколько стабильны указанные коэффициенты. В частности, если один из них изменял знак десять раз на основе обучающего набора, то мы вполне можем ожидать, что хорошая оценка для него равна нулю, а не так называемая оптимальная в конце данных. Конечно, в зависимости от переменной мы можем думать о законном основании, позволяющем ей на самом деле менять знак с течением времени.

В целом данные в пределах выборки должны приходить перед данными за ее пределами, чтобы избежать проблем причинности, как показано на рис. 6.7.

Сделаем одно заключительное замечание о причинно-следственном моделировании и данных в пределах выборки против данных за ее пределами. Оно не противоречит разработке программного кода, потому что мы всегда действуем (при обучении и моделировании «за пределами выборки») так, будто используем нашу модель в промышленной эксплуатации и видим, как она работает. Конечно, мы приспосабливаем нашу модель под выборку, поэтому ожидаем, что она будет лучше работать, чем при промышленной эксплуатации.

Другими словами — как только у нас будет модель в эксплуатации, нам придется принимать решения о будущем, основываясь *только на том, что знаем сейчас* (это определение причинности), и мы захотим обновить нашу модель всякий раз, когда собираем новые данные. Поэтому коэффициенты нашей модели — живые организмы, которые постоянно развиваются. Так и должно быть — в конце концов, мы моделируем реальность и со временем все меняется.

Рис. 6.7. Данные в пределах выборки должны приходить перед данными за ее пределами в наборе временных рядов

Подготовка финансовых данных

Мы часто «готовим» данные, прежде чем вводить их в модель. Обычный способ подготовки связан со взятием среднего значения или дисперсии данных либо иногда с преобразованием неким способом, например с помощью логарифмирования (с последующим взятием среднего значения или дисперсии этих преобразованных данных). Полученные данные в конечном итоге становятся подмоделью нашей модели.

ПРЕОБРАЗОВАНИЕ ВАШИХ ДАННЫХ

Помимо контекста финансовых данных, подготовка и преобразование информации тоже важная часть процесса. Существует несколько возможных методов преобразования данных, чтобы они лучше «себя вели»:

- нормализовать данные, вычитая среднее значение и деление на стандартное отклонение;
- нормализовать или масштабировать, разделив на максимальное значение;

- взять журнал данных;
- разделить на пять равномерно расположенных интервалов (либо число, отличное от пяти) и создать из этого категориальную переменную;
- выбрать значащий порог и преобразовать данные в новую двоичную переменную со значением 1, если точка данных больше порогового значения или равна ему, и 0, если меньше его.

Как только мы оценим наше среднее значение y и дисперсию σ_y^2, можем нормализовать следующую точку данных с этими оценками так же, как делаем, чтобы получить из гауссова или нормального распределения стандартное нормальное распределение со средним значением = 0 и стандартным отклонением = 1:

$$y \mapsto \frac{y - \overline{y}}{\sigma_y}.$$

Конечно, мы можем отслеживать и другие моменты, а также готовить данные, запускать другие подмодели нашей модели. Например, можем выбрать для рассмотрения только «новую» часть чего-то, которая эквивалентна попытке предсказать нечто наподобие $y_t - y_{t-1}$ вместо y_t. Или можем обучить подмодель, чтобы выяснить, какая часть y_{t-1} предсказывает y_t, например, подмодель, которая является одномерной регрессией или еще чем-нибудь.

Здесь много вариантов, которые всегда будут зависеть от ситуации и вашей цели. Имейте в виду: это все имеет причинно-следственная связь, так что вам нужно быть осторожными, когда вы обучаете итоговую модель вводить вашу следующую точку данных. Убедитесь, что все шаги идут в хронологическом порядке и вы никогда не обманываете и не заглядываете вперед по времени в данные, которые еще не произошли.

В частности, и так происходит все время, нельзя нормализовать по среднему, вычисляемому с помощью обучающего набора. Вместо этого имейте *текущую оценку среднего значения*, которую знаете в данный момент, и делайте нормализацию с ее учетом.

Чтобы понять, почему это настолько опасно, представьте себе крах рынка в середине вашего обучающего набора. Среднее значение и дисперсия ваших результатов сильно зависят от подобного события, и выполнение чего-то столь же безобидного, как средняя оценка, позволяет предупреждать о крахе, прежде чем он произойдет. Такая не имеющая причины интерференция имеет склонность помогать модели и, вероятно, может помочь плохой модели выглядеть хорошо (или, что более вероятно, сделать модель, которая представляет собой равномерный шум, хорошей).

Логарифмическая доходность

В финансах мы рассматриваем доходность ежедневно. Другими словами, заботимся о том, насколько акция (или фьючерс, или индекс) меняется изо дня в день. Это может означать, что мы измеряем движение с открытия в понедельник до открытия во вторник, но стандартным подходом является забота о ценах закрытия в последующие торговые дни.

Обычно мы рассматриваем не процентные доходы, а скорее *логарифмическую доходность*: если F_t обозначает закрытие в день t, то *логарифмическая доходность* в этот день определяется как $\log(F_t / F_{t-1})$, в то время как процентный доход будет рассчитан как $100((F_t / F_{t-1}) - 1)$. Чтобы упростить обсуждение, сравним логарифмическую доходность с *масштабируемой процентной доходностью*, которая равна процентной доходности, только без множителя 100. Идея такова — это различие не сказывается на скалярности.

Есть несколько причин, по которым мы используем логарифмическую доходность вместо процентной. Например, логарифмическая является аддитивной, а масштабируемая процентная — нет. Другими словами, пятидневная логарифмическая доходность — сумма пяти однодневных логарифмических доходностей. Это часто удобно для вычислений.

Аналогично логарифмическая доходность является симметричной по отношению к прибыли и убыткам, тогда как на процентные доходы влияет прибыль. Например, если цена на нашу акцию снизится на 50 % или будет иметь коэффициент масштабирования в размере –0,5 %, а затем повысится на 200 %, то коэффициент масштабирования будет 2,0; теперь мы там, откуда начинали. Но, работая по тем же сценариям с логарифмической доходностью, мы сначала увидим логарифмическую доходность, равную $log(0{,}5) = -0{,}301$, а затем $log(2{,}0) = 0{,}301$.

Тем не менее эти два вида доходности близки друг к другу в случае малых объемов, так что при работе с короткими временными горизонтами, такими как ежедневные или еще короче, это не имеет большого значения. Данное утверждение можно легко проверить: установить $x = F_t / F_{t-1}$, масштабированный процентный доход равен $x - 1$, а логарифмическая доходность равна $log(x)$ и имеет следующее выражение Тейлора:

$$\log(x) = \sum_n \frac{(x-1)^n}{n} = (x-1) + (x-1)^2 / 2 + \ldots$$

Другими словами, первый член ряда Тейлора согласуется с процентной доходностью. Таким образом, пока второй член мал по сравнению с первым, что обычно справедливо для ежедневной доходности, мы получаем довольно хорошее приближение процентной доходности с помощью логарифмической доходности.

На рис. 6.8 приведена иллюстрация того, насколько близки графики этих двух функций с учетом того, что при $x = 1$ вообще нет никаких изменений цены.

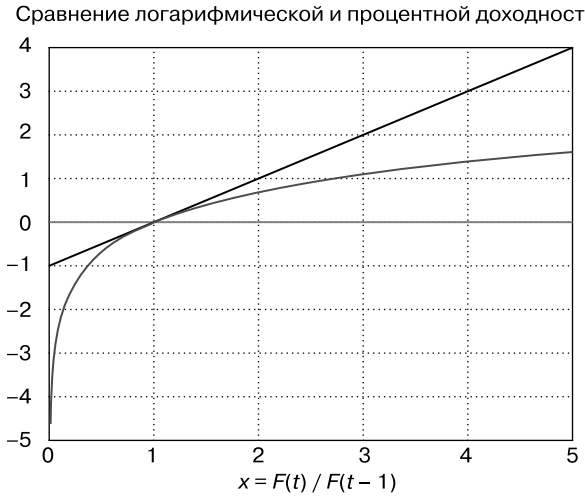


Рис. 6.8. Сравнение логарифмической и процентной доходностей

Пример: индекс S&P

Разберем простенький пример. Если вы начинаете с уровней закрытия S&P, как показано на рис. 6.9, то получите логарифмическую доходность, изображенную на рис. 6.10.

Рис. 6.9. Уровни закрытия S&P показаны во времени

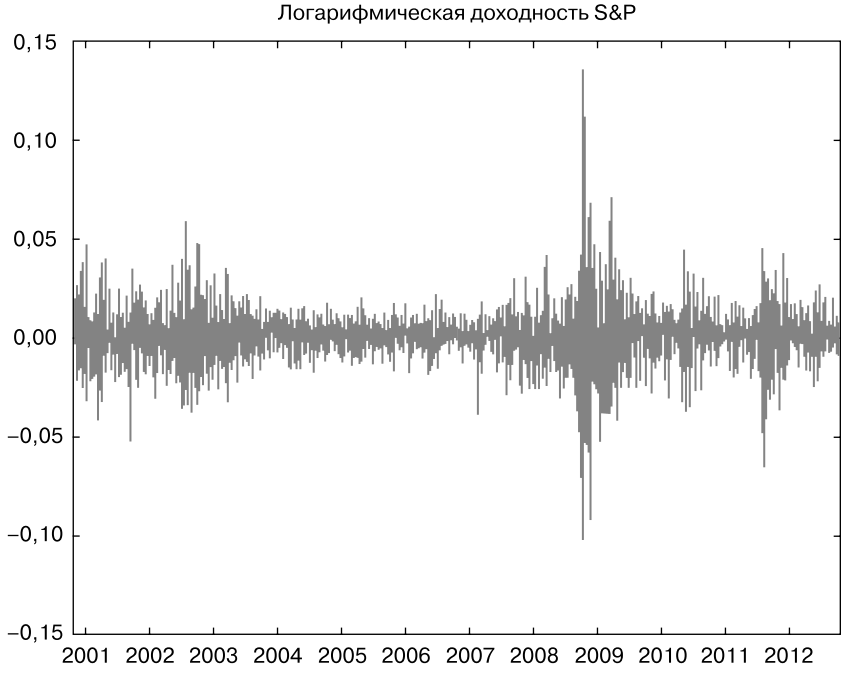


Рис. 6.10. Логарифмическая доходность S&P показана во времени

Что за беспорядок? Это сумасшедшая волатильность, вызванная финансовым кризисом. Мы иногда (не всегда) хотим учитывать эту волатильность, проводя нормализацию с ее учетом (как было описано ранее). Когда мы это сделаем, получим нечто похожее на рис. 6.11, которое ведет себя явно лучше.

Разработка измерения волатильности

Как только мы определили нашу доходность, можем хранить текущую оценку того, насколько сильно изменилась доходность за последнее время, которая обычно измеряется как стандартное отклонение и называется *оценкой волатильности*.

Важное решение для измерения волатильности — выбор окна ретроспективного обзора, являющегося отрезком времени прошлого, из которого мы будем брать нашу информацию. Чем длиннее окно, тем больше информации нужно для нашей оценки. Однако чем оно короче, тем быстрее оценка волатильности отвечает на новую информацию. Иногда вы можете подумать об этом так: если происходит довольно крупное рыночное событие, то сколько времени нужно, чтобы рынок «забыл о нем»? Этот пример довольно расплывчат, но может дать представление на соответствующей длине ретроспективного окна. Например, это определенно

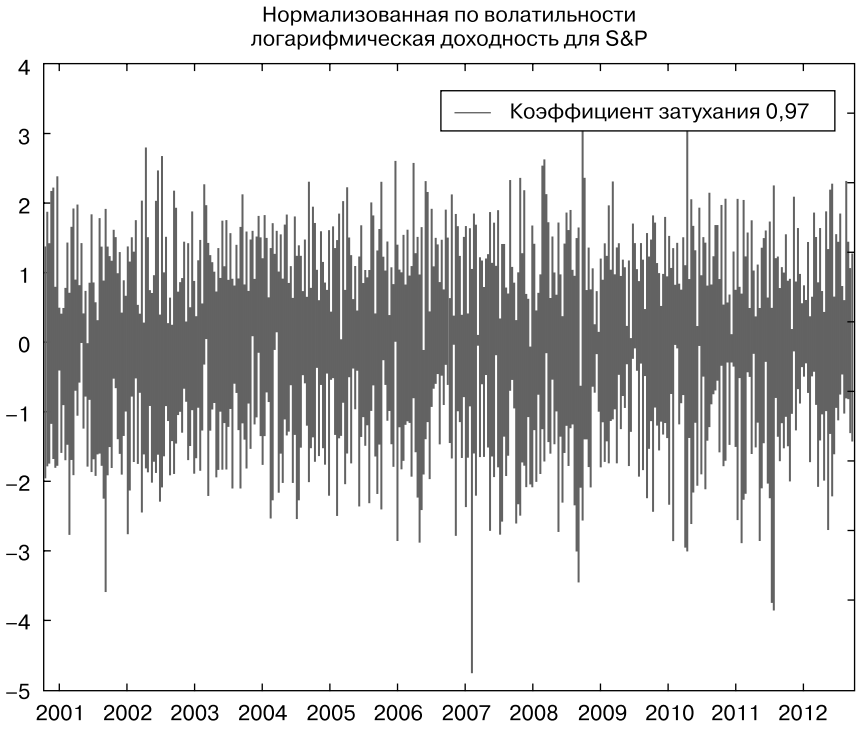


Рис. 6.11. Логарифмическая доходность S&P, нормализованная с учетом волатильности, показана во времени

больше недели, иногда менее четырех месяцев. Конечно, все также зависит от того, насколько велико событие.

Затем нужно решить, как мы используем важность данных за последние несколько дней. Самый простой подход — взять строго скользящее окно, что означает следующее: мы одинаково взвешиваем каждый из предыдущих n дней, а доходность данного дня учитывает указанные n дней, а затем выпадает с обратной стороны окна. Плохая новость об этом легком подходе заключается в том, что большая доходность будет считаться большой до последнего момента и затем полностью исчезнет. Это не значит, что люди с легкостью забывают о вещах, — обычно они дают информации постепенно исчезнуть из их памяти.

У нас есть окно с непрерывным ретроспективным просмотром, в котором мы экспоненциально уменьшаем вес старых данных, и используется концепция «периода полураспада» данных. Это говорит о том, что влияние прошлых значений доходов масштабируется в зависимости от того, как далеко они находятся в прошлом, каждый день умножаясь на некоторое число меньше 1 (называемое спадом). Например, если мы возьмем число, равное 0,97, то доходность пятидневной давности

умножается на скаляр, равный 0,975. Затем будем делить на сумму весов и в итоге принимаем средневзвешенное значение доходностей, где весы являются просто степенью чего-то похожего на 0,97. «Период полураспада» в этой модели можно вывести из числа 0,97, используя эти формулы так: $-\ln(2) / \ln(0{,}97) = 23$.

Теперь, когда мы выяснили, какой вес хотим давать каждой доходности предыдущего дня, вычисляем дисперсию как просто взвешенную сумму квадратов предыдущих доходностей. Затем берем квадратный корень в конце, чтобы оценить волатильность.

Обратите внимание: мы только что дали формулу, которая включает все предыдущие доходности. Это потенциально бесконечный расчет, хотя и с экспоненциально уменьшающимися весами. Но есть классный трюк: чтобы на самом деле вычислить это, нужно всего лишь хранить одну промежуточную сумму от всей суммы до текущего момента и суммировать ее с новым значением доходности, возведенным в квадрат. Таким образом можем обновить нашу vol (так называют волатильность (volatility) сведущие люди) с помощью одного значения в памяти и одной простой средневзвешенной величины.

Для начала делим на сумму весов, но весами являются степени некоторого числа s, так что это геометрическая сумма, и в пределе она имеет вид $1 / (1 - s)$.

Экспоненциальное понижающее взвешивание

Описанная техника называется экспоненциальным понижающим взвешиванием, это удобный способ сжатия данных в одно значение, которое можно обновить, не прибегая к необходимости сохранять весь набор данных.

Далее, предположим, что у нас есть текущая оценка дисперсии такого вида:

$$V_{old} = (1-s)\cdot\sum_i r_i^2 s^i$$

и у нас есть новое значение доходности r_0 для добавления в ряд. Тогда нетрудно показать, что мы хотим:

$$V_{new} = s\cdot V_{old} + (1-s)\cdot r_0^2.$$

Обратите внимание: мы сказали, что будем использовать стандартное отклонение выборки, но формула для этого обычно включает удаление среднего значения перед получением суммы квадратов. Здесь мы игнорируем среднее значение главным образом потому, что обычно берем ежедневную волатильность, где среднее значение (которое трудно предвидеть в любом случае!) намного меньше, чем шум, ввиду чего мы можем рассматривать его по большому счету как нуль. Если бы мы измеряли волатильность в более длительных временных масштабах, таких как кварталы или годы, то, вероятно, не игнорировали бы среднее значение.

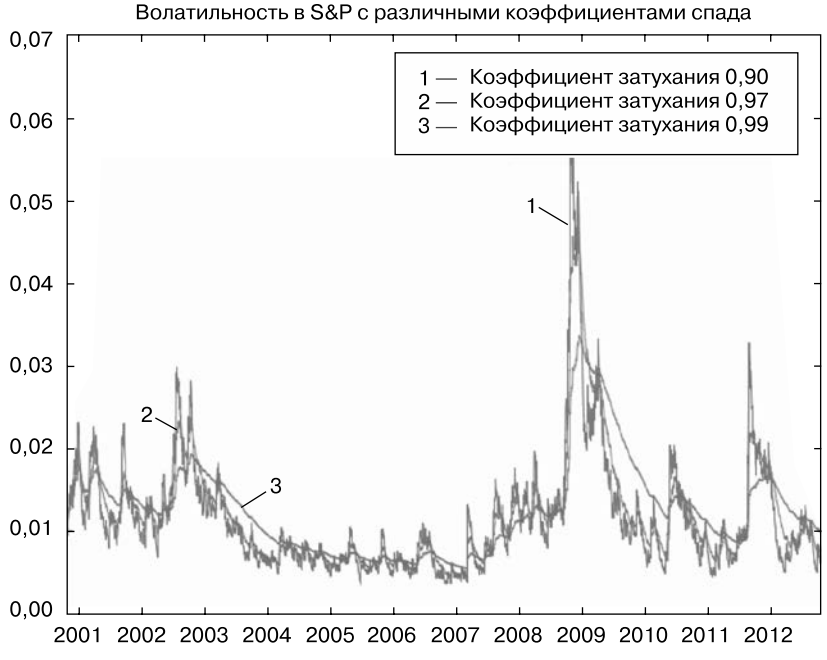


Рис. 6.12. Волатильность в S&P с различными коэффициентами затухания

На самом деле важно, какой понижающий вес коэффициент вы используете, это показано на рис. 6.12.

В действительности мы можем обмануть оценку риска (и людей тоже), выбирая такой понижающий вес фактор, который уменьшает степень риска.

Экспоненциальное понижающее взвешивание

Мы уже видели пример экспоненциального понижающего взвешивания в случае сохранения текущей оценки волатильности доходности S&P.

Главная формула для понижающего веса некоторой текущей аддитивной оценки E довольно проста. Мы придаем свежим данным больший вес, чем более старым, присваиваем значение понижающего взвешивания более старых данных s и используем его как параметр. Он называется *затуханием*. В простейшей форме мы получаем:

$$E_t = s \cdot E_{t-1} + (1-s) \cdot e_t,$$

где e_t — новый член.

Аддитивные оценки

Нам нужно, чтобы каждая из наших оценок была аддитивной (поэтому у нас есть оценка текущей дисперсии, а не среднего квадратического отклонения). Если то, к чему мы стремимся, является средневзвешенным, то следует иметь текущую оценку как числителя, так и знаменателя.

Если мы хотим быть осторожными в отношении плавности в начале (что более важно при наличии нескольких сотен точек данных или их меньшего количества), то фактически изменим параметр s через его обратную величину, которую можем считать как вид полураспада. Мы начинаем с полураспада 1 и наращиваем его до асимптотического «настоящего» периода полураспада $N = 1 / s$. Таким образом, при получении вектора v значений e_t, индексированных по дням t, мы делаем нечто наподобие этого:

```
true_N = N
this_N_est = 1.0
this_E = 0.0
for e_t in v:
    this_E = this_E * (1-1/this_N_est) + e_t * (1/this_N_est)
    this_N_est = this_N_est*(1-1/true_N) + N * (1/true_N)
```

Финансовое моделирование петли обратной связи

Есть некий аспект, который любые специалисты по количественному анализу или обработке данных должны понимать в финансовом моделировании, — это существование петли обратной связи. Найденный способ зарабатывать деньги в конечном итоге станет неактуальным — иногда люди называют это явление «рынок учится со временем».

Один из способов увидеть это заключается в следующем: со временем ваша модель сводится к пониманию того, что некий продукт (допустим) станет дорожать в будущем, так что вы покупаете его до подорожания, ждете, а затем продаете с прибылью. Но если вы думаете об этом, то ваша покупка фактически изменила процесс через влияние на рынок и снизила сигнал, который вы ожидали, по крайней мере, если другие участники рынка купили ее, поскольку она выглядела дешевой для них относительно предыдущей цены. Вы немного подняли цену, вследствие чего можете ожидать, что участники будут меньше покупать в ответ, а это значит, что общий сигнал меньше. Конечно, покупая только одну акцию в ожидании увеличения, вы оказываете минимальное влияние. Но если ваш алгоритм работает очень хорошо, то вы склонны делать ставки все больше и больше, воздействуя все сильнее. Действительно, почему бы вам не рискнуть большей суммой? Но нет. Через какое-то время вы узнаете оптимальную сумму, которой можно рискнуть, по-прежнему зарабатывая хорошие деньги, и эта оптимальная сумма достаточно велика, чтобы серьезно повлиять на рынок.

Так рынок учится — через комбинации множества алгоритмов ожидания событий и ухода их прочь.

Следствием данного обучения с течением времени является то, что существующие сигналы очень слабы. Вещи, которые были видны (по прошествии времени) невооруженным глазом в 1970-х годах, больше не доступны, поскольку все они понятны и ожидаемы участниками рынка (хотя могут появиться новые участники).

Главная суть в том, что в настоящее время мы довольны, имея 3 % корреляции для моделей с горизонтом один день (горизонт для модели — то, как долго вы ожидаете, что ваш прогноз будет хорошим). Это означает не слабый сигнал, а большое количество шума! Тем не менее вы все равно можете зарабатывать, если имеете такое преимущество и ваши торговые расходы достаточно малы.

В частности, большинство «метрик успеха» машинного обучения для моделей, таких как измерения точности или безошибочности, в этом контексте не очень актуальны.

Поэтому вместо измерения безошибочности мы обычно рисуем график для оценки моделей, как показано на рис. 6.13, а именно (кумулятивный) *PnL*-график модели. PnL означает Profit-and-Loss (прибыль и убыток) — это изменение день за днем (разница, а не отношение), или сегодняшнее значение минус вчерашнее.

Рис. 6.13. График кумулятивных PnL двух теоретических моделей

Подобную методику можно обобщить практически на любую модель — строите ли вы кумулятивную сумму продукта *заниженного* прогноза или заниженной реали-

зации. (Заниженное значение — такое, при котором среднее вычитается.) Другими словами, вы видите, что ваша модель регулярно работает лучше, чем «самая глупая» модель, считающая, будто все является средним.

Если на подобном графике кривая перемещается в верхний правый угол, то все в порядке. Слишком зазубренный график означает, что риски вашей модели слишком велики и она нестабильна.

Почему регрессия?

Итак, мы теперь знаем — в финансовом моделировании сигнал слабый. Если вы представите, будто есть некая сложная базовая взаимосвязь между вашей информацией и тем, что вы пытаетесь предсказать, то не пытайтесь выяснить суть этого — слишком много шума, чтобы его отыскать. Вместо этого подумайте о том, что функция, возможно, сложная, но непрерывная, и представьте, будто записали ее как ряд Тейлора. Тогда вы не можете рассчитывать на получение чего-то практического, кроме линейных членов.

Не думайте об использовании логистической регрессии, поскольку вам нужно будет игнорировать размер, который важен в финансах, — это имеет значение, если рынок растет на 2 % вместо 0,0 %. Но при логистической регрессии вам понадобится своеобразный двухпозиционный переключатель, что возможно, но приведет к потере массы информации. Учитывая тот факт, что мы всегда находимся в слабоинформированной среде, это плохая идея.

Обратите внимание: хоть мы и утверждаем, что вы, возможно, захотите использовать линейную регрессию в шумной среде, сами фактические члены не обязательно должны быть линейными в вашей информации. Вы всегда можете брать произведение разных членов как x в вашей регрессии, но по-прежнему описываете линейную модель в нелинейных членах.

Добавление гипотез

Одна из интерпретаций гипотез состоит в том, что их можно рассматривать как мнения, которые математически сформулированы и включены в наши модели. Фактически мы уже сталкивались с обычной гипотезой в форме понижающего взвешивания старых данных. Гипотезу можно описать как «новые данные важнее старых».

Кроме того, мы можем также решить рассмотреть нечто наподобие «коэффициенты меняются плавно». Это актуально, когда мы решаем, скажем, использовать большое количество старых значений некоего временного ряда, чтобы предсказать следующее значение, получая в результате примерно такую модель:

$$y = F_t = \alpha_0 + \alpha_1 F_{t-1} + \alpha_2 F_{t-2} + \in,$$

что является лишь примером, где мы берем последние два значения временного ряда *F* для предсказания следующего. Конечно, мы могли бы использовать более двух значений. Если бы мы применили много запаздывающих значений, то могли бы усилить нашу гипотезу, чтобы уравновесить введение такого большого количества степеней свободы. По сути, *гипотезы уменьшают степени свободы*.

Принцип, по которому мы размещали гипотезы о взаимосвязях между коэффициентами (в данном случае последовательные с задержкой времени точки данных), заключается в добавлении матрицы к нашей ковариационной матрице при выполнении линейной регрессии. Подробнее об этом на сайте https://mathbabe.org/2012/06/04/combining-priors-and-downweighting-in-linear-regression/.

Детская модель

Допустим, мы нарисовали график во временном ряду и нашли, что имеем сильную, но затухающую автокорреляцию до первых 40 лагов или нечто показанное на рис. 6.14.

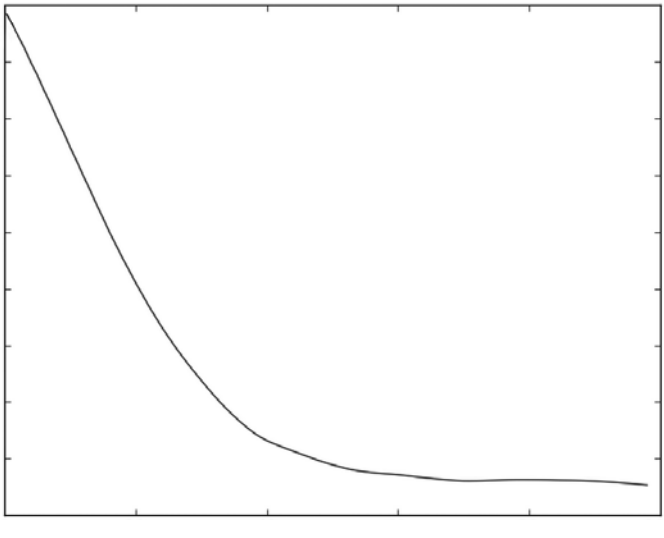


Рис. 6.14. Взгляд на автокорреляцию до 100 лагов

Мы можем рассчитать автокорреляцию, имея данные временных рядов. Мы создаем второй временной ряд, который является одним и тем же вектором данных, сдвинутых на один день (или некий фиксированный период времени), а затем вычисляем корреляцию между этими двумя векторами.

При желании предсказать следующее значение мы бы захотели использовать сигнал, который уже существует, зная последние 40 значений. С другой стороны, мы не

хотим делать линейную регрессию с 40 коэффициентами, поскольку это повлечет слишком много степеней свободы. Здесь идеальное место для гипотезы.

Хороший способ подумать о гипотезе — добавить член в функцию, которую мы стремимся минимизировать, измеряющий точность нашего подбора. Это называется штрафной функцией, а при отсутствии гипотезы является просто суммой квадратов ошибки:

$$F(\beta) = \sum_i (y_i - x_i\beta)^2 = (y - x\beta)^\tau (y - x\beta).$$

Если мы хотим минимизировать F, то берем ее производную по отношению к вектору коэффициентов β, приравниваем к нулю и решаем для β — вот единственное решение:

$$\beta = (x^\tau x)^{-1} x^\tau y.$$

Теперь, добавив стандарную гипотезу в виде штрафного члена для больших коэффициентов, мы имеем:

$$F_1(\beta) = \frac{1}{N}\sum_i (y_i - x_i\beta)^2 + \sum_j \lambda^2\beta_j^2 = \frac{1}{N}(y - x\beta)^\tau (y - x\beta) + (\lambda I\beta)^\tau (\lambda I\beta).$$

Вышеописанное можно решить и с помощью вычислений, и мы решаем для β, чтобы получить:

$$\beta_1 = (x^\tau x + N \cdot \lambda^2 I)^{-1} x^\tau y.$$

Другими словами, добавление штрафного члена для больших коэффициентов приводит к добавлению скалярного кратного матрицы тождественного преобразования к ковариационной матрице в закрытой форме решения β.

Если мы сейчас хотим добавить другой штрафной член, представляющий собой гипотезу «плавно меняющихся коэффициентов», то можем думать об этом, как требует такое понятие: *«соседние коэффициенты должны не слишком отличаться друг от друга»*, которое может быть выражено в следующей штрафной функции с новым параметром μ:

$$F_2(\beta) = \frac{1}{N}\sum_i (y_i - x_i\beta)^2 + \sum_j \lambda^2\beta_j^2 + \sum_j \mu^2(\beta_j - \beta_{j+1})^2 =$$

$$= \frac{1}{N}(y - x\beta)^\tau (y - x\beta) + \lambda^2\beta^\tau\beta + \mu^2(I\beta - M\beta)^\tau (I\beta - M\beta),$$

где M — матрица, содержащая нули всюду, кроме нижних элементов, находящихся вне диагоналей, где она содержит l. Тогда $M\beta$ — вектор, который является результатом сдвига коэффициентов β на единицу и замены последнего коэффициента на 0. Матрица M называется *оператором сдвига*, а разность $I - M$ можно рассматривать как *дискретный оператор производной* (получить дополнительную информацию о дискретном исчислении можно на сайте https://en.wikipedia.org/wiki/Finite_difference).

Поскольку это самая сложная версия, то рассмотрим ее подробно. Вспомним наше векторное исчисление: производная скалярной функции $F_2(\beta)$ по отношению к вектору β является вектором и удовлетворяет совокупности свойств, которые происходят на скалярном уровне, включая тот факт, что она аддитивна и линейна и что:

$$\frac{\partial u^{\tau} \cdot u}{\partial \beta} = 2\frac{\partial u^{\tau}}{\partial \beta} u.$$

Применяя предыдущие правила, мы имеем:

$$\frac{\partial F_2(\beta)}{\partial \beta} = \frac{1}{N}\frac{\partial (y - x\beta)^{\tau}(y - x\beta)}{\partial \beta} + \lambda^2 \cdot \frac{\partial \beta^{\tau}\beta}{\partial \beta} + \mu^2 \cdot \frac{\partial ((I - M)\beta)^{\tau}((I - M)\beta)}{\partial \beta} =$$

$$= \frac{-2}{N} x^{\tau}(y - x\beta) + 2\lambda^2 \cdot \beta + 2\mu^2 (I - M)^{\tau}(I - M)\beta \quad .$$

Приравнивая это к 0 и решая для β, получаем:

$$\beta_2 = (x^{\tau}x + N \cdot \lambda^2 I + N \cdot \mu^2 \cdot (I - M)^{\tau}(I - M))^{-1} x^{\tau} y.$$

Другими словами, у нас есть еще одна матрица, добавленная к нашей ковариационной матрице, которая выражает гипотезу о плавном изменении коэффициентов. Обратите внимание: симметричная матрица $(I - M)^{\tau}(I - M)$ имеет 1 вдоль своих под- и наддиагоналей, но также имеет 2 вдоль диагонали. Другими словами, нужно скорректировать λ, когда мы корректируем μ, поскольку эти члены взаимодействуют.

Гипотезы и производные высших порядков

При желании вы можете добавить гипотезу о второй производной (или других высших производных), возведя в квадрат производную оператора (I – M) (или возведя в более высокие степени).

Итак, какова модель? Помните, что мы будем выбирать экспоненциальный понижающий весовой член y для наших данных и станем хранить текущую оценку как $x\tau x$ и $x\tau y$, подобно рассмотреному выше. Тогда y, λ и μ являются гиперпараметрами нашей модели. Обычно мы имеем представление о том, насколько большой y должен основываться на рынке, а два других параметра зависят друг от друга и от самих данных. Именно здесь ремесло превращается в искусство — необходимо в некоторой степени оптимизировать эти варианты в имеющихся рамках; главное, не перестараться и не достичь состояния переобучения.

Упражнение: GetGlue и данные о событиях с метками даты/времени

GetGlue любезно предоставил набор для изучения их данных, которые содержат события с метками времени пользователей, вошедших в систему и оценивающих телевизионные шоу и фильмы.

Необработанные данные (http://getglue-data.s3.amazonaws.com/getglue_sample.tar.gz) на основе каждого пользователя с 2007 по 2012 год показывают только рейтинги и отметки телевизионных шоу и фильмов. Это составляет менее 3 % от их общих данных, и даже в таком случае они большие, а именно 11 ГБайт без сжатия.

Здесь код на языке R для просмотра первых десяти строк:

```
#
# Автор: Джаред Ландер (Jared Lander)
#
require(rjson)
require(ptyr)

# расположение данных
datapath <- "http://getglue-data.s3.amazonaws.com/ getgtue_sampte.tar.gz"

# создать соединение, которое может разархивировать файл
theCon <- gzcon(url(dataPath))

# прочитать десять строк данных
n.rows <- 10
theLines <- readLines(theCon, n=n.rows)

# проверить структуру данных
str(theLines)

# обратите внимание, что первый элемент отличается от остальных
theLines[1]

# используем fromJSON на каждом элементе вектора, кроме первого
theRead <- lapply(theLines[-1], fromJSON)

# направляем их все в data.frame
theData <- ldply(theRead, as.data.frame)

# смотрим, как мы это сделали
View(theData)
```

Начните с шагов, представленных ниже.

1. Загрузите 1000 строк данных и некоторое время изучайте их зрительно. Для всего задания работайте с этой 1000 строк, пока ваш код в хорошей форме. Затем вы можете увеличить до 100 тыс. или 1 млн строк.
2. Объедините и подсчитайте данные. Найдите ответы на следующие вопросы.
   - Сколько уникальных действий может совершить пользователь? И сколько действий каждого типа в этом наборе данных?
   - Сколько уникальных пользователей в этом наборе данных?
   - Как называются десять самых популярных фильмов?
   - Сколько событий в этом наборе данных произошло в 2011 году?

3. Предложите пять новых вопросов, которые, по вашему мнению, интересны и заслуживают изучения.
4. Исследуйте/ответьте на ваши вопросы.
5. Визуализируйте. Предложите одну визуализацию, которая, по вашему мнению, захватывает нечто интересное в этом наборе данных.

Упражнение: финансовые данные

Следующее упражнение поможет изучить концепции этой главы.

1. Получите данные: перейдите к Yahoo! Finance и скачайте ежедневные данные биржи, которая имеет данные не менее чем за восемь лет; убедитесь, что они идут от более ранних к поздним. Если не знаете, как это сделать, то используйте Google.
2. Создайте временные ряды ежедневной логарифмической доходности биржевого курса.
3. Для сравнения сделайте то же самое для данных объема биржевых торгов (то есть создайте временные ряды ежедневных логарифмических изменений объема торгов).
4. Затем попробуйте установить модель линейной регрессии, которая использует последние два возврата значения для прогнозирования следующего значения. Запустите ее и посмотрите, сможете ли вы с ее помощью заработать деньги. Попробуйте как для доходов, так и для объемов торгов. Бонусные очки, если вы сделаете причинно-следственную модель, нормализуйте по волатильности (стандартное отклонение) или добавьте экспоненциальное затухание для старых данных.
5. Начертите кумулятивные графики прибыли и убытков (прогнозный и полученный) и посмотрите, движутся ли они вверх.

7 Извлечение смысла из данных

Каким образом компании извлекают смысл из данных, которые у них есть?

В этой главе мы послушаем двух специалистов с очень разными подходами к данному вопросу, а именно Уильяма Кукерски (William Cukierski) из Kaggle (https://www.kaggle.com/) и Дэвида Хаффакера (David Huffaker) из Google.

Уильям Кукерски

Уилл получил степень бакалавра по физике в Корнеллском университете (Cornell) и степень доктора философии по биомедицинской инженерии в Ратгерском университете (Rutgers). Он специализировался на исследованиях рака, изучении изображений патологии. Работая над написанием диссертации, он все чаще и чаще участвовал в соревнованиях, проводимых Kaggle (подробнее об этой компании будет рассказано чуть позже), неоднократно занимал призовые места и теперь работает на Kaggle.

После того как Уилл предоставит общую информацию о соревнованиях по анализу данных и краудсорсингу, он объяснит, как его компания работает не только для участников платформы, но и для более крупного сообщества.

Затем Уилл уделит внимание *выделению* и *выбору признаков*. Вкратце о выделении признаков: вы берете необработанный дамп имеющихся данных и очень тщательно его проверяете, чтобы избежать «мусора на входе, мусора на выходе», который получаете, если просто загружаете необработанные данные в алгоритм без предварительного анализа. Выбор признаков — это процесс построения подмножества данных или функций данных, которые станут показателями или переменными для ваших моделей и алгоритмов.

Общая информация: соревнования по анализу данных

В области машинного обучения существует история соревнований по анализу данных: люди или команды соревнуются в течение нескольких недель или месяцев, чтобы разработать алгоритм предсказания. То, что он предсказывает, зависит от конкретного набора данных, но некоторые примеры включают такие прогнозы, как: может ли данный человек попасть в автокатастрофу или понравится ли ему определенный фильм. Предоставляются обучающий набор, заранее определенная метрика оценки, а также некий набор правил, например, как часто участники соревнований могут представлять свои прогнозы, могут ли команды сливаться в более крупные и т. д.

Примеры соревнований по компьютерному обучению включают ежегодное соревнование по обнаружению знаний и интеллектуальному анализу данных (Knowledge Discovery and Data Mining, KDD), однократно проводившийся Netflix, приз — 1 млн долларов (конкурс, который длился два года), и, как мы узнаем немного позже, сам Kaggle.

Сделаем ряд замечаний об оправданности соревнований по анализу данных. Во-первых, такие соревнования являются частью экосистемы науки о данных — одной из культурных сил, играющих роль в современном ландшафте этой науки, вследствие чего исследователи данных должны иметь о них представление.

Во-вторых, создание этих соревнований ставит человека в положение систематизирующего науку о данных или определяющего ее сферу. Осмысление проблем, которые они порождают, дает нам набор примеров для изучения центрального вопроса нашей книги: что такое наука о данных? Это не говорит о том, что мы однозначно примем подобное определение, но мы можем хотя бы использовать его в качестве отправной точки: какие атрибуты существующих соревнований затрагивают науку о данных и какие ее аспекты отсутствуют?

Наконец, участники различных соревнований попадают в рейтинг, и поэтому одним из показателей «топового» исследователя данных может быть его позиция на таких соревнованиях. Но обратите внимание: многие ведущие специалисты, особенно женщины, и в том числе авторы этой книги, не принимают в них участия. На самом деле на верхних позициях мало женщин, и мы считаем, что это явление необходимо хорошо обдумать, если ожидаем, что лидерство в рейтинге будет выступать в роли посредника для талантов в области науки о данных.

Соревнования по анализу данных убирают весь мусор

Соревнования можно рассматривать как формализованные, сухие и синтетические по сравнению с тем, с чем вы сталкиваетесь в нормальной жизни. Соревнования удаляют мусор, прежде чем вы начнете строить модели, — задаются хорошие вопросы, собираются и очищаются данные и т. д. Команда исследова-

телей данных Kaggle на самом деле тратит много времени на создание набора данных и метрик оценки и выяснение того, какие вопросы задавать, поэтому вопрос заключается в следующем: пока они занимаются наукой о данных, являются ли они участниками соревнований?

Общая информация: краудсорсинг

Существует две модели краудсорсинга. Во-первых, есть *дистрибутивная* модель, наподобие «Википедии», которая предназначена для относительно упрощенных, но широкомасштабных пополнений. В этой онлайн-энциклопедии любой человек в мире может внести вклад в содержание и существует система регулирования и контроля качества, созданная добровольцами. Конечный результат — довольно качественный сборник всех человеческих знаний (более или менее).

Далее, есть особые, узкоспециализированные, сложные проблемы, которыми занимаются Kaggle (https://www.kaggle.com/), DARPA (https://en.wikipedia.org/wiki/DARPA#Active_projects), InnoCentive (http://www.innocentive.com/) и др. Эти компании ставят задачу перед общественностью, но, как правило, конкурируют только люди с высокоспециализированными навыками. Обычно денежный приз, слава или уважение, которые достаются вашему сообществу, связаны с победой.

Исторически сложилось так, что у проектов краудсорсинга есть ряд проблем, которые могут повлиять на их полезность. Несколько аспектов влияет на вероятность участия людей. Во-первых, многим не хватает метрики оценки. Как вы решаете, кто победит? В ряде случаев метод оценки не всегда объективен. Может присутствовать субъективная оценка, когда судьи решают, что ваш дизайн плох, или у них просто другой вкус. Это приводит к высокому порогу вхождения, поскольку люди не доверяют критерию оценки. Кроме того, никто не получает признания до тех пор, пока не выиграет или по крайней мере не будет оценен по достоинству. Данные обстоятельства приводят к большим невозмещаемым расходам для участников, о которых люди знают заранее, — это может стать еще одним препятствием для участия.

Организационные факторы также могут послужить барьером на пути к успеху. Плохо организованные соревнования объединяют участников с механическими турками (https://www.mturk.com/): другими словами, предполагают, что участники будут бестолковыми, и предоставляют некорректные вопросы и скудные призы. Этот прецедент просто плох для всех, поскольку деморализует исследователей данных и не помогает предприятиям отвечать на более важные вопросы, которые выжали бы максимум из их данных. Еще одна распространенная проблема заключается в том, что соревнования не разбивают работу на удобные небольшие части. Либо вопрос слишком велик, чтобы за него взяться, либо слишком мал, чтобы быть интересным.

Наученные этими ошибками, мы ожидаем, что хорошие соревнования будут иметь приемлемый интересный вопрос с метрикой оценки, которая является прозрачной и объективной. Даются задание, набор данных, метрика успеха. Кроме того, призовой фонд установлен заранее.

Приведем немного исторического контекста для краудсорсинга, поскольку идея не нова. Вот несколько примеров.

- В 1714 году британский военно-морской флот не смог измерить долготу (http://www.crowdsourcing.com/cs/2008/05/chapter-7-what.html) и выставил приз в размере 6 млн сегодняшних долларов за помощь. Джон Харрисон, неизвестный столяр-краснодеревщик, выяснил, как сделать часы для решения этой проблемы (https://www.goodreads.com/book/show/4806.Longitude).
- В 2002 году телевизионная сеть Fox выпустила приз для следующего сольного поп-исполнителя, в результате чего появилось телешоу American Idol («Американский идол») (https://ru.wikipedia.org/wiki/American_Idol), в котором участники соревнуются в конкурсе по вокалу, проводимом в духе отборочного турнира.
- Существует также компания X-Prize (https://www.xprize.org/), которая предлагает «мотивирующие призовые соревнования... чтобы совершать радикальные прорывы в интересах человечества, тем самым вдохновляя формирование новых отраслей и оживление рынков». Приз Ansari X-prize в области космических полетов составлял в общей сложности 10 млн долларов, при этом суммарные инвестиции участников соревнования в решение задачи составили 100 млн долларов. Обратите внимание: вышеописанное показывает, что такой процесс в целом не всегда оправдан, но, с другой стороны, может быть очень эффективным для людей, предлагающих приз за решение.

ТЕРМИНОЛОГИЯ: КРАУДСОРСИНГ И МЕХАНИЧЕСКИЕ ТУРКИ

Поговорим о паре терминов, которые стали проникать в разговорный язык в течение последних нескольких лет.

Хотя концепция краудсорсинга — использование многих людей независимо друг от друга для решения проблемы — не нова, этот термин был введен в обращение в 2006 году. Основная идея заключается в том, что ставится задача, а конкурсанты соревнуются, чтобы найти лучшее решение. *The Wisdom of Crowds* («Мудрость толпы») — книга, написанная Джеймсом Суровецки (James Suriowiecki) (Anchor, 2004). Ее центральным тезисом является феномен, связанный с тем, что в среднем толпы людей будут принимать более рациональные решения, чем эксперты, только при определенных условиях (независимость отдельных лиц вместо группового мышления, где группа разговаривающих между собой людей может повлиять на принятие друг другом совершенно неверных решений), когда группы людей могут прийти к правильному решению. И только некоторые проблемы хороши для этого подхода.

Mechanical Turk от Amazon — онлайн-служба краудсорсинга, где людям задают задачи. Например, набор изображений, которые нужно пометить как «радостные» или «грустные». Эти метки могут затем использоваться в качестве основы обучающего набора для задачи обучения под наблюдением. Затем алгоритм можно было бы обучать по изображениям, отмеченным человеком, для автоматической маркировки новых изображений. Таким образом, центральная идея Mechanical Turk заключается в следующем: люди выполняли бы довольно рутинные задания, чтобы помочь машинам, а затем машины автоматизировали бы задачи, помогающие людям! Любой исследователь, которому требуется автоматизировать выполнение задания, может использовать Mechanical Turk от Amazon, если предоставляет людям вознаграждение. И любой человек может зарегистрироваться и стать частью службы краудсорсинга, хотя есть некоторые проблемы контроля качества: если исследователь осознает, что человек просто отмечает каждый следующий рисунок как «радостный», а не смотрит на изображения, то этого человека больше не задействуют в работе.

Mechanical Turk — пример искусственного искусственного интеллекта (да, дважды «искусственного») в том смысле, что люди помогают машине, которая помогает людям.

Модель Kaggle

> Быть исследователем данных — это как изучать все больше и больше о все большем и большем количестве предметов, пока не будете знать ничего ни о чем.
>
> *Уилл Кукерски*

Kaggle — компания, чей рекламный слоган гласит: «Мы делаем науку о данных спортом». Kaggle устанавливает связи с компаниями и с исследователями данных. За вознаграждение Kaggle проводит соревнования для предприятий, которые, по существу, хотят использовать ресурсы (или более широкое сообщество по обработке данных) для решения своих проблем с данными. Kaggle обеспечивает инфраструктуру и привлекает таланты в даталогии.

У компании есть также группа штатных первоклассных исследователей данных, включая самого Уилла. Компании являются их клиентами, приносящими доход, и предоставляют наборы данных и проблемы с данными, которые хотят решить. Kaggle привлекает для решения задачи исследователей данных по всему миру. Поучаствовать может любой. Сначала опишем опыт в Kaggle для исследователя данных, а затем обсудим клиентов.

Единственный участник

В соревнованиях Kaggle вам предоставляется обучающий набор, а также тестовый, в котором скрыты игреки, но иксы даны, поэтому вы просто используете свою модель в целях получения предсказанных значений иксов для набора тестов и загрузки их в систему Kaggle, чтобы увидеть свой оценочный балл. Таким образом, вы не делитесь своим фактическим кодом с Kaggle, если, конечно, не выиграете приз (и Kaggle не должен беспокоиться о том, какую версию Python вы используете). Обратите внимание на то, что даже выдача только иксов — это реальная информация, в частности, она говорит о том, например, для каких размеров иксов должен быть оптимизирован ваш алгоритм. Кроме того, для целей конкурса существует третий контрольный набор, к которому у участников нет доступа. Вы не видите иксы или игреки, служащие для определения победителя в конце соревнований.

В Kaggle участникам предлагается представить свои модели до пяти раз на день во время соревнований, которые продолжаются несколько недель. Когда конкурсанты представляют прогнозы, таблица лидеров Kaggle немедленно обновляется, чтобы отобразить текущую метрику оценки участника в тестовом наборе. При наличии достаточного количества конкурентов мы видим, что они совершают скачки, как показано на рис. 7.1, где один получил пятипроцентное преимущество, дав другим стимул работать больше. Это также устанавливает диапазон точности для задачи, которого у вас вообще нет. Другими словами, не имея какой-либо другой информации, поскольку над проблемой, над которой вы трудитесь, не работает больше никто, вы не знаете, действительно ли ваша модель, правильная на 75 %, является наилучшей.

Этот эффект скачков и хорош и плох. Он побуждает людей создавать более эффективные модели, возможно, с риском переобучения, но также имеет тенденцию усложнять модели по мере их улучшения. Одна из причин, по которой нежелательны слишком длительные соревнования, заключается в том, что спустя энное время единственным способом повысить эффективность становится доведение сложности до абсурда. Например, изначальные соревнования Netflix Prize продолжались два года, а окончательная победившая модель была слишком сложной для ее фактического ввода в производство.

Их клиенты

Так почему компании будут платить за работу с Kaggle? Ниша, которую заполняет Kaggle, — существующее отсутствие согласованности между теми, кто нуждается в анализе, и теми, у кого есть навыки. Несмотря на то что компании отчаянно нуждаются в анализе, они, как правило, имеют тенденцию накапливать данные; это самое большое препятствие на пути к успеху для тех компаний, которые даже проводят конкурсы Kaggle. Во многих компаниях соревнования не проводятся по аналогичным причинам. Инновация Kaggle заключается в том, что они убеждают

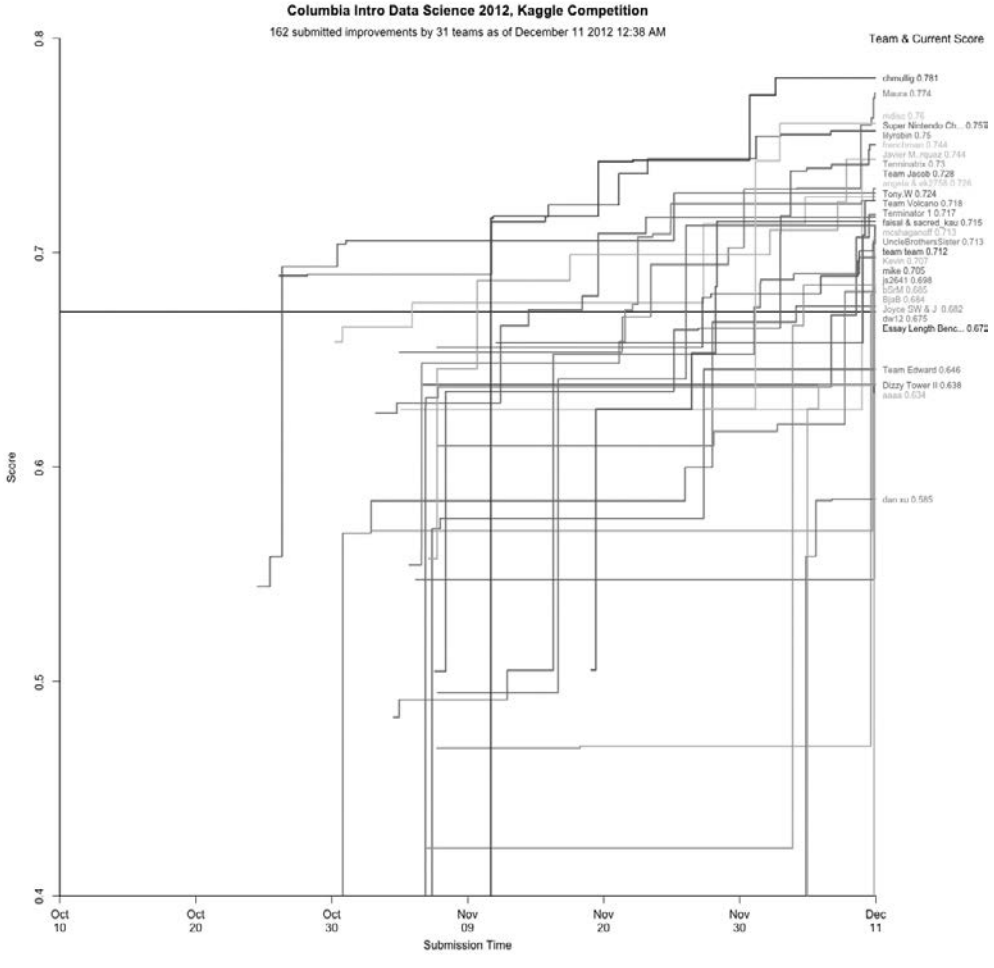


Рис. 7.1. Крис Миллиган (Chris Mulligan), студент из класса Рэйчел, создал эту визуализацию скачков, чтобы отобразить, как в реальном времени проходили соревнования в течение всего семестра

предприятия делиться собственными данными ради того преимущества, которое получат, если их большие проблемы с данными будут решаться для них через краудсорсинг десятков тысяч специалистов, сотрудничающих с Kaggle по всему миру.

Участники Kaggle уже дали хорошие результаты. Allstate, компания по страхованию автомобилей, у которой и так есть хорошая команда, поставила перед участниками в соревнованиях по анализу данных задачу улучшить свою модель страхования, которая, учитывая характеристики водителей, приближенно определяет вероятность аварии автомобиля. Двести два конкурента улучшили модель

Allstate, снизив на 271 % нормализованный коэффициент Джини. Другой пример включает компанию, где приз для участников соревнований составил 1000 долларов, а компания получила прибыль в размере порядка 100 000.

СПРАВЕДЛИВО ЛИ ЭТО?

Справедливо ли это по отношению к исследователям данных, которые уже трудятся в компаниях, сотрудничающих с Kaggle? Часть из них могут потерять работу, например, если результат соревнования лучше, чем внутренняя модель. Справедливо ли заставлять людей, по существу, работать бесплатно и в конечном счете приносить прибыль коммерческой компании? Неужели это приводит к тому, что исследователи данных теряют свою справедливую рыночную стоимость? Kaggle взимает плату за проведение соревнований и предлагает четко определенные призы, так что исследователь данных всегда может выбрать, принимать ли участие. Достаточно ли этого?

Похоже, что это может быть отличной возможностью для компаний, но только пока исследователи данных в мире не осознают свою ценность и располагают свободным временем. Как только люди начинают выше оценивать свои навыки, они могут дважды подумать, работать ли им (почти) бесплатно, если только это не касается дела, в которое они действительно верят.

Когда в Facebook нанимали исследователей данных, компания организовала соревнования Kaggle, где премией выступало *интервью*. Было 422 участника. Мы считаем, что для Facebook удобно проводить интервью на должность исследователя данных с позиции благодарности за простое интервью. Кэти думает, что это отвлекает специалистов от задавания неудобных вопросов о том, что такое информационная политика и лежащая в основе этика компании.

СОРЕВНОВАНИЕ ПО ОЦЕНКЕ ЭССЕ НА ОСНОВЕ ПЛАТФОРМЫ KAGGLE

Частью выпускного экзамена для курса в Колумбийском университете был конкурс по оценке эссе. Студенты должны были его создать, обучить и тестировать, как и любое другое соревнование в Kaggle; поощрялась групповая работа. Подробности конкурса эссе обсуждаются ниже, а доступ к данным вы получите на сайте https://www.kaggle.com/competitions?sortBy=grouped&group=inClass.

Вам предоставляется доступ к рассортированным вручную эссе, чтобы вы могли создавать, тренировать и тестировать автоматический механизм подсчета баллов. Успех зависит от того, насколько вы сможете приблизить ваши оценки к тем, которые были поставлены реальными экспертами-экзаменаторами.

Для этого конкурса есть пять наборов эссе. Каждый из них был создан по одному вопросу. Средняя длина отобранных эссе варьируется от 150 до 550 слов на ответ. Одни эссе зависят от исходной информации, а другие — нет. Все ответы были написаны студентами, расположенными по уровню обучения от 7 до 10. Все ответы были

написаны студентами разных годов обучения, от 7 до 10. Все эссе были рассортированы вручную и дважды оценены. Каждый из наборов данных имеет уникальные характеристики. Подобное разнообразие важно для проверки пределов возможностей вашего механизма оценки. Данные содержат следующие столбцы:

- id — уникальный идентификатор для каждого набора эссе отдельного студента;
- 1–5 — id для каждого из наборов эссе;
- essay — текст в кодировке ASCII ответа студента;
- rater1 — оценка первого эксперта;
- rater2 — оценка второго эксперта;
- grade — итоговая оценка экспертов.

Мысленный эксперимент: каковы этические последствия использования робота-оценщика?

Уилл попросил студентов рассмотреть вопрос о том, хотят ли они, чтобы их эссе автоматически оценивались с помощью базового компьютерного алгоритма, и каковы этические последствия автоматического выставления оценок. Вот некоторые из мыслей студентов.

- *Оценщики-люди не всегда справедливы*. В случае с медиками были проведены исследования, где одному врачу показывают один и тот же слайд с интервалом два месяца и врач ставит разные диагнозы. Мы противоречим сами себе, даже если думаем, что это не так. Будем помнить об этом, когда говорим о «справедливости» использования алгоритмов машинного обучения в сложных ситуациях. Машинное обучение применялось для исследования рака, где ставки намного выше, хотя, вероятно, нужно меньше усилий для того, чтобы их разыграть.
- *Являются ли вещи, созданные машинами, более структурированными и не препятствует ли это творчеству?* Некоторые могут утверждать, что люди *хотят*, чтобы вещи были стандартизированы. (Это также зависит от того, насколько вы действительно беспокоитесь о своей оценке.) Это дает постоянство, которое нам нравится. Например, людям не нужны художественные машины; они хотят безопасные автомобили. Тем не менее разумно ли переходить от «человеческой» версии того же самого предмета к машинной в любом случае? Существует ли универсальный ответ, или этот вопрос зависит от каждого конкретного случая?
- *Что является целью: написание хорошего эссе или успешное выполнение стандартного теста?* Если подразумевается последнее, то вы можете также рассматривать тест как отбор: следуете инструкциям и получаете оценку в зависимости

от того, насколько хорошо это делаете. Кроме того, реальное самостоятельно получающее прибыль подразделение стандартизованного тестирования, возможно, продает книги, рассказывающие вам о приемах прохождения тестов. Как это трактуется здесь? Одним из возможных способов истолкования было бы владение алгоритмами, которые обходили бы оценочные алгоритмы, создавая эссе, получающие хорошие оценки, но не написанные вручную. Тогда мы могли бы видеть, что образование превращается в войну машин, между алгоритмами, имеющимися у студентов, и алгоритмами, имеющимися у учителей. Вероятно, в этой войне мы бы сделали ставку на студентов.

ЗНАНИЯ ПРЕДМЕТНОЙ ОБЛАСТИ ИЛИ АЛГОРИТМЫ МАШИННОГО ОБУЧЕНИЯ

Это ложное разделение. Это не «или-или». Вам нужно и то и другое для решения задач науки о данных. Однако президент Kaggle Джереми Ховард (Jeremy Howard) очень разозлил некоторых узких специалистов, дав интервью журналу New Scientist («Новый ученый») в декабре 2012 года и Питеру Алдхаусу (Peter Aldhous) под заголовком *Specialist Knowledge is Useless and Unhelpful* («Знания специалистов ненужные и бесполезные»). Ниже приведен отрывок.

ПА: Что отделяет победителей от проигравших?

ДХ: Разница между хорошими участниками и плохими — информация, которой они снабжают алгоритмы. Вы должны решить, как абстрагироваться от данных. Победителями соревнований Kaggle, как правило, являются любопытные и творческие люди. Они придумали дюжину совершенно новых способов поразмышлять о проблеме. Самое приятное в таких алгоритмах, как случайный лес, — то, что вы можете вбросить в них как можно больше безумных идей, и алгоритмы определяют, какие из них работают.

ПА: Это совсем не похоже на традиционный подход к созданию моделей прогнозирования. Как отреагировали эксперты?

ДХ: Эти идеи некомфортны для многих людей. Они провокационны, поскольку мы говорим: «Десятилетия, потраченные вами на приобретение специальных знаний, не только не нужны, но и бесполезны; ваши сложные методы хуже, чем обобщенные». Это сложно для людей, привыкших к науке старого образца. Они тратят столько времени, чтобы обсудить, имеет ли идея смысл. Они проверяют визуализацию и обмозговывают ее. Все это практически бесполезно.

ПА: Отводите ли вы какую-либо роль экспертным знаниям?

ДХ: Некоторые специалисты требуются на раннем этапе, когда вы пытаетесь понять, какую проблему решаете. Необходимый вам опыт — это стратегический опыт в ответе на данные вопросы.

ПА: Можете ли вы увидеть какие-либо недостатки подхода на основе управляемого данными черного ящика, который доминирует в Kaggle?

ДХ: Некоторые люди считают, что вы не получите более глубокого понимания проблемы. Но это совсем не так: алгоритмы говорят, что важно, а что — нет. Вы можете спросить, почему эти вещи важны, но я думаю, что это менее интересно. В итоге вы получаете рабочую модель прогнозирования. Тут не о чем спорить.

Выбор признаков

Идея выбора признаков — это идентификация подмножества данных или преобразованных данных, которые вы хотите поместить в свою модель.

До работы в Kaggle Уилл занимал высокие места в соревнованиях (так он получил работу), поэтому он знает не понаслышке, что нужно для создания эффективных моделей прогнозирования. Выбор функции полезен не только для победы в соревнованиях — это важная часть построения статистических моделей и алгоритмов в целом. Наличие данных не означает, что *все* они должны войти в модель.

Так, возможно, что в ваших исходных данных много избыточных или коррелированных переменных, и поэтому вы не хотите включать их в модель. Аналогично вы можете построить новые переменные, преобразовывая старые путем логарифмирования, например, или превращая непрерывную переменную в двоичную, прежде чем вводить их в модель.

Терминология: признаки, объясняющие переменные, показатели

Различные области научных знаний используют разные термины для описания одного и того же понятия. Статистики говорят «объясняющие переменные», или «зависимые переменные», или «показатели», когда описывают подмножество данных, которое подается на вход модели. Специалисты в области компьютерных наук говорят «признаки».

Выделение и выбор признаков — это наиболее важные, но недооцененные шаги машинного обучения. Лучшие признаки лучше самых прекрасных алгоритмов.

Уилл Кукерски

У нас нет лучших алгоритмов, у нас просто есть больше данных.

Питер Норвиг (Peter Norvig), директор по исследованиям в Google

Возможно, рассуждает Уилл, Норвиг на самом деле хотел сказать, что у нас есть *лучшие признаки*? Понимаете, больше данных — иногда это просто больше данных (например, я могу записывать броски кубиков до бесконечности, но через некоторое время не получу никакого дополнительного значения, поскольку мои

признаки будут сходиться). Однако для более интересных проблем, с которыми сталкивается Google, рельеф признаков достаточно сложен/богат/нелинеен для того, чтобы извлечь выгоду из сбора данных, позволяющих получить эти признаки.

Почему? Мы получаем все бóльшие и бóльшие наборы данных, но это не всегда полезно. Если количество признаков больше, чем наблюдений, или есть проблема разреженности, то большой размер не обязательно будет хорошим. И если огромный размер данных просто затрудняет действия с ними по причинам, связанным с вычислениями (например, они не могут уместиться на одном компьютере, вследствие чего должны быть сегментированы на нескольких машинах), не улучшая наш сигнал, то это чистый минус.

Чтобы улучшить эффективность моделей прогнозирования, нужно усовершенствовать процесс выбора признаков.

Пример: привлечение пользователей

Приведем пример, чтобы вы взяли его на заметку, прежде чем мы вникнем в ряд возможных методов. Предположим, у вас есть приложение, которое вы создали, назовем его Chasing Dragons («Погоня за драконами», рис. 7.2), и пользователи платят ежемесячную абонентскую плату за его применение. Чем больше у вас пользователей, тем больше денег вы получаете. Предположим, вы понимаете, что только 10 % новых пользователей когда-либо возвращаются после первого месяца. Таким образом, у вас есть два варианта увеличения вашего дохода: найти способ увеличить процент сохранения существующих пользователей или приобрести новых. Как правило, дешевле сохранить существующего клиента, чем найти и привлечь новых. Но, отбросив этот конкретный анализ рентабельности приобретения или удержания, сосредоточимся на ситуации с сохранением вашего пользователя. Мы построим модель, которая *предсказывает*, вернется ли новый пользователь в следующем месяце в зависимости от его поведения в текущем. Вы могли бы построить такую модель, чтобы *понять* вашу ситуацию с удержанием, но вместо этого сосредоточимся на построении алгоритма, который является очень точным при *прогнозировании*. Возможно, вы захотите задействовать эту модель, чтобы дать бесплатный месяц пользователям, которых, как вы предполагаете, нужно дополнительно поощрить, чтобы они приняли решение задержаться.

Хорошей, грубой, простой моделью, с которой вы могли бы начать, была бы логистическая регрессия, впервые представленная в главе 4. Она даст вероятность того, вернется ли пользователь во втором месяце в зависимости от его деятельности в течение первого месяца. (Существует богатый набор статистической литературы под названием *Survival Analysis* («Анализ выживаемости»), который также может хорошо работать, но в данном случае это не обязательно — то, на чем мы хотим сосредоточиться здесь, — не часть, связанная с моделированием, а данные.) Вы фиксируете поведение каждого пользователя в течение первых 30 дней после

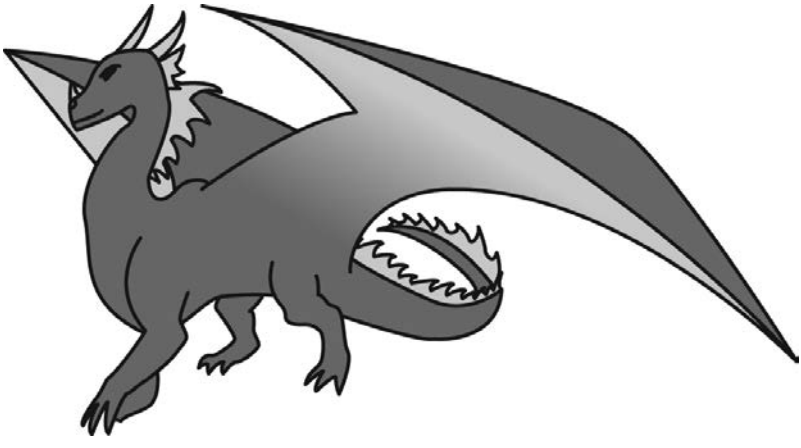


Рис. 7.2. Chasing Dragons, разработанное вами приложение

регистрации. Вы можете регистрировать каждое действие, которое пользователь произвел, с отметками времени: нажал кнопку «Уровень 6» в 05:22 утра, убил дракона в 05:23 утра, получил 22 балла в 05:24 утра, пользователю была показана реклама дезодоранта в 05:25 утра. Это будет этап сбора данных. Любое действие, которое пользователь может предпринять, регистрируется.

Обратите внимание: одни пользователи могут иметь тысячи таких действий, а у других их может быть только несколько. Все они будут храниться в журналах событий с метками времени. Затем вам необходимо преобразовать эти журналы в наборы данных со строками и столбцами, где каждая строка была бы пользователем, а каждый столбец — признаком. На данный момент вы не должны быть избирательными; вы находитесь в фазе генерации признаков. Таким образом, ваша научная команда (гейм-дизайнеры, разработчики программного обеспечения, статистики и специалисты по маркетингу) может произвести мозговой штурм признаков. Вот несколько примеров:

- количество дней в первом месяце, когда пользователь посетил сайт;
- время до второго посещения;
- количество очков в j-й день для $j = 1...30$ (это будет 30 отдельных признаков);
- общее количество очков за первый месяц (сумма других признаков);
- заполнил ли пользователь профиль в Chasing Dragons (по двоичной шкале — 1 или 0);
- возраст и пол пользователя;
- размер экрана устройства.

Используйте воображение и придумайте как можно больше признаков. Обратите внимание: среди них присутствуют избыточность и корреляции; ничего страшного.

ГЕНЕРАЦИЯ ПРИЗНАКОВ ИЛИ ИЗВЛЕЧЕНИЕ ПРИЗНАКОВ

Мы только что осуществили данный процесс, формируя список признаков для Chasing Dragons, — это процесс *генерации признаков* (feature generation) или *извлечения признаков* (feature extraction). Он является настолько же искусством, насколько наукой. Хорошо иметь в распоряжении эксперта в области данных процессов, но также полезно использовать свое воображение.

В сегодняшней технологической среде мы находимся в состоянии, когда можем генерировать множество признаков путем ведения журнала. Например, сравните это с другими ситуациями, такими как опросы, — вам повезло, если найдете респондента, который ответит на 20 вопросов, не говоря уже о сотнях.

Но сколько из этих признаков просто шум? В таких условиях, когда вы можете получить множество данных, не все из них могут быть действительно полезной информацией.

Имейте в виду, что в конечном счете доступные вам признаки ограничиваются следующими двумя соображениями: возможно ли вообще получить информацию и пришло ли вообще вам в голову попытаться ее добыть. Вы можете думать о том, что информация попадает в следующие интервалы.

- *Релевантная и полезная, но ее невозможно получить.* Вы должны иметь в виду, что существует много информации о пользователях, которую вы *не* собираете: сколько свободного времени у них на самом деле? Какие другие приложения они скачали? Они безработные? Они страдают от бессонницы? У них есть склонность к зависимости? Снятся ли им кошмары с драконами? Что-то из этой информации может дать лучший прогноз того, вернутся ли они в следующем месяце или нет. Вы не так много можете сделать с этими сведениями, кроме того, что часть данных, которые вы способны добыть, выполняют функцию посредников, поскольку сильно коррелируют с этими ненаблюдаемыми фрагментами информации: например, если они всегда играют в три часа ночи, то могут страдать бессонницей или работать в ночную смену.
- *Релевантная и полезная, ее можно зарегистрировать, и вы это сделали.* К счастью, вам пришло в голову зарегистрировать информацию во время сеанса мозгового штурма. Это замечательно, однако только то, что вы решили ее зарегистрировать, не значит, что вы знаете, релевантна она или полезна, поэтому хотели бы, чтобы ваш процесс выбора признаков обнаружил это.
- *Релевантная и полезная, ее можно зарегистрировать, но вы этого не сделали.* Возможно, вы не подумали о том, чтобы фиксировать, загрузили ли пользователи фотографию в свой профиль, а это действие значительно прогнозирует их вероятность возврата. Вы человек, поэтому иногда в конечном итоге упускаете действительно важные вещи, но данный факт показывает, что ваше собственное воображение является ограничением в выборе признаков. Один из ключевых способов избежать нехватки полезных признаков — проведение исследований удобства использования (которые будет обсуждать Дэвид Хаффакер позже в главе), чтобы помочь проанализировать взаимодействие с пользователем и те его аспекты, которые вы хотели бы охватить.

- *Нерелевантная и бесполезная, но вы не знали об этом и зарегистрировали ее.* Вот то, что касается выбора признаков, — вы зарегистрировали информацию, но вам она действительно не нужна и вы хотели бы иметь способность знать это.
- *Нерелевантная и бесполезная, но вы не способны ее получить или этого с вами не произошло.* И это хорошо! Она не занимает места и не нужна вам.

Итак, вернемся к логистической регрессии для вашего прогноза удержания в игре. Пусть $c_i = 1$, если i-й пользователь вернется к применению Chasing Dragons в любое время в последующем месяце. Это грубо — вы можете выбрать следующую неделю или последующие два месяца, не имеет значения. Сначала нужно получить рабочую модель, а затем вы можете ее усовершенствовать.

В конечном счете вы хотите, чтобы ваша логистическая регрессия имела вид:

$$\text{logit}(P(c_i = 1 | x_i) = \alpha + \beta^T \cdot x_i.$$

Итак, что нужно делать? Вбросить все сотни признаков, которые вы создали, в одну большую логистическую регрессию? Можно. Это не страшно, но если вы захотите ее расширить или запустить данную модель в производство и получить максимальную прогнозирующую способность, которую вы можете извлечь из данных, то поговорим о том, как вы можете улучшить данный список признаков.

Мы сочли знаменитую статью Изабель Гуйон (Isabelle Guyon), опубликованную в 2003 году под названием *An Introduction to Variable and Feature Selection* («Введение в выбор переменных и признаков»), полезным ресурсом (http://jmlr.org/papers/volume3/guyon03a/guyon03a.pdf). В статье основное внимание уделяется построению и выбору подмножеств признаков, которые *полезны* для создания хорошего показателя. Это контрастирует с проблемой поиска или ранжирования всех потенциально релевантных переменных. В данной работе она изучает три категории методов выбора признаков: фильтры, обертки и встроенные методы. Держите в уме пример прогнозирования для Chasing Dragons во время чтения.

Фильтры

Фильтры упорядочивают возможные признаки в отношении ранжирования на основе метрик или статистики, например корреляции с выходной переменной. Это иногда хорошо на первом проходе по пространству признаков, поскольку они затем учитывают прогнозирующую способность отдельных из них. Однако проблема с фильтрами заключается в том, что вы получаете коррелированные признаки. Другими словами, фильтр не беспокоится об избыточности. И, рассматривая признаки как независимые, вы не учитываете возможные взаимодействия.

Это не всегда плохо и не всегда хорошо, как объясняет Изабель Гуйон. С одной стороны, два избыточных признака могут быть более мощными при совместном использовании; а с другой — что-то кажущееся бесполезным в одиночку может реально помочь в сочетании с другим, вероятно бесполезным признаком.

Вот пример фильтра: для каждого признака запускайте линейную регрессию, используя только этот признак в качестве показателя. Каждый раз обращайте внимание либо на *p*-величину, либо на *R*-квадрат и на порядок рангов в зависимости от самого низкого *p*-значения или самого высокого значения *R*-квадрата (подробнее об этих двух понятиях рассказано в пункте «Критерий выбора» подраздела «Обертки» текущего раздела).

Обертки

Обертка для выбора признаков пытается найти подмножества признаков определенного фиксированного размера, которые будут давать результат. Однако, как знает любой, кто изучил комбинации и перестановки, число возможных размеров k подмножеств n вещей, называемое $\binom{n}{k}$, растет экспоненциально (https://en.wikipedia.org/wiki/Binomial_coefficient#Bounds_and_asymptotic_formulas и https://ru.wikipedia.org/wiki/Биноминальный_коэффициент). Таким образом, есть опасная перспектива переобучения при выполнении обертки.

Для обертки необходимо учитывать два аспекта: 1) выбор алгоритма, используемого для выбора признаков; и 2) принятие решения о выборе критерия или фильтра, показывающего, что ваш набор признаков является «правильным».

Выбор алгоритма

Сначала поговорим о ряде алгоритмов, относящихся к категории *пошаговой регрессии*, методу выбора признаков, который включает в себя выбор признаков в соответствии с неким критерием путем их систематического либо добавления, либо удаления в регрессионной модели.

Существует три основных метода пошаговой регрессии: прямой выбор, обратное исключение и комбинированный подход (прямо и обратно).

- *Прямой выбор*. Вы начинаете с модели регрессии, не содержащей признаков, и постепенно добавляете по одному за раз, в зависимости от того, какой признак сильнее всего улучшает модель в соответствии с критерием выбора. Это выглядит так: постройте все возможные модели регрессии с одним показателем. Выберите лучшую. Теперь попробуйте все возможные модели, которые включают в себя лучший и второй показатели. Выберите лучшую из них. Вы постоянно добавляете по одному признаку и останавливаетесь, когда ваш критерий выбора больше не улучшается, а становится хуже.

- *Обратное исключение*. Вы начинаете с модели регрессии, которая включает в себя *все* признаки и постепенно удаляете их по одному за раз в соответствии с тем, удаление какого признака дает наибольшее улучшение критерия выбора. Вы прекращаете удаление, если это ухудшает критерий.
- *Комбинированный подход*. Большинство методов подмножества улавливают некий дух минимальной избыточности — максимальной релевантности (https://en.wikipedia.org/wiki/Minimum_redundancy_feature_selection). Например, у вас может быть жадный алгоритм, который начинает с лучшего признака, получает несколько более высокую оценку, удаляет худшее и т. д. Это гибридный подход с использованием метода фильтрации.

Критерий выбора

Существует ряд критериев, из которых можно выбирать. В качестве исследователя данных вы должны выбрать, какой критерий выбора использовать. Да! Для выбора критерия выбора вам нужен критерий выбора.

Часть того, что мы хотим вам сообщить, состоит в следующем: на практике, несмотря на теоретические свойства этих различных критериев, выбор, который вы делаете, несколько произволен. Один из способов борьбы с этим — попробовать различные критерии и посмотреть, насколько надежным является ваш выбор модели. Различные критерии выбора могут производить совершенно разные модели, и это часть вашей работы — решать, что оптимизировать и почему:

- *R-квадрат*. С учетом формулы $R^2 = 1 - \frac{\sum_i (y_i - \hat{y}_i)^2}{\sum_i (y_i - \bar{y})^2}$ его можно определить как долю дисперсии, объясняемую вашей моделью.
- *p-значения*. В контексте регрессии, где вы пытаетесь оценить коэффициенты (β_s), чтобы думать в категориях *p*-значений, вы делаете предположение о наличии *нуль-гипотезы* о том, что β_s равны нулю. Для любого заданного β *p*-значение описывает вероятность наблюдения данных, которые вы наблюдали, и получение тестовой статистики (в данном случае оценочное значение $\hat{\beta}$), полученной *по нуль-гипотезе*. В частности, при наличии низкого значения *p* маловероятно, что вы будете наблюдать такую тестовую статистику, если нуль-гипотеза действительно выполняется. Это значит, что (с определенной долей уверенности) коэффициент, скорее всего, будет отличным от нуля.
- *AIC* (Akaike Information Criterion) — информационный критерий Акаике. Определяется по формуле $2k - 2ln(L)$, где k — количество параметров в модели, а $ln(L)$ — «максимизированное значение логарифма вероятности». Цель состоит в минимизации AIC.
- *BIC* (Bayesian Information Criterion) — байесовский информационный критерий. Определяется по формуле $k \cdot ln(n) - 2ln(L)$, где k — количество параметров модели, n — количество наблюдений (элементов данных или пользователей)

и $ln(L)$ — максимизированное значение логарифма вероятности. Цель — минимизация BIC.

- *Энтропия*. Будет рассмотрена подробнее в подразделе «Встроенные методы: деревья решений» текущего раздела.

Практика

Как уже упоминалось, пошаговая регрессия исследует большое пространство всех возможных моделей, и поэтому существует опасность переобучения — она часто будет намного лучше в выборке, чем при новых данных за пределами последней.

Вам не нужно переучивать модели на каждом шаге этих подходов, поскольку существуют прекрасные способы увидеть, как ваша целевая функция (иначе говоря, критерий выбора) меняется по мере изменения поднабора признаков, которые вы проверяете. Они называются «конечными разностями» и, по сути, основываются на разложении в ряд Тейлора целевой функции.

Последнее замечание: если в вашей команде есть эксперт, то не входите в кроличью нору машинного обучения при выборе признаков, прежде чем задействуете своего специалиста полностью!

Встроенные методы: деревья решений

Деревья решений интуитивно привлекательны, поскольку за пределами контекста науки о данных в нашей повседневной жизни мы можем думать о том, чтобы разбить большие решения на ряд вопросов. На рис. 7.3 представлено дерево решений студентки колледжа, стоящей перед очень важным решением: как провести время.

Данное решение на самом деле зависит от множества факторов: есть ли какие-нибудь вечеринки или сроки сдачи, насколько эта студентка ленива и что ей больше всего нравится (вечеринки). Возможность интерпретации — одна из лучших характеристик деревьев решений.

В контексте задачи обработки данных дерево решений является алгоритмом классификации. Возьмем пример Chasing Dragons; вы хотите классифицировать пользователей как «Да, вернется в следующем месяце» или «Нет, не вернется в следующем месяце». На самом деле это не решение в обиходном понимании, пусть это вас не смущает. Вы знаете, что класс любого пользователя зависит от многих факторов (количество драконов, которых убил пользователь, его возраст, сколько часов он уже играл в игру). И вы хотите это классифицировать на основе данных, которые собрали. Но как вы построите деревья решений из данных и какие математические свойства можете ожидать от них?

В конечном счете вам нужно дерево, чем-то напоминающее изображенное на рис. 7.4.

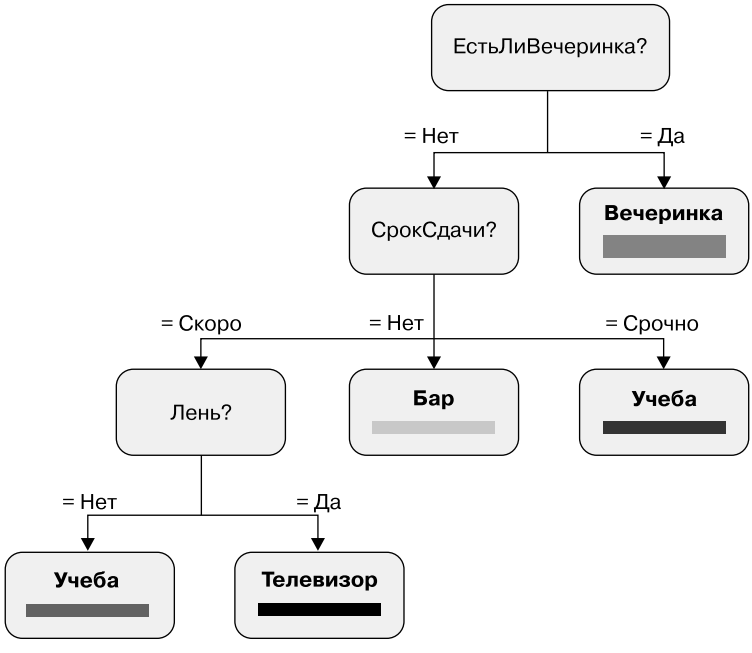


Рис. 7.3. Дерево решений для студентки колледжа, также известное как дерево вечеринки (взято с разрешения Стефена Марсланда (Stephen Marsland) из книги Machine Learning: An Algorithmic Perspective («Машинное обучение: алгоритмическая перспектива») (Chapman and Hall/CRC)

Но вы хотите, чтобы дерево было основано на данных, а не только на том, что вы чувствуете. Выбрать признак на каждом шаге — это как успешно пройти игру 20 Questions («20 вопросов»). Сначала вы берете то, что является *наиболее информативным*. Формализуем это: нам нужно понятие «информативный».

Для этого обсуждения предположим, что разбиваем сложные вопросы на несколько требующих ответа «да» или «нет» и обозначаем ответы 0 или 1. Учитывая случайную величину X, обозначим как $p(X = 1)$ и $p(X = 0)$ вероятность того, что X истинно или ложно соответственно.

Энтропия

С целью количественно определить, что является наиболее «информативным» признаком, мы определяем энтропию — фактически меру того, насколько что-то смешано, — для X следующим образом:

$$H(X) = -p(X = 1)\ log_2(p(X = 1)) - p(X = 0)\ log_2(p(X = 0)).$$

Обратите внимание: когда $p(X = 1) = 0$ или $p(X = 0) = 0$, энтропия исчезает; это согласуется с тем, что:

Рис. 7.4. Дерево решения для Chasing Dragons

$$\lim_{t \to 0} t \cdot \log(t) = 0.$$

В частности, если любой из вариантов имеет нулевую вероятность, энтропия равна 0. Более того, поскольку $p(X = 1) = 1 - p(X = 0)$, энтропия симметрична относительно 0,5 и максимальна в точке 0,5, что легко может быть подтверждено путем несложных вычислений. На рис. 7.5 показана картина данного явления.

Математически мы в каком-то смысле получаем это. Но что оно означает в словесном выражении и почему мы называем его энтропией? Раньше мы говорили, что энтропия — измерение того, насколько что-то смешано.

Например, если параметр X обозначает случай, когда родившийся ребенок оказался мальчиком, то мы ожидаем, что X будет истинным или ложным с вероятностью, близкой к $1/2$; это соответствует высокой энтропии, то есть тому, что младенцы в мешке, из которого выбираем ребенка, сильно смешаны.

Но если X обозначает случай выпадения осадков в пустыне, то это низкая энтропия. Другими словами, содержимое мешка дневных погодных явлений не очень смешивается в пустынях.

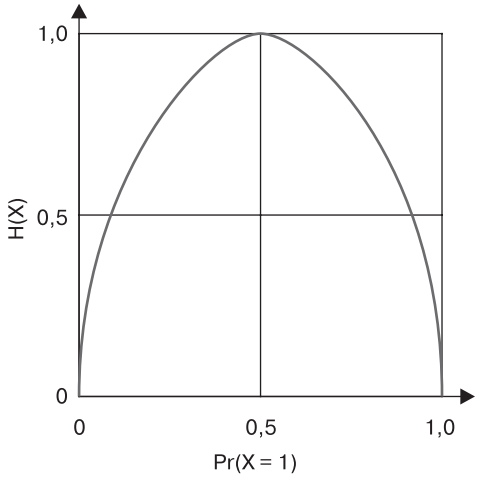


Рис. 7.5. Энтропия

Используя данную концепцию энтропии, мы будем думать об X как о *цели* нашей модели. Таким образом, X может быть событием, когда кто-то покупает что-то на нашем сайте. Нам хотелось бы узнать, какой атрибут пользователя даст больше информации об этом событии X. Мы определим *прирост информации*, обозначенный как $IG(X,a)$, для данного атрибута a как энтропию, которую *теряем*, если знаем значение последнего:

$$IG(X,a) = H(X) - H(X \mid a).$$

Чтобы вычислить его, нужно определить $H(X \mid a)$. Мы можем сделать это в два шага. Для любого действительного значения a_0 атрибута a можем вычислить *определенную условную энтропию* $H(X \mid a) = a_0$, как вы могли ожидать:

$$H(X \mid a = a_0) = -p(X = 1 \mid a = a_0) \cdot log_2(p(X = 1 \mid a = a_0)) - p(X = 0 \mid a = a_0) \cdot \\ \cdot log_2(p(X = 0 \mid a = a_0))$$

и затем суммировать для всех возможных значений a, чтобы получить условную энтропию $H(X \mid a)$:

$$H(X \mid a) = \sum_{a_i} p(a = a_i) \cdot H(X \mid a = a_i).$$

В словесном выражении условная энтропия спрашивает: насколько смешано содержимое нашего мешка, если мы знаем значение атрибута a? И тогда информация может быть описана так: сколько информации мы получаем об X (или о том, сколько энтропии мы теряем), как только нам известно a?

Возвращаясь к тому, как мы используем концепцию энтропии для построения деревьев решений: она помогает решить, какой признак разбить в дереве, или, другими словами, какой наиболее информативный вопрос задать.

Алгоритм дерева решений

Вы строите дерево решений итеративно, начиная с корня. Вам нужен алгоритм для определения того, какой атрибут следует разбить; в частности, какой узел должен быть определен следующим. Вы выбираете этот атрибут, чтобы максимизировать прирост информации, поскольку *получаете максимальную выгоду*. Вы продолжаете идти до тех пор, пока все точки на завершающей стадии не окажутся в одном классе или слева не останется никаких признаков. В таком случае вы принимаете решение большинством голосов.

Часто затем дерево «обрезают», чтобы избежать переобучения. Это означает отрезать его ниже определенной глубины. В конце концов по своему замыслу алгоритм становится слабее и слабее по мере того, как вы строите дерево, и хорошо известно, что дерево, построенное целиком, часто менее точное (для новых данных), чем обрезанное.

Это пример встроенного алгоритма выбора признаков. (Почему встроенного?) В данном случае вам не нужно использовать фильтр, поскольку метод получения информации производит выбор признаков.

Предположим, у вас есть набор данных для Chasing Dragons. Ваша выходная переменная — `Return`: двоичная переменная, которая фиксирует, возвращается ли пользователь в следующем месяце, и у вас есть множество показателей. Вы можете применить библиотеку rpart из R и функцию `rpart`, и код будет выглядеть так:

```
# Дерево классификации с rpart
library(rpart)

# Рост дерева
model1 <- rpart(Return ~ profile + num_dragons + num_friends_invited + gender + age
+ num_days, method="class", data=chasingdragons)

printcp(model1) # отобразить результаты
plotcp(model1)  # визуализировать результаты перекрестной проверки summary(model1)
                # подробная сводка пороговых значений, выбранных для
                # преобразования в двоичные

# Вывести дерево на печать
plot(model1, uniform=TRUE,
main="Дерево классификации для Chasing Dragons") text(model1, use.n=TRUE, all=TRUE,
cex=.8)
```

Обработка непрерывных переменных в деревьях решений

Пакеты, которые реализуют деревья решений, способны обрабатывать непрерывные переменные. Таким образом, вы можете предоставить непрерывные признаки, и они определят оптимальный порог для превращения непрерывной переменной в двоичный показатель. Но если *вы* сами строите алгоритм дерева решений, то в случае непрерывных переменных вам нужно определить правильный порог значения, что-

бы его можно было рассматривать как двоичную переменную. Таким образом, вы можете разделить количество убийств драконов пользователем на «менее 10» и «не менее 10», и вернетесь к варианту с двоичной переменной. В этом случае требуется дополнительная работа, чтобы принять решение о приросте информации, поскольку он зависит как от порогового значения, так и от признака.

Фактически вы могли бы подумать о решении, где порог должен существовать как отдельная подмодель. Можно оптимизировать данный выбор, максимизируя энтропию по отдельным атрибутам, но это, конечно, не лучший способ работы с непрерывными переменными. В самом деле, такой вопрос способен быть столь же сложным, как и сам выбор признаков, — вместо одного порога вы можете пожелать создать, например, подборку значений вашего атрибута. Что делать? Это всегда будет зависеть от ситуации.

ВЫЖИВАНИЕ НА «ТИТАНИКЕ»

Ради развлечения Уилл показал нам это дерево решений для выживания на «Титанике» с сайта BigML (https://bigml.com/user/rachel/gallery/model/5182d391ce5680547f0000f4). Начальные данные взяты из исходного кода Encyclopedia Titanica (https://www.encyclopedia-titanica.org/), и они доступны там. На рис. 7.6 представлен только его снимок, но если вы перейдете на сайт, то оно будет интерактивным.

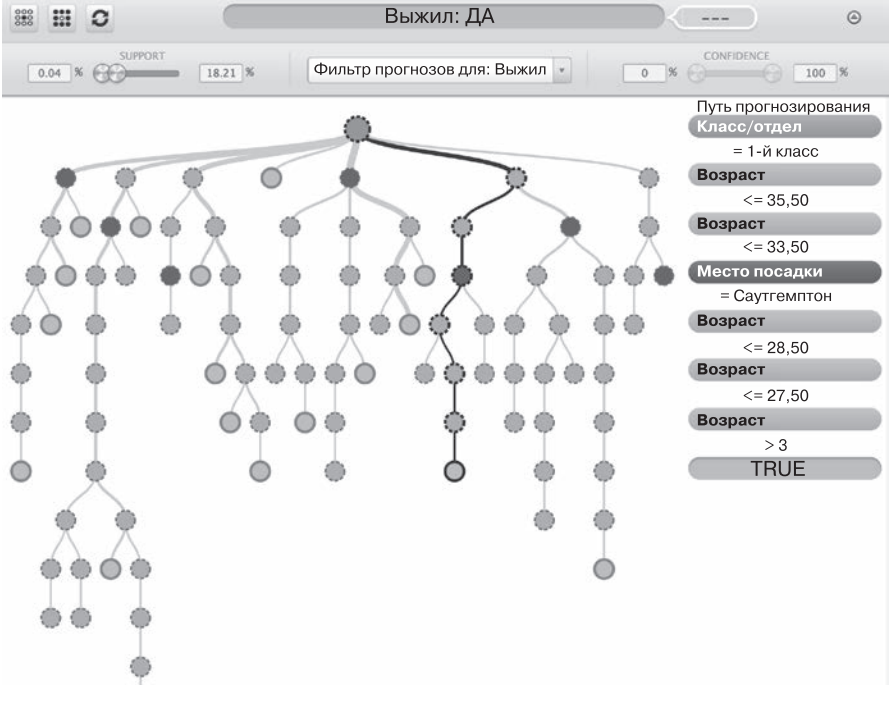


Рис. 7.6. Выживание на «Титанике»

Случайные леса

Перейдем к другому алгоритму выбора признаков. Случайные леса обобщают деревья принятия решений с бэггингом (bagging, сокращенно от *bootstrap aggregating*) (https://en.wikipedia.org/wiki/Bootstrap_aggregating). Ниже мы расскажем о бэггинге более подробно, но эффект от его использования заключается в том, чтобы сделать ваши модели более точными и надежными, но за счет интерпретируемости — случайные леса, как известно, сложны для понимания. Наоборот, их легко определить двумя гиперпараметрами: вам просто нужно указать количество деревьев, которые вы хотите разместить в вашем лесу, допустим *N*, а также количество признаков, случайным образом выбираемых для каждого дерева, например *F*.

Прежде чем мы войдем в заросли алгоритма случайного леса, рассмотрим бутстрэппинг (bootstrapping). *Выборка методом бутстрэпа* (bootstrap sample) — выборка с возвращением; это значит, что мы могли бы выбирать одну и ту же точку данных более одного раза. Обычно размер выборки составляет 80 % от размера всего набора данных (обучающего), но, конечно, этот параметр можно скорректировать в зависимости от обстоятельств. Фактически это третий гиперпараметр нашего алгоритма случайного леса.

Теперь перейдем к алгоритму. Чтобы построить случайный лес, вы строите *N* деревьев решений следующим образом.

1. Для каждого дерева берете выборку методом бутстрэпа из ваших данных и для каждого узла произвольно выбираете F признаков, например 5 из 100 общих.
2. Затем используете свой механизм определения прироста информации на основе энтропии, описанный в предыдущем подразделе, чтобы решить, какой из этих признаков разделите в дереве на каждом этапе.

Обратите внимание: вы могли заранее решить, насколько глубоким должно получиться дерево, или можете обрезать деревья по факту, но деревья обычно *не* обрезают в случайных лесах, поскольку замечательная особенность последних заключается в том, что они способны включать в себя характерный шум.

Код будет выглядеть следующим образом.

```
# Автор: Джаред Ландер (Jared Lander)
#
# мы будем использовать данные для бриллиантов из ggplot
require(ggptot2)

# загрузить и просмотреть данные для бриллиантов
data(diamonds)
head(diamonds)
# построить гистограмму с линией, отмечающей 12 000 долларов
ggplot(diamonds) + geom_histogram(aes(x=price)) + geom_vline(xintercept=12000)

# создать переменную, принимающую значения TRUE/FALSE, которая
```

```
# показывает то, что цена выше нашего порогового значения
diamonds$Expensive <- ifelse(diamonds$price >= 12000, 1, 0) head(diamonds)

# вернуть идентификатор строки для столбца цен
diamonds$price <- NULL

## glmnet
require(glmnet)
# построить матрицу прогноза, мы пропускаем последний столбец,
# который является нашим ответом
x <- model.matrix(~., diamonds[, -ncol(diamonds)])
# построить вектор ответов
y <- as.matrix(diamonds$Expensive)
# запустить glmnet
system.time(modGlmnet <- glmnet(x=x, y=y, family="binomial"))
# вывести траекторию коэффициента
plot(modGlmnet, label=TRUE)

# это показывает, что установка начального значения позволяет вам
# воссоздавать случайные результаты, запускать их по нескольку раз
set.seed(48872)
sample(1:10)

## дерево решений
require(rpart)
# выполнить простое дерево решений
modTree <- rpart(Expensive ~ data=diamonds)
# вывести разделения
plot(modTree)
text(modTree)

## бэггинг (bagging или bootstrap aggregating)
require(boot)
mean(diamonds$carat)
sd(diamonds$carat)
# функция для бутстрэппинга mean
boot.mean <- function(x, i)
{
    mean(x[i])
}
# позволяет найти варьируемость mean
boot(data=diamonds$carat, statistic=boot.mean, R=120)
require(adabag)
modBag <- bagging(formula=Species ~ ., iris, mfinal=10)

## повышение
require(mboost)
system.time(modglmBoost <- glmboost(as.factor(Expensive) ~ .,
        data=diamonds, family=Binomial(link="logit"))) summary(modglmBoost)
?blackboost

## случайные леса
require(randomForest)
system.time(modForest <- randomForest(Species ~ ., data=iris,
        importance=TRUE, proximity=TRUE))
```

КРИТИКА ВЫБОРА ПРИЗНАКОВ

Рассмотрим общую критику выбора признаков. А именно, это не лучше, чем драгирование данных (data dredging). Если мы просто возьмем любой полученный ответ, который коррелирует с нашей целью, насколько бы нерелевантным он ни был, то могли бы подумать, что производство масла в Бангладеш предсказывает рейтинг S&P (Standard & Poor's) (http://nerdsonwallstreet.typepad.com/my_weblog/files/dataminejune_2000.pdf). Как правило, мы хотели бы сначала изучить потенциально подходящие признаки, по крайней мере до некоторой степени. Конечно, чем больше наблюдений мы производим, тем меньше нужно заниматься помехами.

Существует хорошо известный компромисс между систематической ошибкой и дисперсией (bias-variance tradeoff) (https://en.wikipedia.org/wiki/Supervised_learning#Bias-variance_tradeoff): модель имеет «большую систематическую ошибку», если слишком проста (признаки не выделяют достаточную информацию). В этом случае намного большее количество данных не улучшает модель. С другой стороны, если модель слишком сложна, то «высокая дисперсия» приводит к переобучению. В таком случае мы хотим уменьшить количество используемых признаков.

Удержание пользователей: интерпретируемость и прогнозирующая способность

Итак, допустим, вы построили дерево решений и оно довольно хорошо делает прогнозы. Но следует ли его интерпретировать? Можете ли вы попытаться найти в нем смысл?

Возможно, в основном дерево говорит вам: «Чем больше пользователь играет в первом месяце, тем больше вероятность его возвращения в следующем», что бесполезно, и это происходит, когда вы делаете анализ. Кажется цикличным, *конечно*: чем больше пользователям нравится приложение сейчас, тем больше вероятность, что они вернутся. Но может быть и так, что дерево сообщает вам, что показ объявлений пользователям в первые пять минут *уменьшает* их шансы на возвращение. В то же время нормально показывать рекламу после первого часа и не имеет смысла делать это в первые пять минут! Теперь, чтобы узнать больше, понадобится провести своего рода A/B-тестирование (см. главу 11), но эта первоначальная модель и выбор признаков помогут определить приоритеты типов тестов, которые вы, возможно, захотите выполнить.

Кроме того, стоит отметить, что признаки, связанные с поведением пользователя (пользователь играл десять раз в этом месяце), качественно отличаются от тех, которые имеют отношение к вашему поведению (вы показали десять объявлений и изменили цвет дракона на красный вместо зеленого). Здесь имеет место проблема причинности/корреляции. Означает ли корреляция между получением большого количества очков в первый месяц и возвратом к игре в следующем,

что если вы просто *дадите* пользователям большое количество очков в текущем месяце, даже когда они вообще не играют, то они вернутся? Нет! Не количество очков побуждает их вернуться, а то, что они действительно играют (искажающий фактор); это коррелирует с их возвращением, *а также* с получением большого количества очков. Как следствие, вы будете производить выбор признаков для *всех* переменных, но затем сосредоточьтесь на тех, с которыми можете что-то сделать (например, показывать меньше объявлений) в зависимости от атрибутов пользователя.

Дэвид Хаффакер: гибридный подход к проведению социологических исследований Google

Дэвид сосредоточен на эффективных сочетаниях как качественных, так и количественных исследований, а также больших и малых данных. Крупные массивы больших количественных данных можно извлечь более эффективно, если вы вначале потратите время на тщательное осмысление в меньшем масштабе, а затем используете то, что изучили, в более широком. И наоборот, вы можете найти закономерности в большом наборе данных, которые хотите изучить, все больше углубляясь, проводя интенсивные исследования удобства использования на небольшой группе людей, добавляя больше красок наблюдаемому явлению или проверяя взаимосвязи, согласовывая ваш разведочный анализ большого набора данных с соответствующей научной литературой.

Дэвид был одним из коллег Рэйчел в Google. Они успешно сотрудничали — начиная с наборов взаимодополняющих навыков, всплеска ценных качеств, проявившихся при совместной работе над Google+ (социальной сетью Google), вместе с другими людьми, а именно с инженерами-программистами и специалистами в области компьютерных наук. Дэвид привносит точку зрения социологов в анализ социальных сетей. Он хорошо владеет количественными методами понимания и анализа социального поведения в сети. Он получил степень доктора философии в области средств массовой информации, технологий и общества в Северо-Западном университете (Northwestern University). Дэвид рассказал о подходе Google к социальным исследованиям, чтобы побудить класс к образу мыслей, который сочетает качественное и количественное, а также мелко- и крупномасштабное.

Google делает хорошую работу по объединению людей. Они размывают границы между исследованиями и разработками. Они даже писали об этом в статье за июль 2012 года Google's Hybrid Approach to Research («Гибридный подход Google к исследованиям») (https://cacm.acm.org/magazines/2012/7/151226-googles-hybrid-approach-to-research/fulltext). Их исследователи внедрены в группы разработчиков программного обеспечения. Работа носит итеративный характер, и инженеры в команде стремятся

получить код, близкий к производственному, с первого дня проекта. Они используют техническую инфраструктуру для организации экспериментов в своей массивной базе пользователей и быстрого развертывания масштабированного прототипа. Учитывая масштабы пользовательской базы Google, редизайн по мере расширения не является приемлемым вариантом. Вместо этого они проводят эксперименты с небольшими группами пользователей.

Переход от описаний к прогнозам

Дэвид предположил, что в качестве исследователей данных мы рассмотрим, как приступить к экспериментальному проектированию с тем, чтобы перейти к причинным требованиям между переменными, а не к описательной взаимосвязи. Другими словами, наша цель — перейти от описания к прогнозам.

В качестве примера он рассказал о возникновении «круга друзей», функциональной возможности Google+. Google знает, что люди хотят делиться избирательно; пользователи могут отправлять фотографии своей семье, тогда как, скорее всего, с большей долей вероятности отправят шуточные послания своим друзьям. Google придумал идею кругов, но было неясно, будут ли люди применять их. Как Google может ответить на вопрос: будут ли люди задействовать круги для организации своей социальной сети? Важно знать, что является мотивом для пользователей, когда они решают поделиться.

В Google применили *смешанный подход*, а это значит, что они использовали несколько методов для того, чтобы рассмотреть под разными углами экспериментальные данные и результаты анализов. Некоторые из их методов были небольшими и качественными, другие — более крупными и количественными.

Взяв случайную выборку из 100 000 пользователей, они задались целью определить популярные имена и категории имен, присвоенных кругам. Они выявили 168 активных пользователей, которые заполнили опросы, и провели более длительные интервью с 12 опрошенными. Степень охвата этих интервью была сопоставлена с систематической ошибкой выборки, сопряженной с поиском людей, готовых пройти собеседование.

Они обнаружили, что большинство людей участвуют в выборочном обмене, используют круги, и имена последних чаще всего связаны с работой или учебой, и у них есть элементы сильной ссылки (epic bros — «эпические братья») или слабой ссылки (acquaintances from PTA — «знакомства в группе защиты учителей и родителей»).

Они спросили участников опроса, почему те обмениваются контентом. Ответы в основном касались трех категорий. Во-первых, желание поделиться информацией о себе: личный опыт, мнения и т. д. Во-вторых, обмен мнениями: люди хотят участвовать в разговоре. В-третьих, пропаганда: людям нравится распространять информацию.

Затем они спросили участников, почему те выбирают свою аудиторию. И снова выявились три категории: во-первых, конфиденциальность — многие люди были публичными или скрытными по определению. Во-вторых, тематическое соответствие: люди хотели поделиться только с теми, кто может быть заинтересован, и не хотели загрязнять поток данных других. В-третьих, распространение: некоторые люди просто хотят максимально расширить свою потенциальную аудиторию.

В результате этого исследования выяснилось, что людям нравится выборочно делиться контентом, в зависимости от контекста и аудитории. Так что компании Google пришлось разработать для своего продукта возможности, связанные с учетом контента, контекста и аудитории. Google хотела бы учитывать полученные результаты не только при проектировании, но и при выполнении масштабного исследования данных.

МЫСЛЕННЫЙ ЭКСПЕРИМЕНТ: АНАЛИЗ КРУПНОМАСШТАБНЫХ СЕТЕЙ

Мы углубимся в сетевой анализ в главе 10 с Джоном Келли (John Kelly). Но пока подумайте о возможностях применения результатов исследований удобства использования Google+ и изучения выборочного обмена контентом при огромном масштабе используемых данных. Вы можете задействовать большие данные и рассматривать связи между акторами как граф. Для Google+ пользователи — это узлы, а ребра (направленные) находятся «в одном круге». Подумайте над тем, какие данные хотите зафиксировать, а затем над тем, как проверить некоторые гипотезы, возникшие при разговоре с небольшой группой вовлеченных пользователей.

Как исследователю данных, вам может быть полезно подумать о разных структурах и представлениях данных, и как только вы начнете мыслить в категориях сетей, сможете увидеть их повсюду.

Другие примеры сетей представлены ниже.

- Узлы являются пользователями в Second Life («Вторая жизнь») (https://secondlife.com/), а ребра соответствуют взаимодействиям между ними. Обратите внимание: игроки могут взаимодействовать в этой игре более чем одним способом, что приводит к потенциально различным типам ребер.
- Узлы — сайты, направленные ребра — ссылки.
- Узлами являются теоремы, а направленными ребрами — взаимозависимости (http://www.math.columbia.edu/~dejong/plaatje.png).

Социальность в Google

Как вы могли заметить, «социальность» — слой для всего Google. Поиск теперь включает данный слой: если вы ищете что-то, то можете увидеть, что ваш друг это опубликовал. Это называется *социальным комментированием*. Оказывается, люди

внимательнее относятся к комментариям, исходящим от какого-либо эксперта в предметной области, чем от тех, с кем они очень близки. Таким образом, когда дело доходит до покупки вина, для вас может быть более интересным узнать мнение специалиста по винам, чем вашей мамы.

Обратите внимание: это звучит как очевидное, но если вы начали, наоборот, с расспросов тех, кому доверяете, то можете начать с вашей мамы. Другими словами, «тесные связи» — даже если вы можете определить их, — не самая лучшая характеристика для ранжирования комментариев. Но напрашивается вопрос: что тогда? Как правило, в подобной ситуации исследователи данных могут использовать скорость нажатий или то, как долго приходится нажимать.

В общем, вам нужно всегда учитывать количественную метрику успеха. Она определяет вашу успешность, так что вы должны быть внимательны.

Конфиденциальность

Перед ориентированной на человека технологией стоит острая проблема конфиденциальности, что затрудняет работу. Google провела опрос о том, насколько люди беспокоятся по поводу контента. Они спросили: «Как это влияет на ваше участие? Какова природа вашей конфиденциальности?»

Оказывается, существует сильная корреляция между проблемой конфиденциальности и низким уровнем участия, что неудивительно. Это также связано с тем, насколько хорошо вы понимаете, какой информацией делитесь, и с таким вопросом: когда вы что-то публикуете, куда оно идет, и насколько вы можете его контролировать? Сталкиваясь с огромным количеством сложных настроек, вы, как правило, начинаете терять интерес.

Результаты опроса выявили следующие вызывающие беспокойство категории:

- *хищение конфиденциальных данных* — денежные потери;
- *компьютерный мир:*
  - доступ к персональным данным;
  - я находил действительно конфиденциальную информацию;
  - нежелательный спам;
  - провокационные фотографии (о боже, мой босс это видел);
  - нежелательные вымогательства;
  - нежелательная рассылка рекламных объявлений;
- *физический мир:*
  - преследования/домогательства вне сети;
  - вред для моей семьи;

- упорные преследователи;
- риски, связанные с работой.

Мысленный эксперимент: что является наилучшим способом снизить беспокойство и повысить понимание и контроль?

Таким образом, исходя из понятной заботы пользователей о конфиденциальности, студенты в классе Рэйчел провели мозговой штурм отдельных потенциальных решений, которые могла бы реализовать Google (или мог бы рассматривать любой, кто имеет дело с данными пользовательского уровня).

Возможности:

- ❑ написать и опубликовать манифест вашей политики данных. Google опробовала это, но, оказывается, никто не любит читать манифесты;
- ❑ обучать пользователей вашим политикам на манер фишки Netflix: «поскольку вам понравилось вот это, мы думаем, вам может понравиться и вот это». Но не всегда так легко объяснять вещи в сложных моделях;
- ❑ просто избавиться от всех сохраненных данных через год. Но вам все равно нужно объяснить, что вы делаете.

Вероятно, мы могли бы перефразировать вопрос: как вы разрабатываете настройки конфиденциальности, чтобы облегчить их для людей? В частности, как вы делаете процесс прозрачным? Вот несколько идей по данному направлению:

- ❑ создать рисунок или граф, показывающие, куда идут данные;
- ❑ позволить людям управлять конфиденциальностью;
- ❑ предоставить доступ к быстрым настройкам;
- ❑ создать настройки, которые вы показываете людям, разделенным на категории «где у вас нет права выбора» и «где вы можете выбирать», для ясности;
- ❑ лучшее, что вы могли бы сделать, — создать разумные настройки по умолчанию, чтобы людям не пришлось об этом беспокоиться.

Дэвид сказал нам мудрую вещь: по мере продвижения вперед и доступа к большим данным вы действительно должны дополнять их качественными подходами. Используйте смешанные методы, чтобы лучше понять картину происходящего. Качественные опросы действительно могут помочь.

8 Рекомендательные механизмы: создание ориентированных на пользователя масштабируемых информационных продуктов

Рекомендательные механизмы, также называемые рекомендательными системами, являются квинтэссенцией информационных продуктов и хорошей отправной точкой для объяснения ученым, не использующим данные, чем вы занимаетесь или что такое наука о данных. Это связано с тем, что многие люди взаимодействовали с рекомендательными системами, когда им предлагали книги на Amazon.com, или получили рекомендованные фильмы в Netflix. Однако они вряд ли задумывались о проектировании и алгоритмах, лежащих в основе этих рекомендаций, а также о том, что их поведение при покупке книги или оценке фильма генерирует данные, которые затем возвращаются в рекомендательный механизм и приводят (как хотелось бы надеяться) к улучшению рекомендаций для них самих и для других людей.

Не считая того, что рекомендательные системы являются явным примером продукта, который буквально использует данные в качестве своего топлива, еще одна причина, по которой мы называем эти системы квинтэссенцией, заключается в том, что для создания надежной рекомендательной системы на всех стадиях требуется понимание линейной алгебры и способность писать код. Рекомендательные системы также иллюстрируют проблемы, возникающие с большими данными при решении интуитивно понятной задачи, усложняющейся при реализации решения для случая больших масштабов данных.

В данной главе Мэтт Гэттис (Matt Gattis) рассказывает о том, что потребовалось ему для создания рекомендательной системы для Hunch.com, включая то, почему он принял определенные решения и как он делал выбор между различными алгоритмами при построении крупномасштабной технической системы и инфраструктуры, которые обеспечивают поддержку ориентированного на пользователя продукта.

Мэтт закончил факультет информатики Массачусетского технического университета, работал в SiteAdvisor и является одним из основателей Hunch, выступая в качестве его технического директора. Hunch — это сайт, который дает рекомендации любого рода. Сначала они работали так: задавали людям множество вопросов (люди, похоже, любят отвечать на них), а затем кто-то может задать механизму такие вопросы, как «Какой мобильный телефон стоит купить?», «Куда мне отправиться в путешествие?», и тот даст им совет. Сотрудники Hunch используют машинное обучение, чтобы давать все лучшие и лучшие советы. Роль Мэтта заключалась в проведении научных исследований и разработке базового рекомендательного механизма.

В первую очередь они сосредоточились на том, чтобы сделать вопросы максимально интересными. Затем, конечно же, увидели вещи, о которых нужно спросить, так как те чрезвычайно информативны, вследствие чего были добавлены. Затем сотрудники обнаружили, что могут задать всего 20 вопросов, а затем предсказать остальные из них с точностью до 80 %. Были вопросы, которые вы могли бы себе представить, и вопросы, вызывающие удивление, например: были ли люди склонными к соревновательности, интроверсии или экстраверсии, рациональности или действиям по обстоятельствам и т. д. — очень похоже на индикатор типов Майерс — Бриггс (https://en.wikipedia.org/wiki/Myers–Briggs_Type_Indicator и https://ru.wikipedia.org/wiki/Типология_Майерс_—_Бриггс).

Постепенно Hunch от опросов мнений людей перешел к модели API (https://en.wikipedia.org/wiki/Application_programming_interface) для сканирования данных с сайтов. Этот сервис могут также использовать сторонние организации для персонализации контента каких-либо сайтов — выгодное коммерческое предложение, в результате которого Hunch.com был приобретен компанией eBay.

Мэтт создавал код с детства, поэтому считает, что разработка программного обеспечения является его сильной стороной. Hunch требует многопрофильного опыта, так вот Мэтт не считает себя экспертом в какой-либо узкоспециализированной области, за исключением самих рекомендательных систем.

Самая лучшая цитата Мэтта: «Формирование команды по обработке данных — что-то вроде планирования ограбления». Это значит, что вам нужны люди, имеющие всевозможные навыки, и один человек, вероятно, не сможет сделать все сам. (Думаю, как в фильме «Одиннадцать друзей Оушена» (https://www.imdb.com/title/tt0240772/), но еще лучше.)

Реальный рекомендательный механизм

Рекомендательные механизмы используются все время: какой фильм вы захотите посмотреть, зная другие фильмы, которые вам понравились? Какую книгу вы хотели бы, исходя из прошлых покупок? Как вы, скорее всего, проведете отпуск, учитывая прошлые поездки?

Существует множество различных способов построения такой модели, но они кажутся очень похожими без реализации. В этой главе мы расскажем, как создать одну относительно простую, но полную версию.

Чтобы настроить рекомендательный механизм, предположим: у вас есть *пользователи*, образующие набор *U*, и *элементы* для рекомендаций, составляющие набор *V*. Как сказал Кайл Тиг в главе 6, вы можете показать это как двудольный граф (рис. 8.1), если каждый пользователь и каждый элемент имеют узел для его представления — есть линия от пользователя к элементу, при условии, что данный пользователь высказал мнение о данном элементе. Обратите внимание: этот элемент может не всегда *нравиться* пользователям, вследствие чего ребра могут иметь вес: быть положительными, отрицательными или иметь значение по непрерывной шкале (или дискретной, но с множеством значений, как звездная система). Последствия данного выбора могут быть тяжелыми, но здесь мы не будем слишком углубляться в них — для нас это численные показатели.

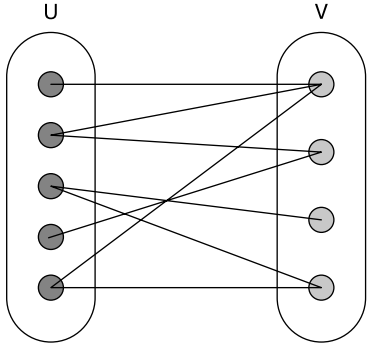


Рис. 8.1. Двудольный граф с пользователями и элементами (телевизионные шоу) в качестве узлов

Далее, у вас есть обучающие данные в виде ряда предпочтений — вы знаете некое мнение тех или иных пользователей относительно некоторых элементов. По этим данным вы хотите предсказать другие предпочтения своих пользователей. Все описанное, по сути, является выводом для механизма рекомендаций.

У вас также могут быть метаданные о пользователях (то есть мужчины они или женщины и т. д.) или объектах (цвет продукта). Например, пользователи приходят

на ваш сайт и настраивают учетные записи, поэтому вы можете знать о каждом из них как минимум пол, возраст и предпочтения.

Вы представляете данного пользователя как вектор признаков, иногда включающий только метаданные или предпочтения (что приведет к разреженному вектору (https://en.wikipedia.org/wiki/Sparse_vector/), поскольку не знаете мнение пользователя по поводу всего), а иногда и то и другое, в зависимости от действий, которые вы совершаете с вектором. Кроме того, вы иногда можете связать все векторы пользователя, чтобы получить большую матрицу пользователя, которую мы называем U, злоупотребляя нотацией (https://en.wikipedia.org/wiki/Abuse_of_notation/).

Обзор метода k-ближайших соседей

Рассмотрим метод k-ближайших соседей (обсуждавшийся в главе 3): если вы хотите предсказать, понравится ли пользователю А какой-то предмет, то смотрите на пользователя Б, *ближайшего* к пользователю А, у которого есть мнение; тогда вы предполагаете, что мнение А совпадает с мнением Б. Другими словами, как только идентифицируете аналогичного пользователя, вы найдете то, что пользователь А не оценил (что, как вы предполагаете, значило следующее: он никогда не видел этот фильм или не покупал этот предмет), но оценил пользователь Б и ему понравилось, и используете это в качестве рекомендации для пользователя А.

Как было сказано в главе 3, для реализации всего вышеописанного вам потребуется метрика, чтобы вы могли измерять расстояние. Пример двойственности мнения: расстояние Жаккара (https://en.wikipedia.org/wiki/Jaccard_index и https://ru.wikipedia.org/wiki/Коэффициент_Жаккара), то есть 1 – количество вещей, которые нравятся им обоим, делится на количество вещей, которые нравятся одному из них. Другие примеры включают косинусный коэффициент (https://en.wikipedia.org/wiki/Cosine_similarity) или евклидово расстояние (https://en.wikipedia.org/wiki/Euclidean_distance и https://ru.wikipedia.org/wiki/Евклидово_пространство).

Какая метрика лучше?

Вы можете получить разные ответы в зависимости от выбранной метрики. Но это хорошо. Попробуйте множество различных функций для определения расстояния, посмотрите, как ваши результаты меняются, и подумайте почему.

Некоторые проблемы, связанные с методом k-БС

Таким образом, вы можете использовать ближайших соседей; интуитивно понятно, что вы хотите рекомендовать предметы людям, найдя похожих людей и используя их мнение, чтобы генерировать идеи и рекомендации. Но есть ряд проблем, которые порождает ближайший сосед. Рассмотрим их.

- *Проклятие размерности*. Чересчур много измерений, поэтому ближайшие соседи слишком далеки друг от друга, чтобы в действительности считаться близкими.
- *Чрезмерная аппроксимация*. Это еще одна проблема. Например, одна точка может располагаться ближе всех, но может быть белым шумом. Как учесть подобную ситуацию? Одна из идей заключается в использовании метода k-БС, скажем, с $k = 5$, а не с $k = 1$, увеличивающим шум.
- *Взаимосвязанные признаки*. Кроме того, есть множество признаков, которые тесно связаны друг с другом. Например, вы можете представить, что по мере взросления становитесь более консервативными. Но тогда вычисление и возраста, и политических убеждений означало бы, что вы в некотором смысле дважды считаете один и тот же признак. Это повлечет ухудшение эффективности, поскольку вы используете избыточную информацию и, по существу, дважды устанавливаете вес для некоторых переменных. Предпочтительнее производить построение с пониманием взаимосвязанности и создавать проект в пространстве меньшей размерности.
- *Относительная важность признаков*. Одни признаки информативнее других. Поэтому определение веса признаков может быть полезным: вероятно, ваш возраст не имеет ничего общего с вашим предпочтением в отношении элемента номер 1. Вы, возможно, будете использовать что-то вроде смешанного второго момента, чтобы выбрать свои веса.
- *Разреженность*. Если ваш вектор (или матрица, при условии, что вы собрали векторы) слишком разрежен или у вас много недостающих данных, то большинство вещей неизвестно, а расстояние Жаккара не имеет значения, поскольку нет перекрытия.
- *Ошибки измерений*. Существует ошибка измерения (также называемая ошибкой отчетности): люди могут врать.
- *Сложность вычислений*. Есть нормативные издержки — сложность вычислений.
- *Чувствительность метрик расстояния*. Евклидово расстояние также имеет проблему масштабирования: расстояния для возраста перевешивают расстояния для других признаков, если сообщаются как 0 (для «не нравится») или 1 (для «нравится»). По существу, это значит, что евклидово расстояние в сыром виде не имеет большого смысла. Кроме того, старые и молодые люди могут думать одно, а люди среднего возраста — что-то другое. Кажется, мы предполагаем линейную связь, но ее может и не быть. Следует ли вам, к примеру, производить сортировку по возрастным группам?
- *Предпочтения меняются с течением времени*. Предпочтения пользователей также могут меняться со временем, что выходит за рамки модели. Например, они могли купить на eBay принтер и теперь на короткое время их интересуют только чернила.
- *Стоимость обновлений*. Кроме того, обновлять модель дорого, поскольку вы добавляете больше данных.

Самые большие проблемы — первые две в списке, а именно выбросы и проклятие размерности. Как с ними бороться? Вернемся к методу, с которым вы уже знакомы, — линейной регрессии — и начнем построение оттуда.

За рамками метода k-БС: классификация машинного обучения

Сначала мы рассмотрим упрощенный алгоритм машинного обучения, а именно построим отдельную модель линейной регрессии для каждого элемента. С помощью каждой модели мы могли бы затем предсказать для данного пользователя, зная его атрибуты, понравится ли ему элемент, соответствующий этой модели. Таким образом, одна модель может предсказывать, придется ли вам по вкусу сериал «Безумцы» (Mad Men), а другая предназначена для предсказания того, понравится ли вам Боб Дилан.

Обозначим через $f_{i,j}$ то, что пользователь указал свое предпочтение для j-го элемента, при условии его наличия (или пользователь является атрибутом, если j-й элемент есть элемент метаданных, например возраст или is_logged_in (зарегистрирован)). Это тонкий вопрос, можно слегка запутаться, если не усвоить следующее: здесь вы обрабатываете метаданные так, словно это были элементы. Мы упомянули об этом раньше, но ничего страшного, если вы не поняли, надеемся, сейчас данная информация стала доступнее. Сказав, будто способны предсказать, что вам может понравиться, мы также говорим, что можем использовать это с целью предсказания ваших атрибутов. То есть, если мы не знали, мужчина вы или женщина, поскольку этих данных у нас не было или мы никогда не спрашивали вас об этом, мы могли бы это предсказать.

Чтобы позволить данной идее закрепиться еще больше, предположим, будто у нас есть три числовых атрибута для каждого пользователя, поэтому мы имеем $f_{i,1}$, $f_{i,2}$ и $f_{i,3}$. Затем с целью угадать, что пользователь предпочитает новый элемент (временно обозначим эту оценку как p_i), мы можем искать лучший выбор для β_k таким образом:

$$p_i = \beta_1 f_{i,1} + \beta_2 f_{i,2} + \beta_3 f_{i,3} + c.$$

Хорошая новость: вы знаете, как вычислить коэффициенты с помощью линейной алгебры, оптимизации и статистического анализа, в частности линейной регрессии.

Плохая новость: эта модель работает только для одного элемента, и для того чтобы она была полной, вам нужно построить столько моделей, сколько есть элементов. Более того, вы вообще не используете информацию о других элементах с целью создания модели для данного элемента, поэтому не привлекаете другие части информации.

Но подождите, есть еще одна хорошая новость: модель решает проблему «взвешивания признаков», которую мы обсуждали ранее, так как коэффициенты линейной регрессии *являются* весами.

Однако выбросы *по-прежнему* являются проблемой, и она приходит в виде огромных коэффициентов, когда у вас недостаточно данных (то есть недостаточно данных по этим элементам).

Сделаем более строгим предыдущее утверждение о том, что огромные коэффициенты подразумевают присутствие выбросов или, возможно, даже плохую модель. Например, если две из ваших переменных одинаковы или почти одинаковы, то коэффициент для одной может составлять 100 000, а для другой может быть –100 000, и на самом деле они ничего не добавляют к модели. В общем, вы всегда должны иметь некоторые предварительные сведения о том, каким будет разумный размер для ваших коэффициентов, что вы делаете, нормализуя все ваши переменные и представляя, что «важный» эффект станет переводиться с точки зрения размера коэффициентов, — все, что значительно больше данной величины (ее абсолютного значения), будет подозрительным.

Для решения этой проблемы выбросов вы налагаете байесовское априорное распределение вероятностей таким образом, что данные веса не должны располагаться слишком далеко от правильного значения, — это выполняется за счет добавления штрафного члена для больших коэффициентов. Фактически это соответствует добавлению априорной матрицы к ковариационной (https://mathbabe.org/2013/02/24/the-overburdened-prior/). Данное решение зависит от одного параметра, который традиционно называется λ.

Но вышесказанное вызывает вопрос: как вы выбираете λ? Вы можете сделать это экспериментально: используйте некоторые данные в качестве обучающего набора, оцените, насколько хороший результат получили при определенных значениях λ, и сделайте поправку. Так происходит в реальной жизни, хотя обратите внимание на то, что описанное не совсем соответствует идее оценки того, какой размер будет разумным для вашего коэффициента.

Вы не можете использовать данный штрафной термин для больших коэффициентов и предполагать, что проблема «взвешивания признаков» все еще решена, так как на самом деле вы станете штрафовать некоторые коэффициенты намного больше, чем другие, если их начальные значения будут иметь разный масштаб. Самый простой способ обойти это обстоятельство — нормализовать переменные перед введением их в модель, подобно тому, как мы делали в главе 6. Если у вас есть причина полагать, что некоторые переменные должны иметь большие коэффициенты, то вы можете нормализовать разные переменные с помощью различных средних значений и отклонений. В конечном счете ваш способ нормализации снова эквивалентен наложению априорного распределения вероятностей.

Последняя проблема с априорным распределением: хотя задача будет иметь уникальное решение (как если бы штрафное значение было единственным

минимумом), сделав λ достаточно большим, к тому времени вы не сможете решить стоящую перед вами задачу. Подумайте об этом: если вы сделаете значение λ абсолютно огромным, то коэффициенты станут равны нулю и у вас не будет модели вообще.

Проблема размерности

Итак, мы взялись за решение проблемы выбросов, поэтому теперь подумаем о сверхразмерности, то есть о том, что у вас могут быть десятки тысяч элементов. Обычно мы используем сингулярное разложение (Singular Value Decomposition, SVD) (https://ru.wikipedia.org/wiki/Сингулярное_разложение) и метод главных компонент (Principal Component Analysis, PCA) (https://ru.wikipedia.org/wiki/Метод_главных_компонент) для решения данной проблемы, и вскоре покажем как.

Чтобы понять принцип, прежде чем погрузимся в математику, подумаем о том, как уменьшаем количество измерений и каждый день на внутреннем уровне создаем скрытые признаки. Например, люди изобретают такие понятия, как «крутизна», но мы не можем *напрямую* измерить, насколько «крут» кто-то. Другие люди демонстрируют различные закономерности поведения, которые мы внутренне сопоставляем или сводим к нашему измерению «крутизны». Таким образом, крутизна является примером скрытого признака, поскольку ненаблюдаема и не поддается измерению напрямую, и ее можно рассматривать как метод понижения размерности, ведь, возможно, она представляет собой комбинацию многих наблюдаемых «признаков» человека, которые мы незаметно для себя взвешивали в уме.

Здесь происходят две вещи: размерность сводится к одному признаку и скрытому аспекту последнего.

Но в данном алгоритме мы не решаем, каким скрытым факторам нужно уделить внимание. Вместо этого мы позволяем машинам выполнять работу по выяснению того, какие скрытые признаки являются важными. «Важные» в текущем контексте означает следующее: они объясняют расхождения в ответах на различные вопросы, другими словами, эффективно моделируют ответы.

Наша цель — создать модель, имеющую представление в подпространстве с низкой размерностью, в котором собирается информация о вкусах, чтобы генерировать рекомендации. Поэтому мы говорим, что вкус является *скрытым*, но его можно аппроксимировать, собрав всю поддающуюся наблюдению информацию, которую мы *имеем* о пользователе.

Кроме того, считайте, что в большинстве случаев рейтинговые вопросы бинарны (да/нет). С целью преодоления данного обстоятельства в Hunch создали отдельную переменную для каждого вопроса. Они также обнаружили следующее: сравнительные вопросы могут быть лучше при выявлении предпочтений.

ВРЕМЯ ОСВЕЖИТЬ ВАШЕ ЗНАНИЕ ЛИНЕЙНОЙ АЛГЕБРЫ, ЕСЛИ ВЫ ЕЩЕ ЭТОГО НЕ СДЕЛАЛИ

Оставшаяся большая часть этой главы, вероятно, не будет значима (а мы хотим, чтобы она имела смысл для вас!), если вы не знаете линейную алгебру и не понимаете терминологию и геометрическую интерпретацию слов типа *«ранг»* (подсказка: определение данного слова в линейной алгебре не имеет ничего общего с алгоритмами ранжирования), *«ортогональность»*, *«транспозиция»*, *«базис»*, *«пространство столбцов»* и *«матричное разложение»*. Размышление о данных в матрицах как о точках в пространстве и о том, что означает преобразование этого пространства или получение подпространства, может дать правильное представление о ваших моделях и о том, почему они ломаются или как повысить эффективность кода. Это не просто математическое упражнение ради себя самого, хотя в нем есть элегантность и красота, — оно может быть разницей между неудачным стартом проекта и проектом, который купят на eBay. Мы рекомендуем отличный бесплатный, доступный онлайн вводный курс в линейную алгебру от Khan Academy (https://www.khanacademy.org/math/linear-algebra), если вам нужно освежить знания по данному предмету.

Сингулярное разложение (SVD)

Надеемся, вы получили некоторое представление о том, что мы будем делать. Итак, перейдем к математике, начав с сингулярного разложения. Если имеется матрица X размерностью $m \times n$ ранга k, то по теореме линейной алгебры (https://en.wikipedia.org/wiki/Singular_value_decomposition#Statement_of_the_theorem) мы всегда можем представить ее как произведение трех матриц следующим образом:

$$X = USV^{\tau},$$

где матрица U имеет размерность $m \times k$, S — $k \times k$ и V — $k \times n$, столбцы матриц U и V попарно ортогональны (https://en.wikipedia.org/wiki/Orthogonality и https://ru.wikipedia.org/wiki/Ортогональность). Обратите внимание, что стандартная формулировка SVD несколько отличается: U и V — это квадратные унитарные матрицы, а средняя «диагональная» матрица является прямоугольной. Мы будем использовать эту форму, поскольку собираемся брать приближения для X все более низкого ранга. Вы можете найти доказательство существования данной формы в качестве шага в доказательстве существования общей формы здесь: https://en.wikipedia.org/wiki/Singular-value_decomposition#Statement_of_the_theorem.

Применим предыдущую матричную декомпозицию к нашей ситуации. X — наш исходный набор данных, который состоит из пользовательских рейтингов элементов. У нас есть m пользователей, n элементов, а k будет рангом X и, следовательно, также верхней границей числа d скрытых переменных, которые мы хотим учитывать. Обратите внимание: мы выбираем d, тогда как m, n и k определены нашим обучающим набором. Как и в методе k-БС, где k — параметр настройки (совершенно другое k — не путайте!), в данном случае d является аналогичным параметром.

Каждая строка U соответствует *пользователю*, тогда как в V есть строка для каждого *элемента*. Значения, расположенные на диагонали матрицы S, называются «сингулярными». Они определяют важность каждой скрытой переменной — наиболее важная скрытая переменная имеет наибольшее сингулярное значение.

Важные свойства SVD

Поскольку столбцы матриц U и V ортогональны друг другу, то можно упорядочить столбцы по сингулярным значениям с помощью операции перехода от одного базиса к другому. Таким образом, если поместить столбцы в порядке убывания их соответствующих сингулярных значений (что вы и делаете), то размеры будут упорядочены по важности от наивысшего до самого низкого. Вы можете принять приближение X с более низким рангом, отбросив часть S. Другими словами, замените S на подматрицу, взятую из верхнего левого угла S.

Конечно, если вы убираете часть S, то вам придется одновременно отбросить части U и V, но это нормально, поскольку вы отсекаете *наименее важные векторы*. По сути, так вы выбираете количество скрытых переменных d — у вас больше нет исходной матрицы X, а только ее приближение, ввиду того что d обычно намного меньше k, но все еще довольно близко к X. Это то, что люди имеют в виду, когда говорят о сжатии, если вы когда-либо слышали данный термин. Часто существует важная интерпретация значений элементов матриц U и V. Например, с помощью SVD вы можете увидеть, что «самый важный» скрытый признак часто представляет собой что-то вроде показателя того, мужчина перед нами или женщина.

Как бы вы на самом деле использовали эти скрытые характеристики для рекомендаций? Вы должны взять матрицу X, заполнить все пустые ячейки средними значениями рейтинга для данного элемента (не нужно заполнять его нулями, поскольку это может означать нечто в рейтинговой системе, а SVD не может обрабатывать отсутствующие значения), а затем вычислить SVD. Теперь, когда вы произвели такое разложение, это значит, что вы захватили скрытые характеристики, которые при желании можете применять для сравнения пользователей. Но вам нужно не это — вы хотите предсказать. Перемножив U, S и V^T, вы получите приближение A к X или предсказание $\hat{X}$, таким образом, можете прогнозировать оценку, просто просматривая запись для соответствующей пары «пользователь/элемент» в матрице X.

Возвращаясь к нашему первоначальному списку проблем, приведенному в подразделе «Некоторые проблемы, связанные с методом k-БС» текущего раздела, вспомним, что мы хотим избежать проблемы с отсутствующими данными, но она не устраняется с помощью предыдущего подхода, основанного на SVD, равно как и проблема с вычислительной сложностью. На самом деле SVD чрезвычайно вычислительно затратно. Итак, посмотрим, как снизить эту затратность.

Метод главных компонент (PCA)

Рассмотрим другой подход к прогнозированию предпочтений. В этом случае вы по-прежнему ищете U и V, но вам больше не нужно S, поэтому следует просто искать U и V, такие, что:

$$X \equiv U \cdot V^{T}.$$

Ваша проблема оптимизации заключается в том, что вы хотите свести к минимуму расхождение между фактическим X и приближением к X через оценку значений U и V с помощью квадратичных ошибок:

$$argmin\Sigma_{i,j}(x_{i,j} - u_i \cdot v_j)^2.$$

Здесь через u_i обозначена строка матрицы U, соответствующая i-му пользователю, и аналогично через v_j обозначена строка матрицы V, соответствующая j-му элементу. Как обычно, элементы могут включать информацию о метаданных (поэтому вектором возраста всех пользователей будет строка в V).

Тогда значение скалярного произведения $u_i \cdot v_j$ является *предсказанным предпочтением* i-го пользователя для j-го элемента, и вы хотите, чтобы оно было как можно ближе к *действительному предпочтению* $x_{i,j}$.

Итак, вы хотите найти лучшие возможные варианты для матриц U и V, которые сводят к минимуму квадратичную ошибку между прогнозом и наблюдением касательно всего, что вы на самом деле знаете. Идея заключается в том, что если это действительно хорошо для уже известных данных, то будет хорошо и для тех, которые вы угадываете. Описание должно показаться вам знакомым — это среднеквадратичная погрешность, аналогичная той, которую мы использовали для линейной регрессии.

Теперь можете выбрать параметр, а именно число d, определяемое тем, *сколько скрытых признаков вы хотите использовать*. Матрица U будет содержать строку для каждого пользователя и столбец для каждого скрытого признака, а матрица V — строку для каждого элемента и столбец для каждого скрытого признака.

Как вы выбираете d? Как правило, это значение около 100, поскольку больше 20 (как уже говорилось, в ходе разработки продукта мы обнаружили, что довольно хорошо понимали людей, задав им 20 вопросов), и это столько, сколько нужно добавить, чтобы вычисления не стали слишком сложными.

Полученные скрытые признаки — базис четко определенного подпространства полного n-мерного пространства потенциальных скрытых признаков. Нет оснований полагать, что данное решение является единственным, если в вашей матрице ответов не хватает большого количества значений. Но это не обязательно важно, поскольку вы просто ищете решение.

Теорема: полученные скрытые признаки будут некоррелированы. Мы уже обсуждали, что корреляция была проблемой, связанной с методом k-БС, а кто хочет иметь избыточную информацию в своей модели? Таким образом, хороший аспект этих скрытых признаков заключается в том, что они некоррелированы. Ниже приведено краткое доказательство.

Пусть мы нашли матрицы U и V с фиксированным произведением $U \cdot V = X$, поэтому квадратичное рассогласование сведено к минимуму. Следующий шаг — найти лучшие U и V такие, что их элементы малы — фактически мы минимизируем сумму квадратов элементов U и V. Но мы можем преобразовать U с помощью любой обратимой матрицы G размерностью $d \times d$ при условии, что преобразуем матрицу V, используя инверсию G: $U \cdot V = (U \cdot G) \cdot (G^{-1} \cdot V) = X$.

Предположим, что производим преобразования при равном 1 определителе матриц G, то есть ограничимся преобразованиями, не меняющими объем. Если теперь проигнорируем размер элементов V и сосредоточимся лишь на размерах элементов U, то минимизируем площадь поверхности d-мерного параллелепипеда в n-мерном пространстве (в частности, в том, которое порождается столбцами U), где объем фиксирован. Это достигается за счет взаимной ортогональности сторон параллелепипеда, что равносильно фразе, будто скрытые признаки будут некоррелированными.

Но не забывайте, что мы не учли матрицу V! Тем не менее, оказывается, строки V тоже будут взаимно ортогональными, когда мы привели к этому столбцы U. Это легко увидеть, если вы держите в уме, что у X есть SVD, как обсуждалось ранее. На самом деле между SVD и данной формой произведения $U \cdot V$ много общего, и некоторые люди просто называют последнюю алгоритмом SVD, пусть и не совсем верно.

Теперь мы допускаем преобразования с нетривиальным определителем, поэтому, например, пусть G — некая масштабированная версия матрицы тождественного преобразования. Тогда, если мы произведем некоторые вычисления, окажется, что лучший выбор скалярного значения (то есть минимизация суммы квадратов элементов U и V) на самом деле является *геометрическим средним* двух указанных величин, и это просто замечательно. Другими словами, мы минимизируем их среднее арифметическое с единственным параметром (скалярным), а ответ является геометрическим средним.

Вот такое доказательство. Вы верите нам?

Вариант метода наименьших квадратов

Но как вы это делаете? Как на самом деле находите U и V? В реальности, что будет видно далее, вы не минимизируете сначала квадратичную ошибку, а затем размер элементов матриц U и V. Вы фактически выполняете оба действия одновременно.

Таким образом, ваша цель — найти U и V, решив описанную ранее задачу оптимизации. У данной оптимизации нет хорошей замкнутой формулы, как, например,

в обычном методе наименьших квадратов с одним набором коэффициентов. Вместо этого вам нужен итеративный алгоритм, такой как градиентный спуск. Пока ваша задача выпуклая, решение будет хорошо сходиться (то есть вы не обнаружите себя в локальном, но не глобальном максимуме) и вы можете заставить вашу задачу быть таковой, используя регуляризацию.

Ниже приведен алгоритм.

- Выберите случайную матрицу *V*.
- Оптимизируйте *U* при фиксированной *V*.
- Оптимизируйте *V* при фиксированной *U*.
- Продолжайте выполнять два предыдущих шага, пока все полностью не изменится. Точнее, вы выбираете значение ϵ, и если ваши коэффициенты изменяются меньше чем на ϵ, то объявляете свой алгоритм конвергентным.

Теорема без доказательства: предыдущий алгоритм будет конвергентным, если вы взяли достаточно большое априорное значение. Если вы увеличите свое априорное значение, то упростите оптимизацию, поскольку вы искусственно создаете более выпуклую функцию. С другой стороны, если ваше априорное значение огромно, то все ваши коэффициенты будут равны нулю, так что это действительно не поможет в любом случае. Как следствие, вы, вероятно, не захотите увеличить свое априорное значение. Оптимизация вашего априорного значения философски отвергается, поскольку что это за *априорное* значение, если вы его подгоняете так, чтобы оно совершало необходимые вам действия? Плюс вы в некоторой степени смешиваете здесь схемы, осуществляя поиск точного приближения *X* параллельно с минимизацией коэффициентов. Чем больше вы печетесь о коэффициентах, тем меньше заботитесь о *X*. Однако на самом деле вам нужно заботиться только о *X*.

Фиксируйте V и скорректируйте U

При данном способе вы производите оптимизацию по каждому пользователю. То есть для *i*-го пользователя хотите найти:

$$\arg\min_{u_i} \sum_{j \in P_i} (p_{i,j} - u_i \cdot v_j)^2,$$

где значения v_j фиксированы. Другими словами, в данный момент времени вы заботитесь только об *одном пользователе*.

Но подождите минутку, это совпадает с линейным методом наименьших квадратов и имеет замкнутое решение! Другими словами, набор:

$$u_i = (V_{*,i}^{\tau} V_{*,i})^{-1} V_{*,i}^{\tau} P_{*i},$$

где $V_{*,i}$ является выборкой из *V*, для которой у вас есть предпочтения *i*-го пользователя. Произвести инверсию просто, поскольку это размерность $d \times d$, что является

маленькой величиной. И здесь не так много предпочтений на одного пользователя, поэтому найти данное решение много раз действительно не так уж сложно. В целом у вас есть выполнимый способ корректировки для U.

Фиксация U и оптимизация V происходят аналогично — вам только нужно учитывать пользователей, оценивших конкретный фильм, что может быть довольно большой величиной для популярных фильмов, но в среднем это не так; однако даже в подобном случае вы только инвертируете матрицу размерностью $d \times d$.

Еще одна интересная вещь: поскольку каждый пользователь зависит только от своих предпочтений, то вы можете распараллелить эту коррекцию U или V. Вы можете запускать ее на стольких разных машинах, насколько быстро хотите выполнить ее.

Последние размышления о данных алгоритмах

Существует много разных вариантов этого подхода, но мы надеемся, что дали представление о компромиссе между данными методами и о том, как их можно использовать для составления прогнозов. Иногда вам необходимо расширить метод, чтобы он работал в вашем конкретном случае.

Например, вы можете добавлять новых пользователей или данные, а также оптимизировать U и V. Вы можете выбрать, какие пользователи, по вашему мнению, нуждаются в большем количестве коррекций, чтобы сэкономить время вычислений. Или если у них достаточно рейтингов, то можете решить не корректировать остальных.

Как и в любой модели машинного обучения, вы должны выполнить перекрестную проверку (https://en.wikipedia.org/wiki/Cross-validation_(statistics)) для данной модели — оставить немного данных и проверить, насколько хорошо работает модель, — мы говорили об этом на протяжении всей книги. Это способ проверки наличия проблем переобучения.

Мысленный эксперимент: фильтр для пузырей

Каковы последствия использования минимизации ошибок для прогнозирования предпочтений? Как представление рекомендаций влияет на собранные данные о реакции?

Например, можете ли вы оказаться в локальных максимумах с эффектом «богатые становятся богаче»? Другими словами, показывают ли определенные элементы, что вначале им было дано необоснованное преимущество перед другими элементами? И поэтому некоторые вещи просто становятся популярными или нет случайным образом?

Как вы это скорректируете?

Упражнение: постройте собственную рекомендательную систему

В главе 6 мы провели некий разведочный анализ набора данных GetGlue. Теперь у вас есть возможность создать рекомендательную систему с этим набором. Следующий код не для GetGlue, но код Мэтта иллюстрирует реализацию рекомендательной системы на относительно небольшом наборе. Ваша задача — настроить его для работы с данными GetGlue.

Пример кода на Python

```
import math,numpy

pu    =    [[(0,0,1),(0,1,22),(0,2,1),(0,3,1),(0,5,0)],[(1,0,1),
(1,1,32),(1,2,0),(1,3,0),(1,4,1),(1,5,0)],[(2,0,0),(2,1,18),
(2,2,1),(2,3,1),(2,4,0),(2,5,1)],[(3,0,1),(3,1,40),(3,2,1),
,(3,4,0),(3,5,1)],[(4,0,0),(4,1,40),(4,2,0),(4,4,1),
],[(5,0,0),(5,1,25),(5,2,1),(5,3,1),(5,4,1)]]

pv    =    [[(0,0,1),(0,1,1),(0,2,0),(0,3,1),(0,4,0),(0,5,0)],
[(1,0,22),(1,1,32),(1,2,18),(1,3,40),(1,4,40),(1,5,25)],
[(2,0,1),(2,1,0),(2,2,1),(2,3,1),(2,4,0),(2,5,1)],[(3,0,1),
,(3,2,1),(3,3,0),(3,5,1)],[(4,1,1),(4,2,0),(4,3,0),
,(4,5,1)],[(5,0,0),(5,1,0),(5,2,1),(5,3,1),(5,4,0)]]

V = numpy.mat([[ 0.15968384,    0.9441198 ,    0.83651085],
               [ 0.73573009,    0.24906915,    0.85338239],
               [ 0.25605814,    0.6990532 ,    0.50900407],
               [ 0.2405843 ,    0.31848888,    0.60233653],
               [ 0.24237479,    0.15293281,    0.22240255],
               [ 0.03943766,    0.19287528,    0.95094265]])
print V

U = numpy.mat(numpy.zeros([6,3]))
L = 0.03

for iter in xrange(5):

    print "\n     ITER %s     "%(iter+1)

    print "U"
    urs = []
    for uset in pu:
        vo = []
        pvo = []
        for i,j,p in uset:
            vor = []
            for k in xrange(3):
                vor.append(V[j,k])
            vo.append(vor)
            pvo.append(p)
```

```
        vo = numpy.mat(vo)
        ur = numpy.linalg.inv(vo.T*vo +
            L*numpy.mat(numpy.eye(3))) *
            vo.T * numpy.mat(pvo).T
        urs.append(ur.T)
    U = numpy.vstack(urs)
    print U

    print "V"
    vrs = []
    for vset in pv:
        uo = []
        puo = []
        for j,i,p in vset:
            uor = []
            for k in xrange(3):
                uor.append(U[i,k])
            uo.append(uor)
            puo.append(p)
        uo = numpy.mat(uo)
            vr = numpy.linalg.inv(uo.T*uo + L*numpy.mat(num py.eye(3))) *
                uo.T * numpy.mat(puo).T
        vrs.append(vr.T)
    V = numpy.vstack(vrs)
    print V

    err = 0.
    n = 0.
    for uset in pu:
        for i,j,p in uset:
            err += (p – (U[i]*V[j].T)[0,0])**2
            n += 1
    print math.sqrt(err/n)

print
print U*V.T
```

9 Визуализация данных и выявление мошенничества

У главы два автора: Марк Хансен (Mark Hansen), профессор Колумбийского университета, и Ян Вонг (Ian Wong), ученый-аналитик в Square. (Так было, когда он пришел в класс в ноябре 2012 года. На данный момент он работает в Prismatic.) Докладчики и разделы не объединены одной темой, кроме того, что будут обсуждать визуализацию данных... и оба жили в Калифорнии! Говоря более серьезно, оба являются рассудительными людьми (как и все наши рассказчики), которые глубоко задумываются над следующими темами и вопросами: что делает код хорошим, характер языков программирования как форма выражения — и над главным вопросом этой книги: что такое наука о данных?

История визуализации данных

Первым выступит Марк Хансен (http://www.stat.ucla.edu//~cocteau/), который благодаря творческому отпуску в The New York Times R&D Lab (научно-исследовательская лаборатория New York Times) недавно перешел из Калифорнийского университета в Лос-Анджелесе (University of California at Los Angeles, сокр. UCLA) в Колумбийский университет (Columbia University), где, совмещая журналистику и статистику, возглавляет Брауновский институт инноваций в области средств массовой информации (Brown Institute for Media Innovation). Имеет степень доктора философии по статистике в Беркли и несколько лет перед своим назначением в UCLA работал в Bell Labs (и снова она!), где занимался статистикой и (на добровольных началах) электротехникой, дизайном и медиаискусством. Марк — известный эксперт по визуализации данных, а также энергичный и интересный рассказчик.

Марк проведет нас через ряд ориентиров и представит исторический контекст для своих проектов по визуализации данных, о которых расскажет подробнее

в конце. Проекты Марка — настоящие произведения искусства: инсталляции, появляющиеся в музеях и общественных местах. Рэйчел пригласила его, поскольку его работа и философия вдохновляют и выступают тем, к чему нужно стремиться. Он установил собственный курс, определил поле, взорвал границы и постоянно бросает вызовы статус-кво. Марк занимался визуализацией данных еще до того, как она стала популярной, или, другими словами, мы считаем его одним из основоположников этого направления. Чтобы на практике улучшить ваши навыки в визуализации данных, в конце главы мы представим некоторые идеи и направления.

Габриэль Тард

Марк начал с краткого рассказа о Габриэле Тарде (Gabriel Tarde) (https://ru.wikipedia.org/wiki/Тард,_Габриель), который был социологом, верившим, что обществоведение способно произвести значительно больше данных, чем естествознание.

По мнению Тарда, естественные науки *наблюдают на расстоянии*: обычно они моделируют или объединяют модели, описывающие совокупность каким-либо образом, например, биолог может говорить о функции совокупности наших клеток. Тард указал на это как на недостаток, обычно вызванный нехваткой информации. Согласно Тарду, наоборот, нужно отслеживать каждую клетку.

Аналогично можно поступить в социальной сфере, заменив клетки людьми. Можно собрать огромное количество информации об отдельных лицах, особенно если они предлагают ее с помощью Facebook.

Минутку, разве мы не потеряли за деревьями лес, сделав это? Другими словами, если мы фокусируемся на микроуровне, то можем упустить огромную культурную ценность социального взаимодействия. Бруно Латур (Bruno Latour), современный французский социолог, высказал мнение по поводу Тарда в работе Tarde's Idea of Quantification («Идея квантификации Тарда») (http://www.bruno-latour.fr/sites/default/files/116-CANDEA-TARDE-FR.pdf):

> «Но целое сейчас является не более чем предварительной визуализацией, которая в любой момент может быть модифицирована и реверсирована с возвращением к индивидуальным компонентам и последующим поиском других средств для перегруппировки тех же самых элементов в альтернативные совокупности».

В 1903 году Тард даже предугадал появление Facebook (https://archive.org/details/lawsofimitation00tard) как своего рода «ежедневной прессы»:

> «Если статистика продолжит развиваться, как это было в последние несколько лет, если относительно информации, которую она выдает, продолжат возрастать корректность, скорость передачи, объем и регулярность, то может настать время, когда после каждого общественного мероприятия числовые данные будут выдаваться сразу же

> автоматически, чтобы, как говорится, занять свое место в статистических регистрах, которые будут непрерывно передаваться общественности и распространяться за границу в графической форме с помощью ежедневной прессы. Затем на каждом шагу, при каждом взгляде, брошенном на плакат или газету, нас будут как бы атаковать статистические факты, обладающие точностью и концентрированным знанием обо всех особенностях реальных социальных условий, о коммерческих успехах и поражениях, о возникновении или распаде отдельных политических партий, о развитии или разрушении какой-либо доктрины и т. д., тем же самым способом, каким мы бываем атакованы, когда открываем глаза из-за колебаний эфира, сообщающих нам о приближении или отдалении какого-либо так называемого тела и многих других вещей аналогичного характера».

Затем Марк сформулировал тему своей лекции:

> «Измени́те инструмент — и вы изме́ните всю социальную теорию, связанную с ним».

Подобно пресловутому коту физиков (https://ru.wikipedia.org/wiki/Шрёдингер,_Эрвин), Марк (и Тард) хочет, чтобы мы заново рассмотрели оба явления: способ, благодаря которому мы наблюдаем изменения в структуре общества, и осмысление связей индивидуума с общностью.

Другими словами, характер ранее существовавших методов сбора данных вынуждал рассматривать сводные статистические данные как то, что можно (вполне обоснованно) вычислять по выборке, например по средним значениям. Но теперь, когда действительно возможно получить все данные и работать с ними, больше нет необходимости концентрироваться только на статистических показателях, имеющих смысл в совокупности, равно как и на собственной индивидуально составленной статистике (например, взятой из взаимодействий наподобие графов), что возможно на данный момент благодаря высокой степени контроля. Не позволяйте догме, которая возникла из-за прошлых ограничений, направлять ваше мышление, когда этих ограничений уже нет.

Мысленный эксперимент Марка

Раз данные становятся более персонифицированными, а мы собираем больше данных об индивидууме, то какие новые методы или средства необходимы для выражения фундаментальных связей между нами и нашими сообществами, нашими сообществами и нашими странами, нашими странами и миром?

Будем ли мы когда-нибудь удовлетворены результатами опроса или рейтингами доверия к президенту, когда сможем увидеть полную траекторию общественного мнения, как индивидуального, так и взаимодействующего?

Добавим: хотим ли мы на самом деле жить в культурной среде, где подобная информация так легко отслеживается и доступна?

Что такое возрожденная наука о данных

Марк пересмотрел вопрос, который Рэйчел поставила и на который попыталась ответить в главе 1, поскольку увлекается переосмыслением вообще всего. Он начал разговор со следующей цитаты Джона Тьюки (https://ru.wikipedia.org/wiki/Тьюки,_Джон):

> «Лучшее в профессии статистика — то, что вы можете поиграть на заднем дворе у каждого».

Еще раз подумаем — так ли хорошо данное обстоятельство? Оправданно ли оно? В определенном смысле, думая о себе как об играющих во дворе у *других* людей с *их* игрушками, мы проводим черту между «традиционной областью данных» и «всем остальным».

Возможно, это даже подразумевает, что вся наша магия исходит из традиционных областей данных (математики, статистики, информатики) и мы являемся некими суперлюдьми, поскольку чрезвычайно педантичны. Конечно, так удобно считать с высоты наших эго, но утверждение слишком ограниченно и высокомерно.

И вышеописанное вызывает вопрос: что такое «традиционные» и «все остальные» вообще?

По мнению Марка, во «все остальные» следует включить области от общественных и естественных наук до образования, дизайна, журналистики и медиаискусства. Наша деятельность подразумевает нечто более важное, чем быть технологами, и нам нужно понимать, что сама по себе технология проистекает из естественных потребностей дисциплины. Например, геоинформационная система (GIS) возникла из географии, а интеллектуальный анализ текстовых данных — из цифровых гуманитарных наук.

Другими словами, не математики управляют миром, а скорее практические сферы были сформированы техниками, естественным образом выросшими из этих областей. Когда данные взаимодействуют с их методами, каждый метод обучается отдельно; их проблемы уникальны для конкретного метода.

Достоверная наука о данных объединяет эти уроки, и это не чисто математическая интеграция. Она может быть способом описания событий. В частности,

мы говорим, что наука о данных — это не обязательно что-то поддающееся измерению.

Подведем итог: возможно, у языка науки о данных есть *нечто* связанное с общественными науками точно так же, как и с математикой.

Возможно, вы не удивитесь, услышав, что Марк, рассказывая о своем профиле исследователя данных, ввел термин «экспансионист».

Processing

Затем Марк описал язык программирования под названием Processing (обработка) (https://processing.org/) в контексте класса программирования, который предоставил художникам и дизайнерам. Он использовал его в качестве примера разницы между дизайнером и инженером, когда первый принимается за изучение данных или начинает кодировать. Хороший язык, по сути, структурирован или предназначен для выражения определенных задач и способа мышления людей, которые его применяют.

Одним из подходов, ведущих к пониманию этой разницы, является еще один мысленный эксперимент. А именно, что является вариантом использования в языке для художников? Сравните это с тем, что язык, такой как R, должен охватывать для статистиков или исследователей данных (например, стохастичность, распределения, векторы и данные).

В языке для художников вы захотите иметь возможность определять фигуры, точно отображать любую визуальную вещь, которую задумали, рисовать, вероятно, в 3D, анимировать, обрабатывать взаимодействия и (самое главное) *публиковать*.

Обработка может выполняться на языке Java, например, с простой кнопкой Publish (Опубликовать). Язык адаптирован для задач художников. Марк упомянул, что обучение дизайнеров коду означало для него шаг назад и рассказ об итерациях, операторах и т. д. — другими словами, о вещах, которые казались ему очевидными, но не были таковыми для художника. Марк должен был объяснять все свои допущения, и это самое интересное в обучении непосвященных.

Франко Моретти

Марк перешел к обсуждению медленного чтения по сравнению с быстрым. Он упомянул Франко Моретти (Franco Moretti) (https://ru.wikipedia.org/wiki/Моретти,_Франко), литературоведа из Стэнфорда.

Франко рассуждает о быстром чтении, которое означает попытку понять суть написанного, не читая строку за строкой. Это приводит к мышлению PCA-типа (https://en.wikipedia.org/wiki/Principal_component_analysis), своего рода уменьшению размеров романов (методы уменьшения размеров мы рассматривали в главе 8).

Марк считает это замечательным примером того, как в идеале наука о данных объединяет способы, которые уже используются специалистами в различных областях. Другими словами, мы не просто приходим в дворы других людей и играем; возможно, вместо этого мы заходим и наблюдаем за их игрой, а затем формализуем и делаем вклад в их процесс с помощью собственных колокольчиков и свистков. Таким образом, другие могут научить нас новым играм, фактически расширяющим наши фундаментальные концепции данных и подходы, которые нужны нам для их анализа.

Примеры проектов визуализации данных

Вот некоторые из любимых проектов визуализации Марка, и о каждом из них он спрашивал нас: это ваша идея визуализации данных? Что является данными?

На рис. 9.1 представлена проекция на паровое облако электростанции. Размер зеленой проекции соответствует количеству энергии, используемой городом.

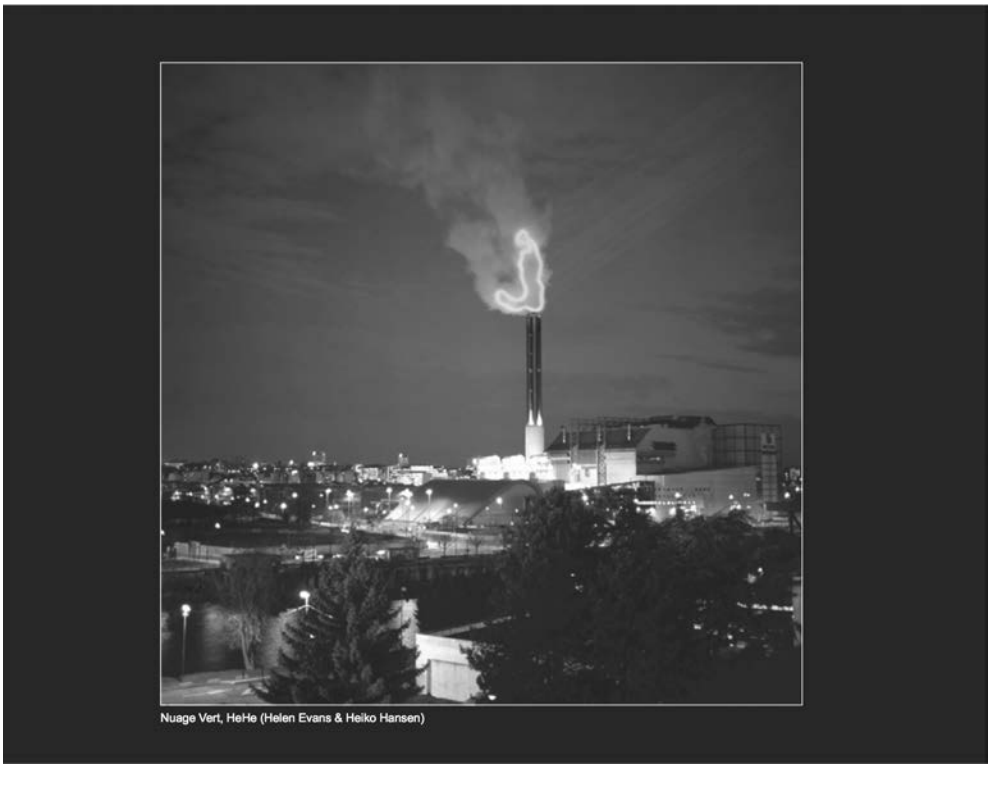


Рис. 9.1. «Зеленое облако» (Nuage Vert) от Хелен Ивенс (Helen Evans) и Хейко Хансен (Heiko Hansen) (https://www.youtube.com/watch?v=I_4rTQCWItw&feature=youtu.be)

В «Одном дереве» (One Tree) (рис. 9.2) художник клонировала деревья и сажала генетически идентичные семена в нескольких областях. Снимок показывает в том числе условия окружающей среды в каждой области, где они были высажены.

Рис. 9.2. «Одно дерево» от Натальи Еремиенко (Natalie Jeremijenko) (https://boingboing.net/2003/05/16/natalie-jeremijenkos.html)

На рис. 9.3 показан «Пыльный рельеф» (Dusty Relief), в котором строение собирает вокруг себя загрязнение, отображаемое как пыль.

«Проект “Явление”» (Project Reveal) (рис. 9.4) — это своеобразное волшебное зеркало, которое беспроводным образом соединяется с используемой технологией распознавания лиц и предоставляет вам информацию о вас самих. По словам Марка, стоя у зеркала утром, вы получаете тот самый момент откровения.

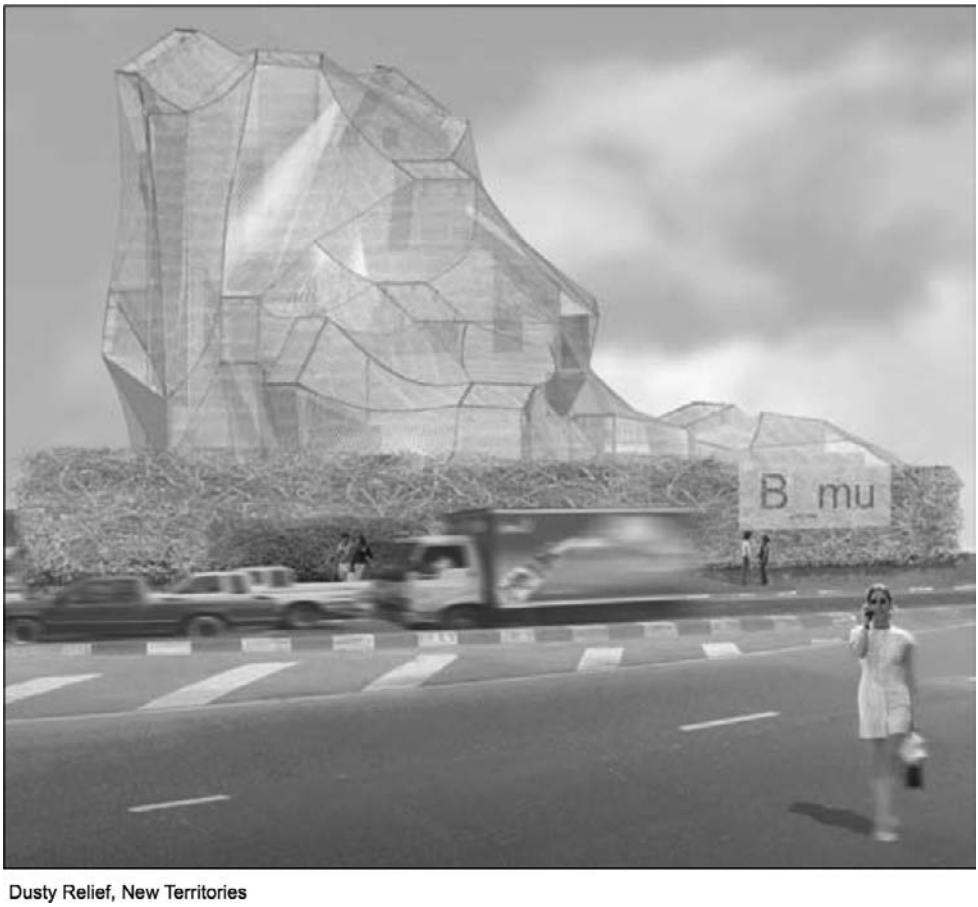


Рис. 9.3. «Пыльный рельеф» от New Territories (http://www.new-territories.com/roche2002bis.htm)

Лора Курган (Laura Kurgan) возглавляет SIDL (Spatial Information Design Lab). Во фрагменте, показанном на рис. 9.5, она перевернула статистику преступности, предоставленную Google. Лора нашла данные для лиц, отбывающих тюремное заключение, и для каждого заключенного проверила его домашний адрес, измеряя в перерасчете на дом, сколько денег государство тратит на то, чтобы содержать в тюрьме людей, живших в этих домах. Она показала, что некоторые блоки тратили на содержание 1 000 000 долларов. Мораль проекта: не следует наносить что-то на карту просто потому, что вы можете это сделать. Это не значит, что есть новая история. Чтобы ее получить, иногда нужно копать глубже и все переворачивать.

Рис. 9.4. «Проект "Явление"» от научно-исследовательской лаборатории The New York Times (http://nytlabs.com/projects/mirror.html)

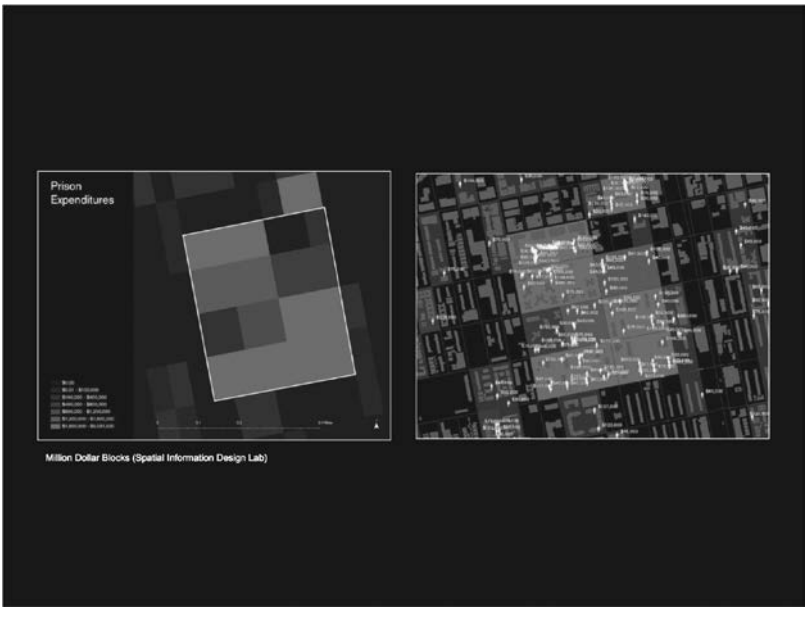


Рис. 9.5. Блоки на миллион долларов от Spatial Information Design Lab (http://www.spatialinformationdesignlab.org/)

Проекты визуализации данных от Марка

Теперь, когда мы слегка познакомились с влиянием и философией Марка, посмотрим на некоторые из его проектов, чтобы увидеть, как он воплощает их в жизнь.

Фойе The New York Times: «Наборный шрифт»

Марк провел нас через проект, который выполнил совместно с Беном Рубином (Ben Rubin) — медиахудожником, долгие годы сотрудничающим с Марком, — по поручению The New York Times. (В дальнейшем Марк прибыл в научно-исследовательскую лабораторию The New York Times во время творческого отпуска.) На рис. 9.6 показан этот проект, установленный в фойе штаб-квартиры The New York Times в среднем Манхэттене, расположенном на 8-й авеню и 42-й улице.

Рис. 9.6. «Наборный шрифт», фойе здания The New York Times, от Бена Рубина и Марка Хансена

Проект состоит из 560 текстовых дисплеев (две стены с 280 дисплеями на каждой), и они прокручивают различные сцены, у каждой из которых есть тема и базовая модель анализа данных.

На одном из них цифровые кадры наподобие бегущей строки волна за волной оставляют за собой блоки текста, где каждый из блоков представляет различные истории из газеты. В тексте данной истории выделены фразы, которые делают ее отличной от других в информационно-теоретическом смысле.

На другом изображении подсвечиваются числа из историй, так что на данном экране можно увидеть «18 горилл». В третьем эпизоде кроссворды разгадывают сами себя под аккомпанемент звуков карандашей, пишущих по бумаге.

На рис. 9.7 показан пример демонстрационного бокса, который предназначен для передачи ретрозвучания. В каждом боксе встроены Linux-процессор под управлением Python и звуковая карта, которая воспроизводит различные звуки: щелчки, набор на клавиатуре, колебания — в зависимости от того, какая сцена проигрывается.

Рис. 9.7. Демонстрационный бокс проекта «Наборный шрифт»

Данные собраны из текстов статей The New York Times, блогов и с помощью информационно-поисковых систем. Каждое предложение проанализировано с применением стэнфордских технологий обработки естественного языка (https://nlp.stanford.edu/), которые представляют предложения в графической форме.

Проект насчитывает около 15 сцен, написанных на коде, позволяющем добавлять новые. На YouTube можно посмотреть (https://www.youtube.com/watch?v=WfZQf1983iw) интервью об их выставке с Марком и Беном.

Проект «Каскад»: жизнь на экране

Затем Марк рассказал нам о проекте «Каскад» — его совместной работе с Джером Торпом (Jer Thorp) (http://blog.blprnt.com/about) (художником в The New York Times) в партнерстве с bit.ly (https://bitly.com/). «Каскад» возник в результате размышлений о том, как люди делятся ссылками на The New York Times в Twitter.

Идея состояла в сборе количества данных, достаточного для того, чтобы можно было видеть, как люди просматривают, кодируют ссылку в bit.ly, делятся ею в Twitter, видеть, что другие люди щелкают на этом твите, наблюдать, как bit.ly декодирует ссылку, а затем видеть, как эти люди просматривают страничку The New York Times. На рис. 9.8 показана визуализация всего процесса, во многом аналогичного тому, как Тард предложил делать это.

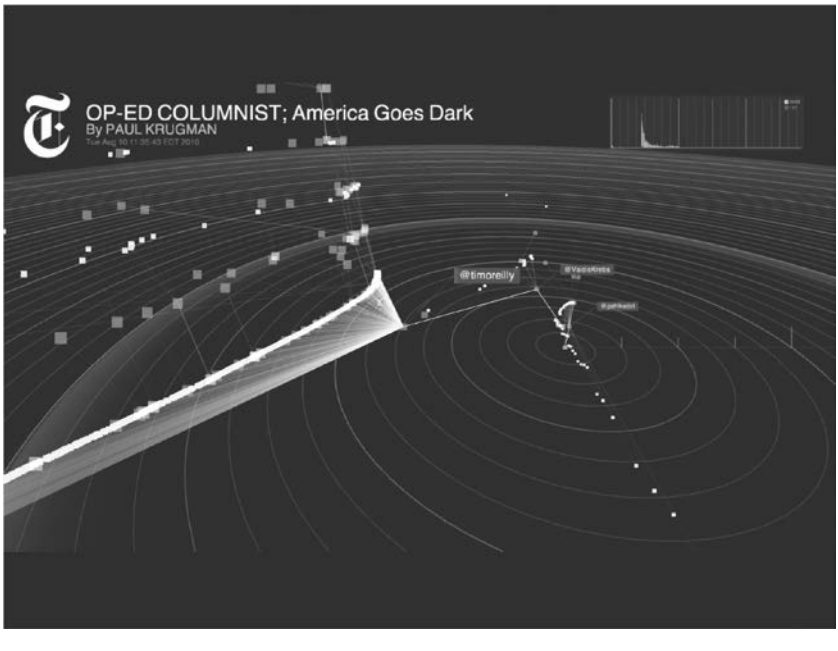


Рис. 9.8. Проект «Каскад» от Джера Торпа и Марка Хансена

Разумеется, пришлось принимать решения по данным: например, свободное совпадение твитов и нажатий во времени. Если 17 разных твитов предоставили один и тот же URL, то узнать, какую ссылку или твит нажал тот или иной человек, невозможно, в связи с чем они предполагаемые (предположение фактически подразумевает вероятностное сопоставление с метками времени, таким образом, оно по крайней мере обоснованно). Авторы использовали карту Twitter, показывающую, кто на кого подписан, — если кто-то, чьим подписчиком вы являетесь, «твитнул» что-либо раньше вас, то это считается ретвитом.

На сайте https://www.youtube.com/watch?v=yQBOF7XeCE0 представлено видео о проекте «Каскад» от The New York Times.

Оно было записано два года назад, и Twitter с тех пор стал несколько больше.

Кронкайт Плаза

Затем Марк рассказал нам кое о чем, над чем он работал с обоими художниками: Джером и Беном. Это также связано с новостями, но подразумевало демонстрацию чего-либо снаружи здания, а не в фойе; в частности, это здание службы связи в Кронкайт Плаза в Техасском университете в Остине (University of Texas at Austin, сокр. UT at Austin), изображенное на рис. 9.9.

Каждый вечер в Cronkite Plaza можно видеть сцены, проецируемые на здание с помощью шести разных проекторов. Большая часть демонстрируемого текста поступает из новостных передач Уолтера Кронкайта (Walter Cronkite), но авторы использовали и местные источники новостей с закрытыми заголовками. В одной сцене извлекались вопросы, заданные во время показа местных новостей, такие как «Как она отреагировала?» или «Какого вида собаку вы бы получили?».

Транзакции eBay и Books

И снова работая совместно с Джером Торпом, Марк изучил дневные транзакции в eBay, проведенные через Paypal, и, по одному ему известной причине, два года продаж книг. Как вы себе это представляете? Взгляните на их художественную визуализацию — инсталляцию данных, заказанную eBay для биеннале[1] ZERO1, проходившего в 2012 году (рис. 9.10), и на послужившие основой данные (рис. 9.11).

Их оригинальный подход заключался в следующем: они начали с текста пьесы Death of a Salesman («Смерть коммивояжера») Артура Миллера (Arthur Miller)

[1] Мероприятие (выставка), проводимое с периодичностью раз в два года. — *Примеч. пер.*

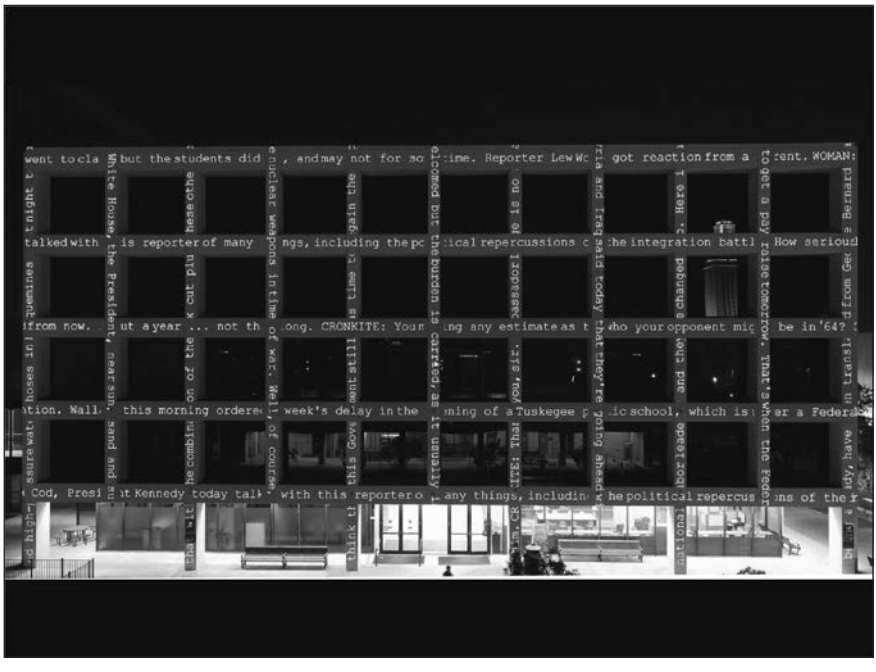


Рис. 9.9. «Вот такие дела» от Джера Торпа, Бена Рубина и Марка Хансена

Рис. 9.10. «Перед нами дом коммивояжера (2012)» от Джера Торпа и Марка Хансена

```
4106.00,CHINESE OLD SIGNED CERAMIC SHUDEI BONSAI POT KYUSU NR
712.50,Omron HEM-650 Wrist Blood Pressure Monitor with APS
499.99,Panasonic TC-P42X3 42" Viera Plasma HDTV
491.00,$500 Best Buy Gift Card for $491.00
372.23,Droid Incredible 2 (Verizon)
346.99,Hondo Chiquita Guitar
329.70,IMATION CD-RW 48 SPINDLES
324.25,1979 Camellias of Yunnan Souvenir Miniature Stamp sheet
287.00,DIESEL New $690 Leather Jacket Coat M
279.79,NEW CALLAWAY TOP FLITE COMPLETE MRH GOLF CLUBS SET BLUE
275.88,AMAZING LABRADORITE 925 SILVER CUFF BRACELET BA31
227.16,NEW Chefs Banquet Survival Emergency Food 330 servings
212.30,Authentic Prada Blue Bag
203.50,Superb Chinese Carved Jade Plaque 18th C.
189.00,NEW IDF Tactical Combat Vest with Removable Backpack
187.98,HP OfficeJet Pro 8500A PLUS AlO Printer   A910g    NEW
182.90,Cisco Small Business RV042 Dual WAN VPN Router -  RV042
174.95,PHILIPPE STARCK Black VEILED Stainless VEIL NEW! PH5022
173.23,14K. GOLD CHANDELIERS EARRINGS NATURAL BLUE TOPAZ GEMS
137.99,FUJI FINEPIX JZ300 10x 12MP DIGITAL CAMERA SILVER NEW
126.00,New Stock White/Ivory Wedding Dress Sz:6/8/10/12/14/16
124.64,Lot of 700 HP 02 Mixed Colors Empty Ink Tank Cartridges
116.40,Goddess of Wisdom 2001 Barbie Doll NIB Mint Condition
112.08,Littmann Cardiology III Stethescope
111.11,ALLEN EDMONDS PARK AVENUE OX BLACK 8 D MEDIUM $325
105.95,REPRODUCTION GERMAN WWII WOOL SOCKS SIZE 2 RING (9-10)
99.00,TOSHIBA SATELLITE L305-S5875 15.4" WXGA  LCD SCREEN
97.58,calvin klein men's slim fit plaid suit pants
90.00,Pear Shape Diamond
80.30,HAPPY CHLOE DUCK 2010 SWAROVSKI  RETIRED  #1041293
79.99,Cisco 2600 series 2611 2-port Ethernet Router CCNA CCNP
77.77,PRO BICYCLE MECHANICS XLC TOOL KIT 33pc BIKE REPAIR SET
74.90,NEW LG VX10000 VOYAGER QWERTY TXT MP3 PHONE VERIZON
69.94,Modern Jacquard Bedding Comforter Set Queen Black/Grey
67.49,pearl gameboy advanced SP metal case 18 gamesECT.
```

Рис. 9.11. Данные, на основе которых была создана инсталляция для eBay

(https://ru.wikipedia.org/wiki/Смерть_коммивояжера). И воспользовались механизмом mechanical turk (мы обсуждали, что это значит, в главе 7) для обнаружения в тексте объектов, которые можно приобрести на eBay.

Найденный объект перемещается в специальный контейнер, например «стул», или «флейта», или «стол». Когда собралось несколько предметов для покупки, данный механизм берет эти объекты и смотрит, где все они продаются, в дневных транзакциях на eBay, а также просматривает подробную информацию об отличающихся

значениях и т. д. После изучения продаж код найдет почтовый индекс в каком-нибудь тихом месте вроде Монтаны.

Затем Марк переключается на данные о продажах книг, просматривает все книги, купленные или проданные по этому почтовому индексу, выбирает книгу (которая также находится в Project Gutenberg (http://www.gutenberg.org/)) и начинает ее читать и собирать в ней «покупаемые» объекты. И все повторяется. Здесь можно посмотреть видео, показывающее процесс: https://vimeo.com/50146828.

Общественный театр «Машина для Шекспира»

Последнее, что показал нам Марк, — это его совместная работа с Рубином и Торпом, установленная в фойе Общественного театра (the Public Theater) (https://www.publictheater.org/), которая показана на рис. 9.12. Данное произведение представляет собой овальную структуру с 37 узкими, вытянутыми, похожими на лезвия светодиодными экранами, которая подвешена над стойкой бара в театре.

Рис. 9.12. «Машина для Шекспира» от Марка, Джера и Бена

Каждой пьесе Шекспира соответствует одно «лезвие». Более длинные пьесы расположены в удлиненной части овала — при входе вы увидите «Гамлета».

Входящими данными являются тексты каждой из пьес. В каждом изображении происходит что-то особенное, например, могут собираться именные конструкции, имеющие отношение к персонажу каждой из пьес, то есть на «лезвии» для «Гамлета» будут показаны только фразы о персонажах из данного произведения. В другом изображении используются различные виды комбинаций или лингвистические конструкции, такие как фразы из трех слов наподобие «высокие и могущественные» либо «добрые и любезные» или фразы со сложным словом: «дьявольски-священная», «сердечная боль» или «с грубыми чертами лица»[1].

Следует отметить, что цифровые гуманитарные науки через MONK Project (http://monkproject.org/) активно предлагают описание пьес в формате XML (https://ru.wikipedia.org/wiki/XML). Каждому отдельному слову уделяется повышенное внимание, и выделяется порядка 150 различных частей речи.

Как сказал Марк, это Шекспир, поэтому он остается потрясающим вне зависимости от того, что вы делаете. Но также слова поочередно рассматриваются в качестве символов, в разрезе определенной тематики или как части речи.

Итак, снова рассмотрим вопрос, который задал Марк, прежде чем показать нам эти визуализации: что является данными? *Все* представленное — данные.

Последний совет от Марка о том, как добывать данные: будьте хорошим исследователем — обычно проще получить данные, если запросить их тихим вежливым голосом.

Цели этих экспозиций

Данные экспозиции должны быть изящными и художественными, но также должны учить чему-то или рассказывать историю. В то же время не стоит перебарщивать с нравоучительностью. Цель заключается в том, чтобы оставаться между искусством и информацией. Это забавное положение: все чаще мы видим эффект сглаживания, когда инструменты становятся цифровыми и доступными, так что статистики могут писать код как дизайнеры — мы можем создавать вещи, которые выглядят как дизайн, но действительно ли они им являются, — и аналогично дизайнеры могут сделать нечто выглядящее как данные или статистика, но таково ли оно на самом деле?

Наука о данных и риски

Наш следующий гость, Ян Вонг (Ian Wong), прибыл из Сан-Франциско, чтобы рассказать о том, как наука о данных справляется с рисками. Ян — ученый-аналитик

[1] В оригинале devilish-holy, heart-sore и hard-favoured соответственно. — *Примеч. пер.*

в Square и ранее отказался от программы доктора наук по электротехнике в Стэнфорде, где занимался исследованиями в области машинного обучения. (Он получил несколько степеней магистра в области статистики и электротехники за это время.) Поговорив с классом, Ян покинул Square и теперь работает в Prismatic, настраиваемом под нужды клиента сервере новостей.

Ян начал с трех ключевых моментов.

- *Машинное обучение != написание R-сценариев.* Машинное обучение (machine learning, ML) основано на математике, выражено в коде и собрано в программное обеспечение. Вам необходимо разработать качественные методы проектирования ПО и научиться писать читаемый и многоразовый код. Пишите код для читателей, а не для автора, так как производственный код перечитывается и гораздо чаще перестраивается другими людьми, чем вами.
- *Визуализация данных != построение красивой диаграммы.* Следует широко использовать и внедрять визуализации в инфраструктуру каждой хорошей компании. Интегрировать их в продукты и процессы. Они должны помогать в работе.
- *ML и визуализация данных совместно дополняют человеческий интеллект.* Будучи людьми, мы имеем ограниченные познавательные способности. Но, основываясь на данных, мы можем создавать экзоскелеты (http://seangourley.com/2012/08/high-frequency-trading-and-the-new-algorithmic-ecosystem/), которые позволяют понимать информационный мир и ориентироваться в нем.

О Square

Компания была основана в 2009 году Джеком Дорси (Jack Dorsey) и Джимом Маккелви (Jim McKelvey). Square разрослась с 50 сотрудников в 2011 году до более чем 500 в 2013-м.

Миссия компании — сделать торговлю легкой для всех. Как считают основатели Square, транзакции необоснованно сложны. Коммерсанту слишком трудно понять, как принимать платежи. В равной мере и покупателям сложно совершить платеж. В Square намеревались дать ответ на вопрос: «Как сделать транзакции простыми и легкими?»

Вот как они это делают. Продавцы могут подписаться на Square, загрузить приложение Register и получить считыватель кредитных карт по почте. Затем они могут подключить считыватель к телефону, открыть приложение и совершить платежи. Маленький пластиковый квадрат позволяет мелким предпринимателям (на самом деле не только мелким) принимать транзакции по кредитным картам. Местные хипстерские кофейни, похоже, были первопроходцами, если Портленд и Сан-Франциско что-то значат. Потребителям нужно просто предоставить свои кредитные карты. Они не будут испытывать ничего необычного, хотя и подписываются на iPad, а не на листе бумаги.

С помощью Square можно совершать покупки, используя гарнитуру громкой связи (hands-free). Когда покупатель решает заплатить через Square Wallet со своего телефона, его имя появляется в приложении Register продавца, и все, что должен сделать последний, — нажать данное имя.

Square хочет облегчить для продавцов подписку на собственные сервисы и принятие платежей. Конечно, вероятно, что кто-то может зарегистрироваться и попытаться злоупотребить данной службой. Поэтому в Square действуют очень осторожно, чтобы не потерять деньги на сделках с мошенниками или неудачных бизнес-моделях.

Проблема рисков

Создавая бесконфликтный опыт применения для покупателей и продавцов, Square должна относиться с особым вниманием к категории пользователей, которые злоупотребляют данным сервисом. Подозрительная или нежелательная активность, такая как мошенничество, не только подрывает доверие клиентов, но и является незаконной и оказывает отрицательное влияние на итоговую прибыль компании. Таким образом, создание надежной и высокоэффективной системы управления рисками выступает основой роста платежной компании.

Но каким образом Square удается эффективно отслеживать некорректное поведение? Ян объяснил это инвестированием в машинное обучение с разумной долей визуализации.

Обнаружение подозрительной активности с помощью машинного обучения. Начнем с вопроса: что значит подозрительный? Если мы посмотрим, скажем, на множество микротранзакций или на мгновенность, высокую частоту транзакций либо на изменчивую их частоту, то наши брови сами собой поползут вверх.

Приведем пример. Допустим, у Джона есть автокафе и через несколько недель после открытия он начинает проводить транзакции на сумму 1000 долларов через Square. (Одно возможное объяснение: Джон — сумасшедший, который накрывает гамбургеры золотыми листочками (https://mathbabe.org/2012/07/30/the-douche-burger-and-putting-a-ruler-to-the-dick/).) С одной стороны, если мы разрешим перевести деньги, то Square окажется в затруднительном положении в случае некорректного платежа. С технической точки зрения мошенник, которым в данном случае, вероятно, является Джон, будет нести ответственность, но наш опыт говорит о том, что обычно мошенники неплатежеспособны, поэтому все закончится тем, что Square придется оплатить счет.

С другой стороны, приостановка платежа, который окажется реальным, будет свидетельством плохого обслуживания клиентов. В конце концов, что, если Джон невиновен, а Square отклоняет оплату? Вероятно, Джон будет злиться на компанию

и может даже попытаться публично запятнать ее репутацию, но в любом случае его доверие будет утрачено.

Этот пример выявляет важнейшие проблемы, с которыми сталкивается Square: ложные срабатывания подрывают доверие пользователей, мошенничество стоит денег компании.

Если честно, то на самом деле есть *два* типа мошенничества, о которых нужно беспокоиться: мошенничество со стороны продавца и со стороны покупателя. Мы сосредоточимся на мошенничестве со стороны продавца, как было в истории с Джоном.

Поскольку Square обрабатывает продажи на миллионы долларов в день, то должна систематически и автоматически оценивать достоверность платежей. Нужно оценивать уровень риска для каждого события и субъекта в системе.

Так что же делает компания? Прежде чем начать, Ян сделал набросок части своей схемы данных, показанной на рис. 9.13, 9.14 и 9.15.

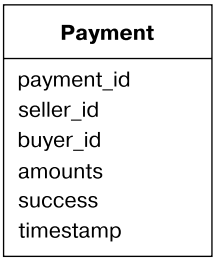


Рис. 9.13. Схема платежей

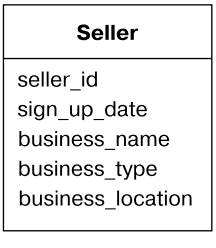


Рис. 9.14. Схема продаж

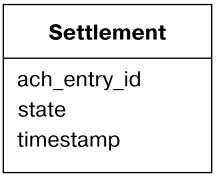


Рис. 9.15. Схема оплаты

Ниже представлено три типа данных:

- данные платежа, где мы можем присваивать значения полям `transaction_id`, `seller_id`, `buyer_id`, `amount`, `success` (`0` или `1`) и `timestamp` (id транзакции, id продавца, id покупателя, количество, успех и метка времени);
- данные продажи, где мы можем присваивать значения полям `seller_id`, `sign_up_date`, `business_name`, `business_type` и `business_location` (id продавца, дата регистрации, наименование предприятия, вид деятельности, место нахождения предприятия);
- данные оплаты, где мы можем присваивать значения полям `settlement_id`, `state` и `timestamp` (id оплаты, статус, метка времени).

Важно отметить, что Square рассчитывается со своими клиентами в течение всего дня после первоначальной транзакции, поэтому процесс не должен принимать решения за доли микросекунд. Компании, конечно, хотелось бы сделать это побыстрее, но в некоторых случаях имеется достаточно времени для того, чтобы позвонить по телефону и все проверить.

На рис. 9.16 представлен данный процесс: получая множество (порядка миллионов) событий для платежей и связанных с ними дат (как показано в схеме данных ранее), компания пропускает каждое из них через модели оценки риска, а затем некоторые, подозрительно выглядящие, отправляются на «ручной разбор». Затем команда оперативного отдела рассмотрит все случаи в индивидуальном порядке. В частности, все, что выглядит неприемлемым, передается в оперативный отдел, следящий за торговцами. Все одобренные транзакции оплачиваются (запись добавляется в таблицу `settlement`).

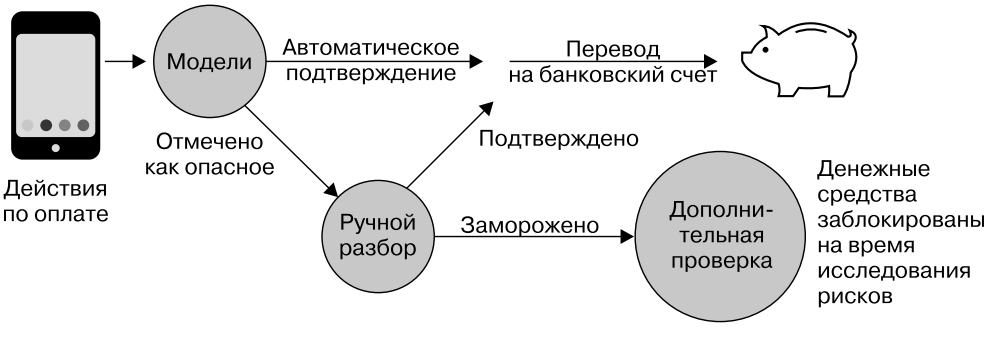


Рис. 9.16. Механизм оценки рисков

Учитывая предыдущий процесс, сфокусируемся на том, как в Square настраивают модели оценки рисков. Вы можете представить модель как функцию, сопоставляющую платежи и метки (например, «хорошая» или «плохая»). Подобная постановка вопроса звучит как задача обучения под наблюдением. И хотя этот способ решает *часть* данной задачи, все, конечно, не так просто — платеж не отклоняется, а затем просто остается с этой меткой, поскольку, как уже говорилось, отправляется команде оперативного отдела для независимой оценки. Таким образом, в действительности существует довольно сложный набор меток, который включает случаи, когда платеж первоначально был отклонен, но позже было решено, что все в порядке; или он был принят, но при дальнейшем рассмотрении мог оказаться некорректным; или его некорректность *подтвердилась*; или было подтверждено, что все в порядке; данный список можно продолжить.

Технически мы бы назвали это проблемой *обучения под частичным наблюдением*, охватывающей сферы обучения под наблюдением и неконтролируемого обучения. Но следует отметить, что «метка отказа» устанавливается через несколько месяцев, когда получено огромное количество возвратных платежей, вследствие чего данную проблему можно рассматривать как обучение под строгим наблюдением, если вернуться достаточно далеко назад по времени. Поэтому пока нельзя доверять меткам последних данных; в целях этого обсуждения Ян описывает более простой случай решения для контролируемой части.

Теперь, когда мы подготовили почву для указанной проблемы, Ян перешел к описанию способа обучения под наблюдением, как этому обычно учили в школе.

- Получить данные.
- Выявить признаки.
- Обучить модель.
- Оценить эффективность.
- Опубликовать модель!

Но перенести этот способ в реальный мир не так просто. На самом деле даже не ясно, является ли порядок верным. Ян выступает за то, чтобы в первую очередь подумать о *цели*, а это значит, что оценка эффективности переместится в верхнюю часть списка.

Проблема оценки эффективности

Итак, сделаем это: сфокусируемся на оценке эффективности. Прямо сейчас Ян определит три области, где можно столкнуться с проблемами.

Определение метрики ошибки

Как измерить, насколько хорошо смоделирована наша задача обучения? Вспомним о различных возможных случаях, используя таблицу истинности (табл. 9.1).

Таблица 9.1. Таблица сравнения значений прогноза с правильными, также называемая матрицей различий

	Actual = True	Actual = False
Predicted = True	TP (true positive)	FP (false positive)
Predicted = False	FN (false negative)	TN (true negative)

Наиболее простой метрикой эффективности является *Правильность* (Accuracy), которая определяется с помощью предшествующей записи как отношение:

$$Правильность = (TP + TN) / TP + TN + FP + FN.$$

С другой стороны, правильность можно рассматривать как вероятность того, что ваша модель получит верный ответ. С учетом очень малого количества положительных примеров мошенничества (по крайней мере, по сравнению с общим числом транзакций) правильность не является хорошей метрикой успеха, поскольку модель «все выглядит хорошо» (или эквивалентная модель «ничего не выглядит как мошенничество») ни о чем не говорит, но имеет высокую правильность.

Взамен мы можем оценить эффективность, используя *Точность* (Precision) и *Полноту* (Recall). Точность определяется как отношение:

$$Точность = TP / (TP + FP)$$

или вероятность того, что транзакция с меткой «мошенническая» является таковой на самом деле.

Полнота определяется как отношение:

$$Полнота = TP / (TP + FN)$$

или вероятность того, что мошенническая транзакция будет перехвачена моделью.

Решение о том, какая из этих метрик является оптимальной, зависит от затрат на не перехваченные некорректные транзакции, которые легко измерить, в сравнении с чрезмерным количеством приостановленных транзакций, которые намного сложнее подсчитать.

Определение меток

Метки — то, что Ян считал забытой половиной данных. При обучении с неполной статистикой и в соревнованиях по интеллектуальному анализу данных наличие меток часто воспринимается как нечто само собой разумеющееся. Однако на самом деле метки трудно определить и зафиксировать, в то же время они жизненно важны. Определение меток — не просто что-то связанное с целевой функцией; это и *есть* целевая функция.

В настройках Square определение метки подразумевает точное указание следующих параметров.

- Что считается подозрительной деятельностью?
- Каким должен быть уровень детализации? Событие или субъект (или и то и другое)?
- Можем ли мы надежно зафиксировать метку? Какие еще системы нужно интегрировать для получения этих данных?

Наконец, Ян кратко упомянул о том, что шум для метки может резко влиять на проблемы прогнозирования, вызывая возникновение *высокого уровня дисбаланса* (допустим, очень мало положительных примеров).

Проблемы, связанные с признаками и обучением

Ян сказал, что признаки классифицируют ваши знания предметной области. Как только процесс машинного обучения запущен и выполняется, бóльшую

ЧТО ТАКОЕ МЕТКА

Приведем еще один пример трудностей, связанных с метками. На конференции DataEDGE, проводимой ежегодно в Школе информации Калифорнийского университета в Беркли (UC Berkeley's School of Information), в разговоре с Майклом Чуи (Michael Chui) из Глобального института Маккинси (McKinsey Global Institute) Итамар Розенн (Itamar Rosenn), первый исследователь данных из Facebook (нанятый в 2007 году), описал трудности в определении «вовлеченного пользователя». Что значит *вовлеченный*? При желании предсказать, вовлечен ли пользователь, вам нужно иметь об этом некое понятие, если вы собираетесь указывать пользователей как вовлеченных или нет. Очевидное определение отсутствует, и на самом деле множество определений может работать в зависимости от контекста — нет основополагающей истины! Некоторые определения вовлеченности могут зависеть от частоты или ритма, с которыми пользователь заходит на сайт, или от того, насколько часто он создает или потребляет контент. Это проблема обучения под частичным наблюдением, в которой вы одновременно пытаетесь определить метки, а также предсказывать их.

часть энергии моделирования следует направить на поиск наилучших способов описания предметной области (то есть встает вопрос о новых признаках). Но вы должны знать, когда в действительности произойдет обучение этих признаков.

Точнее, если вы столкнулись с проблемой дисбаланса класса, то нужно проявить осторожность в отношении переобучения. Размер выборки, необходимый для изучения какого-либо признака, пропорционален размеру интересующей генеральной совокупности (которой в данном случае является класс «мошенников»).

Например, это может привести к работе с категориальными переменными с множеством категорий. Имеющийся почтовый индекс не является достаточной информацией, так как очень немногие нечестные продавцы предоставляют его. В данном случае вы можете захотеть использовать какую-нибудь умную сортировку почтовых индексов. Иногда Ян и его команда создают подмодель внутри модели лишь для того, чтобы уменьшить размер определенных признаков.

Второй вопрос, связанный с разреженностью данных, — это проблема *холодного запуска* для новых продавцов. У вас нет однотипной информации для всех ваших продавцов, в частности для тех, которые являются новыми. Но если вы будете чрезмерно осторожны, то рискуете произвести негативное впечатление на новых клиентов.

Наконец, и это типично для алгоритмов прогнозирования, вам необходимо настроить ваш алгоритм, чтобы он соответствовал постановке задачи, что в данном случае сродни поиску иглы в стоге сена. Например, вам следует подумать о том,

взаимодействуют ли признаки линейно или нелинейно, и о настройке обучения модели для учета дисбаланса класса: нужно ли настраивать вес для каждого класса? Как насчет плана выборочного контроля обучаемого в ансамбле (ensemble learner)?

Вы также должны знать о состязательном поведении, которое является способом сказать, что кто-то на самом деле пытается вам навредить. Пример из электронной коммерции: если злонамеренный покупатель выясняет, что вы выявляете мошенничество по разрешению адреса с помощью контроля точного совпадения строк, то может просто зарегистрироваться под десятью новыми учетными записями, в адресе каждой из которых будут содержаться небольшие орфографические отклонения от первоначального написания. Теперь вам нужно знать, как выделять эти варианты адресов, и предвидеть эскалацию состязательного поведения. Поскольку модели с течением временем ухудшаются по мере того, как люди учатся их обманывать, то вам необходимо постоянно отслеживать эффективность и переобучать свои модели.

Советы по построению моделей

Ниже представлено несколько полезных рекомендаций по созданию хороших производственных моделей.

- *Модели — это не черные ящики.* Вы не можете построить хорошую модель, предположив, что алгоритм сам со всем разберется. Например, вам нужно знать, почему вы неправильно классифицируете определенных людей, поэтому придется засучить рукава и заглянуть в модель с целью посмотреть, что произошло. Вы, по сути, создаете концепцию на основе ошибок.
- *Развивайте способность модели выполнять быстрые итерации.* Рассматривайте это как эксперименты, которые вы производили бы в научной лаборатории. Если вы не уверены, попробовать A или B, то попробуйте оба. Конечно, всему есть предел, но большую часть времени люди ошибаются из-за того, что «делают недостаточно».
- *Модели и пакеты не являются панацеей.* Если вы слышите, как некто говорит: «Так какие модели или пакеты вы используете?» — то вы столкнулись с тем, кто ничего не понимает в этом.

Ян заметил, что отправная точка многих дискуссий по компьютерному обучению так или иначе связана с тем, какой алгоритм или пакет используют люди. Например, если вы в R, то люди зацикливаются на том, применяете ли вы `randomForst`, `gbm`, `glmnet`, `caret`, `ggplot2` либо `rocr`; или в `scikit-learn` (Python), независимо от того, задействуете ли `RandomForestClassifier` или `RidgeClassifier`. Но все это деревья, за которыми не видно леса.

Читаемость кода и возможность повторного использования

Если речь идет не о моделях, тогда о чем речь на самом деле? Дело касается вашей способности однократно и повторно применять пакеты, чтобы иметь возможность легко менять любую из этих моделей. Ян призывает людей сосредоточиться на читаемости, повторном использовании, корректности, структуре и гигиене этих кодовых баз.

И если вы когда-нибудь углубитесь в работу настолько, что реализуете алгоритм самостоятельно, то приложите все усилия, чтобы создать понятный и расширяемый код. Если при написании алгоритма случайного леса вы жестко закодировали количество деревьев, то тем самым загоняете себя (и всех, кто использует этот алгоритм) в угол. Поместите там параметр, чтобы люди могли его повторно использовать. Сделайте его настраиваемым. Пойдите на компромисс. И пишите тесты, ради бога. Чистый код и ясность мысли идут рука об руку.

В Square стараются поддерживать повторное использование и читаемость, размещая структурированный код в разных папках с различными, многократно применяемыми компонентами, которые обеспечивают семантику разных частей построения модели машинного обучения:

- `Model` (Модель) — обучающие алгоритмы;
- `Signal` (Сигнал) — получение данных и расчет признаков;
- `Error` (Ошибка) — оценка эффективности;
- `Experiment` (Эксперимент) — сценарии для разведочного анализа данных и экспериментов;
- `Test` (Тест) — тестирование всего.

Они только пишут сценарии в папке эксперимента, где либо связывают компоненты модели, сигнал и ошибку, либо проводят разведочный анализ данных. Каждый раз, когда они пишут сценарий, это больше, чем просто фрагмент кода, который предадут забвению. Это эксперимент, к которому возвращаются снова и снова, чтобы найти новые закономерности.

Что дает такой строгий подход? С каждым запуском эксперимента ваши знания должны постепенно расти. В противном случае эксперимент бесполезен. Этот подход помогает убедиться в отсутствии повторного выполнения той же работы. Без этого вы даже не сможете понять, что вы или кто-то еще попытались сделать. Ян далее утверждает: «Если вы не пишете производственный код, то непродуктивны».

Более подробно о том, что должен содержать каждый каталог проекта, см. в Project Template (http://projecttemplate.net/) Джона Майлса Уайта (John Myles White). Тем студентам, которые используют R для своих классов, Ян рекомендует исследовать и активно читать GitHub-репозиторий кода для R. Он советует попытаться

написать собственный R-пакет и обязательно прочитать страничку Хэдли Уикхема (Hadley Wickham) в «Википедии», посвященную инструментальным средствам разработки (devtools wiki) (http://adv-r.had.co.nz/). Кроме того, Ян говорит, что развитие эстетического чувства для кода аналогично приобретению вкуса к красивым доказательствам; это достигается с помощью строгой практики и обратной связи с коллегами и наставниками.

Дополнительно Ян рекомендует сравнить реализации пакета caret package (https://cran.r-project.org/web/packages/caret/index.html) с scikit-learn (https://github.com/scikit-learn/scikit-learn). Который из них является более расширяемым и многоразовым? Почему?

Получите оба!

Научиться правильно писать код — сложная задача. И еще труднее разобраться в этом одному. Представьте, что пытаетесь выучить испанский (или ваш любимый язык) и не имеете возможности практиковаться с другим человеком.

Найдите партнера или команду, которые будут готовы к совместному использованию программы и проведению строгих разборов кода. В Square каждый отдельный фрагмент кода просматривается по крайней мере еще одной парой глаз. Это важный способ не только проверить ошибки, но и обеспечить совместное использование кода и высокий уровень качества.

Как только вы найдете соратника по программированию, попробуйте сделать следующее. Определите общую проблему. Настройте рабочую станцию с одним монитором и двумя наборами из клавиатуры и мыши. Подумайте об этом как о совместной работе по решению проблемы: сначала обсудите общую стратегию решения проблемы, а затем перейдите к его фактической реализации. Вы оба по очереди будете оператором или наблюдателем. Оператор пишет код, а наблюдатель его просматривает и разрабатывает план действий. В то время как оператор занят набором на клавиатуре, наблюдателю следует постоянно спрашивать себя: «Я понимаю этот код?» и «Как сделать код понятнее?». Когда возникает путаница, найдите время, чтобы совместно выяснить недоразумения (или даже устранить отсутствие понимания). Будьте открыты для того, чтобы учиться и учить. Вы быстро соберете крупицы знаний, начиная с горячих клавиш редактора и заканчивая логически последовательной организацией кода.

Роли оператора и наблюдателя нужно периодически менять в течение дня. Если все будет сделано правильно, то вы почувствуете себя истощенными через несколько часов. Практикуйте, чтобы улучшить выносливость при работе в паре.

И когда вы не можете трудиться над программой в паре с кем-либо, выработайте привычку проверять код с помощью Git. Узнайте о последовательности выполняемых действий Git и отправляйте напарнику конструктивную критику через

запросы на включение кода (pull requests). Рассматривайте это как экспертный обзор в научных кругах.

Создание моделей машинного обучения

Ниже представлены некоторые из самых сложных проблем при ведении компьютерного обучения в реальном мире.

1. Как «создается» модель?
2. Как для данных моделей осуществляется расчет признаков в реальном времени?
3. Как мы убеждаемся в том, что «видимое нами есть то, что мы получим»? То есть минимизируем расхождения между эффективностью в автономном режиме и при подключении к сети.

На большинстве курсов и соревнованиях по машинному обучению прогностические модели противопоставляются статическому набору отложенных данных, которые умещаются в памяти, и модели доступны до тех пор, пока необходимо их выполнение. При наличии таких слабых ограничений разработчики моделей довольствуются тем, что берут данные, выполняют операцию за время `O(n^3)` для получения признаков и пропускают их через модель. Комплексное проектирование признаков часто приветствуется в педагогических целях. Ниже представлен ряд примеров.

- Большая размерность? Не о чем волноваться, мы просто выполним сингулярное разложение, сохраним матрицы преобразования и умножим их на отложенные данные.
- Преобразование часто поступающих данных? Подождите, дайте сначала настроить модель Пуассона в соответствии с ранее полученными данными и набором отложенных данных.
- Временн*ы*е ряды? Введем некоторые коэффициенты Фурье.

К сожалению, в реальной жизни мы не располагаем таким временем и пространством. Прогностические модели сталкиваются с постоянно растущим набором данных, поступающих онлайн. Во многих случаях ожидается получение прогнозов в течение миллисекунд после передачи данных. Вся эта сложная деятельность по созданию модели не имеет смысла, если та не сможет обработать трафик.

Имейте в виду, что многие модели сводятся к скалярному произведению характеристик с весами (обобщенные линейные модели (GLM), метод опорных векторов (SVM)) или к ряду конъюнкций с пороговыми значениями, которые можно выразить в виде поиска в массиве или блока операторов if-else по отношению к характеристикам (деревья решений). Таким образом, сложная часть упрощается до вычисления признаков.

Существуют различные подходы к вычислению признаков. В зависимости от сложности модели, времени задержек и требований к объему признаки вычисляются как в пакете, так и в реальном времени. Модели могут выбирать для использования только признаки в пакете, только признаки реального времени или и то и другое. В некоторых случаях модель реального времени задействует признаки в режиме реального времени плюс выходные данные моделей, обучаемых с помощью пакетов.

Такие фреймворки, как MapReduce, часто используются для вычисления признаков в пакетах. Но при более строгих требованиях к времени задержек Ян и команда машинного обучения в Square работают в системе для вычисления признаков в реальном времени.

Важная цель проектирования подобной системы — обеспечение того, чтобы характеристики, полученные ранее и онлайн, вычислялись аналогичным образом. Другими словами, между такими характеристиками не должно быть систематических расхождений. Создателям моделей следует быть уверенными в том, что эффективность моделей при работе онлайн соответствует ожидаемой.

Визуализация данных в Square

Затем Ян рассказал о том, для каких целей в Square команда, осуществляющая контроль рисков, использует визуализацию. Это:

- включение эффективного контроля;
- выявление закономерностей для отдельных клиентов и сегментов клиентов;
- оценка здоровья предприятия;
- обеспечение анализа среды.

Ян описал инструмент управления последовательностью выполняемых действий для обзора пользователей, демонстрирующих признаки продавца, включая историю продаж и географическую информацию, обзоры, контактную информацию и др. Подумайте о контроле выполнения операции. Указанный инструмент — не что иное, как вид визуализации данных. Оперативной группе, которой поручено проверить подозрительную деятельность, отведено ограниченное время для выполнения работы, поэтому создатели и исследователи данных должны сотрудничать с командой оперативного отдела, чтобы выяснить способы наилучшего представления данных. С помощью этих визуализаций они пытаются повысить интеллект оперативной команды, создать экзоскелет для выявления закономерностей подозрительной деятельности. Ян считает, что именно здесь появится много интересных разработок — органичный союз машинного обучения и визуализации данных.

Визуализации различных сегментов клиентов часто отображаются на различных телевизорах в офисе (в сообществе Square ласково называемых информационными излучателями). Эти визуализации не обязательно пытаются предсказать мошенничество как таковое, а скорее обеспечивают способ наблюдения, позволяющий находить тенденции и закономерности в динамике.

Это связано с концепцией анализа окружения (ambient analytics), которая заключается в обеспечении среды для постоянного и пассивного получения данных с целью дать вам возможность с их помощью развить внутреннее чутье. В конце концов, благодаря тому, что мы очень хорошо знакомы с нашими данными, мы иногда узнаем, какие закономерности необычны или какие сигналы заслуживают собственной модели или монитора. Команда по контролю рисков Square приложила много усилий для разработки настраиваемых и обобщаемых информационных панелей.

Помимо необработанных транзакций, есть показатели риска, за которыми Ян внимательно следит. Например, он ежедневно отслеживает явные и замороженные показатели, а также то, сколько событий нужно просмотреть. Используя сложную систему наблюдения, он может перейти к тому, что аналитики заморозили сегодня, сколько времени потребовалось на просмотр каждой учетной записи и на какие факторы указывает длительный процесс просмотра.

В целом сотрудники Square — большие поклонники визуализации бизнес-показателей (регистрации, активации, активных пользователей) на информационных панелях. Они считают, что прозрачность ведет к большей подотчетности и вовлеченности. Эти люди постоянно делают своего рода ЭКГ их бизнеса в рамках глобальной аналитики. Группа по контролю рисков, в частности, является твердым сторонником идеи «тем, что поддается измерению, можно управлять».

Ян закончил представлять свой профиль исследователя данных и напоследок дал несколько советов. Он думает, что график `plot(skill_level ~ attributes | ian)` следует отображать с помощью логарифмической шкалы, поскольку на то, чтобы начать хорошо в чем-то разбираться, уйдет не слишком много времени, чего не скажешь о процессе «из хорошего специалиста стать прекрасным». Ян считает: измерять производительность также нужно по логарифмической шкале, и его аргумент заключается в том, что ведущие разработчики программного обеспечения выдают пакеты гораздо более высокими темпами, чем другие люди.

И напоследок Ян призывает вас:

- экспериментировать с реальными данными;
- заложить основы математики, статистики и информатики в школе;
- пройти стажировку;
- быть грамотными не только в статистике;
- оставаться любопытными!

Мысленный эксперимент Яна

Предположим, вы узнаете о каждой одиночной транзакции, как только она происходит. Как вы примените эти данные?

Визуализация данных для остальной части

Не все из нас могут создавать визуализации данных, считающиеся произведениями искусства, достойными музеев. Однако стоит развивать свою способность использовать визуализацию для общения, рассказывания историй и передачи заложенного в данных смысла. Визуализация данных, как и наука о данных, — это нечто большее, чем набор инструментов, но для того чтобы стать мастером, нужно сначала освоить технику. Ниже приведены уроки и книги, которые, как мы считаем, помогут росту ваших навыков в визуализации данных.

- Неплохая ознакомительная лекция о конструктивных элементах визуализации данных от Михаила Дубокова (Michael Dubokov) на https://www.targetprocess.com/articles/visual-encoding/.
- У Натана Яу, который был аспирантом Марка Хансена в Калифорнийском университете в Лос-Анджелесе, есть сборник руководств по созданию визуализаций в R, расположенный по адресу http://flowingdata.com/. Натан Яу также издал две книги: *Visualize This: The Flowing Data Guide to Design, Visualization, and Statistics* (Wiley)[1] и *Data Points: Visualization That Means Something* (Wiley) («Точки ввода данных: осмысленная визуализация»).
- Скотт Мюррей (Scott Murray), художник в области кодирования, представляет серию обучающих программ, описывающих d3, по веб-адресу http://aligned-left.com/tutorials/d3/. Они были разработаны в книге Interactive Data Visualization («Интерактивная визуализация данных») (O'Reilly) (http://shop.oreilly.com/product/0636920026938.do).
- Хэдли Уикхем, разработавший пакет ggplot2 в R на основе Grammar of Graphics («Грамматика графики») Леланда Уилкинсона (Leland Wilkinson) (https://www.amazon.com/Grammar-Graphics-Statistics-Computing/dp/0387245448), написал соответствующую книгу *ggplot2: Elegant Graphics for Data Analysis (Use R!)* («ggplot2: Элегантная графика для анализа данных (Используйте R!)») (Springer).
- Классикой визуализации данных считаются несколько книг. Например, *The Visual Display of Quantitative Information* («Визуальное представление больших объемов информации» (Graphics Pr)) Эдварда Тафти (Edward Tufte). Он статистик, широко известный как один из основоположников визуализации данных;

[1] На русском языке книга была выпущена в 2013 году под названием «Искусство визуализации в бизнесе: Как представить сложную информацию простыми образами» («Манн, Иванов и Фербер»). — *Примеч. пер.*

мы знаем, что уже говорили это о Марке Хансене — они из разных поколений. Эдвард делает меньший акцент на инструментах и больший — на принципах хорошего дизайна. Кроме того, Уильям Кливленд (о котором мы упоминали в главе 1 в связи с его предложением о расширении области статистики в науку о данных) написал две книги: *Elements of Graphing Data* («Элементы графической обработки данных») (Hobart Press) и *Visualizing Data* («Визуализация данных») (Hobart Press).

- Последние книги издательства O'Reilly: *R Graphics Cookbook* («Основы графики в R») (http://shop.oreilly.com/product/0636920023135.do), *Beautiful Data* («Красивые данные») (http://shop.oreilly.com/product/9780596157128.do) и *Beautiful Visualization* («Красивая визуализация») (http://shop.oreilly.com/product/0636920000617.do).
- Нельзя не упомянуть, что в художественных школах есть отделы графического дизайна и книги, посвященные принципам дизайна. Образование в области визуализации данных, которое не учитывает это, как и принципы журналистики и повествования, а фокусируется только на инструментах и статистике, даст лишь половину картины, не говоря уже о психологии человеческого восприятия.
- Доклад Брета Виктора (Bret Victor) (http://hci.stanford.edu/courses/cs547/speaker.php?date=2013-02-01) *Describing Dynamic Visualizations* («О динамических визуализациях») (https://vimeo.com/66085662) настоятельно рекомендует Джефф Хир (Jeff Heer), профессор Стэнфордского университета, который создал d3 совместно с Майклом Бостоком (Michael Bostock) (ранее работал в Square, а теперь перешел в The New York Times). Джефф описал этот доклад как представление альтернативного взгляда на визуализацию данных.
- Сотрудничайте с художником или графическим дизайнером!

Упражнение по визуализации данных. Студенты курса, как и вы, читатели, имели разные уровни подготовки и опыта в визуализации данных, поэтому Рэйчел предложила тем, кто чувствовал себя новичками, выбрать два учебника Натана Яу и выполнить задания, а затем поразмышлять о том, помогло ли это и что они хотели бы сделать, чтобы улучшить навыки в визуализации.

Более продвинутым студентам предоставили возможность участвовать в соревнованиях по визуализации данных в Hubway (http://hubwaydatachallenge.org/). Hubway — это программа для обмена велосипедами в Бостоне, где был опубликован набор данных и проведен конкурс по их визуализации. Набор данных по-прежнему доступен, так почему бы не попробовать? Два ученика в классе Рэйчел, Эври Ким (Eurry Kim) и Каз Сакамото (Kaz Sakamoto), выиграли в номинации «Лучший рассказ о данных» на конкурсе; Рэйчел очень гордится ими. Пропущенный через призму романтических отношений, их визуальный дневник (рис. 9.17) отображает ревизию первых 500 000 совместных поездок жителей Бостона.

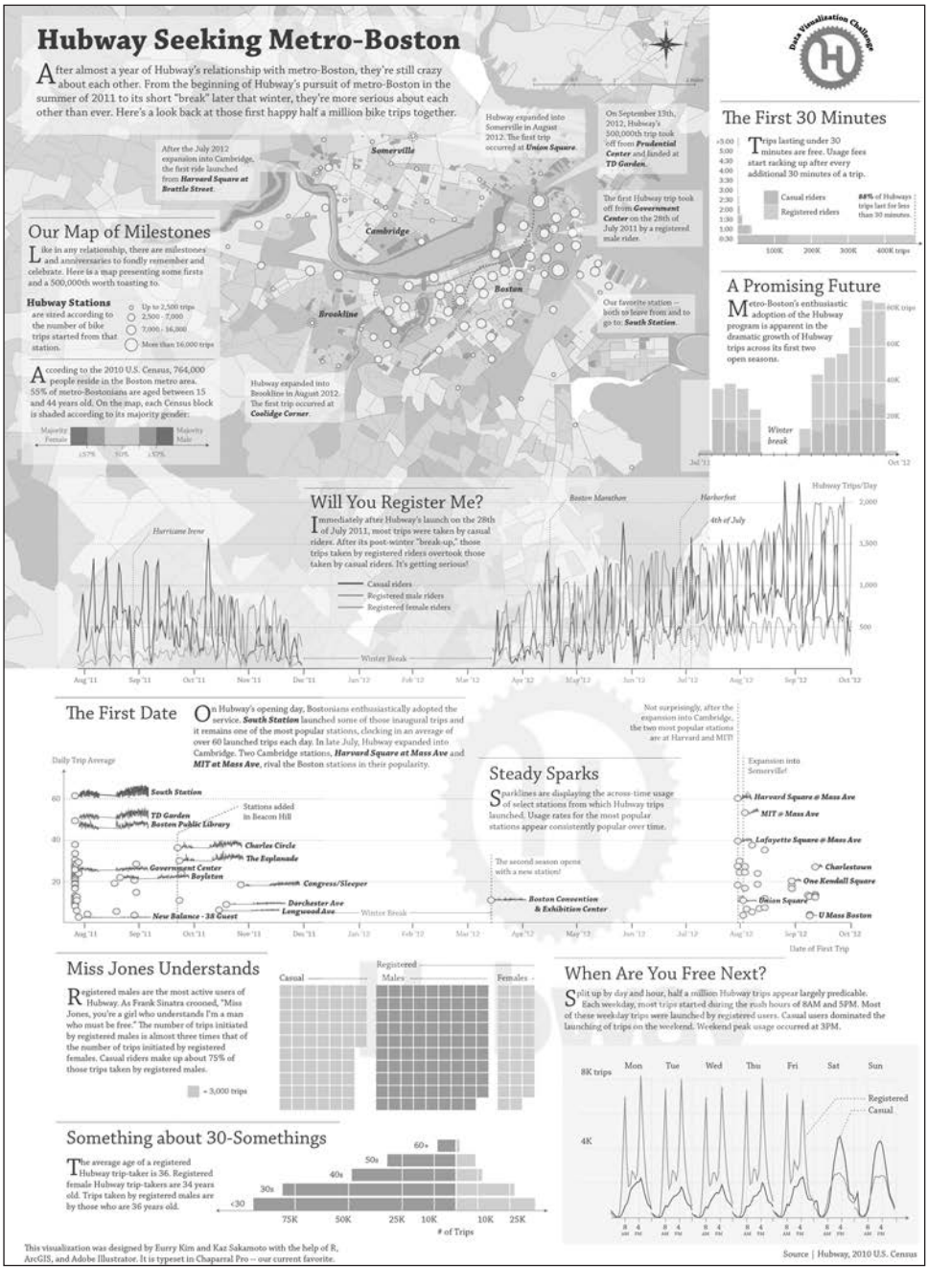


Рис. 9.17. Визуализация от Эври Кима и Каз Сакамото программы по обмену велосипедами Hubway и ее освоения замечательными людьми в районе бостонского метро

10 Социальные сети и журналистика данных

В этой главе мы рассмотрим две темы, которые стали особенно актуальными за последние 5–10 лет: социальные сети и журналистика данных. Социальные сети (не обязательно *онлайновые*) изучались кафедрами социологии на протяжении десятилетий, равно как и их коллегами в области информатики, математики и статистики — теории графов. Однако появление социальных сетей в Интернете, таких как Facebook, LinkedIn, Twitter и Google+, дало новый богатый источник данных, который открывает множество исследовательских проблем, как с точки зрения общественных наук, так и с количественной/технической.

Сначала мы услышим о том, как одна компания, Morningside Analytics, визуализирует и находит смысл в данных социальной сети, а также о некоторых из основополагающих теорий социальных сетей. Мы взглянем на построение историй, которые могут быть рассказаны на основе данных социальной сети, что является формой журналистики данных. Размышляя о профилях исследователей данных (и в этом случае генная экспрессия является подходящей аналогией), можно отметить, что сочетания математики, статистики, информационного взаимодействия, визуализации и программирования, необходимые для работы в области науки о данных или журналистики данных, несколько отличаются, но основные навыки те же. В основе обеих областей лежит способность задавать хорошие вопросы, отвечать на них с помощью данных и обмениваться своими находками. В связи с этим мы кратко коснемся журналистики данных с точки зрения Джона Брунера (Jon Bruner), редактора O'Reilly.

Анализ социальных сетей в Morning Analytics

Первым рассказчиком в этой главе является Джон Келли (John Kelly) из Morningside Analytics, который поговорит с нами о сетевом анализе.

У Келли есть четыре диплома, начиная с диплома бакалавра, полученного в 1990 году в Колумбийском колледже, а затем магистра, магистра философии и доктора философии Школы журналистики Колумбийского университета (Columbia's School of Journalism), где сосредоточился на сетевой социологии и статистике в области политологии. Келли также прошел несколько курсов в Стэнфорде, изучая дизайн и теорию игр и др. Он занимался магистерской работой с Марком Смитом (Marc Smith) из Microsoft (http://videolectures.net/marc_smith/). Темой они избрали то, как политические дискуссии развиваются в виде информационных сетей. После колледжа и до окончания школы Келли был художником, который пользовался компьютерами для создания звукового дизайна. Он три года работал в качестве директора по цифровым СМИ в Школе искусств Колумбийского университета (Columbia School of the Arts) (https://arts.columbia.edu/). Кроме того, Келли программист: он самостоятельно изучил Perl и Python, когда провел год во Вьетнаме вместе со своей женой.

Келли рассматривает математику, статистику и информатику (в том числе машинное обучение) в качестве инструментов, которые ему следует использовать и которые должны быть достаточно хороши для того, чтобы делать то, что он действительно хочет. Как шеф-повару на кухне, ему нужны хорошие кастрюли и сковородки и острые ножи, но настоящий продукт — это еда.

Так что же он преподносит на своей кухне? Келли хочет понять, как люди сходятся вместе и, когда делают это, какое влияние оказывают на политику и общественное мнение. В числе клиентов его компании, Morningside Analytics, аналитические центры и политические организации. Они, как правило, хотят знать, как социальные сети влияют на политику и порождают ее.

Информационное посредничество и презентации — то, на чем Келли зарабатывает деньги. Визуализации являются неотъемлемой частью как знаний предметной области, так и информационного взаимодействия; таким образом, его опыт заключается в сочетании визуализаций и извлечения из них выводов. В конце концов, Morningside Analytics не получает плату просто за то, что находит интересную информацию, а скорее помогает людям использовать ее.

Случайно-атрибутивные данные и данные социальных сетей. Келли не моделирует данные стандартным способом с помощью случайно-атрибутивных данных. Понятие «случайно-атрибутивный» связано с тем, что обычно создаются модели с различными «случаями», которые могут относиться к людям или событиям, у каждого из которых есть различные «атрибуты», могущие иметь отношение к возрасту, операционной системе или истории поиска.

Моделирование с помощью случайно-атрибутивных данных началось в 1930-х годах с ранних исследований рынка и вскоре было применено как к маркетингу, так и к политике.

Келли отмечает, что существует огромное смещение в сторону моделирования со случайно-атрибутивными данными. Одно из объяснений этого смещения — ин-

формацию в подобном виде легко хранить в базах данных или *собирать*. Келли считает, что в любом случае таким образом мы упускаем из виду суть вопросов, на которые пытаемся ответить.

Он упомянул Пола Лазарсфельда (Paul Lazarsfeld) (https://ru.wikipedia.org/wiki/Лазарсфельд,_Пол) и Элиу Каца (Elihu Katz) (https://en.wikipedia.org/wiki/Elihu_Katz), двух социологов-новаторов, приехавших из Европы и разработавших *анализ социальных сетей*, подход, основанный не только на отдельных людях, но и на отношениях между ними.

Чтобы понять, почему анализ сетей иногда превосходит анализ случайно-атрибутивных данных, подумайте о следующем примере. Федеральное правительство потратило деньги на опрос людей в Афганистане. Идея заключалась в следующем: увидеть, чего хотят граждане, чтобы предвидеть события в будущем. Но, как указывает Келли, грядущее происходящее не является простой функцией от того, что думают люди; на самом деле вопрос в том, кто имеет власть и что думают *эти люди*.

Аналогично представьте себе перемещение в прошлое и проведение научного опроса граждан Европы в 1750 году для определения будущей политики. Если бы вы знали, что делаете, то смотрели бы на то, кто с кем заключал браки в королевской семье.

В некотором смысле текущая фокусировка на случайно-атрибутивных данных — проблема поиска чего-то «под фонарем»: своего рода отклонение при проведении наблюдений, когда люди привыкли совершать некие действия определенным (зачастую более простым) способом, вследствие чего продолжают делать это, даже если не получают ответы на интересующие их вопросы.

Келли утверждает, что мир — это сеть, которая намного больше, чем множество случаев с атрибутами. Если вы понимаете только поведение отдельно взятых людей, то как устанавливаете взаимосвязи?

Анализ социальных сетей

Этот анализ происходит из двух источников: теории графов (https://ru.wikipedia.org/wiki/Теория_графов), в которой Леонард Эйлер (Leonhard Euler) (https://ru.wikipedia.org/wiki/Эйлер,_Леонард) решил задачу о семи кенигсбергских мостах (https://ru.wikipedia.org/wiki/Задача_о_семи_кёнигсбергских_мостах), и социометрии (https://ru.wikipedia.org/wiki/Социометрия), основанной Якобом Морено (Jacob Moreno) (https://ru.wikipedia.org/wiki/Морено,_Якоб_Леви) в 1970-х годах, когда ранние компьютеры получили хорошие результаты для крупномасштабных вычислений на больших наборах данных.

Анализ социальных сетей был основан Харрисоном Уайтом (Harrison White), заслуженным профессором Колумбийского университета, одновременно с коллегой по университету социологом Робертом Мертоном (Robert Merton). Их идея

заключалась в том, что действия людей должны быть связаны с их атрибутами, но для того, чтобы действительно их понимать, вам необходимо рассматривать и сети (так называемые системы), которые *позволяют им совершать некие действия*.

Как мы привносим эту идею в наши модели? Келли хочет, чтобы мы рассматривали явление, которое он называет разрывом между «микро» и «макро» или индивидуальным и системным: как мы преодолеем этот разрыв? Вернее, как он преодолевается в разных контекстах?

Например, в США есть формальные механизмы для преодоления этих микро- и макроразрывов, а именно, рынки в случае разногласий при покупке товаров и выборы для политических разногласий. Но большая часть мира не владеет этими формальными механизмами, хотя часто имеет фиктивную тень этих вещей. По большей части нам нужно иметь достаточные сведения о реальной социальной сети, чтобы знать, кто имеет власть и влияние и может вносить изменения.

Терминология из социальных сетей

Базовые единицы сети называются *акторами* либо *узлами*. Они могут быть людьми, или сайтами, или любыми «вещами», которые вы рассматриваете, и часто указываются как одна точка при визуализации. Отношения между акторами называются *реляционными связями* или *ребрами*. Например, событие, связанное с тем, что кто-то поставил кому-то лайк или стал другом, будет показано ребром. Мы говорим о парах акторов как о *диадах*, а о тройках акторов — как о *триадах*. Например, при наличии ребра между узлами A и B и ребром между узлами B и C *триадным замыканием* будет существование ребра между узлами A и C.

Иногда мы рассматриваем *подгруппы*, также называемые *подсетями*, которые состоят из подмножества целого набора акторов вместе с их реляционными связями. Конечно, это значит, что мы принимаем во внимание и *группу* как таковую, то есть всю совокупную «сеть». Обратите внимание: это относительно простая концепция применительно, скажем, к сети Twitter, но в случае «либералов» возникают значительные сложности.

Мы говорим об *отношении* в основном как о способе иметь реляционные связи между акторами. Например, лайкнуть другого человека — это отношение, но это кем-то было принято. *Социальная сеть* — совокупность некоего набора акторов и отношений.

На самом деле существует несколько разных типов социальных сетей. Например, самый простой случай: у вас есть группа акторов, соединенных связями. Это конструкция, которую вы будете использовать для отображения графика Facebook: любые два человека являются друзьями или нет и любые два человека теоретически могут быть друзьями.

В *двудольных графах* соединения существуют только между двумя формально отделенными классами объектов. Так, одним классом объектов могут быть люди, а другим — компании и людей можно связать друг с другом по признаку работы на компанию. Или у вас могут быть люди и вещи, которые способны их заинтересовать, и если это действительно так, то их можно соединить.

Наконец, существуют *эго-сети*, обычно формируемые как «часть сети, окружающая одного человека». Например, это может быть «подсеть моих друзей в Facebook», которые также могут знать друг друга в определенных случаях. Исследования показали, что люди с более высоким социально-экономическим статусом имеют более сложные эго-сети, и вы можете определить личный уровень социального статуса, взглянув на свою эго-сеть.

Показатели центральности

Первый вопрос, который люди часто задают при предоставлении социальной сети: *кто здесь значимый?*

Конечно, есть разные способы быть значимым, и различные определения, пытающиеся зафиксировать что-то вроде значимости, которые приводят к различным *показателям центральности*. Здесь мы приводим ряд часто используемых примеров.

Итак, есть понятие *степени*. Используется для подсчета количества людей, связанных с вами. Таким образом, на языке Facebook это количество друзей, которые у вас есть.

Затем у нас есть концепция *близости*: буквально, если вы «близки» со всеми, то у вас должен быть высокий показатель близости.

Для большей точности нам нужно иметь представление о расстоянии между узлами в *связном графе*; в случае сети друзей это значит, что все связаны со всеми остальными через цепочку общих друзей. Расстояние между узлами x и y, обозначаемое $d(x, y)$, определяется просто как длина кратчайшего пути между двумя узлами. Теперь, когда у вас есть это обозначение, вы можете определить близость узла x как сумму:

$$C(x) = \Sigma 2^{-d(x,y)},$$

где суммирование производится по всем узлам y, отличным от x.

Далее, существует показатель центральности, называемый *посредничеством*, который измеряет степень, в которой люди в вашей сети знают друг друга через вас, или, точнее, будут ли кратчайшие пути между ними проходить через вас. Идея здесь заключается в том, что если у вас есть высокий уровень посредничества, то информация, вероятно, протекает через вас.

Точнее, для любых двух узлов x и y в одной и той же связанной части сети определим $\sigma_{x,y}$ как *число кратчайших путей между узлами x и y*, а $\sigma_{x,y}(v)$ — как *число кратчайших путей между узлами x и y, проходящих через третий узел v*. Тогда показатель посредничества в зависимости от v определяется как сумма:

$$B(v) = \Sigma\, \sigma_{x,y}(v) / \sigma_{x,y},$$

где суммирование осуществляется по всем различным парам узлов x и y, которые отличаются от v.

Последний показатель центральности, который мы подробно рассмотрим в подразделе «Представление сетей и характеристическое число центральности» раздела «Дополнительные сведения об анализе социальных сетей с точки зрения статистики» текущей главы после введения понятия матрицы смежности, называется *характеристическим числом центральности*. Буквально, *у популярного человека, имеющего популярных детей,* это число высоко. PageRank от Google является примером такого показателя центральности.

Индустрия показателей центральности

Важно сделать оговорку о слепом применении предыдущих показателей центральности. А именно: «любители измерений» образуют индустрию, в которой каждый пытается продать себя как *авторитет*. Но опыт говорит нам, что у каждого есть свои слабые и сильные стороны. Главное — выбрать нужную сеть или подсеть.

Например, если вы изобразите 100 лучших блогеров на каком-либо крупном графе и в поисках влиятельного блогера из «Братьев-мусульман» начнете с верхней части списка, а затем спуститесь вниз, то это не сработает: вы найдете тех, кто влиятелен в большой сети и ведет блоги для «Братьев-мусульман», но они будут иметь влияние не на «Братьев-мусульман», а скорее на транснациональную элиту из более крупной сети. Другими словами, вы должны иметь в виду локальную окрестность графика.

С показателями центральности связана другая проблема: опыт говорит о том, что разные контексты требуют разных инструментов. Некие утилиты могут работать с блогами, но для работы с данными из Twitter вам нужно получить нечто совершенно другое.

Одной из причин тому являются разные данные, но другая — это различные способы, которыми люди обманывают показатели центральности. Например, с помощью Twitter люди создают 5000 ботов Twitter, подписывающихся друг на друга, а некоторые по определенной стратегии выбирают других (реальных) людей, чтобы заставить их выглядеть влиятельными согласно какому-либо показателю (возможно, характеристическому числу центральности). Но, конечно, это не точные данные; просто кто-то предоставляет неверные сведения.

Некоторые сетевые пакеты уже существуют и могут вычислять различные показатели центральности, упомянутые ранее. Например, смотрите NetworkX (http://www.lanl.gov/errors/system-notification.php); или igraph (http://igraph.org/redirect.html), если используете Python; или statnet (http://statnet.org/) для R; или NodeXL (https://www.microsoft.com/en-us/research/project/nodexl-network-overview-discovery-and-exploration-in-excel/?from=http%3A%2F%2Fresearch.microsoft.com%2Fen-us%2Fprojects%2Fnodexl%2F), если предпочитаете Excel. И наконец, следите за появлением пакета для C от Юре Лесковеца (Jure Leskovec) из Стэнфорда (https://cs.stanford.edu/people/jure/).

Мысленный эксперимент

Вы работаете в элитарном, хорошо финансируемом мозговом центре в Вашингтоне. Вы можете нанять людей и потратить 10 млн долларов. Ваша задача — эмпирически предсказать будущую политическую ситуацию в Египте. Какие политические партии будут в стране? Как Египет будет выглядеть через 5, 10 или 20 лет? У вас есть доступ к двум из следующих наборов данных для всех египтян: сети Facebook или Twitter, полных записей о том, кто с кем учился в школе, текстовые или телефонные записи всех, адреса всех или сетевые данные о членах всех формальных политических организаций и частных компаний.

Прежде чем вы приступите к анализу, имейте в виду, что со временем ситуация изменится — люди могут выйти из Facebook или политические обсуждения, вероятно, должны будут уйти в подполье, если блоги окажутся слишком публичны. Кроме того, Facebook дает много информации, но иногда люди будут пытаться скрыться — возможно, именно те, в слежке за которыми вы больше всего заинтересованы. По этой причине телефонные записи могут быть более репрезентативными.

Если вы считаете данный сценарий амбициозным, то должны знать, что он уже был реализован. Например, компания Siemens из Германии продала программное обеспечение Ирану для мониторинга иранских национальных мобильных сетей (https://en.wikipedia.org/wiki/Nokia_Networks#Iran_monitoring_controversy). Фактически правительства, по большому счету, вкладывают больше энергии в налаживание связей со своими союзниками и меньше — в закрытие территорий: Пакистан нанимает американцев для ведения своего пропакистанского блога, а россияне помогают сирийцам.

И последнее: вы должны рассмотреть возможность изменения стандартного направления своего мышления. Многие люди спрашивают: «Что мы можем узнать из того или иного источника данных?» Взамен подумайте об этом по-другому: что означает предсказание политики в обществе? И какие данные вам нужно иметь, чтобы сделать это?

Другими словами, сначала выясните суть вопросов, а затем ищите данные, которые помогут на них ответить.

Morningside Analytics

Келли показал нам карту сети из 14 крупнейших в мире сообществ блогеров (блогосфер — blogosphere). Чтобы понять рисунки, представьте себе, что есть сила, подобная ветру, которая отбрасывает узлы (блоги) к краю, но тогда есть противодействующая сила, а именно связи между блогами, скрепляющие их друг с другом. На рис. 10.1 показан пример арабского сообщества блогеров.

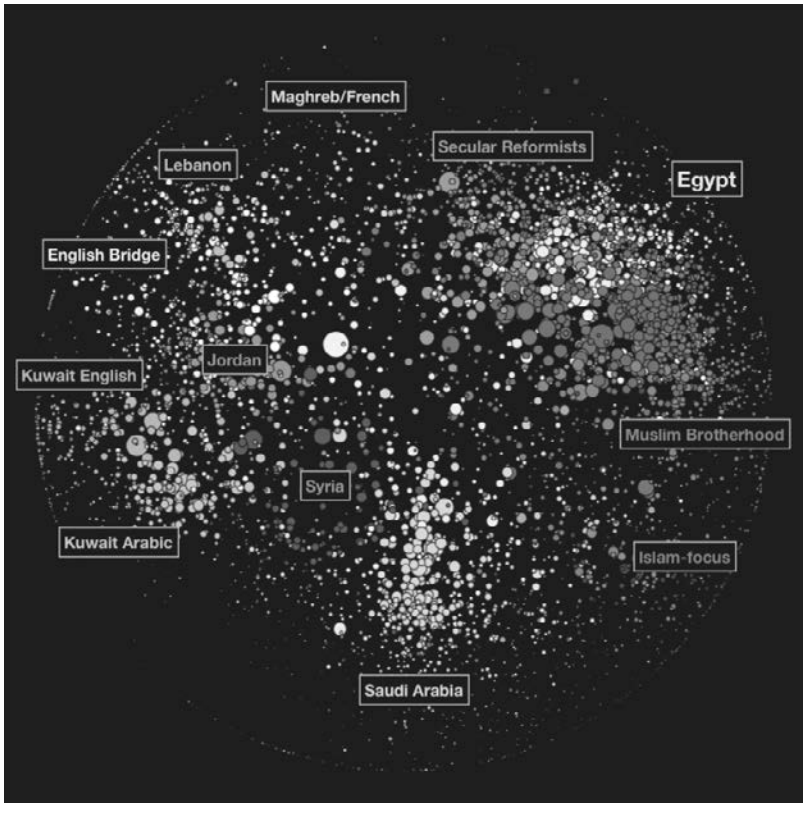


Рис. 10.1. Пример арабской блогосферы

Различные цвета представляют страны и кластеры блогов. Размер каждой точки — это центральность в зависимости от степени, то есть количество ссылок на другие блоги в сети. Физическая структура блогосферы может дать нам представление.

Если мы анализируем текст с помощью обработки текстов на естественном языке (natural language processing, NLP), думая о публикациях в блоге как о нагромождении текста или текстовой реке, то видим только микро- или макрокартину — теряем самую важную историю. Здесь не хватает анализа социальных сетей (social network

analysis, SNA), который помогает отображать и анализировать закономерности взаимодействия. Например, 12 различных международных блогосфер отличаются друг от друга. Мы можем заключить, что разные общества имеют отличающиеся интересы, которые порождают различные закономерности.

Но почему они разные? В конце концов, они представляют собой нечто более высокой размерности, проецируемое на два измерения. Разве не может быть так, что они только нарисованы по-разному? Да, но мы можем выполнить большой объем текстового анализа, который убеждает нас, что эти картинки действительно информативны. Мы прилагаем усилия к качественному толкованию контента.

Например, во французской блогосфере мы видим кластер, в котором обсуждается кулинария для гурманов. В Германии мы наблюдаем различные кластеры, обсуждающие политику и множество безумных хобби. Среди английских блогов мы видим два больших кластера. Оказывается, что это блоги консерваторов и либералов.

Российские сети блогеров, как правило, стараются обеспечить лояльность своей аудитории, поэтому мы очень четко видим определенные кластеры.

Объединение в группы по близости производится с помощью алгоритма Фрухтермана — Рейнгольда (https://github.com/gephi/gephi/wiki/index.php), где пребывание в одном районе означает связь ваших соседей с другими соседями, так что это действительно отражает коллективный феномен влияния. Далее мы интерпретируем сегменты. На рис. 10.2 показан пример англоязычных блогов.

Как визуализации помогают находить стаи рыб. Все компании, владеющие социальными сетями, основываются на том, что у них есть либо данные, либо инструментарий — запатентованный механизм анализа тональности высказываний или нечто в этом роде: *машина, которая говорит «Пинь»*[1]. Однако имейте в виду: социальные сети в значительной степени являются продуктом организаций, оплачивающих продвижение своего дела, — другими словами, *играют с машиной, которая говорит «Пинь»*. Чтобы верить в то, что вы видите, вам нужно опережать эту игру, то есть следует разобраться с ней и понять, как она работает. Это значит, что необходима визуализация.

Пример: если вы думаете о выборах, то посмотрите блоги «мамочек» или «спортивных фанатов». Это более информативно, чем просмотр блогов партийных активистов, где ответ известен заранее.

Вот еще один пример: Келли подробно рассказал о проведенном после разделения блогосферы на сегменты анализе различных типов ссылок на идеологические видеозаписи, такие как речь Мартина Лютера Кинга «У меня есть мечта» и видео из

[1] Речь идет о скетче группы Монти Пайтон: https://www.youtube.com/watch?v=tKodtNFpzBA. — *Примеч. пер.*

Рис. 10.2. Англоязычные блоги

кампании в поддержку Ромни[1]. В случае с выступлением Мартина Лютера Кинга вы видите, что оно публикуется во время всплесков событий избирательного цикла по всему миру, но относительно видеоролика кампании Ромни заметны согласованные усилия консервативных блогеров, направленные на то, чтобы опубликовать видео в унисон.

То есть если бы вы просто смотрели на гистограмму ссылок — чистое число, — то могло бы показаться, что видеоролик Ромни оказался популярным; однако взгляд на него через линзу понятной сегментации блогосферы показывает: это явно запланированная операция для обхода показателя «виральность».

Келли также работает с Центром Беркмана по изучению Интернета и общества при Гарвардском университете (https://cyber.harvard.edu/research). Он проанализировал иранскую блогосферу в 2008 году и снова в 2011 году и нашел, что с точки зрения кластеризации многое осталось без изменений: молодые антиправительственные демократы, поэзия (важная часть иранской культуры) и в оба года доминировали консервативные прорежимные кластеры.

[1] Митт Ромни, был кандидатом в президенты США на выборах 2012 года.

Тем не менее, только 15 % блогов остались прежними в этот временной промежуток.

Таким образом, несмотря на то что люди часто интересуются *индивидуумами* (случайно-атрибутивная модель), отдельные рыбы менее важны, чем их *стаи*. Проводя анализ социальной сети, мы ищем стаи, так как благодаря им узна́ем о значительных интересах общества и о том, насколько эти интересы стабильны с течением времени.

Мораль этой истории состоит в том, что нам нужно сосредоточиться на закономерностях мезоуровня, а не микро- или макроуровня.

Дополнительные сведения об анализе социальных сетей с точки зрения статистики

Один из способов начать анализ социальной сети — это подумать о самой сети как о случайном объекте, подобно случайному числу или случайной переменной. Сеть может быть задумана как результат случайного процесса или исходя из нормального распределения вероятности. Фактически вы можете представить себе образец сетей и в таком случае задавать вопросы наподобие: «Что характеризует сети, которые, гипотетически, могут быть похожими на Twitter? Может ли данная сеть отражать дружеские отношения в реальном мире? Что вообще означает сказать “да” или “нет” в ответ на этот вопрос?»

Это часть основных вопросов дисциплины анализа социальных сетей, которая возникла в научных областях, таких как математика, статистика, информатика, физика и социология, с далеко идущими приложениями в еще большем количестве областей, включая исследования в сфере функциональной магнитно-резонансной томографии, эпидемиологии и изучения онлайновых социальных сетей, таких как Facebook или Google+.

Представление сетей и характеристическое число центральности

В некоторых сетях ребра между узлами направлены: я могу подписаться на вас в Twitter, когда вы не являетесь моим подписчиком, то есть появится ребро, направленное *от меня к вам*. Но другие сети имеют только симметричные ребра: мы либо знаем друг друга, либо нет. Эти последние типы сетей называются *неориентированными*.

Неориентированную сеть с N узлами можно представить матрицей размерностью $N \times N$, состоящей из единиц и нулей, где элемент матрицы с индексом (i, j) равен 1 тогда и только тогда, когда узлы i и j связаны. Данная матрица известна

как *матрица смежности*, или *матрица инцидентности*. Обратите внимание: мы фактически можем также определить это для направленных сетей, но для неориентированных сетей матрица всегда симметрична.

В качестве альтернативы сеть можно представить списком списков: для каждого узла i перечисляются узлы, к которым тот подключен. Это известно как список *смежности*, и обратите внимание: он не зависит от того, является ли сеть ненаправленной. Изображение сети таким способом экономит пространство для хранения — узлы могут иметь атрибуты, представленные в виде вектора или списка. Например, если узлы соответствуют людям, то атрибутами могут быть демографические сведения либо информация об их поведении, привычках или вкусах.

Сами ребра также могут иметь значения или веса/векторы, которые способны фиксировать информацию о характере связи между соединяемыми узлами. Эти значения можно сохранить в матрице размерностью $N \times N$ вместо единиц и нулей, которые просто показывают наличие или отсутствие отношения.

Теперь с учетом матрицы смежности A мы можем наконец определить характеристическое число центральности, о котором впервые упомянули в подразделе «Показатели центральности» раздела «Терминология из социальных сетей» текущей главы. Она сокращенно определяется как единственный вектор x, являющийся решением уравнения:

$$Ax = \lambda x$$

так, что:

$$x_i > 0, i = 1 \ldots N.$$

Как выяснилось, последнее условие эквивалентно выбору наибольшего собственного значения матрицы λ. Итак, для реального алгоритма найдем корни уравнения $det(A - tI)$ и упорядочим их по величине, выберем наибольший и обозначим его λ. Затем решим для x путем решения системы уравнений:

$$(A - \lambda I)x = 0.$$

Теперь у нас есть x, собственный вектор значений центральности.

Обратите внимание: это не позволяет *понять* характеристическое число центральности, даже если дает способ его вычислить. Вы можете получить представление, думая о нем как о пределе простой итерационной схемы, хотя потребуются доказательства, которые можно найти, например, здесь: https://sites.math.washington.edu/~morrow/336_11/papers/leo.pdf.

А именно, начните с вектора, элементами которого являются только степени вершин, возможно масштабированного, так что сумма элементов равна 1. Сами степени не дают реального понимания о характере взаимосвязей данного узла, тем не менее на следующей итерации прибавьте степени всех соседей данного

узла, снова масштабированные. Продолжайте повторять, добавляя степени соседей на каждом следующем шаге. В пределе, поскольку этот итеративный процесс продолжается вечно, то мы получим вектор характеристического числа центральности.

Первый пример случайных графов: модель Эрдеша — Реньи

Разберем простой пример, когда сеть можно рассматривать как единую реализацию базового стохастического процесса. А именно, когда существование данного ребра следует из распределения вероятности *и все ребра рассматриваются независимо*.

Скажем, мы начинаем с N узлов. Тогда есть $D = \binom{N}{2}$ пар узлов, или *диад*, которые могут быть либо связаны (неориентированным) ребром, либо нет. Тогда существует 2^D возможных наблюдаемых сетей. Простейшее базовое распределение, которое можно разместить на отдельных ребрах, называется *моделью Эрдеша — Реньи*, предполагающей, что для каждой пары узлов (i, j) ребро между двумя узлами существует с вероятностью p.

СХЕМА БЕРНУЛЛИ

Не все сети с N узлами встречаются с равной вероятностью в соответствии с этой моделью: наблюдение сети со всеми узлами, связанными со всеми другими узлами, имеет вероятность p^D, а наблюдение сети со всеми рассоединенными узлами — вероятность $(1 - p)^D$. И конечно, существует множество других возможных сетей между этими двумя крайностями. Модель Эрдеша — Реньи также известна как *схема Бернулли*. В литературе по математике данная модель рассматривается как математический объект с интересными свойствами, которые позволяют доказывать теоремы.

Второй пример случайных графов: экспоненциальная модель случайных графов

Есть плохая новость: социальные сети, которые можно наблюдать в реальном мире, как правило, не похожи на схемы Бернулли. Например, сети друзей или сети академического сотрудничества обычно демонстрируют такие признаки, как *транзитивность* (тенденция, в соответствии с которой если A знает B и B знает C, то A знает C), кластеризация (тенденция к тому, чтобы более или менее четко определенные небольшие группы существовали в более крупной сети), взаимность, или обоюдность (в направленной сети тенденция, когда A подписывается на B, если B подписался на A), и посредничество (тенденция к существованию особых людей, через которых проходит информационный поток).

Некоторые из данных наблюдаемых свойств сетей реального мира довольно просто перевести на математический язык. Например, транзитивность можно зафиксировать как количество треугольников в сети.

Экспоненциальная модель случайных графов (exponential random graph models, ERGM) — подход, позволяющий охватить эти реальные свойства сетей; широко используется в социологии.

Общий подход к ERGM заключается в том, чтобы выбрать соответствующую *статистику графа*, такую как количество треугольников, ребер и *звезд* второго порядка (подграфов, состоящих из узла с двумя спицами, таким образом, вершина со степенью 3 имеет три связанные с ней *звезды* второго порядка) с учетом количества узлов, и рассматривать эти данные как переменные z_i вашей модели, а затем скорректировать связанные с ними коэффициенты θ_i так, чтобы они были настроены на определенный тип поведения, который вы наблюдаете или хотите имитировать. Например, если z_1 относится к числу треугольников, то положительное значение для θ_1 указывает на тенденцию к большому числу треугольников.

Дополнительно введенная статистика графов включает звезды k-го порядка (подграфы, состоящие из узла с k спицами, поэтому узел со степенью $k + 1$ имеет $k + 1$ связанных с ним звезд порядка k), степень или *чередующиеся звезды порядка k*, агрегированную статистику по числу звезд k-го порядка для различных значений k. Приведем наглядный пример того, как ERGM будет выглядеть в виде формулы:

$$P_\gamma(Y = y) = (\frac{1}{k})(\theta_1 z_1(y) + \theta_2 z_2(y) + \theta_3 z_3(y)).$$

Здесь мы говорим, что вероятность наблюдения одной конкретной реализации случайного графа или сети, Y, является функцией статистики или свойств графа, которую мы только что описали, обозначив как z_i.

В этой структуре схема Бернулли является частным случаем ERGM, где у нас есть только одна переменная, соответствующая числу ребер.

Выводы для ERGM

В идеале, хотя в ряде случаев это невозможно реализовать на практике, можно было бы наблюдать выборку из нескольких сетей $Y_1 \ldots Y_n$, каждая из которых представлена своей матрицей смежности, скажем, для фиксированного числа N узлов.

Учитывая эти сети, мы могли бы моделировать их как независимые и одинаково распределенные наблюдения, соответствующие одной и той же вероятностной модели. Затем могли бы сделать выводы о параметрах данной модели.

В качестве первого примера, зафиксировав схему Бернулли, которая характеризуется вероятностью p существования любого заданного ребра, мы можем вычислить

вероятность того, что любая из сетей из нашей выборки будет происходить из этой схемы Бернулли следующим образом:

$$L = \prod_i^n p^{d_i}(1-p)^{D-d_i},$$

где d_i является количеством наблюдаемых ребер в i-й сети, а D — общее количество диад в сети, как и раньше. Тогда мы можем избежать оценки p, как показано ниже:

$$\hat{p} = \frac{\sum_{i=1}^n d_i}{nD}.$$

Фактически в литературе по ERGM наблюдается только одна сеть, то есть *мы работаем с единичной выборкой*. В данном единственном примере мы оцениваем параметр для вероятностной модели, которая «сгенерировала» эту сеть. Для схемы Бернулли, даже состоящей только из одной сети, мы могли бы оценить p как удельный вес ребер в общем числе диад, что представляется разумной оценкой.

Но для более сложных ERGM оценка параметров по одному наблюдению сети является грубой. Если это делается с помощью процедуры оценки псевдоправдоподобия, то она иногда дает бесконечные значения (см. статью Марка Хендкока (Mark Handcock) за 2003 год *Assessing Degeneracy of Statistical Models of Social Networks* («Оценка вырождения статистических моделей социальных сетей») (http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.81.5086)). Если вместо этого выполнить оценку с использованием того, что называется *методами MCMC* (Markov chain Monte Carlo methods, методы Монте-Карло с цепями Маркова), то она претерпит так называемое *косвенное вырождение*, где алгоритмы сходятся к вырожденным графам — графам, которые являются полными или пустыми, или алгоритм не сходится устойчиво (также рассмотрено в статье Хендкока).

Другие примеры случайных графов: латентные пространственные модели, сети тесного мира

Проблемы неустойчивости и вырождения модели, возникающие при экспоненциальном моделировании случайных графов, подтолкнули исследователей к введению *латентных пространственных моделей* (latent space models) (см. работу Питера Хоффа (Peter Hoff) *Latent Space Approaches to Social Network Analysis* («Латентно-пространственный подход к анализу социальных сетей») (http://www.dtic.mil/dtic/tr/fulltext/u2/a458734.pdf)).

Латентные пространственные модели пытаются решить следующую проблему: мы наблюдаем некую реальность, но существует соответствующая скрытая реальность, которую мы не можем наблюдать. Например, мы можем наблюдать связи между людьми в Facebook, но не видим, где живут эти люди или другие атрибуты, заставляющие их иметь склонность к тому, чтобы быть друзьями.

Другие исследователи предложили *сети тесного мира* (см. модель Уотта (Watts) и Строгаца (Strogatz) (https://en.wikipedia.org/wiki/Watts%E2%80%93Strogatz_model),

предложенную в их статье 1998 года), которые лежат в середине спектра между полностью случайными и полностью регулярными графами и пытаются охватить реальный феномен шести степеней разделения. Критика этой модели заключается в том, что она создает однородные по степени сети, тогда как наблюдаемые сети реального мира, как правило, не имеют масштаба и неоднородны по степени.

Помимо только что описанных моделей, существуют другие классы моделей, включая случайные поля Маркова, стохастические блочные модели, модели смешанного членства и стохастические блочные модели смешанного членства — каждая из них по-разному моделирует реляционные данные и стремится включать свойства, которые другие модели не учитывают (см., например, статью *Mixed Membership Stochastic Block Models* («Стохастические блочные модели смешанного членства») Эдуардо Эйроли (Edoardo Airoli) (https://dl.acm.org/citation.cfm?id=1442798) и др.).

Ниже перечислены несколько источников, которые могут быть полезны при дальнейшем изучении вопроса.

- *Networks, Crowds, and Markets* (Сети, толпы и рынки) (http://www.cambridge.org/us/academic/subjects/computer-science/algorithmics-complexity-computer-algebra-and-computational-g/networks-crowds-and-markets-reasoning-about-highly-connected-world) (Cambridge University Press) Дэвида Исли (David Easley) и Джона Клейнберга (Jon Kleinberg) с факультета информатики Корнелльского университета (Cornell's computer science department).
- Глава об анализе графов социальных сетей (http://i.stanford.edu/~ullman/mmds/ch10.pdf) в книге *Mining Massive Datasets*[1] (Cambridge University Press) Ананда Раджарамана (Anand Rajaraman), Джеффа Ульмана (Jeff Ullman) и Юре Лесковеца (Jure Leskovec) с факультета информатики Стэнфордского университета (Stanford's computer science department).
- *Statistical Analysis of Network Data* (Статистический анализ сетевых данных) (http://math.bu.edu/people/kolaczyk/softwareSAND.html) (Springer) Эрика Д. Колачика (Eric D. Kolaczyk) из Бостонского университета.

Журналистика данных

Нашим вторым спикером был Джон Брунер, редактор издательства O'Reilly, который ранее работал редактором данных в Forbes (https://www.forbes.com/forbes/welcome/?toURL=https://www.forbes.com/&refURL=&referrer=). Джон обладает широкими навыками: занимается исследованиями и пишет обо всем, что связано с данными.

---

[1] *Лесковец Ю., Раджараман А., Ульман Дж. Д.* Анализ больших наборов данных. — М.: ДМК Пресс, 2016.

Немного из истории журналистики данных

Журналистика данных существует уже некоторое время, но до недавних пор компьютерная отчетность была областью продвинутых пользователей Excel. (Даже сейчас, если вы знаете, как написать программу в Excel, то вы элита.)

В последнее время все начало меняться: нам стало доступно больше данных в виде API (https://ru.wikipedia.org/wiki/API), новых инструментов и менее дорогостоящей вычислительной мощности — почти каждый может анализировать довольно большие наборы данных на ноутбуке. Навыки программирования теперь довольно широко распространены для того, чтобы вы могли найти людей, которые являются и хорошими писателями, и хорошими программистами. Многие люди — специалисты в английском языке — достаточно хорошо разбираются в компьютерах, чтобы вам подойти; или, с другой стороны, вы найдете специалистов в области компьютерных наук, которые могут писать.

В крупных изданиях, таких как The New York Times, практика журналистики данных делится на области: графика и интерактивные функции, исследования, инженеры баз данных, сборщики, разработчики программного обеспечения и писатели-эксперты. Некоторые люди отвечают за правильную постановку вопросов, но предоставляют другим выполнение анализа. Например, Чарльз Духигг (Charles Duhigg) (https://www.nytimes.com/by/charles-duhigg) из The New York Times изучил качество воды в Нью-Йорке и подал запрос в соответствии с Законом о свободе информации в штат Нью-Йорк — он знал достаточно о том, что будет в этом запросе и какие вопросы задать, но фактический анализ сделал кто-то другой.

В организациях меньшего размера все обстоит иначе. Притом что в The New York Times работает 1000 человек на их «этаже» редакции новостей, в The Economist может быть 130, а Forbes задействует 70 или 80 человек в своих отделах новостей. Если вы не работаете ни над чем, кроме национальной ежедневной газеты, то в конечном итоге все делаете сами: задаете вопрос, получаете данные, выполняете анализ, а затем записываете. (Конечно, вы также вольны помогать своим коллегами и сотрудничать с ними, когда это возможно.)

Техническая документация в журналистике: совет профессионала

Джон был математиком в колледже в Чикагском университете, после чего устроился на работу в Forbes, где медленно вернулся к работе с числами. Например, он обнаружил, что использует инструменты теории графов, когда рассматривал вклады миллиардеров в политиков.

Он объяснил термин «журналистика данных» аудитории, прибегнув к своему профилю исследователя данных.

Прежде всего, термин связан с *большим количеством* визуализации данных, поскольку это быстрый способ описать важные аспекты набора данных. Знания в области информатики важны и в журналистике данных. Существуют жесткие сроки, и журналист данных должен хорошо разбираться в инструментах и уметь работать с грязными данными, поскольку даже федеральные данные беспорядочны. Нужно иметь возможность обрабатывать секретные форматы, и часто это означает синтаксический анализ материала в Python. Сам Джон наряду с некоторыми другими инструментами использует JavaScript, Python, SQL и MongoDB.

Статистика, говорит Бруно, определяет ваш способ восприятия мира. Она вдохновляет вас писать: например, *средний человек* в Twitter — женщина с 250 подписчиками, но *медианный человек* имеет 0 подписчиков — данные явно перекошены.

Бруно признает себя новичком в машинном обучении. Тем не менее он считает, что наличие знаний предметной области является критическим в журналистике данных: за исключением людей, которые могут специализироваться на одном предмете, например, правительственном управлении или огромной ежедневной газете, для небольшой газеты вам нужно быть широким специалистом и быстро приобрести базовый уровень опыта.

Разумеется, обмен информацией и презентации абсолютно необходимы журналистам. Их основная специализация — *перевод*: из сложных историй извлекается содержание, понятное для читателей. Журналистам также необходимо предвосхищать вопросы, превращать их в количественные эксперименты и настойчиво отвечать на них.

Совет от Джона для любого, кто начинает проект по журналистике данных: *у вас не должно быть четких убеждений, до того как вы опросите экспертов*. Начните с расплывчатого представления о том, что ищете, и будьте готовы изменить свое мнение и сделать резкий поворот, если эксперты поведут вас в новом и интересном направлении. Звучит как разведочный анализ данных!

11 Причинность

Многие из моделей и примеров в этой книге до сих пор были сфокусированы на фундаментальной проблеме прогнозирования. Мы обсудили примеры, как в главе 8, где ваша цель состояла в том, чтобы построить модель для прогнозирования, может ли человек с большой вероятностью предпочесть определенный предмет, скажем фильм или книгу. В модели могут входить тысячи характеристик, и выбор признаков позволит свести их к минимуму, но в конечном итоге модель оптимизируется для получения максимальной точности. При оптимизации по точности не обязательно беспокоиться о *значении или интерпретации характеристик*; если их тысячи, то они почти не поддаются интерпретации вообще.

Кроме того, вам даже не захочется делать заявление о том, что определенные характеристики *заставляют* человека покупать товар. Например, ваша модель для прогнозирования или рекомендации книги на Amazon может включать в себя характеристику «независимо от того, читали ли вы книгу Уэса Маккинни *Python for Data Analysis* («Python для анализа данных») издательства O'Reilly». Мы не сказали бы, что чтение его книги побудит вас прочесть *нашу* книгу. Это может быть просто хорошим показателем, который был бы обнаружен и проявился бы как таковой в процессе оптимизации по точности. Мы хотели бы подчеркнуть, что это не просто привычный компромисс корреляции-причинности, который вы, вероятно, уже вбили себе в голову. Скорее, ваше *намерение* при построении подобной модели или системы было бы направлено не на понимание причинности вообще, а на *предсказание*. И если бы вы собирались построить модель, *которая* позволила бы добраться до причинности, то действовали бы в этом случае по-другому.

Полностью отличный набор реальных проблем, который фактически использует те же статистические методы (логистическая, линейная регрессии) как часть строительных блоков для решения, — это ситуации, когда вы хотите понять причинность, быть в состоянии сказать, что конкретный тип поведения *вызывает* определенный результат. В подобных случаях ваш способ мышления или цель заключаются не в оптимизации по точности прогноза, а скорее в том, чтобы иметь возможность установить причины.

В этой главе мы разберем тему причинности, и у нас есть два эксперта в данной области в качестве приглашенных докладчиков: Ори Стайтельман (Ori Stitelman) и Дэвид Мэдиган (David Madigan). Биография последнего будет приведена в следующей главе и требует текущей главы в качестве предыстории. Вместо этого мы начнем с Ори, который в настоящее время является исследователем данных в Wells Fargo. Он получил степень доктора философии по биостатистике в Калифорнийском университете в Беркли после работы в консалтинговой фирме, предоставляющей услуги по сопровождению дел в суде. В рамках работы ему нужно было создавать истории на основе данных для экспертов в целях дачи показаний в суде, и таким образом он разработал то, что называет интуицией данных, под воздействием множества разных наборов данных.

Корреляция не подразумевает причинности

Одна из самых больших статистических задач, как теоретической, так и практической, — установление причинно-следственной связи между двумя переменными. Когда одна вещь является причиной другой? Это даже сложнее, чем кажется.

Предположим, мы обнаруживаем корреляцию между продажами мороженого и продажами купальников, которую показываем, изобразив продажи мороженого и купальных костюмов в течение времени на рис. 11.1.

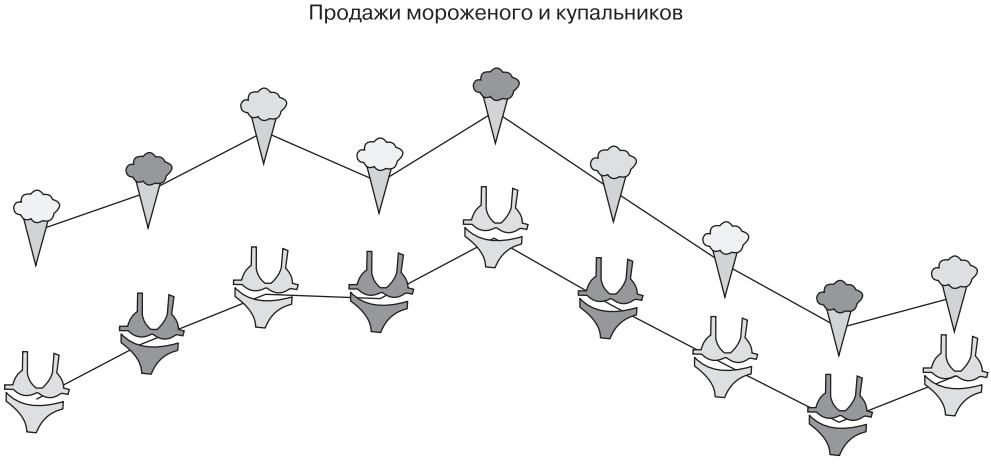


Рис. 11.1. Взаимосвязь между продажами мороженого и купальников

Рисунок демонстрирует тесную взаимосвязь между двумя данными переменными, но не устанавливает *причинности*. Посмотрим на это, делая вид, что ничего

не знаем о ситуации. Здесь могут работать всевозможные объяснения. Неужели у людей возникает непреодолимая тяга к поеданию мороженого, когда они носят купальники? Люди переодеваются в купальные костюмы каждый раз, когда едят мороженое? Или существует некий третий фактор (например, жаркая погода), который мы не считали причиной обоих? Вывод о причинной зависимости — это область, в которой лучше понять условия, побуждающие истолковывать взаимосвязь как причинность.

Задаем причинные вопросы

Естественной формой причинного вопроса является следующая: «Какой эффект *x* оказывает на *y*?»

Некоторые примеры: «Какой эффект *реклама* оказывает на *поведение клиента*?» либо «Какой эффект *препарат* оказал *во время до наступления вирусологической неэффективности*?». Или в более общем случае: «Какой эффект оказал *процесс лечения на результат*?»

Термины «леченый» и «нелеченый» пришли из биостатистики, медицины и области клинических исследований, где пациентам предоставляется врачебное обслуживание; примеры этих терминов мы встретим в следующей главе. Терминология была принята статистической и социологической литературой.

Как оказалось, оценивать причинные параметры сложно. На самом деле эффективность рекламы почти всегда рассматривается как спорный вопрос, поскольку ее очень трудно измерить. Люди обычно выбирают показатели успеха, которые легко оценить, но не измеряют то, что хотят, и все равно принимают решения на их основе, так как это проще. Но у использования подобных показателей есть реальные негативные последствия. Например, маркетологи в конечном итоге получают вознаграждение за продажи через Интернет людям, которые купили бы что-нибудь в любом случае.

Искажающие факторы: на примере сайта знакомств

Рассмотрим пример из мира онлайн-знакомств с участием одинокого парня по имени Фрэнк. Предположим, он просматривает сайт знакомств и находит очень подходящую девушку. Он хочет убедить ее пойти с ним на свидание, но сначала ему нужно написать письмо, которое заинтересовало бы ее. Что ему следует написать в нем? Должен ли он сказать ей, что она красива? Как мы проверяем это с помощью данных?

Подумаем о рандомизированном эксперименте, который мог бы запустить Фрэнк. Он мог бы выбрать несколько красивых девушек и половине из них, отобранной

случайным образом, сказать, что они красивы. Затем он мог бы увидеть разницу в частоте ответов между двумя группами.

Однако по какой-то причине Фрэнк не делает этого — возможно, он слишком романтичен, что заставляет нас попытаться выяснить, является ли комплимент девушке хорошим ходом для него. Пойдет ли Фрэнк на свидание, зависит от нас.

Если бы это было возможно, мы понимали бы будущее как две альтернативные реальности: реальность, где он отправляет электронное письмо, в котором говорит девушке, что та красива, и реальность, в которой он отправляет электронное письмо, но не использует слово «красивый». Но возможна только одна реальность. Итак, как же продолжить?

Запишем наш причинный вопрос в явном виде: какой эффект оказывает Фрэнк, говорящий девушке, что она красива, на получение им положительного ответа?

Другими словами, «лечение» — то, что Фрэнк говорит девушке, что она красива, по электронной почте, а «результат» — положительный ответ в электронном письме или, возможно, его отсутствие. С помощью электронной почты от Фрэнка, который не назовет получателя письма красивой, будет осуществляться контроль этого исследования.

Многое из того, что мы здесь не делаем, можно было бы попробовать. Например, мы не думаем об особенностях Фрэнка. Возможно, он действительно странный, малопривлекательный парень, которого ни одна девушка не захочет позвать на свидание. Возможно, он даже не может произнести по слогам «красивый». И наоборот: а если он эффектный и/или знаменитый и не имеет значения, что он говорит? Кроме того, большинство сайтов знакомств позволяют женщинам общаться с мужчинами так же легко, как и мужчинам с женщинами, поэтому неясно, что наше толкование для «леченых» и «нелеченых» четко определено. Некоторые женщины могут игнорировать свою электронную почту, но спонтанно отправить электронное письмо Фрэнку.

Пример с сайта знакомств OK Cupid

В качестве первого шага к пониманию влияния выбора слова на частоту ответов сайт онлайн-знакомств OK Cupid проанализировал более 500 000 первых контактов, случившихся на нем. Сотрудники рассмотрели ключевые слова и фразы и их влияние на ответы. Результат показан на рис. 11.2.

Ось *Y* показывает частоту ответа. В среднем по *всем* адресам электронной почты она составляла ~32 %. Затем сотрудники сайта взяли подмножество электронных писем, в которые входило определенное слово, такое как «красивая» или «клевая»,

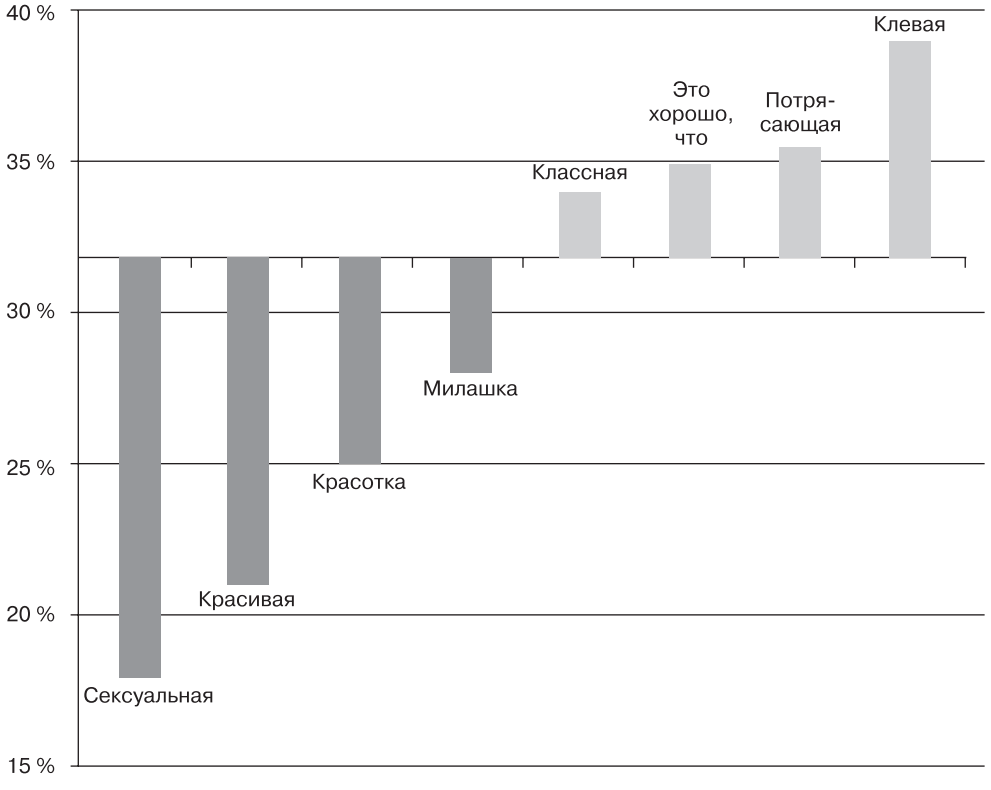


Рис. 11.2. OK Cupid пытается продемонстрировать, что использование слова «красивая» в электронном письме снижает ваши шансы получить ответ

и посмотрели на скорость ответа на эти письма. Записав результат в терминах условных вероятностей, мы бы сказали, что они оценивали их как: P (ответ) = 0,32 против P (ответ | «красивая») = 0,22.

Важная часть информации, отсутствующая на данном графике, — размеры групп. Сколько первых контактов содержало каждое из слов? Это ничего не меняет, за исключением того, что поможет понять следующее: горизонтальная линия на 32 % является средневзвешенной величиной всех этих разных групп электронных писем.

Сотрудники сайта интерпретировали данный график и создали правило под названием «Избегайте физических комплиментов». Они обсуждали это в блоге в статье *Exactly what to say in a first message* («Что точно можно сказать в первом сообщении») (https://theblog.okcupid.com/exactly-what-to-say-in-a-first-message-2bf680806c72), дав

следующее объяснение: «Вы можете подумать, что такие слова, как “эффектная”, “красивая” и “сексуальная”, могут быть для кого-то приятными, но никто не хочет их слышать так быстро. Как мы все знаем, люди обычно любят комплименты, но когда их используют в качестве пикапа, особенно до личной встречи, это неизбежно чувствуется... хм. Кроме того, когда вы говорите девушке, что она красива, скорее всего, про вас такого же нельзя сказать».

Это не эксперимент, а скорее исследование методом наблюдения, о котором мы поговорим более подробно, но на теперешний момент собираем данные, как это, естественно, происходит в дикой природе. Можно ли сделать вывод, глядя на приведенный график, что добавление слова «клевая» в электронное письмо увеличивает скорость ответа или что «красивая» снижает частоту ответа?

Прежде чем дать ответ, рассмотрите следующие три момента.

Во-первых, график мог бы сказать больше о *человеке*, говорящем «красиво», чем о самом слове. Может быть, люди, которые употребляют слово «красивая» в письмах, вообще смешные и слишком недалекие? Во-вторых, люди могут описывать как красивых *себя* или другие вещи, такие как мир, в котором мы живем.

Обе проблемы имеют значение, когда мы пытаемся понять данные, относящиеся ко всему населению, такие как на рис. 11.2, поскольку они затрагивают вопрос о том, действительно ли слово «красивый» в тексте письма фактически означает то, что мы думаем. Но обратите внимание: обе проблемы, если они присутствуют, неизменно реальны для конкретного парня, такого как Фрэнк, пытающегося получить приглашение на свидание. Поэтому если Фрэнк слащав, то он будет теоретически одинаково слащавым в письмах ко всем женщинам, так что, с точки зрения Фрэнка, будет согласованным экспериментом использовать или нет слово «красивая» в своих письмах.

Третий и самый важный вопрос, который следует учитывать, поскольку он *не* остается закономерным для данного парня, заключается в следующем: *конкретные получатели электронных писем, содержащих слово «красивая», могут быть особенными*. Например, могут получать тонны посланий по электронной почте и отвечать только на часть из них, что для Фрэнка значительно снизило бы вероятность вообще получить ответ.

На самом деле если девушка, о которой идет речь, красива (представим, что это четко определенный термин), то данный факт влияет одновременно на два отдельных момента. Во-первых, использует ли Фрэнк слово «красивая» или нет в своих письмах, а во-вторых, на результат, то есть получает ли Фрэнк ответ. По этой причине тот факт, что девушка красива, квалифицируется как *искажающий фактор*, другими словами, переменная, которая влияет или оказывает причинно-следственное влияние на само лечение, а также на результат.

Будем честны в том, что *на самом деле* показывает этот график в противоположность тому, что подразумевал OK Cupid. Он показывает наблюдаемую скорость ответа на электронные письма, содержащие указанные слова. Он *не должен* использоваться и не может быть правильно истолкован как рецепт или предложение о том, как создать электронное письмо, чтобы получить ответ, поскольку после *настройки* для искажающих факторов, которые мы обсудим далее в этой главе, использование слова «красивая» может оказаться лучшим, что мы могли бы сделать. Мы не можем сказать точно, так как не имеем данных, но опишем, какие данные понадобятся и как мы их проанализируем, чтобы провести это исследование должным образом. Совет OK Cupid может быть правильным, но график, который они показали, его не подкрепляет.

Золотой стандарт: рандомизированные клинические испытания

Так что же нам делать? Как люди *вообще* определяют причинность?

Золотой стандарт установления причинности — рандомизированный эксперимент. Это протокол, в соответствии с которым мы произвольно определяем одних людей в группу, которой назначают «лечение», а других — в «контрольную» группу, то есть им ничего *не назначают*. Затем мы получаем некий результат, который хотим измерить, а причинно-следственное влияние — это просто разница между «леченой» и контрольной группами в данном измеряемом результате. Принцип использования экспериментов для оценки причинно-следственного влияния основывается на статистическом предположении, что применение рандомизации для выбора двух групп создает идентичные популяции с точки зрения статистики.

Рандомизация на самом деле хорошо работает: так как мы подбрасываем монетки, все другие факторы, которые могут быть искажающими (например, бывшие курильщики и те, кто курит на данный момент), более или менее устранены, поскольку мы можем гарантировать, что курильщики будут распределены относительно равномерно между этими двумя группами, если в исследовании задействовано достаточно людей.

Поистине гениальный аспект рандомизации заключается в том, что она хорошо соотносится с возможными искажающими факторами, о которых мы подумали, но также уравновешивает *50 млн не учтенных нами моментов*.

Итак, хотя мы можем алгоритмически найти лучшее разбиение для тех факторов, о которых подумали, это, вполне вероятно, не сработает с другими. Вот почему мы в действительности делаем это случайным образом, поскольку данный способ хорошо справляется как с тем, что мы учитываем, так и с тем, что не было учтено.

Но есть и плохие новости относительно рандомизированных клинических испытаний, как мы указывали ранее. И вот первая из них: только с точки зрения этики существует так называемая клиническая эквиполентность (https://en.wikipedia.org/wiki/Clinical_equipoise), означающая, что медицинское сообщество действительно не знает, какое лечение лучше. Если мы знаем, что для кого-то лечение с помощью лекарственных препаратов предпочтительнее, чем без них, то не можем случайным образом не давать людям лекарства.

Например, желая выявить зависимость между курением и сердечными заболеваниями, мы не можем случайным образом назначить кого-нибудь курящим, поскольку известно, что это опасно. Точно так же связь между кокаином и весом при рождении сопряжена с опасностью, равно как и сложная взаимосвязь между диетой и смертностью.

Другая проблема заключается в том, что рандомизированные клинические исследования дороги и громоздки. Их проведение требует много времени и большого количества людей. С другой стороны, отказ от рандомизированных клинических испытаний может привести к ошибочным предположениям, которые также чрезвычайно дороги.

Иногда рандомизированные исследования просто неосуществимы. Вернемся к примеру с OK Cupid, где мы имеем набор данных наблюдений и веские основания предполагать существование искажающих факторов, которые препятствуют нашему пониманию размера эффекта. Как уже отмечалось, в соответствии с золотым стандартом следовало бы провести эксперимент, и, хотя сотрудники OK Cupid *могли бы* провести эксперимент, было бы неразумно делать это — случайная отправка людям электронных писем, в которых говорится, что они «красивые», являлась бы нарушением соглашения с клиентами.

В заключение следует отметить, что, *когда это возможно*, рандомизированные клинические испытания являются золотым стандартом для выяснения причинно-следственных связей. Просто такая возможность существует не всегда.

СРЕДНЕЕ И ИНДИВИДУАЛЬНОЕ

Рандомизированные клинические испытания измеряют влияние определенного лекарственного средства, усредненное по всем людям. Иногда они могут разделить пользователей на группы, чтобы понять средние последствия для мужчин либо женщин или людей определенного возраста и т. д. Но, в конце концов, это все еще усредненный материал, так что мы не знаем, какое влияние будет оказано на определенного человека. В наши дни есть предпосылки для создания персонализированной медицины с ориентиром на генетические данные, а это значит, что мы перестаем смотреть на средние значения, поскольку хотим сделать выводы об одном человеке. Даже применительно к Фрэнку и OK Cupid есть разница между проведением этого исследования среди всех мужчин в сравнении с одним Фрэнком.

A/B-тестирование

В компаниях по разработке программного обеспечения то, что мы описали как случайные эксперименты, иногда упоминается как A/B-тестирование. Фактически мы обнаружили, что слово «эксперименты» для инженеров-программистов означало бы «попробовать что-то новое», а вовсе не базовую статистическую методику применения пользователями разных версий продукта в целях измерения влияния различий на значения метрик. Концепция достаточно интуитивная и кажется простой. По сути, правильная настройка инфраструктуры может свести запуск эксперимента к написанию короткого файла конфигурации с изменением только одного параметра — будь то другой цвет либо макет или базовый алгоритм, — что позволит одним пользователям получить опыт, отличный от опыта других. Таким образом, существуют аспекты проведения A/B-тестирования в информационно-технологической компании, которые значительно его упрощают в сравнении с клиническими испытаниями. И гораздо меньше поставлено под угрозу с той точки зрения, что мы не имеем дело с жизнью людей. Другой удобный момент — отсутствие проблемы соблюдения, поскольку при случайных клинических испытаниях мы не можем контролировать, принимает ли кто-то лекарство, тогда как в онлайн-режиме имеем возможность проверить, что показываем пользователю. Но обратите внимание: мы сказали — *если* правильно настроить экспериментальную инфраструктуру, и это большое ЕСЛИ.

Требуется много работы для правильной настройки и последующего корректного анализа данных. Когда разные команды в компании работают над новыми характеристиками продукта и все хотят опробовать варианты, если вы не будете осторожны, то один пользователь может в конечном итоге испытать сразу несколько изменений. Например, команда по UX (user experience — «опыт взаимодействия») может изменить цвет либо размер шрифта или макет, чтобы увидеть, увеличится ли скорость нажатий. В то же время команда по ранжированию контента может захотеть изменить алгоритм, который выбирает, что рекомендовать пользователям, и команда по рекламе может вносить изменения в свою систему торгов. Предположим, что метрика, которая вас интересует больше всего, — это частота повторных заходов пользователей, и пользователь начинает возвращаться чаще. Вы применяли ее в трех разных комбинациях условий, но этого не знали, поскольку команды не координированы друг с другом. Ваша команда может предположить, что данная комбинация условий — причина бо́льшего количества повторных заходов пользователей, но причина может быть в комбинации всех трех способов.

Существуют различные аспекты инфраструктуры, которые нужно учитывать; более подробно они описаны в статье 2010 года *Overlapping Experiment Infrastructure: More, Better, Faster Experimentation* («Перекрытие экспериментальной инфраструктуры: экспериментируем больше, лучше, быстрее») (https://ai.google/research/pubs/pub36500), написанной сотрудниками Google Дианой Танг (Diane Tang) и др. Ниже представлен отрывок из этой статьи.

ОТРЫВОК ИЗ СТАТЬИ «ПЕРЕКРЫТИЕ ЭКСПЕРИМЕНТАЛЬНОЙ ИНФРАСТРУКТУРЫ: ЭКСПЕРИМЕНТИРУЕМ БОЛЬШЕ, ЛУЧШЕ, БЫСТРЕЕ»

В таком случае цели проектирования нашей экспериментальной инфраструктуры следующие: больше, лучше, быстрее.

- *Больше*. Нам нужна масштабируемость, чтобы проводить больше экспериментов параллельно. Однако нам также требуется гибкость: разные эксперименты нуждаются в разных конфигурациях и разных масштабах, чтобы иметь возможность измерять статистически значимые результаты. В некоторых экспериментах необходимо изменить только выборку по трафику, например лишь японский трафик, и его необходимо соответствующим образом подобрать по размеру. Другие эксперименты могут изменить весь трафик и произвести большие изменения в показателях, и поэтому могут быть запущены с меньшим трафиком.
- *Лучше*. Недостоверные эксперименты не должны запускаться для реального трафика. Достоверные, но некорректные эксперименты (например, ошибочные или непреднамеренно производящие действительно плохие результаты) следует быстро перехватить и отключить. Стандартизованные показатели должны быть легкодоступны для всех экспериментов, чтобы сравнение последних было справедливым: двум экспериментаторам нужно использовать одни и те же фильтры для удаления автоматического трафика при вычислении такой метрики, как CTR.
- *Быстрее*. Эксперимент должен быть простым и быстрым; достаточно простым, чтобы неинженер мог его провести без написания кода. Метрики должны быть доступны быстро, чтобы эксперименты можно было оценить с высокой скоростью. Простым итерациям следует быть быстрыми. В идеальном случае система должна поддерживать не просто эксперименты, но и контролируемое нарастание, то есть постепенно увеличивать изменение по всему трафику систематически и хорошо понятным способом.

Над данной экспериментальной инфраструктурой работает большая команда, которая анализирует результаты экспериментов на постоянной основе, в связи с чем это нетривиально. Еще больше осложняет ситуацию то, что теперь, в эпоху социальных сетей, мы уже не можем считать, будто пользователи независимы (это часть предположения о рандомизации, лежащей в основе экспериментов). Так, например, Рэйчел может быть в получающей «лечение» группе эксперимента, который проводит Facebook (что невозможно, поскольку Рэйчел на самом деле не в Facebook, а просто делает вид). Это позволяет Рэйчел опубликовать какой-то особый магический пост, а Кэти может быть в контрольной группе, но все еще видит специальный магический пост, поэтому фактически получила другую версию «лечения», вследствие чего экспериментальный проект должен учитывать основную структуру сети. Это нетривиальная проблема и все еще открытая исследовательская область.

Второе место: исследования методом наблюдения

Хотя под золотым стандартом, как правило, понимаются рандомизированные эксперименты или A/B-тестирование, они не всегда возможны, поэтому мы иногда используем метод, занимающий второе место, а именно исследования методом наблюдения.

Начнем с определения:

> «Исследование методом наблюдения — это эмпирическое исследование, цель которого — выявление причинно-следственных связей, когда невозможно использование контролируемых экспериментов».

Большая часть деятельности, имеющей отношение к науке о данных, так или иначе связана с данными наблюдений, хотя A/B-тестирование, как вы могли видеть ранее, является исключением из этого правила. В большинстве случаев данные, которые у вас есть, — то, что вы получаете. Например, вы не можете переиграть день на рынке, когда Ромни выиграл президентские выборы.

Спланированные исследования — почти всегда теоретически лучшие тесты, как мы знаем, но есть много примеров, где их проведение неэтично. Исследования методом наблюдения проводятся в контекстах, в которых вы не можете запланировать эксперименты, чтобы выяснить причины и следствия.

В реальности вас иногда не волнуют причинно-следственные связи; вы просто хотите построить прогностические модели. Тем не менее есть много основных проблем, связанных с этим.

Парадокс Симпсона

Ниже описаны всевозможные подводные камни в исследованиях методом наблюдения.

Например, на графике на рис. 11.3 представлена наилучшая эмпирическая кривая, призванная описать, коррелирует ли прием более высоких доз «плохого лекарства» с большей вероятностью сердечного приступа.

Похоже, с этой точки зрения чем выше доза, тем меньше сердечных приступов у пациента. Но есть два кластера, и если вы знаете о них больше, то придете к противоположному выводу, как можно видеть на рис. 11.4.

График был сфальсифицирован, поэтому проблема очевидна. Но, конечно, когда данные многомерны, вы уже не всегда будете рисовать такую простую картину.

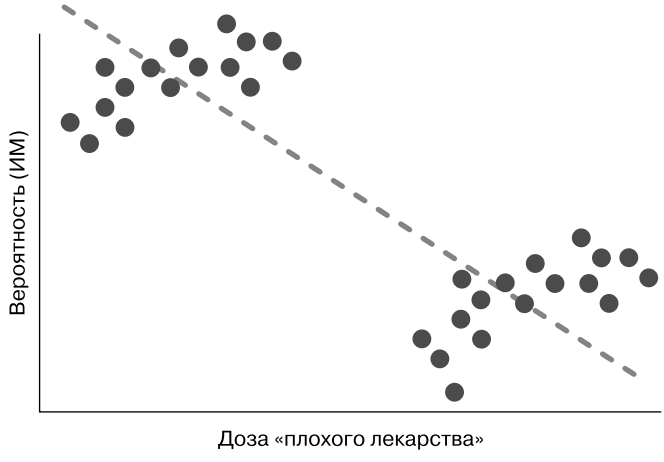


Рис. 11.3. Вероятность получить сердечный приступ (также известный под названием ИМ или «инфаркт миокарда») как функция от размера дозы «плохого лекарства»

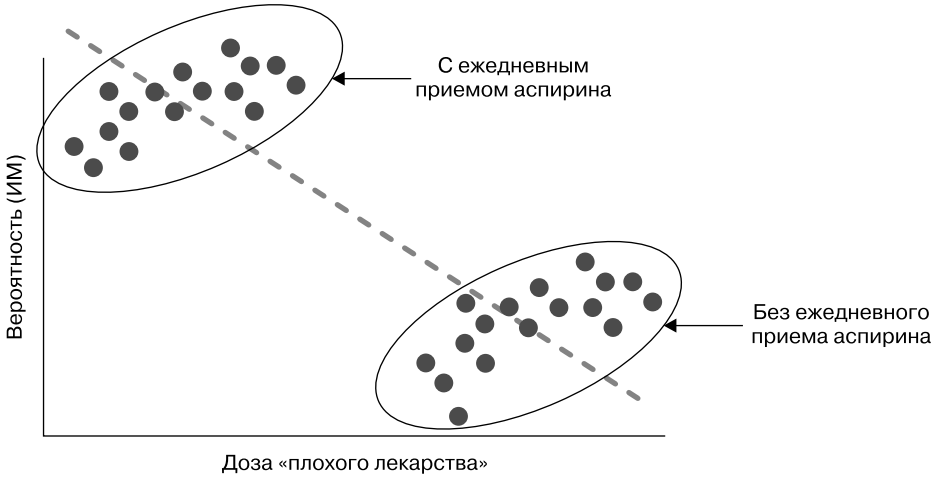


Рис. 11.4. Вероятность получить сердечный приступ как функция от размера дозы «плохого лекарства» и того, принимал ли пациент также аспирин

В данном примере мы бы сказали, что аспирин является искажающим фактором. Мы немного расскажем об этом позже, но пока говорим, что прием аспирина или отсутствие его приема людьми, участвующими в исследовании, не были случайным образом распределены, и это оказало огромное влияние на кажущийся эффект препарата.

Обратите внимание: если вы думаете об исходной линии как о прогностической модели, то она на самом деле *остается* лучшей моделью, которую вы можете

получить, не зная ничего больше о привычке принимать аспирин или половой идентичности привлеченных пациентов. Проблема здесь в том, что вы пытаетесь приписать причинность.

Это основная проблема моделей регрессии по данным наблюдений. Вы понятия не имеете, что происходит. Как описал Мэдиган, «там Дикий Запад».

Возможно, в каждой группе есть мужчины и женщины, и если вы *их* разделите, то увидите: чем больше лекарств они принимают, тем лучше. Поскольку данный человек либо мужчина, либо женщина и либо принимает аспирин, либо нет, то эти параметры действительно важны.

Таким образом, можно сформулировать фундаментальную проблему исследований методом наблюдения: тенденция, которая появляется в разных группах данных, исчезает, когда эти группы объединены, или наоборот. Это иногда называют парадоксом Симпсона (https://ru.wikipedia.org/wiki/Парадокс_Симпсона).

Причинно-следственная модель Рубина

Причинно-следственная модель Рубина (https://en.wikipedia.org/wiki/Rubin_causal_model) является математической основой для понимания того, какая информация нам известна и неизвестна при исследованиях методом наблюдения.

Она предназначена для исследования путаницы, когда кто-то говорит нечто наподобие: «У меня рак легких, поскольку я курил». Это правда? Если да, то вы должны быть в состоянии подтвердить утверждение: «Если бы я не курил, то не получил бы рак легких», но этого никто не знает.

Определим Z_i как применение лечения к i-му единичному элементу (0 = контроль, 1 = лечение), $Y_i(1)$ как реакцию для i-го элемента, если $Z_i = 1$, и $Y_i(0)$ — как реакцию для i-го элемента, если $Z_i = 0$.

Тогда *причинно-следственное влияние единичного уровня*, то, о чем мы заботимся, есть $Y_i(1) - Y_i(0)$, но мы видим только одно из значений $Y_i(0)$ и $Y_i(1)$.

Пример: Z_i — это 1, если я курил, 0 — если я этого не делал (я элемент). $Y_i(1)$ равно 1, если у меня рак и я курил, и 0, если бы я курил и не заработал рак. Точно так же $Y_i(0)$ составляет 1 или 0, в зависимости от того, заболел ли я раком, когда не курил. Общее причинно-следственное влияние на меня — это разность $Y_i(1) - Y_i(0)$. Она равна 1, если у меня действительно есть рак, поскольку я курил; 0, если у меня был рак (или нет), независимо от курения; и –1, если я избежал заболевания раком при курении. Но я никогда не узнаю свое фактическое значение, поскольку знаю только одну величину из двух.

На уровне населения мы знаем, как сделать вывод о том, что среди людей существует немало 1, но *никогда не сможем присвоить данному человеку это число*.

Иногда это называется фундаментальной проблемой причинно-следственной зависимости (https://en.wikipedia.org/wiki/Rubin_causal_model#The_fundamental_problem_of_causal_inference).

Визуализация причинности

Мы можем представить концепции каузального моделирования с помощью того, что называется *каузальным графом*.

Обозначим как W множество всех возможных искажающих факторов. Обратите внимание: это большое допущение, что можно учесть все из них, и вскоре мы увидим, насколько необоснованным оно кажется для эпидемиологических исследований из главы 12.

В нашем примере с Фрэнком мы выделили один момент в качестве потенциального искажающего фактора: девушка, которая, как ему хотелось бы, должна быть красивой. Однако если бы мы подумали об этом, то могли бы придумать другие искажающие факторы, например, привлекателен ли Фрэнк, или он в отчаянии; оба этих фактора влияют как на то, как он пишет девушкам, так и на их положительную реакцию.

Обозначим как A «лечение». В нашем случае оно заключается в том, что Фрэнк использует слово «красивая» во вводном письме. Обычно мы предполагаем, что это имеет двоичный (0/1) статус, поэтому девушке, которой пишет Фрэнк, мы присвоили бы 1, если он употребляет слово «красивая». Просто имейте в виду: если он скажет, что погода прекрасная (it's beautiful weather), то мы тоже будем считать, что это 1.

Обозначим как Y двоичный результат (0/1). Мы должны сделать его четко определенным. Например, можем убедиться, что Фрэнк просит девушек, которым он пишет, дать свой номер телефона, и могли бы определить положительный результат, обозначенный 1, как получение Фрэнком номера. Нам нужно сделать это как можно точнее, вследствие чего, например, мы бы сказали, что это должно произойти на платформе OK Cupid в течение недели после изначального электронного письма Фрэнка. Обратите внимание: мы будем давать 1 женщинам, которые игнорируют его электронные письма, но по какой-то причине отправляют ему электронное письмо со своим номером. Было бы также трудно проверить, что номер не фальшивый.

Узлы в каузальном графе помечены этими наборами искажающих факторов, «лечения» и результатов, а направленные ребра или стрелки указывают на причинность. Другими словами, узел, из которого выходит стрелка, каким-то образом напрямую влияет на узел, в который она входит.

Наш случай показан на рис. 11.5.

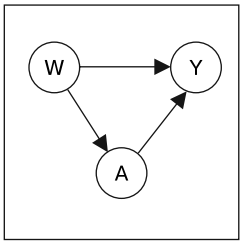


Рис. 11.5. Каузальный граф с одним «лечением», одним искажающим фактором и одним результатом

В случае с примером сайта OK Cupid каузальный граф — простейший из возможных: одно «лечение», один искажающий фактор и один результат. Но каузальные графы могут быть намного сложнее.

Определение: причинно-следственное влияние

Предположим, у нас есть население 100 человек, которые принимают какой-то препарат, и мы проверяем их на рак. Скажем, у 30 из них рак, что дает показатель заболеваемости раком 0,30. Мы хотим спросить: является ли препарат причиной рака?

Для ответа на этот вопрос нам нужно было бы знать, что бы произошло, не принимай эти люди лекарство. Поиграем в бога и условимся: если бы они не принимали препарат, то мы бы увидели, что раком заболели 20, то есть показатель — 0,20. Обычно мы измеряем повышенный риск развития рака как разницу этих двух чисел и называем это *причинно-следственным влиянием*. Следовательно, в данном случае скажем, что причинно-следственное влияние составляет 10 %.

Каузальный эффект иногда определяется как отношение этих двух чисел вместо разности.

Но у нас нет божественных знаний, так что вместо этого мы выбираем другое население с целью сравнить с данным, и видим, заболевают ли *они* раком, *не* принимая лекарство. Скажем, у них естественная частота рака 0,10. Затем мы заключаем, используя их в качестве суррогатного значения, что повышенная частота рака составляет разницу между 0,30 и 0,10, то есть 20 %. Это, конечно, неправильно, проблема в том, что две группы населения имеют ряд основных различий, которые мы не учитываем.

Если бы это были «те же люди» вплоть до химического состава молекул друг друга, то данный суррогатный расчет работал бы отлично. Но, конечно, это не так.

Как же мы в действительности выбираем этих людей? Один из методов состоит в том, чтобы использовать так называемый метод подбора контрольной группы по индексу соответствия (propensity score matching) или моделирование по индексу соответствия. По существу, то, что мы делаем здесь, — это проведение псевдослучайного эксперимента путем создания синтетической контрольной группы с выбором людей, которые *с той же вероятностью* могли бы находиться в группе лечения, но не оказались там. Как нам это сделать? Смотреть на слово «вероятность» в данном предложении? Пришло время для выхода из логистической регрессии. Таким образом, есть два этапа при моделировании по индексу соответствия. Первый — использовать логистическую регрессию для моделирования вероятности того, что каждый человек *получит лечение*. Мы могли бы собрать людей так, чтобы один человек получил лечение, а другой — нет, но было бы *одинаково вероятным* (или близким к одинаково вероятному), что они его получат. Тогда мы можем действовать так же, как если бы в нашем распоряжении был случайный эксперимент.

Например, если бы мы хотели измерить влияние курения на вероятность рака легких, то нам нужно было бы найти людей, имеющих ту же вероятность *курения*. Мы собрали бы столько независимых переменных для людей, сколько смогли бы (возраст, курили ли их родители, курили ли их супруги, вес, диета, физические упражнения, рабочие часы в неделю, результаты анализа крови), и использовали бы в качестве результата, курили ли они. Мы построили бы логистическую регрессию, которая предсказала бы вероятность курения. Затем использовали бы эту модель для присвоения каждому человеку вероятности, которая будет называться их коэффициентом склонности, а затем применили бы ее для сопоставления. Конечно, мы возлагаем особые надежды на то, что *выяснили* и смогли наблюдать *все* независимые переменные, связанные с вероятностью курения, которого, вероятно, не было. И это неотъемлемая трудность указанных методов: мы никогда не узнаем, действительно ли скорректировали все требуемое для адаптации. Тем не менее один из приятных аспектов заключается в следующем: мы увидим, что учет искажающих факторов может иметь большое значение для предполагаемого причинно-следственного влияния.

Выполнить детальную настройку поиска соответствий может оказаться немного сложнее, чем простой поиск попарных соответствий — существуют более сложные схемы, предназначенные для обеспечения равновесия в искусственно сформированных терапевтической (лечебной) и контрольной группах. И есть пакеты в R, которые могут сделать все это для вас автоматически, за исключением того, что вы должны указать модель, которую хотите в первую очередь использовать для сопоставления, чтобы генерировать коэффициенты склонности, и какую переменную хотите применить в качестве результата, соответствующего причинно-следственному влиянию, которое вы оцениваете.

Какие данные нам нужны для измерения причинно-следственного влияния в нашем примере знакомства? Одна из возможностей состоит в том, чтобы иметь некую

третью сторону, mechanical turk; например, пройти через профили знакомств девушек, которым Фрэнк пишет по электронной почте, и отметить красивых. Таким образом, мы могли видеть, насколько искажающим фактором является степень красоты. Этот подход называется *расслоением* и, как мы увидим в следующей главе, может и создавать проблемы, и решать их.

Три совета

Ори на прощанье дал три совета с практическими рекомендациями по моделированию.

Во-первых, при оценке причинно-следственных параметров крайне важно понимать методы генерации данных и распределение, которые, в свою очередь, потребуют получения существенных знаний о предмете. Знание того, как создавались данные, также поможет определить, являются ли выдвигаемые вами предположения разумными.

Во-вторых, первый шаг в анализе данных всегда должен заключаться в том, чтобы вернуться и выяснить, *что вы хотите узнать*. Определите это, а затем найдите и используйте инструменты, которые изучили, для выдачи прямого ответа. Позже убедитесь в этом и вернитесь, чтобы решить, насколько близко подошли к ответу на ваш первоначальный вопрос или вопросы. Кажется очевидным, но вы будете удивлены, как часто люди забывают это делать.

Наконец, не игнорируйте необходимую интуицию в области данных при использовании алгоритмов. То, что ваш метод сходится, еще не значит, что результаты имеют смысл. Убедитесь, что создали разумный рассказ и способы проверить его достоверность.

12 Эпидемиология

В написание этой главы большой вклад внес Дэвид Мэдиган — профессор и завкафедрой статистики в Колумбийском университете. На счету Мэдигана более 100 публикаций в таких областях, как байесовская статистика, глубинный анализ текстов, методы Монте-Карло, фармакологический надзор и вероятностные графические модели.

О Мэдигане

В 1980 году Мэдиган поступил в колледж Trinity города Дублина, где специализировался на математике, за исключением последнего учебного года, когда прослушал несколько курсов по статистике, а также хорошо изучил информатику (Pascal, операционные системы, компиляторы, искусственный интеллект, теорию баз данных) и получил начальные навыки работы на компьютере. Затем он на протяжении шести лет работал в страховой компании и в компании, создающей программное обеспечение, где специализировался на экспертных системах.

Мэдиган работал в среде мейнфреймов и написал код для выставления цен на программы страхования, при этом используя то, что впоследствии будет названо языком написания сценариев. Он также изучил графику, создав графическую презентацию системы очистки воды. Мэдиган научился управлять графическими картами ПК, но все еще не изучал данные.

Далее он получил ученую степень PhD также от Дублинского колледжа Trinity, занялся научно-педагогической работой и стал почетным «пожизненным» профессором в Университете Вашингтона. В этот период начались разработки в сфере машинного обучения и интеллектуального анализа данных, во что он просто влюбился: ко всему прочему руководил программой конференции KDD (http://www.kdd.org/kdd2013/). Мэдиган изучил языки C и Java, R и S+, но *все еще* не работал с данными.

По его собственным словам, он все еще был типичным академическим статистиком: имел навыки работы на компьютере, но все еще не представлял себе, как работать с крупномасштабной медицинской базой данных, 50 таблицами данных, разбросанными по различным базам данных различного формата.

В 2000 году Мэдиган работал в AT&T Labs. Это была «экстремальная академическая среда», где он изучил perl и имел возможность заниматься различными вещами, например веб-скрапингом. Он также изучил awk и получил базовые навыки работы в Unix.

После этого Мэдиган занялся интернет-стартапом, в рамках которого он и его команда создали систему для получения графиков активности потребителя в режиме реального времени.

Начиная с этого времени Мэдиган работает в сфере больших медицинских данных. У него был опыт дачи показаний по показаниям медицинских исследований (слово «показание» используется в данном предложении в двух значениях), что явилось для него откровением в плане объяснения проделанной работы: «Объяснить логистическую регрессию присяжным — задача иного рода, нежели стоять здесь сегодня вечером». По его словам, сверхлегкие графики помогают в этом.

Мысленный эксперимент

На сегодняшний день мы собрали детальные данные длительного наблюдения за десятками миллионов пациентов. Что мы можем делать с этими данными?

Если точнее, то у нас собраны тонны феноменологических данных, то есть индивидуальных данных медицинских историй уровня пациентов. Самые большие базы содержат записи о 80 млн людей: каждое назначенное лекарство, каждое диагностированное состояние, каждое посещение или вызов врача, результаты каждого лабораторного исследования, проведенные процедуры, и все это с метками времени.

Несмотря на все это, мы до сих пор работаем как в Средние века: бóльшая часть диагностики и назначения лечения происходит в мозгу у врача. Можем ли мы усовершенствоваться? Можем ли мы воспользоваться собранными данными для улучшения качества медицинской помощи?

Это очень важная клиническая проблема, особенно если смотреть со стороны страховой компании. Можем ли мы вмешаться и избежать госпитализации?

Так, на платформе Kaggle был размещен конкурс «Улучши здравоохранение — и выиграй 3 млн долларов». Задача конкурса состояла в четком предсказании того, кто обратится в следующем году в больницу за медицинской помощью. Однако

обратите внимание: данные были сокращены, чтобы не нарушать тайну частной жизни.

Указанный набор данных с медицинской историей 80 млн человек окружает большое количество трудных этических вопросов. Представьте, как безнравственно можно распорядиться этими данными! Вместо того чтобы помогать людям оставаться в добром здравии, мы могли бы использовать подобные модели для вычисления больных людей и предложения им программ страхования с огромными страховыми премиями или же вообще для исключения таких людей из страхования.

И это не вопрос моделирования, а скорее вопрос о том, что мы как общество хотим делать с нашими моделями.

Современная академическая статистика

По словам Мэдигана, еще около 20 лет назад академические статистики либо сидели в своих кабинетах и занимались доказательством теорем, не имея под рукой данных (зачастую они даже не знали, как запустить *t*-тест), либо сидели в своих кабинетах, придумывая новый тест или новый способ работы с недостающими данными или чем-то подобным, а затем занимались поисками набора данных, чтобы испробовать свой новый метод. В обоих случаях работа академического статистика не требовала знаний предметной области.

Однако в наши дни все по-другому. Публикации в самых авторитетных статистических журналах отличаются бóльшей глубиной с точки зрения сфер практического применения, написание научных работ подразумевает глубокое сотрудничество с людьми, занятыми в общественных или других прикладных науках. Мэдиган показывает пример путем сотрудничества с медицинским сообществом.

Мэдиган продолжил, сделав замечание относительно современного сообщества, занимающегося машинным обучением, членом которого он является или являлся: это новая сфера с конференциями и журналами, однако, с его точки зрения, для нее характерны явления, наблюдавшиеся в статистике 20 лет назад, — изобрести метод, испытать его на наборах данных. В плане знаний предметной области это шаг назад, а не вперед.

Мы не хотим сказать, что статистика идеальна: только у небольшого количества академических статистиков есть серьезные хакерские навыки, необычным контрдоказательством чего является коллега Мэдигана Марк Хансен (Mark Hansen). По мнению Мэдигана, статистикам не следует выдавать дипломы, если они не владеют подобными навыками.

Медицинская литература и исследования методом наблюдения

Возможно, вы не удивитесь, услышав, что медицинские журналы *полны* исследований методом наблюдения. Результаты таких исследований оказывают значительное воздействие на практическую медицину, на назначаемое докторами лечение и на работу регуляторов.

Например, после прочтения статьи с заголовком Oral bisphosphonates and risk of cancer of oesophagus, stomach, and colorectum: case-control analysis within a UK primary care cohort («Оральные бисфосфонаты и риск рака пищевода, желудка, ободочной и прямой кишки: анализ случай-контроля на выборке первичного выявления в Соединенном Королевстве») (https://www.ncbi.nlm.nih.gov/pubmed/20813820) (написала Джейн Грин (Jane Green) с соавторами) Мэдиган заключил, что мы видим потенциал для такой же проблемы искажающих факторов, как и более ранний пример с аспирином. В статье было вынесено заключение: риск рака увеличивался при десяти и большем количестве назначений оральных бисфосфонатов.

Статья была опубликована на первой странице газеты The New York Times, исследование проводила группа людей без очевидных конфликтов интересов, а препарат принимают миллионы людей, однако очень вероятно, что результаты неверны и противоречат результатам более поздних исследований.

И такому развитию событий есть тысячи примеров. Проблема серьезная, но люди даже не видят ее.

Миллиарды и миллиарды долларов тратятся на медицинские исследования, а от их результатов и интерпретации зависят жизни людей. Мы должны знать наверняка, есть ли смысл в таких исследованиях.

Стратификация не решает проблему искажающих факторов

Эпидемиология пытается подстроиться под потенциальные проблемы искажающих факторов. Плохая новость в том, что у эпидемиологов не очень хорошо это получается. Одна из причин заключается в следующем: наиболее часто используемые методы очень сильно полагаются на *стратификацию*, означающую разделение случаев на подслучаи и изучение этих частных случаев. Таким образом, если исследователи считают, что пол — искажающий фактор, то в статистике сделают поправку на пол — средневзвешенные показатели — это один из способов стратификации.

Однако здесь тоже есть проблема. Стратификация может сделать фундаментальные оценки причинно-следственных связей и хорошими и плохими, особенно когда в эксперименте используются небольшие по объему выборки или генеральная совокупность на самом деле не очень-то однородная.

Например, у нас ситуация, показанная в табл. 12.1. Имейте в виду, что на самом деле мы не можем «видеть» две контрафактивные колонки посередине.

Таблица 12.1. Агрегировано: мужчины и женщины

	Лечение: медикаментозное	**Лечение: контрафактивное**	**Контроль: контрафактивный**	**Контроль: без медикаментов**
Y = 1	30	20	30	20
Y = 0	70	80	70	80
P(Y = 1)	0,3	0,2	0,3	0,2

В данной ситуации в обеих группах (лечебной и контрольной) 100 человек, а причинно-следственная связь — 0,3 – 0,2 = 0,1, или 10 %.

Но, разбив эти группы по полу, мы можем создать проблемную ситуацию, особенно по мере уменьшения количеств, как показано в табл. 12.2 и 12.3.

Таблица 12.2. Стратифицировано: мужчины

	Лечение: медикаментозное	**Лечение: контрафактивное**	**Контроль: контрафактивный**	**Контроль: без медикаментов**
Y = 1	15	2	5	5
Y = 0	35	8	65	15
P(Y = 1)	0,3	0,2	0,07	0,25

Таблица 12.3. Стратифицировано: женщины

	Лечение: медикаментозное	**Лечение: контрафактивное**	**Контроль: контрафактивный**	**Контроль: без медикаментов**
Y = 1	15	18	25	15
Y = 0	35	72	5	65
P(Y = 1)	0,3	0,2	0,83	0,1875

Причинно-следственная связь для мужчин — 0,3 – 0,25 = 0,05, а для женщин — 0,3 – 0,1875 = 0,1125. Заголовок статьи может гласить, что данный препарат вызывает у женщин в два раза более сильные побочные эффекты, чем у мужчин.

Иными словами, стратификация не только не решает проблемы, но и не дает никаких гарантий, что при использовании этого подхода ваши оценки будут более точны. На самом деле, прежде чем задействовать стратификацию, вы должны

обзавестись очень хорошими доказательствами того, что такой подход окажется полезным.

Как люди поступают с искажающими факторами на практике. Несмотря на выдвинутые возражения, эксперты в этой области обязательно используют стратификацию в своих исследованиях. Они работают с искажающими переменными или даже переменными, которые нарекают потенциально искажающими, выполняя стратификацию с учетом этих переменных или прибегая к модельно-ориентированным поправкам иного рода, как, например, метод подбора контрольной группы по индексу соответствия. Таким образом, если считается, что прием аспирина потенциально является искажающим фактором, то выполняется поправка или стратификация с учетом этого фактора.

Например, в случае с данным исследованием (https://www.bmj.com/content/343/bmj.d6423), в котором изучался риск венозной тромбоэмболии, вызванной использованием оральных контрацептивов определенных видов, исследователи выбрали конкретные искажающие факторы, на которые нужно обращать внимание, и выдали следующее заключение:

> «После поправки на длительность применения лица, использовавшие оральные контрацептивы с дезогестрелом, гестоденом или дроспиреноном, подвергались как минимум в два раза большему риску венозной тромбоэмболии по сравнению с лицами, применявшими оральные контрацептивы с левоноргестрелом».

Этот доклад был обнародован на ABC и вызвал серьезный резонанс. Но не обратили бы вы внимание на такие искажающие факторы, как прием *аспирина* в данном случае? Как бы вы выбрали, на какие факторы обращать внимание, а на какие — нет? Или не привлек бы ваше внимание факт, что поведение терапевтов, назначающих эти препараты, отличается в разных ситуациях? Например, могут ли терапевты назначить более новый препарат людям с бо́льшим риском тромбообразования?

Вышло другое исследование по тому же самому вопросу и привело к другим выводам, притом использовались другие искажающие факторы. Исследователи сделали поправку на тромбообразования в анамнезе, что, в общем-то, оправданно, если задуматься. На этих примерах мы можем пронаблюдать иллюстрацию того, насколько сильно могут различаться результаты в зависимости от выбираемых поправок. Начинает казаться, что такая методология действует по принципу «пан или пропал».

Еще один пример — исследование оральных бисфосфонатов, когда были сделаны поправки на курение, алкоголь и индекс массы тела (ИМТ). Каким образом выбирались эти переменные? Фактически существуют сотни примеров, когда две команды исследователей делали радикально различающийся выбор при параллельных исследованиях.

Мэдиган и несколько соавторов протестировали эту мысль, дав нескольким эпидемиологам задачу разработать на высоком уровне пять исследований. Присутствовало низкоуровневое единообразие. Однако дополнительной проблемой стало то, что светила в данной области, услышав об этом, заявляют, что знают «правильный» способ отбора искажающих факторов.

Есть ли лучший способ?

Мэдиган и соавторы изучили 50 исследований, каждое из которых соотносится с другими по препарату и выходной паре (например, антибиотики и желудочно-кишечное кровотечение). Они провели около 5000 анализов каждой пары, а по сути, все возможные эпидемиологические исследования, причем все это было проделано на девяти различных базах данных.

Например, они зафиксировали препарат на ингибиторах АПФ, а последствие — на отеке сердца. Мэдиган и соавторы провели одинаковые исследования на девяти стандартных базах данных, самая маленькая из которых содержала записи о 4 000 000 пациентов, а самая большая — о 80 000 000.

В данном случае для одной базы данных препарат утраивает риск отека сердца, тогда как для другой — увеличивает его в шесть раз. Приведенный пример — один из лучших, хотя как минимум это *всегда плохие новости*, означающие, что данный пример правильный.

С другой стороны, для 20 из 50 пар вы можете перейти от статистически значимого результата в одном направлении до статистически значимого результата в другом. Иными словами, вы можете получить тот результат, *какой захотите*. На рис. 12.1 показана картина, где пример с отеком сердца находится сверху.

Выбор баз данных редко обсуждается или публикуется в работах по эпидемиологии.

Далее они проводили еще более расширенный тест, во время которого испытывали практически все. Иными словами, каждый раз, когда требовалось принять какое-либо решение, они выполняли тест обоими способами. Вот с какими типами решений они проводили настройку: используемая для тестирования база данных; искажающие факторы, на которые следует обращать внимание; рассматриваемый временной промежуток, что актуально для ситуации, когда у пациента случается сердечный приступ через неделю или месяц после прекращения лечения, а также учитывается ли это в исследовании.

Вот что увидели Мэдиган и соавторы: *почти все исследования могут привести к обеим точкам зрения, в зависимости от сделанных выборов*.

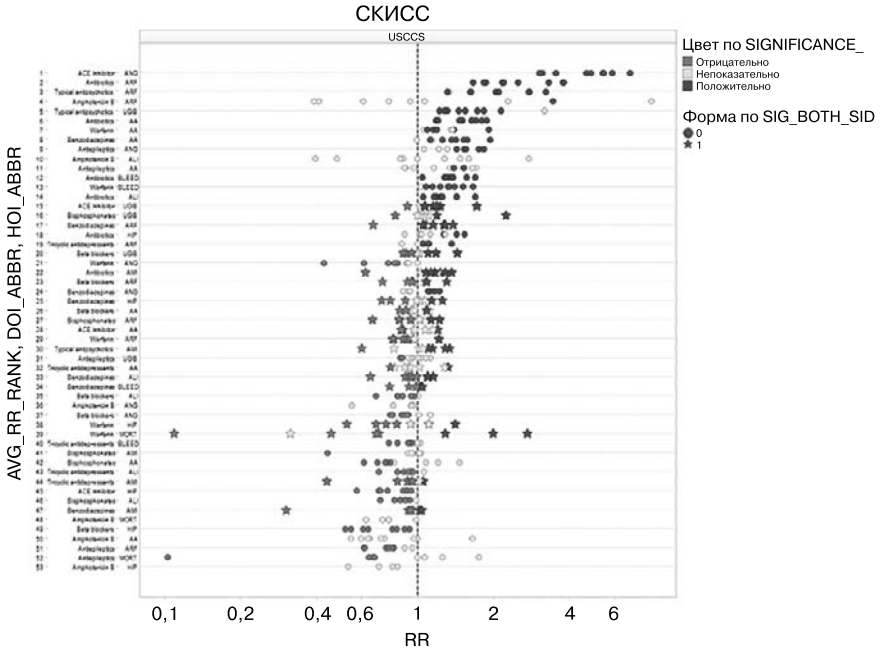


Рис. 12.1. Самоконтролируемое исследование серии случаев

Но вернемся к оральным бисфосфонатам. Определенное исследование (https://www.ncbi.nlm.nih.gov/pubmed/20813820) показало, что эти препараты вызывают рак пищевода. Однако две недели спустя в журнале JAMA была опубликована работа (https://www.ncbi.nlm.nih.gov/pubmed/20699457), посвященная той же самой проблеме, в которой авторы пришли к выводу, что прием вышеуказанных препаратов *не* связан с повышенным риском рака пищевода. Причем авторы второго исследования использовали ту же самую базу данных. Для нас это не стало сюрпризом.

Экспериментальное исследование (партнерство по наблюдению за медицинскими результатами, OMOP)

Чтобы решить данный вопрос прямо или хотя бы вывести на публику ограничения современных методов и результатов, Мэдиган работал ответственным исследователем в рамках исследовательской программы OMOP (https://www.ohdsi.org/), чем внес существенный вклад в методологическую работу по проекту, в том числе в создание, реализацию и анализ различных статистических методов применительно к разным базам наблюдательных данных.

ИНФОРМАЦИЯ ОБ OMOP С ОФИЦИАЛЬНОГО САЙТА

В 2007 году, осознавая, что увеличение использования электронных медицинских историй и доступность на рынке больших наборов медицинских данных привело к увеличению учебных возможностей, Конгресс США поручил Управлению по контролю за продуктами и лекарствами (FDA) создать новую программу надзора за применением лекарственных препаратов в целях выявления потенциальных проблем с безопасностью. Управление запустило несколько инициатив по достижению поставленной цели, в том числе программу Sentinel по созданию сети данных, охватывающую всю страну.

В партнерстве с PhRMA (Ассоциация фармацевтических исследователей и производителей) и FDA Фонд Национальных институтов здоровья запустил публично-частное Партнерство по наблюдению за медицинскими результатами (OMOP). Эта группа специалистов разного профиля взялась за решение на удивление сложной задачи, критически важной для более широких целей исследовательского сообщества: выявление наиболее надежных методов анализа больших объемов данных, полученных из гетерогенных ресурсов.

Используя целый набор подходов, применяемых в таких областях, как эпидемиология, статистика, информатика и др., OMOP пытается найти ответ на критически важные вопросы: что медицинские исследователи могут узнать, оценивая эти новые базы данных по здоровью; может ли единый подход применяться для нескольких заболеваний; можно ли доказать полученные результаты? Успех будет означать возможность для медицинского исследовательского сообщества выполнять больше исследований в более сжатые сроки, используя меньшее количество ресурсов и достигая более существенных результатов. В конце концов, успех будет означать создание лучшей системы по мониторингу лекарств, устройств и процедур, чтобы практикующее медицинское сообщество могло надежно определять риски и возможности улучшения медицинского обслуживания пациентов.

Мэдиган и коллеги использовали десять больших медицинских баз данных, состоящих из смеси заявлений страховых компаний и электронных медицинских историй в совокупности 200 млн человек. И это большие данные; конечно, если вы не разговариваете с астрономом.

Мэдиган и коллеги перенесли эти данные на общую модель данных, а затем применили все методы, используемые в рамках исследований методом наблюдения в здравоохранении. В общей сложности они задействовали 14 общепринятых эпидемиологических конструкций, адаптированных к данным продольного исследования. Исследователи автоматизировали все, что только можно было увидеть. Более того, было применено более 5000 «настроек» 14 методов.

Целью было увидеть, насколько хорошо современные методы могут предсказывать *то, что мы уже знаем*.

Для поиска уже известных фактов исследователи взяли десять классов препаратов (ингибиторы АПФ, бета-блокаторы, варфарин и т. д.) и десять интересующих эффектов (почечная недостаточность, госпитализация, кровотечение и т. д.).

Для некоторых случаев результаты были уже известны. Например, варфарин разжижает кровь и точно вызывает кровотечение. Всего было девять таких известных побочных эффектов.

Было также 44 известных «отрицательных» случая, когда исследователи очень уверены, что прием этих препаратов не нанес никакого вреда, как минимум в плане рассматриваемых побочных эффектов.

Основной эксперимент был таков: провести 5000 общепринятых эпидемиологических анализов, используя все десять баз данных. Насколько хорошо анализы позволяют отличить белое от черного? Это чем-то напоминает тестирование фильтра спама, когда у одного исследователя есть тренировочные электронные письма, заранее известные как спам, а другой хочет знать, насколько хорошо его модель выявляет нежелательные письма, когда спам прогоняется через нее.

Обе модели выводят одно и то же: относительный риск (RR) (измеряется оценкой эффекта причинно-следственной связи, о котором мы говорили ранее) и ошибку.

Исследователи предприняли попытку эмпирически оценить, насколько хорошо функционирует эпидемиология, эдакая количественная версия работы Джона Йоаннидиса (John Ioannidis) (https://alumni.stanford.edu/get/page/magazine/notfound).

Почему это не было сделано раньше?

Для эпидемиологии здесь конфликт интересов: зачем исследователи будут доказывать неработоспособность своих методов? Кроме того, это дорого: данное исследование стоило 25 млн долларов, что, конечно, бледновато в сравнении с деньгами, инвестируемыми в такие исследования.

Исследователи приобрели все данные, автоматизировали работу методов, а также выполнили большое количество вычислений в облаке Amazon cloud. Исходный код находится в открытом доступе. Во второй версии исследователи обнулили четыре конкретных результата и построили так называемую кривую ROC стоимостью 25 млн долларов (рис. 12.2).

Чтобы понять данный график, мы должны определить *пограничное значение*, начиная с 2. Это значит следующее: если риск оценивается больше чем 2, то мы станем называть состояние плохим эффектом, в противном случае — хорошим. Выбор пограничного значения, конечно же, будет иметь значение.

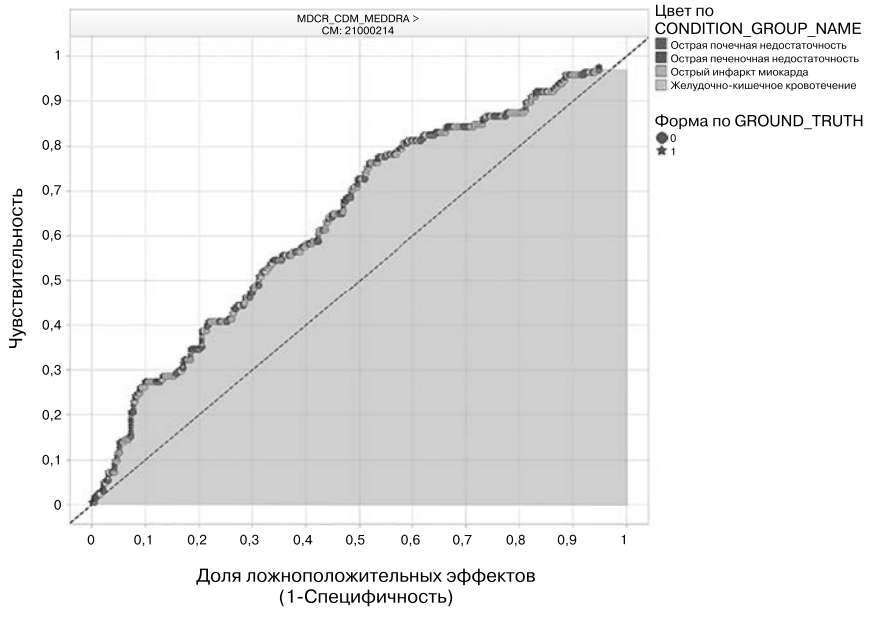


Рис. 12.2. Кривая ROC стоимостью 25 млн долларов

Если мы выберем высокое значение, например 10, то никогда не увидим значение 10, а следовательно, все эффекты будут рассматриваться как хорошие. Более того, мы рассматриваем старые препараты, выведенные с рынка, а значит, чувствительность будет низкая — и вы не найдете ни одной реальной проблемы. Это плохо! Вы должны обнаружить, например, что варфарин вызывает кровотечение.

Есть и хорошие новости: с низкой чувствительностью доля ложноположительных эффектов будет равняться нулю.

Что, если вы установите пограничное значение на действительно низком уровне? Например, –10? Тогда все плохо, у вас будет 100%-ная чувствительность, но высокая доля ложноположительных эффектов.

При изменении пограничного значения от очень низкого до очень высокого вы создаете кривую с точки зрения изменения чувствительности и доли ложноположительных эффектов, и это та кривая, которую мы видим на рисунке. Задается пограничное значение (например, 1,8), для которого доля ложноположительных эффектов и чувствительность равны 30 и 50 % соответственно.

Если вы из FDA, то для вас этот график представляет собой проблему. Доля ложноположительных эффектов 30 % не входит в диапазон параметров, считающихся допустимыми FDA.

Общая «хорошесть» такой кривой обычно измеряется площадью под кривой (AUC): желательно, чтобы она равнялась 1, а если ваша кривая лежит по диагонали, то площадь равняется 0,5. Это равнозначно случайному угадыванию. Таким образом, если площадь под кривой меньше 0,5, то ваша модель ошибочна.

AUC на рис. 12.2 равняется 0,64. Более того, из 5000 анализов, проведенных командой исследователей (включая Дэвида Мэдигана), это наилучший результат.

Однако обратите внимание: этот результат является лучшим, *только если вы можете использовать один и тот же метод для всех тестов*. В данном случае это наилучший результат, но он при этом не многим лучше угадывания.

С другой стороны, ни один эпидемиолог не сделал бы этого. Как следствие, на следующем шаге исследователи *специализировали анализ по базе данных и результату*. В итоге были получены лучшие результаты: для базы данных программы медицинского страхового обеспечения и для острой почечной недостаточности модель, предложенная исследователями, выдавала значение AUC = 0,92, как показано на рис. 12.3. Исследователи смогли достичь чувствительности 80 % при доле ложноположительных эффектов 10 %.

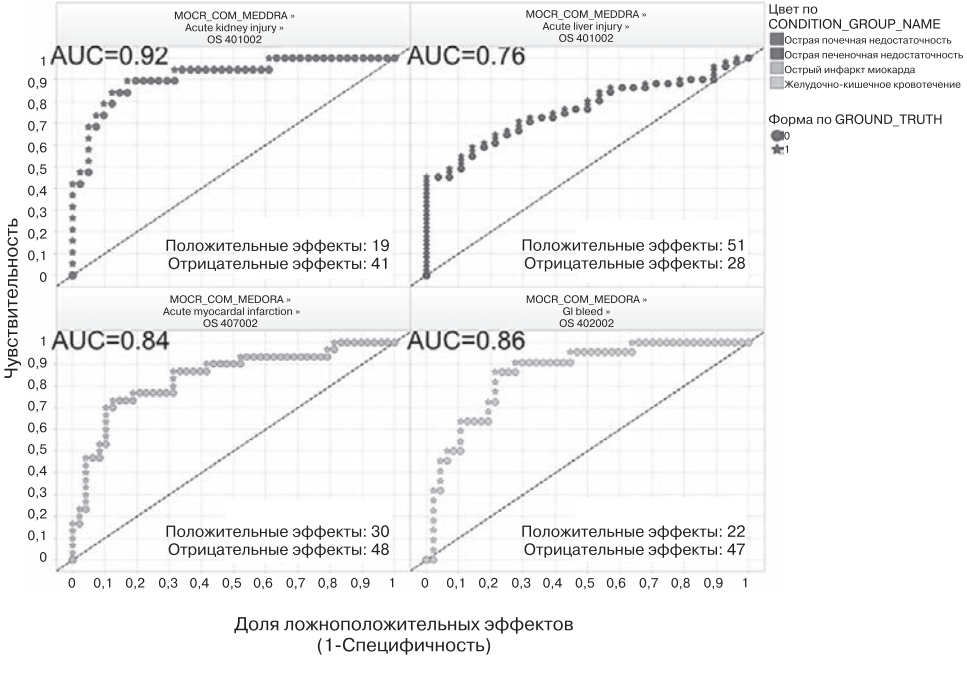


Рис. 12.3. Специализация анализа по базе данных и результату

Вышеописанное было достигнуто методом перекрестной проверки. К разным базам данных прикреплены различные методы. Одним из наилучших был признан

метод под названием OS, подразумевающий сравнение в рамках истории болезни конкретного пациента (таким образом, сравниваются параметры в период, когда пациент принимал препарат и когда не принимал). Сейчас этот анализ не используется широко.

В большинстве своем эпидемиологи не доверяют результатам этого исследования.

Если вы перейдете на сайт http://elmo.omop.org, то сможете просмотреть значения AUC для конкретной базы данных и конкретного метода. Данные, использованные в этом исследовании, были актуальны на середину 2010 года. Для их обновления вам потребовалось бы приобрести последнюю версию базы и повторно провести анализ. Ситуация, возможно, изменилась.

Завершение мысленного эксперимента

В данном исследовании было проведено 5000 различных анализов. Существует ли приемлемый способ сочетания этих анализов для получения более достоверных результатов? Что насчет включения взвешенных средних или применения методов мажоритарной выборки для различных стратегий? Описываемый программный код находится в свободном доступе и может стать хорошей темой для докторской диссертации.

13 Уроки, извлеченные из соревнований по данным: утечка данных и оценка моделей

В написание этой главы внесла вклад Клаудия Перлих (Claudia Perlich) (http://people.stern.nyu.edu/cperlich/). На протяжении последних нескольких лет она занимала должность главного научного консультанта в Media 6 Degrees (M6D) (https://dstillery.com/). До этого Клаудия работала в группе по анализу данных в центре IBM, создавшем Watson (https://en.wikipedia.org/wiki/Watson_(computer) и https://ru.wikipedia.org/wiki/IBM_Watson) — компьютер, победивший в телешоу Jeopardy! (https://ru.wikipedia.org/wiki/Jeopardy!) (хотя не работала над этим проектом). Клаудия — магистр компьютерных наук, а также имеет степень PhD по информационным системам, полученную в Нью-Йоркском университете. Сейчас она преподает студентам, изучающим бизнес, курс науки о данных, в рамках которого обучает, как оценивать работы в даталогии, а также как управлять исследователями данных.

Клаудия также является известным победителем соревнований по интеллектуальному анализу данных. Она выиграла кубок KDD в 2003, 2007, 2008 и 2009 годах, одержала победу в соревновании ILP Challenge (http://ida.felk.cvut.cz/ilp2012/) в 2005-м, INFORMS Challenge (https://www.kdnuggets.com/news/2008/n07/8i.html) в 2008-м и соревнование Kaggle (https://www.kaggle.com/) HIV в 2010-м.

В последнее время Клаудия решила стать организатором соревнований по интеллектуальному анализу данных. Свое первое соревнование она организовала для INFORMS Challenge в 2009-м, а затем для Heritage Health Prize в 2011 году. Клаудия заявляет, что больше не будет участвовать в соревнованиях. К счастью для студентов, она предоставила детальный взгляд изнутри на то, какие уроки можно вынести из соревнований по данным. Из опыта участия во многих соревнованиях она узнала немало конкретики об утечке данных, а также об оценке придумываемых моделей для соревнований.

Профиль Клаудии как исследователя данных

Клаудия начала с вопроса о том, что может служить точкой отсчета для оценки того, куда люди могут поместить свой профиль исследователей данных (ее собственный профиль приведен в табл. 13.1. Применительно к профилю исследователя данных из главы 1 Клаудия сказала так: «Здесь не хватает самого важного и наиболее сложного для описания навыка: данных». Она знакома с несколькими лучшими в мире математиками, экспертами по машинному языку, статистиками и т. д. Сопоставляет ли она себя с тем, что возможно достичь (эксперты), либо только со средним человеком, занятым в ее сфере, либо же просто со средним человеком?

Таблица 13.1. Профиль Клаудии как исследователя данных

	Удовлетворительно	**Хорошие знания**	**Твердые знания**	**Комментарий**
Визуализация	×			Я умею это делать, но не верю в визуализацию
Информатика	×			У меня две степени магистра информатики. Я могу что-нибудь взломать, но не пишу промышленный код
Математика	×			Это было очень давно
Статистика		×		Немного формального обучения, многое узнала в процессе плюс хорошая интуиция
Машинное обучение			×	
Опыт в предметной области				Вы спрашиваете не того человека...
Презентация			×	
Данные			×	

Жизнь главного исследователя данных

По имеющимся сведениям, Клаудия посвятила жизнь прогнозному моделированию, в том числе соревнованиям по интеллектуальному анализу данных, написанию работ для публикаций, конференций, например KDD, журналов, интервью, написанию патентов, преподаванию и работе с данными (ее любимая часть). Клаудия любит выяснять что-либо об этом мире, просто взглянув напрямую на данные.

Набор навыков Клаудии включает в себя 15 лет работы с данными. За это время она развила в себе интуицию к данным благодаря глубокому погружению в *процесс их генерации*. Она посвятила много времени обдумыванию процесса оценки и развитию интуиции к разработке моделей.

Основные навыки Клаудии — работа с данными с помощью таких инструментов, как Unix, sed, awk, Perl и SQL. Она создает модели, используя различные методы, в том числе логическую регрессию, метод k-БС, и, что важно, отводит много времени качественной настройке своих моделей. Около 40 % своего времени она «участник», то есть человек, работающий напрямую с данными, еще 40 % времени «посол» — человек, занимающийся написанием работ, дачей интервью и в основном внешними коммуникациями как представитель M6D, а оставшиеся 20 % она «лидер» своей группы по работе с данными.

О том, каково это: быть женщиной — исследователем данных

Быть женщиной в области науки о данных очень хорошо, так как в этой сфере интуиция не только очень полезна, но и регулярно применяется. Нос человека развивается настолько хорошо, что становится возможным *почувствовать запах* того, что некий элемент ведет себя не так; впрочем, пример несопоставим с возможностью доказать это алгоритмически. Кроме того, люди обычно помнят женщин, даже когда последние уже не помнят их. Клаудия говорит и с радостью признает, что это ей только на руку. Однако она фундаментально погружена в эту сферу благодаря тому, что хороша в ней.

Занимаясь наукой, Клаудия имеет богатый опыт в процессе публикации работ в журналах и подобных изданиях. Она обсудила, слепы ли журналы и конференции к полу человека, отправившего работу с целью опубликовать ее. На протяжении некоторого времени работы подвергались двойному слепому рецензированию, однако сейчас оно зачастую предвзятое. Более того, в 2003 году была опубликована статья Шондры Хилл (Shawndra Hill) и Фостера Провоста (Foster Provost) (https://dl.acm.org/citation.cfm?id=981001), показавшая, что можно угадать, кем написана работа, с точностью 40 %, взглянув лишь на цитации. Точность может быть еще выше, если это не первая публикация автора. Надеемся, что, будучи рецензентами, люди не *используют* такие модели, однако в любом случае результаты исследования означают, что «ослепление» рецензирования не всегда помогает. В последнее время имена не скрываются, и мы надеемся, что это не делает рецензирование менее беспристрастным. Клаудия признает, что сама немного предвзята к нескольким институтам, которые, по ее опыту, подготавливают лучшие работы.

Соревнования по интеллектуальному анализу данных

Клаудия разъяснила различия между разного рода соревнованиями по интеллектуальному анализу данных. Первый тип — так называемые стерильные соревнования, когда выдается чистая, подготовленная матрица данных, стандартная мера

погрешности и зачастую анонимизированные признаки. Это чистая проблема машинного обучения.

Примерами соревнований такого рода могут служить KDD Cup 2009, Netflix Prize и многие соревнования Kaggle. В подобных соревнованиях вы должны в своем подходе делать упор на алгоритмы и вычисления. Победителем, скорее всего, будет представивший сочетания тяжелых машинных вычислений и сложных моделей.

КУБКИ KDD CUPS

Вы можете найти все кубки KDD Cup, задания к ним и соответствующие базы данных на сайте http://www.kdd.org/kddcup/index.php. Ниже мы приводим их список.

- KDD Cup 2010: оценка успеваемости студентов.
- KDD Cup 2009: предсказание отношений с клиентом.
- KDD Cup 2008: рак груди.
- KDD Cup 2007: рекомендации для потребителей.
- KDD Cup 2006: выявление легочной эмболии по наглядным данным.
- KDD Cup 2005: категоризация запросов пользователей сети Интернет.
- KDD Cup 2004: физика частиц плюс предсказание гомологии белков.
- KDD Cup 2003: глубинный анализ сети и анализ журнала использования.
- KDD Cup 2002: документ BioMed плюс классификация ролей генов.
- KDD Cup 2001: молекулярная биоактивность плюс предсказание клеточной локализации белков.
- KDD Cup 2000: анализ истории переходов по сайту онлайн-магазина.
- KDD Cup 1999: обнаружение проникновения в компьютерную сеть.
- KDD Cup 1998: адресный маркетинг для оптимизации прибыли.
- KDD Cup 1997: адресный маркетинг для оптимизации кривой подъема.

С другой стороны, есть соревнования по интеллектуальному анализу данных из «реального мира», где вам выдаются необработанные данные (зачастую в виде большого количества таблиц, которые не так легко свести), вы настраиваете модель самостоятельно и приходите к оценкам, зависящим от задания. Соревнования такого типа более достоверно симулируют обстоятельства реальной жизни, что восходит к мысленному эксперименту Рэйчел, посвященному симуляции сумбурного опыта исследователя данных в классной комнате, упомянутого ранее в этой книге. Вам необходима практика работы с беспорядочностью.

Примерами соревнований второго типа могут служить кубки KDD Cup 2007, 2008 и 2010 годов. Если вы участвуете в соревнованиях подобного рода, то ваш подход к решению задач должен включать понимание предметной области, анализ данных

и создание модели. Победителем в таких соревнованиях может быть человек, понимающий, как обучить модель под заданный вопрос.

Клаудия предпочитает соревнования второго типа, поскольку они гораздо ближе к тому, чем вы занимаетесь в реальной жизни.

Как стать хорошим моделистом

Клаудия считает, что данные и понимание предметной области — единственные наиважнейшие навыки, которыми вы должны обладать как исследователь данных. В то же время этим навыкам невозможно обучить: они могут быть только *культивированы*.

Ниже приведены несколько посвященных интеллектуальному анализу данных уроков, которым, по мнению Клаудии, уделяется недостаточно внимания в академических кругах.

- *Утечка*. Лучший друг участника соревнований и худший кошмар организаторов и практиков. С данными всегда что-то происходит, и Клаудия даже возвела в форму искусства выяснение того, как люди, готовящие соревнования, становятся ленивыми и неаккуратными при обращении с данными.
- *Реалистичные меры эффективности*. Адаптация обучения за пределы стандартных мер оценки моделирования, например среднеквадратичная погрешность (MSE), показатель ошибочной классификации или площадь под кривой (AUC). Так, прибыль может быть примером реалистичной меры эффективности.
- *Конструирование/трансформация признаков*. Реальные данные редко плоские (то есть представленные в красивой матрице) — и хорошие практические решения для этой проблемы до сих пор остаются большой трудностью.

Утечка данных

В публикации для KDD 2011 под названием Leakage in Data Mining: Formulation, Detection, and Avoidance («Утечка в интеллектуальном анализе данных: формулировка, обнаружение и избежание») (https://dl.acm.org/citation.cfm?id=2020496), соавтором которой была Клаудия, она, Шачар Кауфман (Shachar Kaufman) и Сахарон Россет (Saharon Rosset) ссылаются на другого автора — Дориана Пайла (Dorian Pyle), написавшего множество статей по подготовке данных в интеллектуальном анализе данных. В этих публикациях он указывает на феномен, который называет анахронизмами (нечто находящееся в неправильном месте и неправильном времени), и говорит, что «слишком хорошая для правды» эффективность — «явная улика» существования такого феномена. Клаудия и соавторы называют этот феномен

утечкой данных в контексте предсказательного моделирования. Пайл предлагает обратиться к объяснительному анализу данных с целью найти и истребить источники утечки данных. Клаудия и соавторы занялись поисками строгой методологии для работы с утечками.

Термин «утечка» употребляется по отношению к информации или данным, помогающим участникам что-либо предсказать, а использовать такую информацию для предсказаний нечестно. В моделировании это огромная проблема, причем не только для соревнований. Зачастую это артефакт рокировки причины и результата. Пройдемся по нескольким примерам и почувствуем, как подобное может происходить.

Предсказания рынков

Когда-то было объявлено соревнование, где вам требовалось предсказать поведение рейтинга S&P: пойдет ли он на увеличение или снижение. AUC (площадь под кривой ROC (https://en.wikipedia.org/wiki/Receiver_operating_characteristic и https://ru.wikipedia.org/wiki/ROC-кривая)) победителя равнялась 0,999 из 1. Поскольку рынки акций характеризуются довольно случайным поведением, то либо кто-то очень богат, либо что-то пошло не так. (Подсказка: второе.)

В старые добрые времена вы могли победить в соревнованиях, найдя утечку. В данном случае неясно, что стало утечкой, и вы смогли бы об этом узнать, только досконально изучив данные и найдя некую крайне предсказуемую информацию S&P в наборе данных, притом такая информация *не* была бы доступна вам при предсказании рейтинга S&P в режиме реального времени. Мы приводим этот пример, поскольку факт, что достигнутое участниками значение AUC было настолько велико, уже говорит о том, что их модель, должно быть, основывалась на утечке и не сработала бы при реализации.

Кейс Amazon: транжиры

Целью данного соревнования было, используя историю покупок, предсказать клиентов, которые с большой долей вероятности потратят много денег. Набор данных представлял собой данные транзакций в разных категориях. Однако победившая модель, которая определила, что «Бесплатная доставка = Истина», была отличным показателем больших трат. Теперь обратите внимание: вам предлагается бесплатная доставка только *после* того, как вы потратили определенную сумму денег, например более 50 долларов.

Что здесь произошло? Смысл в том, что бесплатная доставка — *результат* больших трат. Однако это не лучший способ смоделировать большие траты, поскольку, в частности, такая модель не работает для новых клиентов или на будущие покупки. Примечание: слабостью здесь стали метки времени. Данные с выражением «Бесплатная доставка = Истина» были одномоментны с продажей, а это ошибка.

Для предсказания будущего нужно использовать только уже имеющиеся данные. Сложность здесь заключалась в том, что информация о бесплатной доставке появилась в наборе собранных данных, следовательно, эту информацию нужно было удалить вручную, что требует тщательного рассмотрения и *понимания* данных со стороны создателя модели. Если бы вы не задумывались об утечке, то вам достаточно было бы лишь ввести переменную бесплатной доставки в модель — и получить хорошо спрогнозированный результат. Однако при реализации данной модели в реальности вам не было бы известно, что клиент *вот-вот* получит бесплатную доставку.

Ювелирные изделия: проблема с выборкой

Снова образцом послужил онлайн-магазин, только в этот раз целью было предсказать клиентов, покупающих ювелирные изделия. Набор данных состоял из транзакций в различных категориях. В очень успешной модели было замечено, что если sum(revenue) = 0, то модель очень хорошо могла предсказать покупателей ювелирных изделий.

Что здесь произошло? Те, кто занимался подготовкой данных для соревнования, удалили информацию о покупке ювелирных изделий и включили только людей, которые просто уже приобрели некую вещь. Факт, что в этот набор данных попали только те, кто уже что-либо приобрел, является странным: в частности, вы не смогли бы использовать подобную модель для клиентов перед тем, как они завершат покупку. Таким образом, модель отрабатывалась на неверных данных, вследствие чего ее невозможно было сделать более полезной. Это проблема составления выборки, и она встречается часто.

ВНИМАНИЕ: ПОЛЬЗОВАТЕЛЯМ ВЫБОРОК

Как уже упоминалось, в данном случае было странным обусловливать анализ только набором людей, совершивших покупку. Вам действительно нужно ограничить свой анализ *только* людьми, которые совершили покупку, или вы хотите проанализировать *всех* людей, перешедших на ваш сайт? Говоря более пространно, в случае с данными уровня пользователей, если вы не проявите должный уровень внимательности и вдумчивости, можете сделать достаточно простую, но серьезную ошибку составления выборки. Например, представим, что вы планируете проанализировать набор данных, состоящий из трафика пользователей вашего сайта за один день. Если вы так сделаете, то относительно пользователей, часто посещающих сайт, вы создаете *выборку с запасом*.

Представьте это следующим образом: предположим, у вас 80 пользователей. Скажем, десять из них посещают ваш сайт каждый день, а остальные заходят лишь раз в неделю. Предположим, что эти пользователи равномерно распределены по семи дням недели. Вы выбираете день. Смотрите на 20 посетителей, пришедших на сайт в этот

день: десять из них посещают сайт ежедневно, а оставшиеся десять — раз в неделю. И здесь вы создаете выборку с запасом для пользователей, посещающих сайт ежедневно. Их поведение на вашем сайте может кардинально отличаться от поведения других пользователей, но при этом они составляют 50 % вашего набора данных и лишь 12,5 % вашей базы данных.

Таргетинг клиентов IBM

В компании IBM целью было предсказать компании, которые захотят приобрести решения линейки WebSphere. Набор данных состоял из транзакционных данных и результата индексирования сайтов потенциальных клиентов. Победившая модель показала, что если термин websphere встречается на сайте компании, то эта компания — отличный кандидат на приобретение данного продукта. Что случилось? Помните: при рассмотрении потенциального клиента нужно использовать определение, что компания *еще* не приобрела продукт WebSphere (иначе IBM не пыталась бы продать его той фирме). Таким образом, на сайте ни у одного *потенциального* клиента не было бы термина websphere. Следовательно, это вообще не показатель. Если бы сотрудники IBM могли попасть в прошлое и использовать тогдашнее состояние сайтов в качестве источника еще до появления продуктов WebSphere, то такие данные могли бы служить показателем. Однако современные данные содержат утечку, поскольку эти компании уже купили продукты WebSphere. К сожалению, проиндексировать *историческое* состояние сайтов невозможно, лишь их сегодняшнее состояние.

Похоже на глупую очевидную ошибку? Возможно. Но это нечто происходящее повсеместно, и вы не можете предугадать такие проблемы, пока не начнете копаться в данных и в действительности понимать *значение* признаков и показателей. Задумайтесь: если такое случилось с чем-то «очевидным», то для менее очевидных случаев необходимо еще более тщательное обдумывание и углубление. Кроме того, это пример чего-то, на что мы еще не обратили достаточного внимания в данной книге. Простой контроль корректности порой может продвинуть вас намного дальше, чем агрегация данных из веба или большой модный машинный алгоритм. Такой подход может показаться невзрачным, но служит примером разумной и хорошей практики. Люди, возможно, не пригласят вас на встречу, чтобы обсудить данный подход. Это может не быть исследованием, пригодным для публикации, но хотя бы является правильной и солидной работой. (Впрочем, опять же благодаря такой практике Клаудия одержала победу в десятках соревнований и все время получает приглашения на различные встречи. Поэтому мы берем свои слова обратно. Хотя нет, не берем. Смысл в том, чтобы хорошо делать свою работу, а остальное приложится. Встречи и слава сами по себе не суть цели, цель — поиск истины.)

Выявление рака груди

Мы пытаемся изучить, у кого рак груди. Взгляните на рис. 13.1. Идентификатор пациента, хоть и кажется безобидным, на самом деле имеет прогностическую силу. Что произошло?

На рис. 13.1 темно-серый (в оригинале — красный) означает наличие рака, светло-серый (зеленый) — отсутствие. Данные разбиты по идентификаторам пациентов. Мы можем четко наблюдать три или четыре интервала идентификаторов пациентов. Результат очень предсказуем в зависимости от интервала. Возможно, это последствие использования нескольких баз данных, относящихся к разным онкоцентрам, пациенты ряда из них пребывают в более тяжелых состояниях: по определению пациенты, которым назначают посещение этих центров, с большей вероятностью больны раком.

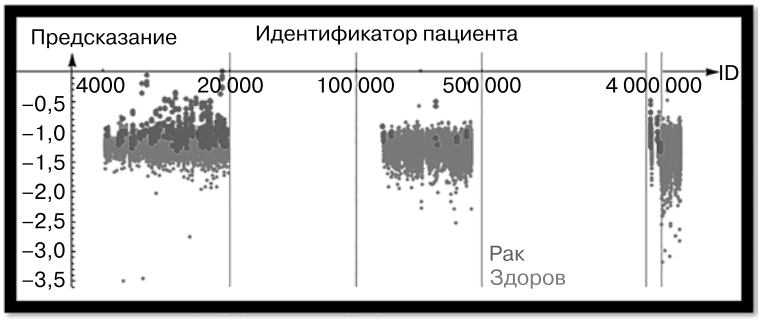


Рис. 13.1. Пациенты, упорядоченные по идентификатору; красный означает наличие рака, зеленый — отсутствие

Данная ситуация привела к интересной дискуссии на занятии.

Студент: Для целей данного соревнования им следовало бы перенумеровать и перемешать пациентов.

Клаудия: Это решило бы проблему? Пациенты могли иметь и другие общие признаки.

Другой студент: Важным вопросом может быть выяснение того, насколько точно мы можем сказать, из какой базы данных получили сведения о конкретном пациенте, используя данные, помимо ID пациента.

Клаудия: Задумайтесь вот над чем: для чего именно нам изначально нужны эти модели? Как точно мы *в действительности* можем предсказать рак?

Получив нового пациента, что бы сделали *вы*? Если по идентификатору новый пациент попадает в пятый интервал, то вы не захотели бы использовать модель с идентификаторами. Но если идентификатор попадает в вышеприведенную схему, то, возможно, модель с идентификаторами и есть наилучший подход.

Эта дискуссия поднимает фундаментальный вопрос: чтобы построить модель и определить ее работоспособность, нам нужно знать цель создания такой модели и то, каким образом она будет использоваться.

Предсказание пневмонии

Во время соревнования INFORMS по предсказанию пневмонии по госпитальным историям болезни, где цель была предсказать, болен ли пациент пневмонией, логистическая регрессия, включавшая номер диагноза в качестве числовой характеристики (AUC = 0,80) сработала хуже, чем логистическая регрессия, включавшая номер диагноза в качестве категорийного признака (https://en.wikipedia.org/wiki/Categorical_variable) (0,90). Что произошло?

Здесь причина крылась в том, как человек приготовил данные для соревнования (рис. 13.2).

icd1x	icd2x	icd3x	icd4x
786	285	459	–1
401	486	–1	–1
401	486	401	–1
599	–1	–1	–1
V22	650	–1	–1
V56	492	586	–1
786	493	285	459

icd1x	icd2x	icd3x	icd4x
786	285	459	–1
401	–1	–1	–1
401	780	–1	–1
599	–1	–1	–1
V22	650	–1	–1
V56	492	586	–1
786	493	285	459

Рис. 13.2. Как были приготовлены данные для соревнования INFORMS

Код диагноза «пневмония» 486, поэтому человек, ответственный за подготовку данных, удалил его (и заменил его на –1), если такой код появлялся в истории болезни (строки соответствуют пациентам, а колонки — диагнозам, максимальное количество диагнозов — четыре, –1 означает отсутствие данных в записи).

Более того, во избежание выдачи пробелов в данных, человек, ответственный за подготовку, сдвинул при необходимости другие диагнозы влево, таким образом, код –1 встречался только в правой части.

Описанный подход вызвал две проблемы:

- если в строке есть только коды –1, то вы знаете, что строка началась только с пневмонии;

- если в строке нет кодов –1, то вы знаете, что пневмония не выявлялась (при условии отсутствия пяти диагнозов, но такое встречается менее часто).

Знания этого факта уже было достаточно для победы на соревнованиях.

Утечки случаются

Одержать победу на соревновании, воспользовавшись утечкой, проще, чем создать хорошие модели. Но даже если вы явно не осознаете и не применяете утечку, то ваша модель сделает это за вас. В любом случае утечка представляет собой огромную проблему для соревнований по интеллектуальному анализу данных в целом.

Как избежать утечки

Эта глава не о том, как победить в прогностическом соревновании по моделированию. Реальность такова, что, как исследователь данных, вы постоянно подвержены риску создания утечки данных в процессе подготовки, чистки данных, приписывания отсутствующих значений, удаления выбросов и пр. Подготавливая данные, вы можете настолько их исказить, что создадите модель, которая станет отлично работать на вашем «чистом» наборе данных, но будет абсолютно непригодна при использовании в реальной ситуации, то есть именно тогда, когда будет нужна. Клаудия дала несколько очень конкретных советов, как избежать утечки. Во-первых, вам необходимо провести строгий временной срез и избавиться от всей информации, полученной до наступления интересующего события. Например, то, что вы знали, *прежде чем пациент был принят на лечение*. Каждая запись должна сопровождаться меткой времени, указывающей, когда вы узнали эту информацию, а не на то, когда произошло событие. Удаление столбцов и строк из таблицы данных может привести к неприятностям, особенно в форме выявляемых несоответствий и противоречий. Лучше всего начинать с нефильтрованных, необработанных данных и их тщательного осмысления. Наконец, вам необходимо знать, каким образом были созданы данные!

В упомянутой ранее работе Клаудия и соавторы описывают предлагаемую методологию избежания утечек, представляющую собой процесс, который протекает в два этапа: пометка каждого наблюдения маркерами правомочности во время сбора данных, а затем наблюдение того, что авторы называют разделением обучения и предсказания.

Оценка моделей

Как понять, насколько хороша ваша модель? В предыдущих главах мы уже обсуждали это, но никогда не будет лишним услышать информацию еще раз от гуру.

Использование мощных алгоритмов для поиска закономерностей моделей всегда сопутствует серьезной опасности чрезмерно близкой подгонки. Это сложная концепция, но общий смысл таков: «при тщательном поиске всегда можно что-нибудь найти», даже если процесс не сопровождается упрощением за пределами конкретных обучающих данных.

Во избежание чрезмерно близкой подгонки мы проводим перекрестную проверку и изначально снижаем сложность модели. На рис. 13.3 изображена стандартная ситуация (впрочем, имейте в виду, что обычно мы работаем в многомерном пространстве и у нас нет приятного рисунка, на который можно было бы взглянуть).

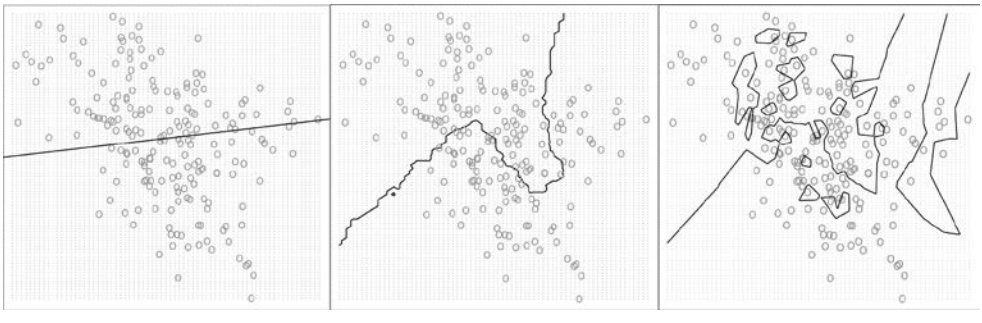


Рис. 13.3. Этот классический рисунок из работы Elements of Statistical Learning («Элементы статистического обучения») Хасти (Hastie) и Тибширани (Tibshirani) (https://web.stanford.edu/~hastie/ElemStatLearn/), изданной в Springer-Verlag, демонстрирует подгонку линейной регрессии к дихотомическому отклику, подгонку 15 ближайших соседей и подгонку одного ближайшего соседа — и все это на одном и том же наборе данных

Рисунок слева — недостаточная подгонка, в центре — хорошая подгонка, а справа — чрезмерно близкая подгонка.

Как показано на рис. 13.4, когда речь заходит о чрезмерно близкой подгонке, имеет значение, какую модель вы используете.

Взглянув на рис. 13.4, можно сделать вывод, что деревья решений без отсечения ветвей наиболее подвержены чрезмерно близкой подгонке. Это хорошо известная проблема деревьев решений без отсечения ветвей, вследствие чего предпочтительно использовать деревья решений с отсечением ветвей.

Точность: фи

Один из параметров оценки модели, которые мы обсуждали в данной книге, — это точность как средство оценки проблем классификации и, в частности, проблем дихотомической классификации. Клаудия отклоняет точность, поскольку это плохой метод оценки. Что не так с точностью? Очевидно, она не подходит не только для регрессии, но даже и для классификации, когда абсолютное большинство резуль-

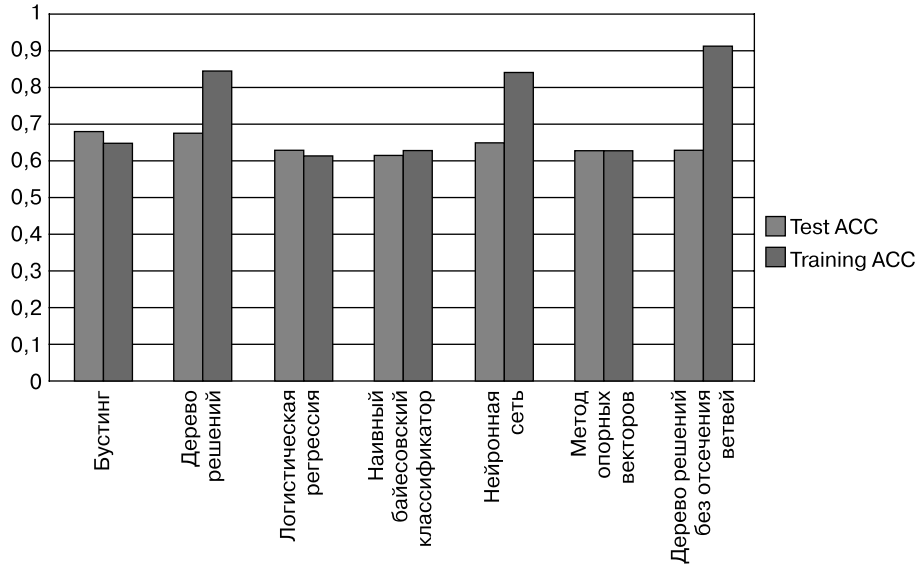


Рис. 13.4. То, какую модель вы используете, имеет значение!

татов — это 1. В подобном случае глупая модель может быть точной, но нехорошей («полагаю, результат всегда 1»), а более хорошая модель способна обладать более низкой точностью.

Вероятности имеют значение, а не 0 и 1

Никто не принимает решение, опираясь только на *сами* бинарные результаты. Так, вам нужно узнать *вероятность* развития у вас рака груди, а не получить ответ «да» или «нет». Чтобы узнать вероятность, требуется бо́льшее количество информации. Людям *важны* вероятности.

Как же, по мнению Клаудии, следует проводить оценку? Клаудия — сторонник раздельной оценки долей и калибровки. Для оценки долей мы строим кривую ROC и вычисляем площадь под ней, значение которой, как правило, находится в диапазоне от 0,5 до 1,0. Это значение не зависит от калибровки и масштабирования. На рис. 13.5 показан пример того, как нарисовать кривую ROC.

Иногда для измерения долей рисуют так называемую кривую подъема (рис. 13.6).

Ключевой момент заключается в том, что подъем здесь вычисляется с учетом базовой линии, которую рисуют на заданной точке, скажем, 10 %. Представим, что 10 % людей показывают рекламу, и пронаблюдаем, сколько людей щелкнут на этой рекламе по сравнению с ситуацией, когда вы случайным образом показываете рекламу 10 % людей. Подъем 3 означает улучшение в три раза.

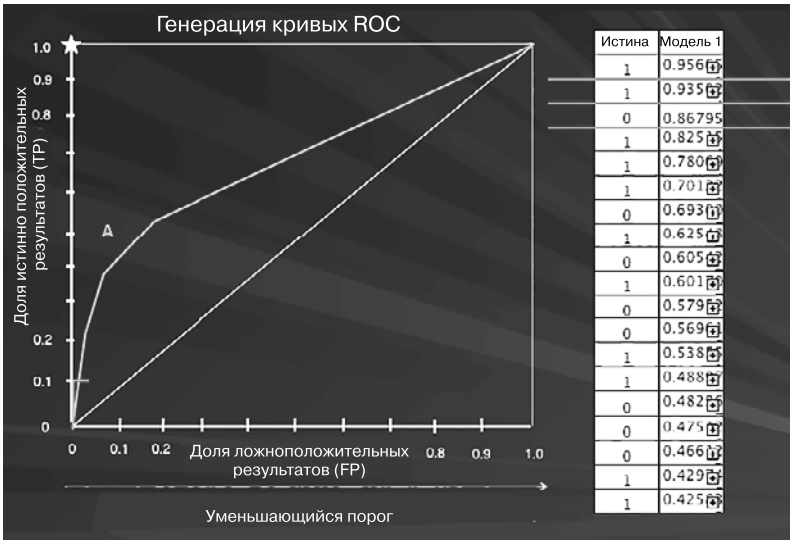


Рис. 13.5. Пример того, как нарисовать кривую ROC

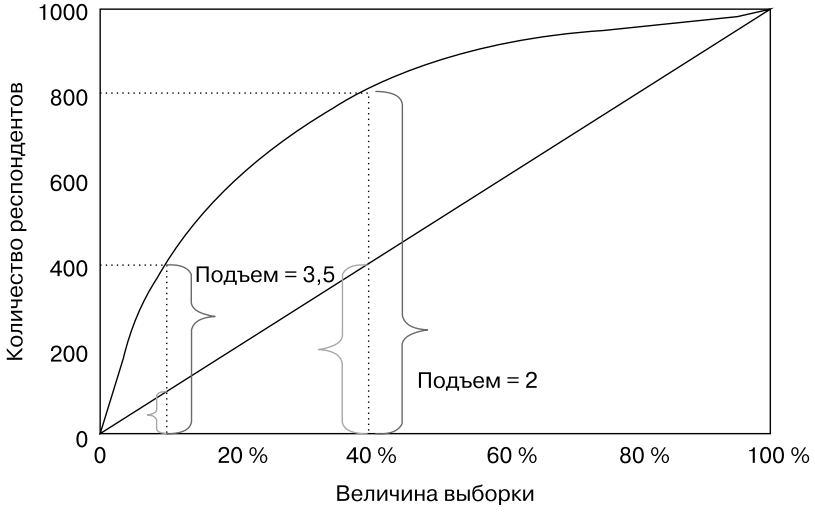


Рис. 13.6. Так называемая кривая подъема

Как измерить калибровку? Точны ли вероятности? Если модель сообщает: вероятность того, что у меня рак, равна 0,57, откуда мне знать, что вероятность действительно 0,57? Мы не можем измерить это напрямую. Мы можем лишь сгруппировать такие предсказания, а затем выполнить совокупное сравнение прогнозной группы (скажем, 0,50–0,55) с действительными результатами для данной группы.

Например, взгляните на рис. 13.7, на котором показано то, что вы получаете, если ваша модель — дерево решений без отсечения ветвей.

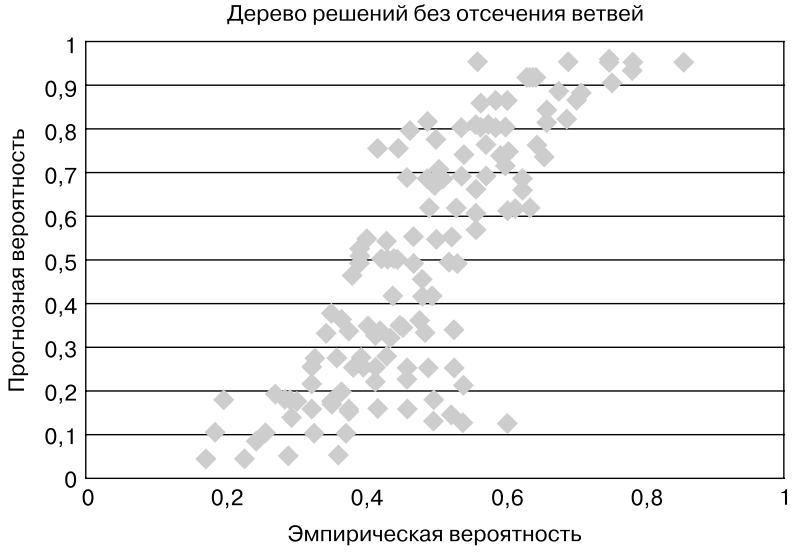


Рис. 13.7. Один из способов измерить калибровку — сгруппировать предсказания и нанести на график прогнозную вероятность в сравнении с эмпирической вероятностью для каждой группы. Здесь мы выполнили это для дерева решений без отсечения ветвей

Ромбики — это группы людей. Значение по оси X — эмпирическая, наблюдаемая подгруппа людей с диагностированным раком в этой группе в качестве примера, тогда как значение по оси Y — среднее прогнозное значение для этого набора людей, полученное с применением дерева решений без отсечения ветвей. Данный рисунок показывает, что деревья решений без отсечения ветвей в целом не очень хорошо справляются с калибровкой.

Хорошая модель показала бы, что группы распределяются вдоль кривой $x = y$, но здесь мы видим следующее: предсказанные значения были гораздо более экстремальными по сравнению с действительными вероятностями. Почему такая закономерность случается с деревьями решений?

Клаудия утверждает, что деревья стремятся к максимальной чистоте: выискивают очаги только с положительными или отрицательными результатами. Поэтому их прогнозы обычно ударяются в крайности, по сравнению с реальным положением дел. В целом про деревья принятия решений можно сказать, что при проверке по эталону они показывают плохие результаты.

Логистическая регрессия обычно выглядит лучше при тестировании калибровки, как показано на рис. 13.8.

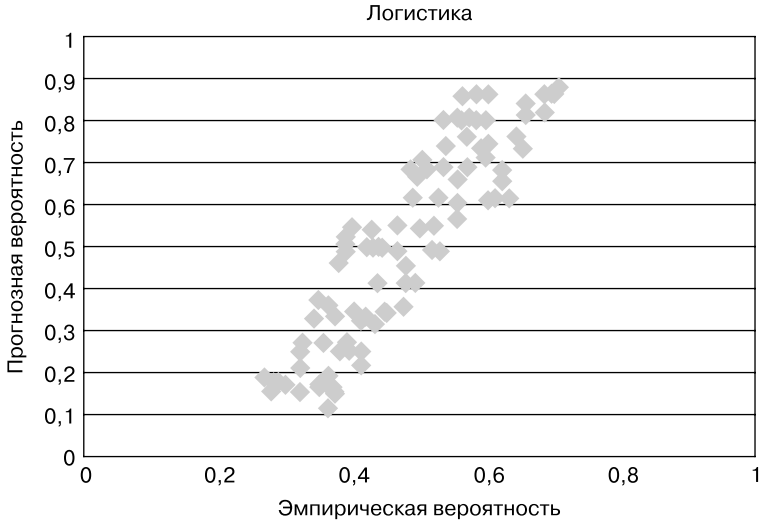


Рис. 13.8. Тестирование калибровки для логистической регрессии

Ромбики, как и ранее, — это группы людей. Данный рисунок показывает, что логистическая регрессия работает лучше при учете калибровки.

Выбор алгоритма

Это нетривиальный вопрос и, в частности, тесты, проводимые на небольших наборах данных, могут ввести вас в заблуждение, поскольку при росте величины выборки, возможно, придется выбирать другой алгоритм. Деревья решения часто работают очень хорошо, но при условии наличия достаточного количества данных.

Обычно выбор алгоритма зависит от размера и природы вашего набора данных. Выбор метода оценки должен частично зависеть от ваших данных и от того, в чем, по вашему мнению, модель должна быть хороша. Если имеющиеся у вас данные можно считать нормальными, то сумма квадратичной погрешности — это функция потерь с наибольшей вероятностью, но при желании оценить медиану вам следует использовать абсолютные погрешности. Для оценки квантиля минимизируйте взвешенную абсолютную погрешность.

Работая над соревнованием по предсказанию рейтингов, которые фильм получит в следующем году, мы воспользовались законом распределения Пуассона. В данном случае наш метод оценки не подразумевал минимизацию суммы квадратичных погрешностей: мы применили нечто иное, найденное нами в литературе по закону распределения Пуассона, зависящего от единственного параметра — λ.

Таким образом, для поиска пригодной для вашей модели оценочной метрики иногда приходится покопаться в литературе.

Последний пример

Сведем все воедино.

Скажем, вам нужно собрать деньги на благотворительность. Если вы отправите письмо всем людям в списке рассылки, то соберете около 9000 долларов. Но вы хотели бы сэкономить деньги, отправив письма только тем, кто внесет пожертвование с наибольшей вероятностью: обычно жертвуют только 5 % людей. Каким образом можно определить, кто они?

Если вы воспользуетесь деревом решений (по стандарту с каким-то отсечением), то вы получите доход 0 долларов: дерево не сможет найти и листика с преобладанием положительных результатов.

Применив нейронную сеть, вы заработаете около 7500 долларов, даже если отправите письма только в тех случаях, когда ожидаете, что выгода превысит себестоимость.

Разобьем эту задачу еще больше. Получив письмо, люди принимают два решения: во-первых, решают, вносить пожертвование или нет, а затем решают, сколько можно внести. Эти два решения можно смоделировать по отдельности:

$$E(\$ \mid человек) = P(ответ = `да` \mid человек) \cdot E(\$ \mid ответ = `да`, человек).$$

Обратите внимание: первая модель должна быть хорошо откалибрована, поскольку вам важно число, а не только ранжирование. Поэтому для первой части можно попробовать логистическую регрессию. Для второй подойдет проведение обучения с помощью специальных примеров, когда пожертвования *были внесены*.

В общей сложности продемонстрированная модель приносит прибыль 15 000 долларов. Разделение упростило для модели перехват сигналов. Обратите внимание: с бесконечными данными все было бы замечательно и вам не потребовалось бы проводить разделение. Но вы работаете с тем, что есть.

Более того, используя данный подход, вы перемножаете ошибки; это может представлять собой проблему, если у вас есть причина полагать, что они коррелированные.

Финальные мысли

По мнению Клаудии, люди не созданы для понимания данных. Последние находятся вне нашей системы чувств, и только небольшое количество людей обладают

почти сенсорной связью с числами. Вместо этого мы созданы для понимания языка.

Мы также не созданы для понимания неопределенности: у нас есть различные особенности, предотвращающие хорошее документирование нас самих. Таким образом, моделирование людей в будущем станет объективно сложнее выяснения того, какую метку присвоить уже произошедшим событиям.

Но несмотря на все это, мы стараемся добиться как можно лучших результатов, тщательно генерируя данные, педантично выясняя, в чем состоит задача, моделируя с помощью данных, как можно более близких к тем, которыми модель будет использоваться, а также следя за тем, что производимая оптимизация соответствует нашим желаниям. Кроме того, мы делаем домашнюю работу, изучая, какие алгоритмы подходят к тем или иным задачам.

14 Проектирование данных: MapReduce, Pregel и Hadoop

В написании этой главы участвовали также Дэвид Кроушоу (David Crawshaw) и Джош Уиллс (Josh Wills). Рэйчел работала с ними обоими в компании Google, в команде исследователей данных Google+, хотя эти двое никогда не работали *вместе*, поскольку Джош Уиллс перешел в компанию Cloudera, и Дэвид Кроушоу заменил его на должности технического руководителя проекта. Их можно назвать инженерами-аналитиками (data engineer), хотя этот термин столь же проблематичен (и, вероятно, перегружен), как и «исследователь данных» (data scientist). Впрочем, достаточно отметить, что оба они работали разработчиками ПО и имели дело с большими объемами данных. Говоря в терминах приведенного в главе 2 технологического процесса исследования данных, Джош и Дэвид отвечали в Google за сбор данных (журналирование клиентской и прикладной части), создание масштабных конвейеров данных, предназначенных для хранения и очистки данных, а также за построение технической инфраструктуры для анализа данных, информационных панелей, аналитики, A/B-тестирования и исследования данных в более широком смысле.

В этой главе узнаем из первых рук, от инженеров компании Google, о разработанном там MapReduce, а затем и о версиях с открытым исходным кодом, созданных другими разработчиками. MapReduce — это алгоритм и фреймворк для обработки больших объемов данных, ставший в последнее время очень популярным. Задача этой главы — немного приоткрыть над ним завесу таинственности. MapReduce стало очень «модным словечком», и во множестве объявлений о вакансиях исследователей данных пишут «знания Hadoop обязательны» (реализация MapReduce с открытым исходным кодом). Мы подозреваем, что эти объявления написаны теми HR-специалистами, которые не понимают на самом деле, для чего MapReduce подходит, и не осознают, что он *требуется* отнюдь не для *всех* задач науки о данных. Но раз уж MapReduce стал настолько неотъемлемым понятием науки о данных,

не помешает объяснить, что это такое и откуда взялось. Знать его нужно, хотя вы не обязаны использовать этот фреймворк — или обязаны, в зависимости от того, чем вы занимаетесь.

ОБЯЗАТЕЛЬНО ЛИ ЗНАТЬ MAPREDUCE, ЧТОБЫ БЫТЬ ИССЛЕДОВАТЕЛЕМ ДАННЫХ?

Забавно было бы посчитать на какой-нибудь конференции, сколько раз люди упомянули слово MapReduce, а затем попросить их объяснить, что данное слово значит на самом деле, и посчитать, сколько из них смогут это сделать. Подозреваем, что немногие. Даже нам пришлось обратиться к экспертам, проведшим в работе с ним бесчисленные часы. В компании Google Рэйчел *приходилось* писать код на языке программирования Sawzall, в основе которого лежит фреймворк MapReduce, с помощью которого производится обработка и очистка данных в целях приведения их в пригодный для анализа и создания прототипов вид. Что касается Кэти, то, работая исследователем данных в компании Intent Media, она использовала версию языка Sawzall с открытым исходным кодом — Pig, точнее, Pig в сцепке с Python в рамках фреймворка Mortar Data. Поэтому мы, пусть и косвенным образом, обе применяли MapReduce и кое-что о нем знаем, хотя и не до такой степени, как те, кто его создавал.

Кроме того, стоит говорить о MapReduce еще и потому, что он иллюстрирует типы алгоритмов, используемые для решения инженерных и инфраструктурных задач при больших объемах данных. Это третий пункт в списке категорий алгоритмов, приведенном нами в главе 3 (два остальных — алгоритмы машинного обучения и алгоритмы оптимизации). В качестве точки отсчета для тех, кому алгоритмическое мышление может быть в новинку, мы опишем еще один алгоритм и фреймворк для проектирования данных, Pregel, предоставляющий возможности широкомасштабных вычислений на графах (это также проект с открытым исходным кодом, который был разработан в компании Google).

О Дэвиде Кроушоу

Дэвид Кроушоу (https://plus.google.com/101314078672406559974/posts) — разработчик программного обеспечения в компании Google, известный тем, что однажды случайно удалил 10 петабайт данных из-за неудачно написанного сценария командной оболочки. К счастью, у него была резервная копия. По образованию Дэвид математик, он работал с Рэйчел над проектом Google+ в Калифорнии, а сейчас создает инфраструктуру для усовершенствования интеллектуального поиска. Он недавно переехал из Сан-Франциско в Нью-Йорк.

Дэвид расскажет о MapReduce и о том, как справиться со слишком большими объемами данных. Прежде чем перейти к этой дискуссии, выполним небольшой мысленный эксперимент.

Мысленный эксперимент

Рассмотрим вопрос конфиденциальности применительно к доступу к медицинским картам.

С одной стороны, следует очень серьезно относиться к вопросам конфиденциальности, когда речь идет о медицинских картах — крайне нежелательно, чтобы любой мог читать чью-либо историю болезни. С другой стороны, определенные виды доступа могут спасти чью-то жизнь.

По некоторым оценкам, в одном маленьком городке от одного до двух пациентов в неделю умирает из-за отсутствия обмена информацией между отделением неотложной помощи больницы и расположенной неподалеку психиатрической больницей[1]. С другой стороны, если бы можно было с легкостью находить соответствующие медицинские карты, то вероятность и других нарушений медицинской тайны повысилась бы. Конечно, даже не зная, сколько именно жизней на кону, принять подобное решение непросто.

Вследствие этого возникают естественные вопросы: какой уровень конфиденциальности в медицине приемлем? У кого должен быть доступ к вашей истории болезни и при каких обстоятельствах?

Можно считать, что конфиденциальность в целом положительная вещь. Например, в ряде стран атеистические воззрения караются смертной казнью, так что лучше в таких странах об этом молчать. Но за конфиденциальность тоже можно поплатиться головой, как мы видим из вышеприведенной истории о смертях в отделении неотложной помощи.

Взглянем и на другие примеры. Известны случаи вопиющих нарушений в работе патрульной полиции, когда люди, имеющие доступ к большой базе данных номерных знаков автомобилей, злоупотребляют своим положением. Хотя можно поспорить, что в данном случае речь идет о *человеческом* факторе, а не о технической проблеме.

Все вышесказанное можно также сформулировать в виде философской проблемы: до какой степени допустимо принимать решения за других?

Остается и вопрос мотивации. Возможно, мы могли бы вылечить рак быстрее при больших количествах медицинских данных, но нельзя не давать лекарство тем, кто не хочет делиться своими данными.

И наконец, для конкретного человека конфиденциальность можно рассматривать как вопрос безопасности. Обычно людей не особо волнует, что у кого-то есть

[1] Эндрю Джельман считает эту «притчу» маловероятной, его отклик вы можете прочитать на странице http://andrewgelman.com/2014/01/24/parables-vs-data/.

информация о них; это начинает их волновать, только если ее можно использовать против них и/или в целях их идентификации.

Возвращаясь к вопросу технологических возможностей: сделать данные полностью деперсонифицированными чрезвычайно сложно. Например, недавнее исследование журнала Nature *Unique in the Crowd: the privacy bounds of human mobility* («Уникальность в толпе: степень конфиденциальности информации о перемещении людей») Ива-Александра де Монтжоя и др. (Yves-Alexandre de Montjoye et al.) (https://www.nature.com/articles/srep01376), проведенное на наборе данных 1,5 млн пользователей мобильных телефонов, показало, что всего лишь четырех точек привязки было достаточно для идентификации 95 % людей.

Недавно мы наблюдали, как люди восприняли в штыки сбор информации о гражданах США агентством NSA (не говоря уже о негражданах США). Когда данная книга готовилась к печати, как раз и произошла утечка данных, инициированная Эдвардом Сноуденом (Edward Snowden). Результатом были громкие публичные дебаты по поводу права на неприкосновенность частной жизни со стороны правительства. Учитывая, сколько информации об отдельных людях продается и покупается в Интернете через торговцев данными вроде Acxiom (не только маркетинговой, но и информации о страховках, карьере и кредитной истории), не помешает поговорить на ту же тему применительно к частным компаниям.

MapReduce

В этом разделе мы узнаем про обычный ход мыслей Дэвида как инженера.

Он задает себе вопрос, который мы уже поднимали в этой книге: *что такое* большие данные? По большей части это просто «модное словечко», но иногда оно может быть полезным. Дэвид использует в качестве рабочего определения следующее.

Если данные, с которыми вы работаете, не помещаются на вашем вычислительном устройстве, то можно рассматривать их как большие данные. Отметим, что согласно этому определению большие данные существуют уже очень давно. Налоговое управление США собирало данные по налогам задолго до появления компьютеров, и в полном соответствии с нашим определением имеющиеся у них данные не помещались на их (несуществующем) вычислительном устройстве.

На сегодняшний день большие данные означают данные, не помещающиеся на одном компьютере. Даже при таком определении размеры этих данных быстро меняются. Ресурсы компьютеров в последние 40 лет растут экспоненциально. И их ждет еще как минимум десять лет экспоненциального роста (причем десять лет назад говорили то же самое).

С учетом вышесказанного, собираются ли большие данные исчезнуть из нашего поля зрения? Можно ли игнорировать это понятие?

Дэвид утверждает, что нет, поскольку, хотя ресурсы отдельного компьютера и растут экспоненциально, эти же самые компьютеры и *продуцируют* данные. Как следствие, скорость появления новых данных тоже растет экспоненциально. Фактически речь идет о двух экспоненциальных кривых, которые в ближайшее время явно не пересекутся.

Что ж, посмотрим на это на примере.

Задача подсчета частот слов

Предположим, нам нужно найти наиболее часто встречающиеся слова в следующем списке: red, green, bird, blue, green, red, red.

Легче всего решить эту задачу, просмотрев список, конечно. Но что делать, если в списке 10 000, 100 000 или 10^9 слов?

Простейший подход: составить список слов и подсчитать частоту появления каждого слова в нем. На рис. 14.1 показан пример фрагмента кода на обожаемом Дэвидом языке Go, который он помогал проектировать и создавать в компании Google.

```
func count (words []string) {
        counts:- map[string]int{}
        for _, word:= range words{
                counts[word] +=1
        }
        printSorted(counts)
}
```

Рис. 14.1. Пример фрагмента кода на языке Go. В силу быстроты подсчета и сортировки этот пример масштабируется до списка размером примерно 100 млн слов. Теперь нас ограничивает только оперативная память компьютера — если подумать, то станет понятно, что слова нужно разместить в памяти дважды: один раз при загрузке списка всех слов, а потом еще раз, когда мы будем соотносить каждое слово с количеством его вхождений

Можно слегка модифицировать этот пример, чтобы избежать необходимости загружать все слова в оперативную память: держать их на диске и организовать их потоковую обработку с помощью канала (https://ru.wikipedia.org/wiki/Канал_(программирование)) вместо списка. Канал — нечто вроде потока данных: сначала читаются первые 100 элементов, обрабатываются, затем читаются следующие 100 элементов.

Но все равно остается *потенциальная* проблема: если список очень велик и все слова в нем уникальны, то выполнение программы все равно завершится аварийным сбоем; список просто слишком велик для имеющегося объема памяти. С другой стороны, *возможно*, практически во всех случаях все будет работать нормально, ведь *почти всегда* слова будут повторяться. На практике программирование — запутанная вещь.

Постойте-ка, ведь сегодня компьютеры — многоядерные машины, давайте использовать все ядра процессоров! Тогда проблемой будет уже полоса пропускания, так что выполним и сжатие входных данных. Это поможет решить проблему. Кроме того, есть и другие, лучшие, но и более сложные варианты. Можно воспользоваться кучей (https://ru.wikipedia.org/wiki/Куча_(структура_данных)) с хешируемыми значениями, размеры которой ограничены, а поведение более предсказуемо. Куча представляет собой некий аналог частично упорядоченного множества, причем очень маленькие элементы можно отбросить, чтобы не держать все в оперативной памяти. Это будет работать в большинстве случаев, хотя и не всегда.

Вы успеваете следить?

Понимать все эти нюансы вам не обязательно, но нам хотелось бы, чтобы вы почувствовали, почему MapReduce просто необходим.

Теперь нам достаточно одного компьютера для обработки массива порядка 10 трлн слов.

Допустим, у нас есть десять машин. Это дает возможность обработать 100 трлн слов. Каждый компьютер получит 1/10 часть входных данных и подсчитает частоту вхождений по своей части слов. Затем каждый из них отправит результаты на одну машину-контроллер. Тот просуммирует их и найдет максимальное значение, которое и будет решением нашей задачи.

Можно сделать это и на основе хешируемых куч, если сначала изучить сетевое программирование.

Теперь возьмем сотню компьютеров. С их помощью можно обрабатывать *тысячу триллионов* слов. Но при объединении на входе, при отправке результатов контроллеру все перестанет работать из-за проблемы с полосой пропускания. Необходимо дерево, в котором каждая группа из десяти машин станет отправлять данные на локальный контроллер, а все они затем будут отправлять данные на главный контроллер.

Но... будет ли такая схема работать при 1000 машин? Нет. При таких масштабах неизбежно произойдет сбой одного или нескольких компьютеров. Если обозначить буквой X переменную, отражающую, работает ли данный компьютер, где $X = 0$ означает, что работает, а $X = 1$ — что поломан, то получим:

$$P(X = 0) = 1 - \epsilon.$$

Но это значит следующее: при 1000 компьютеров вероятность того, что ни один компьютер не сломан, равна:

$$(1 - \epsilon)^{1000}.$$

Это обычно очень маленькое значение даже при малом ϵ. Так, если $\epsilon = 0{,}001$ для каждого отдельного компьютера, то вероятность того, что все компьютеры работают нормально, примерно равна 0,37, меньше половины — ошибкоустойчивость совершенно недостаточная.

Что же делать?

Для решения этой проблемы мы обратимся к понятию отказоустойчивости (https://ru.wikipedia.org/wiki/Отказоустойчивость) при распределенных вычислениях. Обычно оно включает репликацию входных данных (по умолчанию — по три копии всех данных), причем различные копии доступны разным компьютерам, так что если один перестанет работать, то у другого останется доступ к данным. Кроме того, можно встроить в данные контрольные суммы (https://ru.wikipedia.org/wiki/Контрольная_сумма) для аудита ошибок в них с автоматическим мониторингом контроллером (возможно, даже не одним).

В целом нам нужно разработать систему обнаружения ошибок с автоматическим перезапуском в случае такого обнаружения. Для повышения эффективности после завершения работы части машин можно использовать освободившиеся ресурсы для повтора вычислений в целях поиска ошибок.

В: Извините, разве мы не просто подсчитываем количество слов? Похоже, что мы залезли в какие-то жуткие дебри.

О: Так бывает всегда. Не так уж легко говорить об эффективной отказоустойчивости, все очень непросто. И учтите, что эффективность ничуть не менее важна, чем точность, ведь тысяча компьютеров стоят куда больше вашей месячной зарплаты.

В целом получается так:

- первые десять компьютеров — элементарно;
- первые 100 компьютеров — довольно трудно;
- первые 1000 компьютеров — практически невозможно.

Миссия невыполнима.

Или по крайней мере так было еще лет восемь назад. Сегодня Дэвид регулярно использует 10 000 компьютеров в своей работе в компании Google.

Появляется MapReduce. В 2004 году Джефф (Jeff) и Санджей (Sanjay) опубликовали статью MapReduce: Simplified Data Processing on Large Clusters («MapReduce: упрощенная обработка данных на больших кластерах»), а также еще одну статью, посвященную нижележащей файловой системе (https://ai.google/research/pubs/pub51).

Благодаря MapReduce можно больше не думать об отказоустойчивости; MapReduce — платформа, которая берет эти заботы на себя. Программировать из расчета на 1000 компьютеров теперь проще, чем на 100. Это библиотека, позволяющая делать причудливые вещи.

Для использования MapReduce необходимо написать две функции: отображения и свертки. MapReduce выполняет их на нескольких машинах, локальных по отношению к вашим данным. Отказоустойчивость обеспечивается автоматически, достаточно просто передать алгоритм на выполнение фреймворку отображения/свертки.

Функция отображения генерирует на основе каждой точки данных упорядоченную пару (ключ, значение). Затем фреймворк сортирует результаты с помощью «перетасовки», в частности, отыскивая все совпадающие ключи, которые складывает в аккуратные стопки и далее отправляет их на машины для обработки с использованием функции свертки. Результаты выполнения последней имеют вид (ключ, новое значение), где новое значение — некий результат группировки старых значений.

Как же проделать вышеописанное применительно к нашему алгоритму подсчета слов? Просто включаем каждое из слов в упорядоченную пару, где оно станет служить ключом, а значением будет целое число 1. Вот так:

```
red ---> ("red", 1)
blue ---> ("blue", 1)
red ---> ("red", 1)
```

Затем отправляем эти пары на «перетасовку» (с помощью объединения на входе) и получаем стопку из пар `("red", 1)`, которые можно переписать в виде `("red", 1, 1)`. Они отправляются в функцию свертки, просто складывающую все единицы. В итоге получаем `("red", 2)`, `("blue", 1)`.

Суть в том, что *все значения для конкретного ключа* обрабатывает одна функция свертки.

Данных стало больше? Увеличьте количество исполнителей для отображения и свертки. Другими словами, выполняйте вычисления на бóльшем количестве компьютеров. MapReduce выравнивает степень сложности работы с различным количеством компьютеров. Это элегантное решение, которое иногда используют даже в тех случаях, когда не следует (хотя при работе в Google никогда нельзя исключить возможность роста объема данных на два порядка за ночь). Подобно всем утилитам, MapReduce иногда злоупотребляют.

Подсчет количества слов был сначала одной простой функцией, теперь же его пришлось разбить на две. В целом преобразование алгоритма в последовательность шагов MapReduce — зачастую не слишком интуитивно понятная операция.

Для предыдущего примера подсчета слов распределение должно быть равномерным. Если все слова одинаковы, то при перетасовке они попадут на одну машину, что приведет к колоссальным проблемам. Google решил эту проблему с помощью использования куч с хеш-корзинами в функциях отображения на одной из итераций MapReduce. Это решение носит название CountSketch (https://stackoverflow.com/questions/68 11351/explaining-the-count-sketch-algorithm), оно предназначено для работы с нестандартными наборами данных.

В Google есть и утилита для мониторинга заданий MapReduce в режиме реального времени, информационное окно с *сегментами*, соответствующими фрагментам выполняемых на машине вычислений. Она отображает в виде гистограммы ход вычислений на различных машинах. Если все функции отображения работают нормально, то вы увидите сплошную прямую линию. Однако обычно на этапе свертки идет не так все, что только можно, вследствие неоднородности данных, например большого количества значений с одним ключом.

Подготовка данных и запись результатов, выполняемые незаметно для пользователя, занимают длительное время, поэтому лучше сделать все за один подход. Отметим, что предполагается применение распределенной файловой системы — конечно же, для записи данных в распределенную файловую систему нам придется задействовать MapReduce — сказав «А», приходится говорить и «Б».

Занявшись однажды оптимизацией, вы обнаружите в какой-то момент, что настраиваете задания MapReduce, пытаясь сократить длительность повторяющихся процессов на какие-то наносекунды, поскольку речь идет о петабайтах данных. Это порядки величин, достойные физиков. Практически вся эта оптимизация выполнена на C++. Код MapReduce очень сильно оптимизирован, и мы пытаемся выжать из него максимум.

Другие примеры использования MapReduce

Подсчет количества слов — простейший пример использования MapReduce. Рассмотрим еще один, чтобы лучше познакомиться с фреймворком. Для решения задачи с помощью MapReduce важно, чтобы данные можно было распределить по многим компьютерам, а алгоритм мог работать с каждым из них отдельно, то есть чтобы одним компьютерам не требовалась информация о происходящем на других.

Вот еще один пример задачи, для которой можно применить MapReduce. Представьте, что у вас есть масса данных о событиях с метками даты/времени и журналов действий, произведенных пользователями на сайте. Скажем, для каждого пользователя есть следующая информация: `{user_id, IP_address, zipcode, ad_they_saw, did_they_click}`. И допустим, вам требуется подсчитать, сколько неповторяющихся пользователей по каждому почтовому индексу видело рекламу и сколько из них щелкнуло на рекламе хотя бы раз.

Как же сделать это с помощью MapReduce? Можно запустить его на обработку с почтовыми индексами в качестве ключей, чтобы на основе записи о человеке, живущем в районе с почтовым индексом 90210, получался кортеж `(90210,{1,1})`, если этот человек видел рекламу и нажал ее, или `(90210,{0,1})`, если видел и не нажал.

Что это даст? На этапе свертки мы получим общие количества просмотров и нажатий, сгруппированные по почтовому индексу, с результатами вида `(90210,{700,15530})`. Но это не то, что требовалось. Нам нужно было количество уникальных пользователей. Для этого нам понадобятся две операции MapReduce.

Сначала применим кортеж `{zip_code,user}` в качестве ключа, а `{clicks, impressions}` — в качестве значения. Например, `({90210,user_5321},{0,1})` или `({90210,user_5321}, {1,1})`. Функция свертки в результате выдаст таблицу, содержащую количества просмотров и нажатий для каждого пользователя и почтового индекса. Новые записи будут иметь вид `{user, zipcode, number_clicks, number_impressions}`.

Далее для получения количества уникальных пользователей по каждому почтовому индексу и количества нажавших рекламу хотя бы один раз вам понадобится второе задание MapReduce, с `zipcode` и `{1, ifelse(clicks>0)}` в качестве ключа и значения соответственно для каждого пользователя.

Это была вторая демонстрация использования MapReduce для подсчета. А как насчет чего-нибудь посложнее, например реализации с помощью MapReduce статистической модели, скажем линейной регрессии? Возможно ли это?

Оказывается, да. В статье 2006 года рассказывается, как использовать фреймворк MapReduce для реализации разнообразных алгоритмов машинного обучения. Описанный в этой статье общий подход может применяться для алгоритмов вычисления достаточных статистик и суммирования, поскольку эти типы вычислений поддаются пакетной обработке и их можно представить в виде суммы по точкам данных.

На что MapReduce не способен. Чтобы понять *суть* некоего объекта, бывает полезно разобраться, чем он *не является*. Итак, на что MapReduce *не* способен? На множество вещей, например массаж. Но вполне простительно считать MapReduce способным решить практически любую задачу обработки данных.

Однако MapReduce — неидеальное средство для, допустим, итеративных алгоритмов, в которых только что вычисленные значения используются как входные данные для следующего этапа вычислений, — распространенная ситуация в различных алгоритмах машинного обучения, применяющих методы градиентного спуска. Конечно, при необходимости можно задействовать MapReduce, но это требует многоэтапной обработки. Вероятно, здесь лучше подойдут (то есть ока-

жутся эффективнее) другие, более современные подходы, например Spark (http://static.usenix.org/event/hotcloud10/tech/full_papers/Zaharia.pdf).

Pregel

В компании Google был создан и совершенно другой алгоритм для обработки больших объемов данных — Pregel. Он представляет собой основанный на графах вычислительный алгоритм для данных, имеющих графоподобную или сетевую структуру. Этот алгоритм разрешает взаимодействие узлов с узлами, с которыми они соединены ребрами графа. Существуют также узлы-агрегаторы, имеющие доступ к информации всех остальных узлов и возможность, например, суммировать или усреднять все отправляемые им всеми узлами данные.

В основе этого алгоритма лежит последовательность супершагов, в которых чередуются узлы, отправляющие информацию своим соседям, и узлы, отправляющие информацию агрегаторам. Если хотите узнать больше, то исходная статья выложена в Интернете (https://dl.acm.org/citation.cfm?id=1807184). Существует также версия Pregel с открытым исходным кодом — Giraph (http://giraph.apache.org/).

О Джоше Уиллсе

Джош Уиллс — директор Cloudera по направлению «наука о данных», совместно с заказчиками и инженерами разрабатывающий решения на основе Hadoop для множества различных отраслей промышленности. До прихода в Cloudera Джош работал в Google, где занимался системой рекламных торгов, а затем руководил созданием аналитической инфраструктуры, используемой в Google+. Диплом бакалавра математики он получил в Университете Дьюка, а магистерский, в области исследования операций, — в Техасском университете в Остине.

Джош также известен своими емкими высказываниями по поводу науки о данных, например: «Я сделал из данных конфетку» — или уже встречавшееся нам в начале книги: «Исследователь данных (существительное): человек, разбирающийся в статистике лучше любого программиста, а в программировании — лучше любого специалиста по статистике». И еще одна из его жемчужин: «Я — Форест Гамп, у меня есть зубная щетка и много-много данных, которые я ею чищу».

Джош решил начать свой раздел с мысленного эксперимента.

Еще один мысленный эксперимент

Как бы вы создавали самолет на мускульной силе? Что именно делали бы? Как формировали бы команду разработчиков?

Допустим, вы решили посостязаться за соответствующую X-премию (https://www.xprize.org/). Одни команды пытались таким образом выиграть 50 000 долларов в 1950-х[1]. У других это заняло десять лет. История победителя показывает, что проигравшие иногда решали совсем не ту задачу.

А именно: первые несколько команд провели годы за планированием, а потом их планы обратились в прах за считаные секунды. Победившая команда изменила вопрос на следующий: как построить такой самолет, который можно было бы после аварии собрать обратно за четыре часа? Перепробовав множество прототипов, они решили задачу за полгода.

Что значит быть исследователем данных

У Джоша имеется несколько ценных наблюдений относительно работы исследователем данных. Исследователи тратят практически все время на очистку и подготовку данных, добрых 90 % этой работы представляет собой некую разновидность проектирования данных. При выборе между решением конкретных задач и поиском в данных полезной информации исследователи данных выбирают первое. Немного подробнее: начинайте с решения задачи и убедитесь в том, что вам есть что оптимизировать. Распараллеливайте все возможное.

Быть умным — хорошо, но уметь быстро учиться — еще лучше: выполняйте эксперименты как можно скорее, чтобы научиться чему-нибудь быстро.

Избыток и нехватка данных

Большинство людей мыслит категориями нехватки данных. В стремлении к стабильности они отбрасывают все, что считают лишним. Джош же сохраняет все. Он — приверженец воспроизводимых исследований, так что старается иметь возможность повтора любой фазы анализа. Он сохраняет все. Это замечательно по двум причинам. Во-первых, если он допускает ошибку, то ему не приходится начинать все с начала. Во-вторых, при обнаружении новых источников данных становится гораздо проще интегрировать их в наиболее подходящее место технологического процесса.

Проектирование моделей

Модели всегда со временем превращаются в заумные машины Руба Голдберга (Rube Goldberg) (https://ru.wikipedia.org/wiki/Машина_Голдберга) — сборную солянку

[1] Джош ошибается, речь идет о премии Кремера, а не X-премии. Сумма премии также неоднократно менялась, изначально она составляла £5000, а выигравшая первую такую премию команда получила £50 000. — *Примеч. пер.*

из различных моделей. Это не обязательно плохо, поскольку если они работают, то пусть продолжают. Даже начав с очень простой модели, рано или поздно вы будете вынуждены добавить «костыль» для поправки чего-нибудь. Это будет происходить снова и снова, такова уж природа проектирования модели.

Обратите внимание на разницу. Оптимизируемый с помощью вашей модели параметр — вовсе не то же самое, что параметр, оптимизируемый в реальном бизнесе.

Пример: предложения возможных друзей в Facebook не оптимизируют процесс добавления вами друзей, а *максимизируют проводимое вами в Facebook время*! Присмотритесь: в этих предложениях подозрительно часто встречаются симпатичные люди противоположного пола.

Что это может означать в других сферах? В медицине, например, изучение эффективности лекарства вместо здоровья пациентов. Или сосредоточение внимания на успешности операции вместо самочувствия пациента.

Экономическая интерлюдия: Hadoop

Вернемся на минуту к MapReduce и Hadoop. Когда Джош закончил вуз в 2001 году, существовало два варианта хранения файлов — файловые системы и базы данных, — которые Джош сравнил по четырем измерениям: схема, возможности обработки, надежность и стоимость (табл. 14.1).

Таблица 14.1. Варианты хранилищ файлов в 2001 году

Измерения	**Базы данных**	**Файловые системы**
Схема	Структурированы	Схема отсутствует
Обработка данных	Интенсивная обработка производится там же, где хранятся данные	Возможности обработки данных отсутствуют
Надежность	Относительно надежны	Надежны
Стоимость	Высокая стоимость при больших объемах данных	Высокая стоимость при больших объемах данных

С тех пор объемы генерируемых данных значительно повысились, в основном из-за Интернета. Возникает естественная идея ввести экономический показатель данных: *рентабельность байта*. Сколько прибыли можно извлечь из одного байта данных? Во сколько обходится его хранение? Соотношение этих показателей должно быть больше единицы, иначе данные следует выбросить.

Конечно, это не полностью отражает ситуацию. Существует также экономический закон больших данных, гласящий: *отдельные записи особой ценности не несут, но наличие полных данных крайне ценно*. Так, например, наличие у одной компании всех существующих данных означает огромное преимущество в случае веб-индекса,

рекомендательной системы, данных датчиков или онлайн-рекламы, хотя каждая точка данных в отдельности особой ценности не имеет.

Краткое введение в Hadoop

Еще в те времена, когда компания Google имела не так много денег, ее программное обеспечение было не слишком хорошим. Поэтому для работы со всеми данными они решили копировать их на множество серверов. Сначала операция выполнялась вручную, а потом была проведена автоматизация. Формальная автоматизация этого процесса привела к зарождению GFS (https://en.wikipedia.org/wiki/GFS2 и https://ru.wikipedia.org/wiki/Google_File_System).

Ядро Hadoop, который представляет собой версию файловой системы GFS и фреймворка MapReduce с открытым исходным кодом, состоит из двух ключевых компонентов. (Прочитать больше об истории возникновения Hadoop (https://ru.wikipedia.org/wiki/Hadoop) вы можете где-нибудь в других местах, мы лишь подскажем, что тут не обошлось без маленького проекта с открытым исходным кодом Nutch и компании Yahoo!.) Первый компонент — распределенная файловая система (HDFS) (https://ru.wikipedia.org/wiki/Hadoop#HDFS), основанная на файловой системе Google. Данные хранятся в больших файлах при размерах блоков от 64 до 256 Мбайт. Эти блоки реплицируются на несколько узлов в кластере. Главный узел отслеживает отказы отдельных узлов. Второй компонент — MapReduce, о котором нам только что все рассказал Дэвид Кроушоу.

Кроме того, Hadoop написан на Java, а программное обеспечение Google — на C++. Написание операций MapReduce с помощью API Java — удовольствие ниже среднего. Иногда приходится создавать множество операций MapReduce. Однако при использовании потоковой обработки Hadoop (http://hadoop.apache.org/docs/) можно писать код на языках Python, R и других высокоуровневых языках программирования. Это просто и хорошо подходит для распараллеливания заданий.

Cloudera

Основателями компании Cloudera были Дуг Каттинг (Doug Cutting), один из создателей Hadoop, и Джефф Хаммербахер, уже упоминавшийся в главе 1, поскольку именно он придумал название должности «исследователь данных» при работе в Facebook и собрал там команду таких специалистов.

Cloudera — своеобразный Red Hat (https://www.redhat.com/en) для Hadoop. Мы имеем в виду создание компании на основе проекта с открытым исходным кодом. Сделано это было под эгидой Apache Software Foundation (http://www.apache.org/). Код свободно доступен, но Cloudera компонует его воедино, распространяет бесплатно различные дистрибутивы, ожидая, что люди будут платить им за поддержку и обеспечение работоспособности системы.

Apache Hive (http://hive.apache.org/) — система складирования данных, представляющая собой надстройку над Hadoop. В ней используется SQL-подобный язык запросов (с некоторыми специальными расширениями для MapReduce) и реализуются распространенные закономерности соединений и агрегирования. Он удобен и приятен в работе для тех, кто хорошо знаком с базами данных.

Возвращаемся к Джошу: последовательность выполняемых действий

Учитывая подход Джоша к созданию конвейеров данных с помощью MapReduce, неудивительно, что он рассматривает запись в качестве базовой единицы анализа. Мы неоднократно упоминали «данные о событиях с метками даты/времени», которые можно рассматривать как отдельные записи или же можно рассматривать в их качестве записи о транзакциях, обсуждавшиеся в контексте обнаружения мошенничества и операций с платежными картами. Стандартная последовательность выполняемых действий будет таковой.

1. Сформировать (интенсивная MapReduce-обработка) записи, содержащие всю информацию о сущностях (например, о человеке) с помощью Hive (SQL-подобного языка на основе Hadoop).
2. Написать сценарии на языке Python для многократной обработки записей (быстрый итеративный процесс, также выполняемый с помощью MapReduce).
3. Обновлять записи при поступлении новых данных.

Отметим, что в сценариях на этапе 2 обычно выполняется лишь отображение, что упрощает распараллеливание.

Джош предпочитает стандартные форматы данных: текст занимает слишком много места. Thrift (https://ru.wikipedia.org/wiki/Apache_Thrift), Avro (https://en.wikipedia.org/wiki/Apache_Avro) и Protocol Buffers (https://ru.wikipedia.org/wiki/Protocol_Buffers) — более компактные двоичные форматы. Он также рекомендует использовать репозиторий кода и метаданных GitHub. Джош не хранит большие файлы данных в Git.

Как же начать работать с Hadoop

Если вы работаете в компании, в которой есть кластер Hadoop, то, вероятнее всего, сначала столкнетесь с Apache Hive — SQL-подобной абстракцией поверх HDFS и MapReduce. Возможно, ваше первое задание MapReduce будет представлять собой анализ журналов действий пользователей с целью достичь лучшего понимания применения заказчиками продуктов вашей компании.

При самостоятельном изучении Hadoop и MapReduce для создания аналитических приложений можно начать с нескольких отличных стартовых точек. Один из вариантов — создать рекомендательную систему с помощью Apache Mahout, набора библиотек и утилит машинного обучения, работающих с Hadoop. В Mahout имеется Taste — механизм коллаборативной фильтрации, который можно применить для создания рекомендаций на основе заданного файла в формате CSV с идентификаторами пользователей, идентификаторами товаров и необязательными весами, отражающими, насколько тесно пользователь связан с товаром. В Taste задействуются те же алгоритмы выдачи рекомендаций, что и в Netflix и Amazon.

15 Мнения студентов

Всякий алгоритм подобен колонке редактора.

Эмили Белл (директор Центра Тоу цифровой журналистики в Колумбийской последипломной школе журналистики)

Мы предложили написать одну главу нашей книги студентам, прослушавшим курс «Введение в науку о данных» версии 1.0. Они предпочли воспользоваться этой возможностью, чтобы описать свое мнение о курсе и ощущения от него. В написании главы участвовали Александра Богосян (Alexandra Boghosian), Джед Догерти (Jed Dougherty), Эври Ким (Eurry Kim), Альберт Ли (Albert Lee), Адам Обен (Adam Obeng) и Каз Сакамото (Kaz Sakamoto).

Мыслительный процесс

При изучении науки о данных можно начать только с самого переднего края науки.

Вступительный курс физики обычно охватывает механику, электричество и магнетизм и, возможно, потом переходит к некоторым более «современным» темам (например, специальной теории относительности), широко представленным в порядке возрастания сложности. Но такая подача накопленных наукой знаний в виде сгруппированной последовательности не дает никакого понимания того, как, скажем, Ньютон придумал дифференциальное исчисление. Мы не получаем знаний о ходе его работы, о том, как он пришел к своим выводам. Нам ничего не рассказывают о его инструментах или мыслях. Или о том, какие книги он читал и делал ли он записи. Пытался ли он воспроизвести доказательства других ученых? Задумывался ли над задачами, вытекающими из книг его предшественников? Почему на самом деле он решил, что не может обойтись без бесконечно малых величин? Требовались ли ему черновики, или идеи рождались в его голове полностью сформированными при виде падающего яблока? Этому не учат, а изучают самостоятельно; через данный процесс должны пройти все начинающие ученые.

Рэйчел начала первое вводное занятие с того, что настойчиво предостерегла нас: наука о данных — пока еще не вполне устоявшаяся вещь как на практике, так и в теории. В каждой очередной лекции рассказывалось о существующих задачах и выборе задач для изучения. По существу, еженедельные лекции охватывали утилиты и методы исследователей данных, но у каждого преподавателя был свой стиль, свой багаж знаний и подход к каждой задаче. Практически на каждом занятии лектор говорил нечто наподобие «я не знаю, что вы уже изучили, но...» В этом смысле лекции были разрозненными, и нам приходилось интерполировать их в единый связный рассказ о даталогии. Нам приходилось формировать свое понимание курса, подобно тому как исследователи данных продолжают формировать сферу своей деятельности.

Это не значит, что Рэйчел оставила нас блуждать в потемках. В первый же день она предложила рабочее определение науки о данных. «Исследователь данных, — сказала она, — тот, кто обладает познаниями в следующих сферах: математика, статистика, вычислительная техника, машинное обучение, визуализация, средства обмена данными и знания предметной области». Вскоре мы обнаружили, что это было лишь начало нашего формирующегося байесовского понимания. Все студенты и лекторы оценивали себя в смысле этого определения, формируя одновременно картину многообразия сообщества науки о данных и точку отсчета для курса. Среди лекторов были люди из научной среды, финансового дела, известных информационно-технологических компаний и стартапов; бросившие аспирантуру, победители конкурсов Kaggle, специалисты по компьютерной графике. Каждый из них все повышал степень правдоподобия. Сами занятия стали своего рода итеративным определением науки о данных.

Но мы не просто слушали, как разные люди неделя за неделей рассказывали о своей работе. Мы проводили изнурительные часы за домашними заданиями, изучая инструментарий профессии. Иногда в них от нас требовалось реализовать обсуждавшиеся на лекциях методы и принципы. А иногда приходилось обнаруживать и использовать навыки, о существовании которых мы даже не подозревали.

Более того, нас заставляли работать с запутанными реальными данными. Мы часто имели дело с промышленными задачами, причем результаты следовало представить в виде ясного и продуманного отчета — такого, который не стыдно было бы показать профессионалу в данной области. Что важнее всего: зачастую нам ничего не оставалось, кроме как выходить из комфортного личного пространства и обращаться к одногруппникам для выполнения этих заданий. Постоянно подчеркивалась социальная сущность науки о данных, и в дополнение к официальным собраниям группы для раздачи заданий и проектов Рэйчел часто вела присутствующих в находившийся через дорогу бар[1]. Мы работали и пили вместе на протяжении всего курса, учась друг у друга и совместно формируя свои навыки.

---

[1] Примечание от Рэйчел: это был аспирантский курс. Я лично проверяла, все ли студенты совершеннолетние, а также наличие в меню безалкогольных напитков.

Более не наивный

Наш судный день пришелся на третий подпункт второго домашнего задания, раздел «Упражнение от Джейка: использование наивного классификатора Байеса для статей» главы 4. В нем требовалось скачать 2000 статей с сайта The New York Times (который допускал скачивание не более чем 20 статей за раз) и обучить наивный классификатор Байеса сортировать их по разделам газеты. Получение статей являлось лишь половиной задачи. Хотя для многих языков программирования существуют пакеты, реализующие наивный классификатор Байеса, от нас требовали написания своего кода, причем из информации у нас было всего несколько уравнений. Кроме того, использование существующих пакетов нам ничем бы не помогло, поскольку, в отличие от них, у нашей версии должны были быть настраиваемые гиперпараметры регуляризации. Нам также требовалось классифицировать статьи по пяти категориям вместо двух. Как нам рассказали, наивный классификатор Байеса и не слишком наивный, и не слишком байесовский. Вдобавок он оказался и не таким уж простым. Тем не менее те из нас, кто не бросил работу и провел мучительные 40 часов за отладкой 300 строк омерзительно неуклюжего кода, получил истинное удовольствие, наблюдая за 90%-ной точностью прогнозов. Нас сразу же зацепило, причем по-настоящему. Возможно, это было понапрасну потраченное время. Но нас все равно зацепило. Рисунок 15.1 демонстрирует одно из студенческих решений данной задачи.

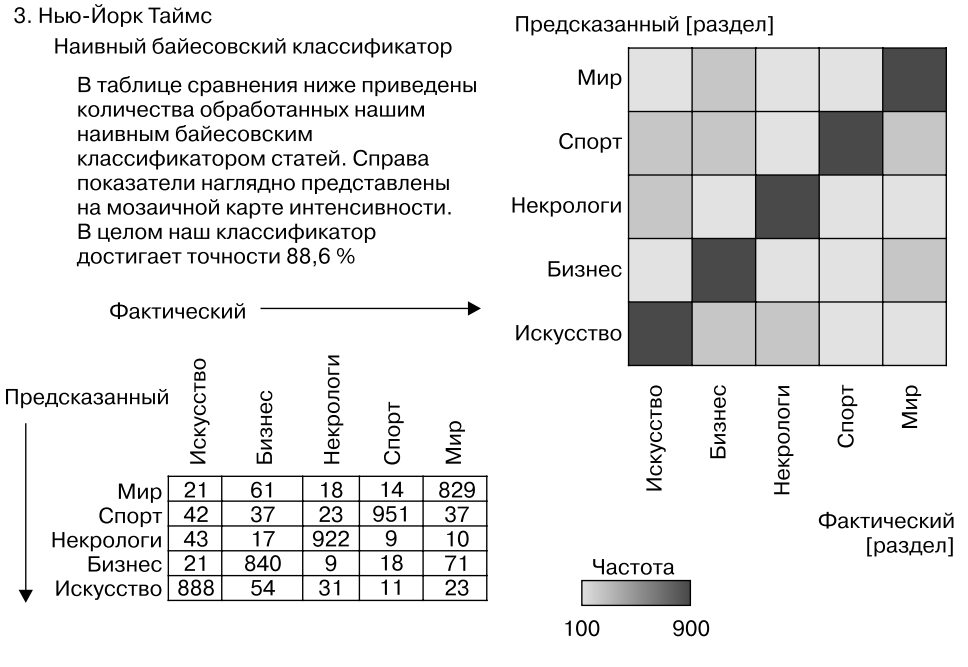


Рис. 15.1. Часть решения домашнего задания одного из студентов

Конкурс Kaggle, составлявший половину нашего итогового проекта, оказался прекрасной возможностью сойти с проторенного пути. Он представлял собой соревнование между студентами группы по созданию алгоритма оценки эссе. Домашние задания часто имитировали реальные промышленные задачи исследователей данных, а соревнования Kaggle можно описать как «конкурс, у кого больше» в науке о данных. Они вдохновляли нас на использование всех изученных рекомендуемых практик даталогии, вводя нас при этом в самую гущу типичной работы исследователя данных. Одно из наших программных решений в конце концов привело к включению в качестве признаков, помимо прочего, количества орфографических ошибок, количества слов из списка Дэйла — Чэла, слов, которые должен знать каждый ученик четвертого класса, векторы TF-IDF 50 самых часто встречающихся слов и корень четвертой степени количества слов в эссе. Не спрашивайте почему. В решении использовались как модель случайных лесов, так и модель градиентного спуска и была реализована вероятностная оптимизация гиперпараметров. Было выполнено обучение 50 000 моделей за тысячи часов работы виртуальных узлов Amazon EC2. Все сработало отлично.

Неоценимая помощь

На одно из первых занятий пришел приглашенный преподаватель Джейк Хофман. Помните первый увиденный вами в жизни фокус? Да, это было очень похоже — ловкость рук, достойная настоящего фокусника. С помощью простых Unix-утилит и исходных данных он создал наивный классификатор Байеса спама прямо на наших глазах. Написав несколько уравнений на доске, он продемонстрировал применение «карате командной строки» для разбора скачанных на ходу общедоступных сообщений электронной почты компании Enron.

На еженедельных занятиях нас ждали «артисты», заслуживающие самых бурных оваций. Но удержаться на плаву при таком быстром темпе занятий нам удалось прежде всего благодаря помощи Джареда Ландера и Бена Редди, которые вели лабораторные занятия. Они познакомили нас с основными практическими приемами исследования данных. Мы охватили целый диапазон понятий из математической статистики, начиная от линейной регрессии и заканчивая функционированием алгоритма случайных лесов. И многие из нас познакомились с новыми для себя утилитами: регулярными выражениями, LaTeX, SQL, R, командной строкой, Git и Python. Мы получили ключи к новым источникам данных в виде различных API и веб-скрапинга.

В целом специалистам в области компьютерных наук в этом курсе пришлось быстро изучить основы теории выбора признаков и языка R. Тем временем социологам пришлось разобраться с функционированием баз данных и различием между глобальными и локальными переменными, а финансистам — изучить этику. У всех нас были свои заботы. Но по мере того как мы все раз за разом ошибочно создавали

в языке R циклы for и поражались удивительно низкой производительности кода, наш багаж знаний понемногу рос. Рисунок 15.2 демонстрирует один из подводных камней этого метода.

```
44  #ПРЕДОСТЕРЕЖЕНИЕ: ВЫПОЛНЕНИЕ ЭТОГО КОДА ЗАНИМАЕТ
      ДЛИТЕЛЬНОЕ ВРЕМЯ И МОЖЕТ ПРИВЕСТИ К ПЕРЕГРЕВУ КОМПЬЮТЕРА
45  detailed_results <- predict(model,as.matrix(testmatrix),type="raw");
```

Рис. 15.2. Урок, извлеченный из ошибки

И по мере роста наших навыков мы могли уделять все больше внимания собственно анализу данных. В конце концов мы вообще не обращали внимания на код, видя только кроющиеся за ним идеи.

Но как бы мы могли разобраться во всем этом самостоятельно? Смог бы хоть кто-то из нас это сделать?

Оценить возможности науки о данных мы смогли, потратив долгие часы на поиск причин ошибок, взбираясь по крутым кривым обучения. Но чтобы все же успеть закончить задание вовремя, нам нужно было учиться у своих сокурсников с другими областями знаний.

На деле для выполнения заданий оказалось критически важно найти сокурсников, которые бы обладали навыками, дополняющими наши. Рэйчел заставила группу работать совместно, не с помощью приказов, а путем объемных, но поддающихся разбиению на составные части заданий. Оказалось, что она стремилась показать нам: по своей сути исследование данных — задача для коллектива. В самом начале курса Рэйчел показала нам сетевую диаграмму типа «звезда». Она собрала нас всех вместе, так что сама была в центре. Каждый из нас был связан с ней. Она надеялась на формирование между нами новых дружеских отношений/идей/проектов/связей за время курса.

Вероятно, в быстро развивающейся сфере важнее, чем в любой другой, быть частью сообщества. В науке о данных это не просто полезно для карьеры, а существенно для ежедневной работы. Если вы не читаете блоги, не подписаны на коллег в Twitter и не посещаете конференции, то откуда же вы узнаете про свежее программное обеспечение для распределенных вычислений или опровержение работоспособности статистического подхода из известной статьи? Сообщество настолько тесно взаимосвязано, что при упоминании MapReduce на конференции в апреле Кэти смогла сразу же переадресовать вопрос одному из слушателей, Нику Автеньеву (Nick Avteniev), простые и быстрые обращения к экспертам в нужной области здесь — норма. Совокупность знаний науки о данных постоянно меняется и носит распределенный характер до такой степени, что единственный способ выяснить, какие знания вам нужны, — посмотреть, что знают другие. Большое количество разных преподавателей очень нам в этом помогло. Все они с удовольствием

отвечали на наши вопросы. Все они дали нам свои адреса электронной почты. Некоторые даже взяли кое-кого из нас на работу.

После прослушивания лекций и общения с этими экспертами у нас родилось еще больше вопросов. Как можно создать объект для временного ряда на языке R? Почему у нас продолжают появляться ошибки при построении графиков матрицы различий? Что такое случайный лес? Ответы на эти вопросы мы не только искали у сокурсников, но и заглядывали на сайты таких социальных сетей, как Stack Overflow, Google Groups и R bloggers. Оказалось, существует обширное сообщество, готовое помочь в отладке кода начинающим исследователям данных вроде нас. И мы не просто получали ответы от других пользователей, столкнувшихся с теми же проблемами до нас. Нет, на эти вопросы отвечали первопроходцы методов. Такие люди, как Хэдли Викхэм (Hadley Wickham), Уэс Маккинни (Wes McKinney) и Майк Бостоk (Mike Bostock), обеспечивали техподдержку написанных ими пакетов. Поразительно!

Длина пройденного пути может варьироваться

Не существует никакого идеального хранилища знаний из сферы науки о данных, которые вы могли бы впитать путем постепенного осознания. Есть разнообразные рекомендуемые практики из разных дисциплин, а также различные терминологии и интерпретации одного метода. (Является ли параметр регуляризации априорной информацией или служит только для сглаживания? Следует ли выбирать его из принципиальных соображений или просто с целью улучшения подгонки модели?) Устоявшихся знаний не существует, поскольку нет организаций, которые эти знания регламентировали бы. Именно поэтому важна структура взаимодействия: у вас появляется возможность создавать свои собственные. Вы сами выбираете, кто влияет на ваши взгляды, как изложил Габриэль Тард, в пересказе Бруно Латура, а затем Марка Хансена:

> «Когда молодой фермер, глядя на закат солнца, не знает, верить ли ему рассказанному школьным учителем о том, что день заканчивается из-за движения Земли, а не Солнца, или поверить своим глазам, которые говорят ему противоположное, в работу вступает луч подражания, через его школьного учителя соединяющий фермера с Галилеем».

Стоять на плечах гигантов, конечно, замечательно, но прежде, чем запрыгивать кому-то на спину, лучше убедиться, что этот кто-то сможет выдержать ваш вес. В мире бизнеса науку о данных используют преимущественно для продажи рекламы. У вас может быть доступ к лучшему набору данных в мире, но если ваши

работодатели хотят всего лишь продать как можно больше ботинок с его помощью, то стоит ли овчинка выделки?

Работая над заданиями и сравнивая свои программные решения, мы воочию убеждались, что результаты нашего анализа могут очень сильно различаться в зависимости от всего лишь нескольких выбранных развилок. Даже если все шаги процесса от формирования гипотезы до результатов фиксированы, количество возможных сочетаний огромно, ведь существует так много различных способов реализации каждого шага.

Клаудия Перлих из компании Media 6 Degrees — обладатель престижного кубка KDD за 2003, 2007, 2008, 2009 годы и сейчас занимается организацией этих соревнований. Она была столь великодушна, что поделилась с нами всеми нюансами комплекса операций науки о данных, а также различными подходами к принятию этих редакторских решений. В одном из соревнований, где нужно было предсказать результаты лечения в больнице, она обратила внимание, что идентификаторы пациентов назначались последовательно, поэтому у всех пациентов из конкретной больницы были последовательные номера. А поскольку различные больницы лечили пациентов с заболеваниями разной тяжести, то идентификатор пациента оказался отличным показателем требуемого результата.

Конечно, подобная утечка данных не была умышленной, ведь задание становилось тривиальным. Но на практике, вероятно, его стоило бы использовать в модели; в конце концов, выбор докторами и пациентами больницы должен учитываться при прогнозе результатов лечения.

Дэвид Мэдиган подчеркивает этические проблемы принимаемых редактором решений в этой быстро развивающейся сфере, демонстрируя, как исследования методом наблюдения в фармацевтической промышленности часто дают совершенно разные результаты (еще один пример — показанный им график для аспирина). Он подчеркивает, насколько важно не отдаляться от реальных данных. Настроить модели и методы и применить их к имеющимся наборам данных — недостаточно.

В академических кругах возникают проблемы, схожие с проблемами бизнеса, но по иным причинам. Фрагменты науки о данных настолько разбросаны по различным областям знаний, что при изучении их по отдельности становится практически невозможно получить целостную картину их сочетания или хотя бы понять, сочетаются ли они вообще. Чисто теоретический подход к науке о данных приводит к ее выхолащиванию и дроблению до такой степени, что в результате получается следующее домашнее задание из главы «Линейные методы регрессии» книги «Элементы статистического обучения».

Упражнение 3.2. По имеющимся данным по двум переменным X и Y подобрать кубическую модель полиномиальной регрессии $f(X) = \sum_{j=0}^{3} \beta_j X^j$. Помимо построения графика подобранной кривой, сформируйте для нее 95%-ный доверительный интервал. Сравните следующие два подхода.

1. В каждой точке x_0 формируется 95%-ный доверительный интервал для линейной функции $\alpha^{\tau}\beta = \sum_{j=0}^{3} \beta_j x_0^j$.
2. Формируется 95%-ное доверительное множество для β, на основе которого генерируются доверительные интервалы для $D = \binom{N}{2} f(x_0)$.

Чем отличаются данные подходы? Какой интервал будет шире? Проведите небольшой модельный эксперимент для сравнения двух этих методов.

Задания подобного типа часто задают в более общих курсах машинного обучения или интеллектуального анализа данных. Наша первая реакция на него как начинающих исследователей данных была довольно скептической. На какой стадии исследования данных могла бы, хотя бы теоретически, возникнуть подобная задача? Сколько всего нужно проделать, чтобы добраться до этой стадии? Почему мы рассматриваем именно две эти переменные? Откуда у нас взялись данные? Кто их нам предоставил? Кто их оплачивает? Почему мы рассчитываем именно 95%-ные доверительные интервалы? Не лучше ли будет использовать другую метрику эффективности? На самом деле, кого волнует, насколько хорошие результаты мы получаем на наших обучающих данных?

Это несправедливо по отношению к Хасти и его соавторам. Вероятно, они бы возразили, что студенты, желающие изучить скрапинг и структурирование данных, могут взять другую книгу, охватывающую эти темы, — различие заданий демонстрирует разительный контраст в подходе данного курса и обычных теоретических вводных курсов. Нам раз за разом навязывали доктрину о том, что статистические инструменты науки о данных без контекста более глобальных решений и окружающих их процессов лишены большей части смысла. Кроме того, нельзя просто рассказать студенту, что реальные данные зашумлены или что на практике никто не подскажет, какую модель регрессии использовать. Такие вещи — и чутье, получаемое при работе с ними, — можно понять только на собственном опыте.

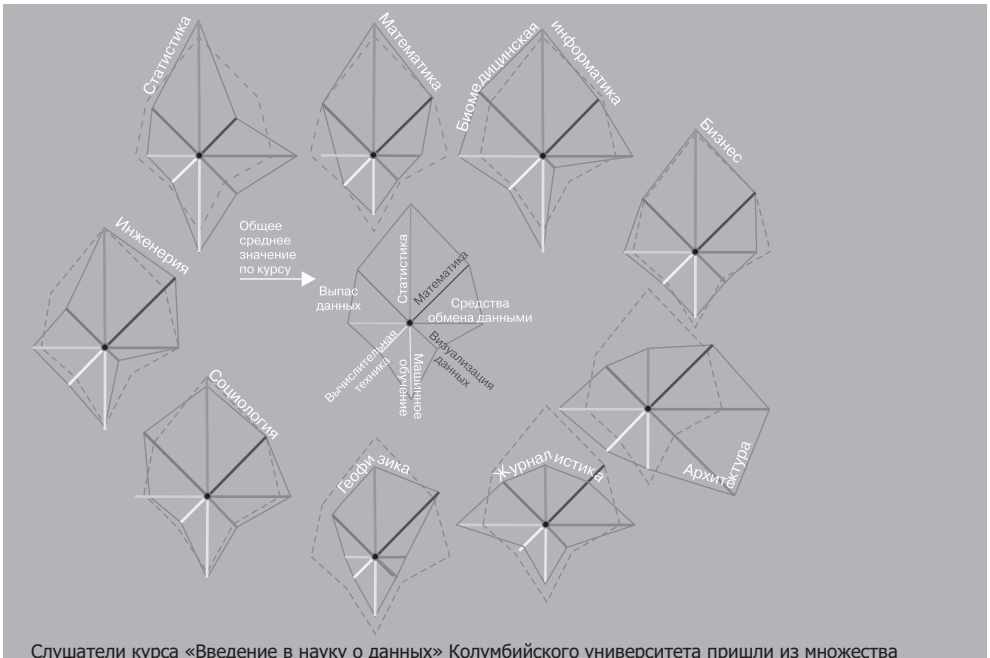
Строим мосты

При всем уважении к Майклу Дрисколлу (Michael Driscoll), мы все же начинающие исследователи данных, а не инженеры-строители. У нас отсутствует полное представление о том, что мы делаем, и нет детальных планов. Исследователи данных по своей натуре искатели приключений: мы знаем, чего хотим, у нас есть определен-

ные инструменты в багаже и, возможно, карта и пара друзей. Добравшись до замка, мы можем обнаружить, что принцесса вовсе не там, но для нас главное другое: по дороге мы затоптали пару троллей и наелись грибов и все еще полны энтузиазма. Если наука — водопровод, то мы вовсе не сантехники. Мы — братья Марио.

Некоторые из наших работ

Студенты усовершенствовали первоначальный профиль науки о данных из главы 1 (рис. 15.3) и создали инфографику для роста популярности даталогии в университетах (рис. 15.4) на основе информации, имевшейся у них по состоянию на конец 2012 года.



Слушатели курса «Введение в науку о данных» Колумбийского университета пришли из множества научных дисциплин. Их навыки показаны на схемах типа «звезда», где лучи отражают уровень их навыков в рамках необходимых для науки о данных: язык R, статистика, математика, средства обмена данными, визуализация данных, машинное обучение, вычислительная техника и выпас данных. Звездчатая схема общего среднего значения по курсу, находящаяся в центре, показывает каждую из областей знаний, так что можно видеть соотношение студентов из каждой области знаний с остальной группой. А как бы вы сформировали свою собственную межгалактическую команду исследователей данных?

* Навыки оценивались на основе опроса, написанного и организованного неким подмножеством студентов курса

Рис. 15.3. «Звезды» науки о данных (совместная работа множества студентов, включая Адама Обена, Эври Кима, Кристину Гутьеррес (Christina Gutierrez), Каза Сакамото и Вайбхава Бхандари (Vaibhav Bhandari))

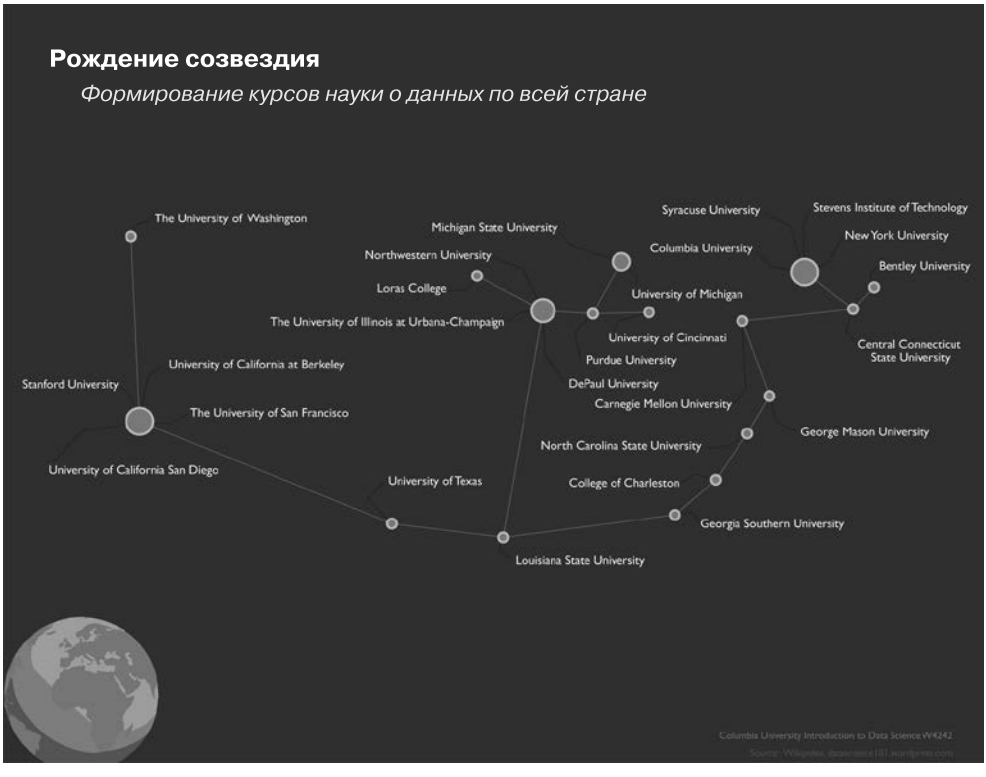


Рис. 15.4. Рождение созвездия (создано Казом Сакамото, Эври Кимом и Вайбхавом Бхандари в рамках более масштабной совместной работы)

16 Исследователи данных нового поколения, завышенная самооценка и этика

Мы хотели бы завершить нашу книгу обсуждением того, что вы обрели благодаря ее прочтению и как, надеемся, будете двигаться дальше.

Что вы обрели

Двумя основными целями данной книги являлось: поговорить о том, что значит быть исследователем данных, и научить вас выполнять хотя бы часть из того, что должен уметь такой специалист.

Мы надеемся, что достигли обеих целей.

Что касается первой цели, то многочисленные специалисты, участвовавшие в написании предыдущих глав, рассказали вам о своем опыте работы исследователем данных. Применительно ко второй: мы гордимся широтой охвата данной книги, даже если не везде смогли охватить все, что хотелось бы.

Вполне вероятно, что можно было выполнить эту задачу и лучше, чем сделали мы, поэтому рекомендуем вам поразмышлять над следующей врезкой.

МЫСЛЕННЫЙ ЭКСПЕРИМЕНТ: ПРЕПОДАВАНИЕ НАУКИ О ДАННЫХ

Как бы вы сами делали учебник по Data Science?

Это не очень четко определенная совокупность знаний, и никакого канонического свода знаний тут нет. Ее живо обсуждают и прославляют в прессе и медиа, но нет

никаких авторитетных источников информации для опровержения неправильных или странных ее описаний. Наука о данных также пересекается со множеством других областей знаний, в сочетании с учебником по машинному обучению подобная книга может оказаться избыточной.

Как можно оценить успешность и влияние на студентов учебника по науке о данных? Хотя бы гипотетически?

Можно ли сформулировать это в виде задачи исследования данных? Или еще лучше: можно ли воспользоваться причинно-следственным моделированием? Для этого вам нужно будет найти людей, которые были бы более или менее похожи на вас, наш дорогой читатель, но не купили эту книгу, после чего применили метод подбора контрольной группы по индексу соответствия. Или можно провести эксперимент и случайным образом лишать людей доступа к данной книге (что будет сделать непросто, ведь к Amazon доступ есть у всех) и проверять, как они будут приходить к тем же выводам другими путями. Но к вопросу это отношения не имеет, разве что позволит с большей уверенностью говорить о том, помогла ли наша книга кому-нибудь.

В коммерческом секторе говорят, что науку о данных нельзя изучить в университете или по книгам, это можно сделать только на практике. Впрочем, возможно, все иначе и наша книга — тому подтверждение. Как вы считаете?

И все-таки что такое наука о данных

Мы снова и снова возвращались к данному вопросу на протяжении всей книги. Это центральный вопрос книги, основная ее тема и своеобразная мантра, которую мы повторяли раз за разом.

Науку о данных можно определить просто как то, чем занимаются исследователи данных, что мы делали ранее, при обсуждении профилей исследователей данных. На самом деле Рэйчел, прежде чем начать читать курс науки о данных в Колумбийском университете, написала список всего, что делают исследователи данных, но никому не хотела его показывать, настолько он был ошеломляющим и беспорядочным. Этот список лег в основу будущих профилей. Но после обсуждения курса с различными людьми она обнаружила, что список им нравится; он представлен ниже.

- Разведочный анализ данных.
- Визуализация (для разведочного анализа данных и создания отчетов).
- Информационные панели и метрики.
- Поиск полезной для бизнеса информации.

- Принятие решений на основе данных.
- Проектирование данных/большие данные (MapReduce, Hadoop, Hive и Pig).
- Получение самих данных.
- Создание конвейеров данных (журналы → отображение/свертка → набор данных → соединение с другими данными → отображение/свертка → скрапинг данных → соединение).
- Создание новых продуктов вместо описания существующих.
- Решение задач нестандартным способом.
- Написание патентов.
- «Детективные» расследования.
- Прогнозирование будущего поведения или эффективности.
- Описание обнаруженного в отчетах, докладах и протоколах.
- Программирование (знание в совершенстве языков R, Python, C, Java и др.).
- Расчет условных вероятностей.
- Оптимизация.
- Алгоритмы, статистические модели и машинное обучение.
- Изложение и интерпретация информации.
- Задание правильных вопросов.
- Изыскания.
- Исследовательская работа.
- Логические выводы из данных.
- Формирование результатов обработки данных.
- Поиск путей обработки данных, их очистки и анализа в больших объемах.
- Контроль корректности.
- Выработка чутья на данные.
- Взаимодействие с экспертами в предметной области (или получение экспертных знаний в предметной области).
- Проектирование и анализ экспериментов.
- Поиск корреляций данных и попытки выявления причинно-следственных связей.

Но теперь мы хотели бы пойти немного дальше и попытаться получить нечто более основательное.

Выйдем в нашем определении науки о данных за пределы набора рекомендуемых практик, используемых в информационно-технологических компаниях. С это-

го определения мы начали нашу книгу. Но эти изыскания показали, что имеет смысл включить в даталогию и другие предметные области: нейробиологию, аналитику в области здравоохранения, обнаружение знаний в электронных данных (eDiscovery), вычислительную социологию, цифровые гуманитарные науки, геномику, политику. Цель этого: охватить все пространство задач, потенциально решаемых с помощью набора рекомендуемых практик из данной книги (часть которых появилась в информационно-технологических компаниях). Наука о данных — как теоретическая, так и практическая область знания, поэтому *где* или *в какой предметной области* она используется — неважно; главное — описать ее как «пространство задач» с соответствующим «пространством решений» в виде алгоритмов, кода и данных.

В этой связи мы начнем со следующего: наука о данных — набор рекомендуемых практик (иногда даже заслуживающий названия «наука»), используемых в информационно-технологических компаниях, которые работают в рамках широкой сферы решаемых на основе данных задач. Но даже при таких условиях зачастую это не более чем чистое очковтирательство, которого лучше избегать и стараться не умножать.

Кто такие исследователи данных нового поколения

> Лучшие умы нашего поколения заняты тем, что придумывают способы заставить людей нажимать на рекламные ссылки... Паршиво.
>
> *Джефф Хаммербахер*

Хочется надеяться, что находящееся сейчас в процессе обучения поколение исследователей данных стремится не просто стать профессионалами и получать хорошую зарплату — хотя и это было бы приятно. Мы хотели бы поощрить стремление исследователей данных нового поколения научиться решать проблемы и задавать вопросы, тщательно продумывать соответствующие план и последовательность работ, а также ответственно подходить к использованию данных и делать мир лучше, а не хуже. Рассмотрим в следующих подразделах эти принципы подробнее.

Умение решать проблемы

Прежде всего поговорим о технических навыках. Для исследователей данных нового поколения важно обладать множеством технических навыков, в числе которых написание кода, знание математической статистики, машинного обучения,

визуализация, владение средствами обмена данными и математикой. Кроме того, чрезвычайно полезными будут основательные знания таких практик, используемых при написании кода, как парное программирование, анализ исходного кода, отладка, а также контроль версий.

Никогда не поздно обратить особое внимание на разведочный анализ данных, описанный в главе 2, а также провести выбор признаков, как предлагает Уилл Кукерски. Брайан Далессандро обращает внимание на выбираемые исследователями данных бесконечные модели. Для их формирования исследователь данных должен выбрать классификатор, признаки, функцию потерь, метод оптимизации и метрику оценки. Дэвид Хаффакер рассказывал о методах формирования признаков или метрик: о преобразовании переменных с помощью журналов, формировании двоичных переменных (например, для действия, которое пользователь выполнил пять раз), агрегирования и подсчета. В силу кажущейся простоты все описанное часто упускают из виду, а ведь это важнейшая составляющая науки о данных. Именно это Далессандро назвал «искусством науки о данных».

Еще одно предостережение: множество исследователей переходят непосредственно от получения набора данных к применению причудливого алгоритма. Но между этими этапами располагается целая пропасть важнейших действий. Нет ничего сложного в том, чтобы выполнить фрагмент кода для прогноза или классификации и заявить об успехе, если алгоритм сходится. Это самое простое. Сложнее — сделать это достаточно хорошо и обеспечить корректность и интерпретируемость результатов.

Как бы поступил исследователь данных нового поколения

Исследователи данных нового поколения не станут пытаться впечатлить заказчиков сложными алгоритмами и моделями, которые не работают. Они проводят существенно больше времени, чем обычно признают, пытаясь преобразовать данные в нужный вид. Наконец, они не привязываются в своей работе к определенным инструментам, методам или разделам науки. Их отличает универсальность и многопрофильный подход.

Развитие личных качеств

Реализовать метод k-средних может множество людей, причем большинство из них — реализовать плохо. На самом деле практически никому не удается с первого раза реализовать этот метод идеально. Но важен не «первый блин», а последующие. Важно оттачивать хорошие привычки и быть открытыми для познания нового.

Вот отдельные черты характера, которые, как нам представляется, помогают решать проблемы[1]: упорство, умение размышлять о процессах мышления, гибкость мышления, стремление к точности и умение сопереживать слушая.

Сформулируем вышесказанное чуть-чуть иначе. В традиционном образовании делается акцент на ответах. Но *желательно бы* сосредоточить внимание или по крайней мере несколько акцентироваться на том, как ведут себя студенты *в случае, когда ответ им неизвестен*. Им необходимы качества, которые помогли бы находить ответы на поставленные вопросы.

Кстати, случалось ли вам удивляться, почему практически никто *не* говорит «я не знаю», когда не знает чего-либо? Это частично объясняется подсознательной склонностью, именуемой эффектом Даннинга — Крюгера (https://ru.wikipedia.org/wiki/Эффект_Даннинга_—_Крюгера).

В основном, плохо владеющие чем-либо люди не подозревают об этом и склонны переоценивать свои силы. Те же, кто очень хорошо что-либо умеет, склонны, напротив, недооценивать собственное мастерство. Имейте это в виду и старайтесь объективно оценивать свои способности — проверяйте на практике возможности запрограммировать излагаемые вами идеи и общайтесь с другими исследователями данных по поводу различных подходов.

ВОЗВРАЩАЕМСЯ К МЫСЛЕННОМУ ЭКСПЕРИМЕНТУ: ОБУЧЕНИЕ НАУКЕ О ДАННЫХ

Как бы вы сформировали курс даталогии с упором на *черты характера*, а не на технические навыки? Как бы определили его объем? Как бы оценивали его? Что могли бы потом ваши студенты написать в своих резюме?

Умение задавать вопросы

Люди склонны переобучать модели. Человеку свойственно хотеть, чтобы его дети были идеальными, и когда вы работаете над чем-либо месяцами, в вас начинает просыпаться материнский (или отцовский) инстинкт.

Человеку свойственно также преуменьшать плохие новости и винить в них других людей, поскольку, с точки зрения родителя, его ребенок не способен совершить плохое, разве что его как-то заставит некто третий. Что же делать с этой человеческой склонностью?

[1] Взято из книги Learning and Leading with Habits of Mind, под редакцией Артура Косты (Arthur L. Costa) и Бены Каллика (Bena Kallick) (ACSD).

В идеале хотелось бы, чтобы исследователи данных были достойны слова «ученый», то есть проверяли гипотезы и радовались трудным задачам и альтернативным теориям. А значит, критиковали собственные идеи, принимая возникающие вызовы и разрабатывая тесты, — выступали в роли ученых, а не просто защищали модели с помощью ораторского искусства. Если кто-то считает, что способен сделать лучше, то предоставьте ему такую возможность, но заранее оговорите метод оценки. Старайтесь сохранять объективность.

Привыкайте всегда проходить по стандартному списку важнейших шагов. Обязательно ли нужно делать именно так? Как это измерить? Какой алгоритм подойдет лучше всего и почему? Как это можно оценить? Обладаю ли я нужными навыками? Если нет, то как им научиться? С кем я могу работать? Кому я могу задать интересующие меня вопросы? И, вероятно, самое важное: как моя деятельность повлияет на реальный мир?

Далее, научитесь задавать вопросы другим людям. Приступая к решению проблемы или ответу на заданный вам вопрос, считайте себя умным и не думайте, что задавший вопрос человек знает больше или меньше вас. Вы ничего никому не доказываете, а просто выясняете истину. Будьте любознательны, как дитя, не бойтесь показаться глупым. Не стесняйтесь попросить пояснений относительно нотации, терминологии или технологического процесса: откуда взялись данные? Как планируется их использовать? Почему именно эти данные? Какие данные мы игнорируем и не больше ли в них признаков? Кто что станет делать? Как будет организована совместная работа?

Наконец, нельзя забывать о таком важнейшем вопросе, как классический статистический принцип «после — не значит вследствие». Не путайте корреляцию с причинно-следственной связью. Лучше ошибиться в сторону предположения, что наблюдаемое явление представляет собой корреляцию.

Как бы поступил исследователь данных нового поколения

Исследователи данных нового поколения склонны к скептицизму относительно самих моделей, их недостатков, их правильного/неправильного использования. Такие специалисты отдают себе отчет о скрытом смысле и результатах создаваемых моделей. Они заранее обдумывают обратную связь для своих моделей.

Моральные принципы исследователей данных

Вы все не просто сумасшедшие гении, тихо сидящие в своей норе. Во время работы вы должны принимать во внимание важные этические вопросы.

У нас есть терабайты маркетинговой информации и сведений о поведении людей. Как исследователи данных мы должны применять не только набор утилит машинного обучения, но и свою человечность при интерпретации данных и поиске их смысла, принимая подобающие решения с учетом данных.

Не забывайте, что произведенные действиями людей данные становятся в дальнейшем «кирпичиками» цифровых продуктов, применяемых *пользователями* и одновременно *влияющих* на поведение последних. Это явление можно наблюдать в рекомендательных системах, алгоритмах ранжирования, предложениях новых друзей и т. п., причем чем дальше, тем больше оно будет встречаться в различных сферах, таких как образование, финансы, розничная торговля и здравоохранение. Подобные циклы обратной связи могут привести к отрицательным последствиям. В качестве примера-предостережения можно привести глобальный финансовый кризис.

Очень много было достигнуто в сфере прогноза будущего (см.: Нейт Сильвер (Nate Silver) (https://slate.com/gdpr?redirect_uri=%2Farticles%2Fbusiness%2Fbooks%2F2012%2F10%2Fnate_silver_s_book_the_signal_and_the_noise_reviewed_.html%3Fvia%3Dgdpr-consent&redirect_host=http%3A%2F%2Fwww.slate.com)), прогноза текущей ситуации (см.: Хол Вэриен (Hal Varian) (https://ai.googleblog.com/2009/04/predicting-present-with-google-trends.html)) и исследований причинно-следственных связей по наблюдаемым данным (прошедшее; см.: Синан Арал (Sinan Aral)).

Следующий логический вывод: модели и алгоритмы способны не только предсказывать будущее, но и влиять на него. Именно на это мы надеемся в наилучшем сценарии и именно этого боимся в наихудшем.

В качестве введения в этические основы указанных вопросов начнем с предложенного Эммануэлем Дерманом (Emanuel Derman) варианта клятвы Гиппократа, предназначенной для финансового моделирования, но прекрасно подходящей и в нашем случае (https://mathbabe.org/2011/10/27/is-big-data-evil/).

- Я торжественно обещаю не забывать, что не я создал мир и он не обязан удовлетворять моим уравнениям.
- Хотя я буду отважно использовать модели для оценки значений, я постараюсь не слишком увлекаться математикой.
- Я никогда не буду жертвовать истиной ради красоты без пояснений, почему я так поступил. И не стану внушать ложных надежд относительно точности моей модели тем, кто ее использует. Вместо этого я буду явным образом описывать ее допущения и недостатки.
- Я понимаю, что результаты моей деятельности могут оказать колоссальное (и зачастую недоступное моему пониманию) влияние на общество и экономику.

Эта клятва не учитывает важный для исследователей данных аспект, а именно принципы поведения при коммерческом использовании. Даже если вы честно

говорите об ограниченности возможностей своей модели, то всегда остается вероятность, что ее применяют неправильно, несмотря на ваши предостережения. Так что вышеприведенный вариант клятвы Гиппократа, к сожалению, недостаточен для реальной работы (хотя и неплох для начала).

Как бы поступил исследователь данных нового поколения

Исследователи данных нового поколения не позволяют деньгам ослепить себя до такой степени, чтобы разрешить использование своих моделей для аморальных целей. Они стремятся решать общественно значимые проблемы и учитывать последствия применения своих моделей.

Наконец, существуют способы сделать доброе дело: записаться волонтером в долгосрочный проект (не просто хакатон на выходных) с помощью DataKind (https://www.datakind.org/).

Существуют также способы повышения прозрачности: Виктория Стодден (Victoria Stodden) работает над проектом RunMyCode (http://www.runmycode.org/home/?/CompanionSite/), нацеленным на обеспечение открытости исходных кодов исследований и их воспроизводимости.

Ненадолго уступим трибуну и позволим кое-кому другому рассказать о важности этических вопросов — и нормализации самооценки — для исследователей данных. Наш курс посетил профессор Мэтью Джонс (Matthew Jones) с кафедры истории Колумбийского университета — специалист по истории науки. Он записал несколько мыслей, возникших у него в связи с данным курсом. Приводим их здесь, чтобы вам было над чем хорошенько поразмыслить.

ДАННЫЕ И ЗАВЫШЕННАЯ САМООЦЕНКА

В результате президентских выборов 2012 года исследователи данных, их близкие и особенно те люди, которые их идеализируют, взорвались потоком злорадства по поводу недостатков традиционной пандитократии. Вычислительная статистика и анализ данных одержали убедительную победу над прогнозами на основе интуиции, внутреннего голоса, многолетнего опыта журналиста и пришедшей в упадок сети инсайдеров в Вашингтоне. Очевидный успех (http://swampland.time.com/2012/11/07/inside-the-secret-world-of-quants-and-data-crunchers-who-helped-obama-win/) в использовании количественных прогнозов командой Барака Обамы открыло новую эру политического анализа. Более старые виды «экспертных знаний», теперь уже в кавычках, отправились на заслуженный отдых отсутствующих данных и на сцену вышел новый, ориентированный на работу с данными политический анализ.

Это захватывающая история с эффектной бифуркацией старых и новых форм знания. Тем не менее хорошие исследователи данных настороженно относились к перспективе полного отказа от существующих знаний предметной области и услуг экспертов.

Предыстория повышает степень доверия к иерархии экспертных знаний. Предыстория интеллектуального анализа данных широко известна, хотя до некоторой степени апокрифична: неожиданное обнаружение с помощью алгоритма сопоставления (https://en.wikipedia.org/wiki/Association_rule_learning) того факта, что мужчины, покупающие подгузники, часто покупают вместе с ними и пиво (https://www.itbusiness.ca/news/behind-the-beer-and-diapers-data-mining-legend/136). Обычные маркетологи с их количественным подходом к психологии покупателей на основе интуиции были разгромлены задолго до появления того, что газеты, вероятно, до сих пор называют «электронным мозгом». Приведенная история следует классическому шаблону. Вероятностные и статистические методы давно уже, с самого их появления (https://press.princeton.edu/titles/4295.html) в эпоху европейского Просвещения, бросали вызов традиционным формам экспертного анализа: формирование цен на страховки и страховые ренты на основе данных, а не анализа черт характера страхующегося, повлекло снижение значимости экспертов и их исчезновение. В книге (http://visualiseur.bnf.fr/Visualiseur?Destination=Gallica&O=NUMM-29058), в которой впервые были введены в вещественный анализ столь любимые многими (и ненавистные многим) эпсилоны и дельты, великий математик Огюстен-Луи Коши (Augustin-Louis Cauchy) обвинял статистиков в Великой французской революции: «Давайте развивать математические науки с рвением, стремясь при этом не выходить за пределы их области применения; невозможно даже представить себе анализ истории с помощью формул или решение моральных дилемм с использованием алгебраической теории или интегрального исчисления».

Эти слова прекрасно сочетаются с празднованием разрушения того, что было так важно для либертарианизма Силиконовой долины, капитализма Шумпетера и некоторых вариантов технической журналистики. Несмотря на успешность при искоренении рентоориентированных видов политического анализа и других дисциплин, при подобном подходе совершенно неверно оцениваются практические навыки и знания, которые зачастую становятся движущей силой науки о данных. Предыдущие главы, посвященные средствам совершенствования многосторонних умений исследователей данных, не оставляют камня на камне от кажущегося противопоставления экспертов по данным и традиционных экспертов. Один из центральных вопросов нашей книги — нормализация повышенной самооценки, особенно завышенной оценки алгоритмов.

Команда по обработке данных Барака Обамы пояснила (https://www.theatlantic.com/technology/archive/2012/11/when-the-nerds-go-marching-in/265325/), что значительной долей своего успеха они обязаны тому, что всерьез отнеслись к опасности завышения самооценки и создали техническую систему, нацеленную на исключение рисков, приводящих к завышению оценки, начиная от подбора и компоновки алгоритмов до обеспечения избыточности прикладных и сетевых систем. «Мне кажется, что республиканцы оплошали с оценкой своих шансов, — объяснил Харпер Рид (Harper Reed) журналисту издания «Атлантик» Алексису Мэдригалу (Alexis Madrigal). — Я знаю, что у нас была лучшая команда по информационным технологиям, с которой мне только доводилось работать, но мы не знали, сработает ли наш метод. Я был совершенно уверен в успехе. Я поставил на кон очень многое. У нас было время. Мы располагали ресурсами. Мы сделали то, что, по нашему мнению, должно было

сработать, и все равно могли не достичь желаемого результата. Что-то могло пойти не так».

Дискуссия по поводу значимости «знаний предметной области» уже давно расколола сообщество исследователей данных на два противоборствующих лагеря. В конце концов, потенциальные возможности неконтролируемого обучения более чем компенсируют деструктивную зависимость от привычных нам категорий социального и научного анализа, как можно видеть в одной из множества статей, восхваляющих (http://articles.latimes.com/print/2012/nov/13/nation/la-na-obama-analytics-20121113) работу команды аналитиков Обамы.

Дэниел Вагнер (Daniel Wagner), 29-летний руководитель команды аналитиков, высказался следующим образом: «Представление о (предвыборной) кампании как поиске групп населения наподобие “мамаша-наседка” или “мамаша-официантка” с тем, чтобы агитировать их голосовать за нужного кандидата, весьма устарело. Предвыборные кампании сейчас ориентируются на обращение в ряды своих сторонников отдельных колеблющихся избирателей. Белые женщины, живущие в пригороде? Это совсем разные люди. Сообщество латиноамериканцев тоже очень многолико и имеет различные интересы. Данные дают возможность просчитать и учесть это разнообразие».

Однако по сравнению с этим путем спасения от отупляющих классификаций движение за возвращение к знаниям предметной области в рамках статистики представляется столь же устаревшим, как формальный интеллектуальный анализ данных.

В ныне печально известной статье из Wall Street Journal (https://www.wsj.com/articles/SB10001424053111904800304576474620336602248) Пегги Нунан (Peggy Noonan) высмеивала объявление о вакансии в аналитическом отделе Обамы: «Производит впечатление политики, какой ее представляют себе марсиане». Кампания была попросту недостаточно человечной, а оперативный центр — одновременно «высокотехнологичным и равнодушным». Она забыла упомянуть, что современные объявления Ромни производят такое же впечатление.

Наука о данных основывается на алгоритмах, но не сводится к ним одним. В основе их использования лежит то, что социологи науки называют неявным знанием (tacit knowledge) (https://www.amazon.com/Tacit-Explicit-Knowledge-Harry-Collins/dp/0226113809), — эмпирические знания, которые непросто или вообще невозможно свести к четко сформулированным правилам. Чтобы успешно применять алгоритмы, нужен человек, это не алгоритмизуется.

Нет ничего важнее, чем предостеречь юных падаванов науки о данных об опасностях переобучения, а также принятия шума в обучающем наборе за полезный сигнал. Во избежание переобучения необходимо обдуманно использовать алгоритмы. Они предоставляют новые возможности, но и думать при этом приходится больше, а не меньше. В 1997 году Петер Хубер (Peter Huber) пояснил: «Задача, как мне представляется, состоит не в замене человеческого гения машинным интеллектом, а в том, чтобы помочь человеческому гению всеми мыслимыми средствами вычислительной техники и искусственного интеллекта, в частности

путем усовершенствования инструментов поиска и использования новейших возможностей анализа». Для описания совершенного владения инструментами контекстного мышления и добродетели обхода рутинных действий отлично подходит слово «импровизация». Необходимо умерять завышенное мнение о возможностях алгоритма, максимально глубоко изучая сам алгоритм и его конкретную реализацию.

Нунан высмеивала размышления о блеске и нищете существующих моделей, образно описанные в опубликованном штабом Обамы объявлении о найме (https://www.kdnuggets.com/jobs/11/07-13-obama2012-predictive-modeling-data-mining-scientists-analysts.html):

- разработка и создание статистических/прогнозных моделей и моделей машинного обучения для работы на местах, в цифровых СМИ, платных СМИ и при сборе пожертвований;
- оценка эффективности предыдущих моделей и принятие решений о необходимости их обновления;
- проектирование и выполнение экспериментов с целью тестирования пригодности и корректности этих моделей на местах.

Непродуманное, механическое использование моделей здесь неуместно — только критический подход и оценка. Никакой незнакомый с местностью марсианин на это не способен: существующие данные — всех видов — слишком важны, чтобы просто их отбросить.

Как же научиться импровизировать? Другими словами, какова оптимальная модель обучения исследователя данных? Чтобы научиться осмысленно импровизировать при работе с алгоритмами и большими объемами данных, нужно вознести в ранг доблести выпас, скрапинг, очистку плохо упорядоченных, неполных и, вероятно, недостаточных данных, умение довести работу с ними до конца. Оптимальная модель обучения требует не узкоспециализированного профессионального образования, а — надо же! — свободных искусств в исходном значении этого термина.

На протяжении столетий искусства, например, математика или музыка назывались свободными потому, что не были чем-то автоматическим, механическим, просто повторяющимся, привычным. Образование в сфере свободных искусств предназначено для свободных людей, то есть людей, свободных размышлять о своих инструментах, действиях и привычках, а не управляемых этими инструментами, а значит, людей, вольных применять их или нет. Это в такой же степени справедливо по отношению к алгоритмам, как и к литературе: при создании достойного своего имени исследователя данных нет места отрыжке. Кроме того, в выборе технологий нет места детерминизму. Ни один исследователь данных не перепутает *возможность* использовать технологию с *необходимостью* ее применять. В разреженном пространстве бесконечного количества операций, которые можно сделать с данными, лишь несколько насыщенных (и интересных) этически приемлемых областей заслуживают наших усилий.

Мэтью Джонс

Советы по профессиональному развитию

У нас найдется немало советов для честолюбивых исследователей данных нового поколения, особенно тех, кто добрался до этой части книги.

В конце концов, множество людей спрашивает нас, имеет ли смысл становиться исследователями данных, поэтому вопрос уже привычен. Мы обычно начинаем консультацию с того, что сами задаем два вопроса.

1. **Какова ваша целевая функция?**

   Для ответа на данный вопрос необходимо определить, что для вас важно. Например, вероятно, важны деньги, поскольку вам потребуется некий их минимум, достаточный для желаемого уровня жизни, и, возможно, вы захотите больше. Это, безусловно, отсекает множество потрясающих проектов, очень полезных для общества, но за которые никто не станет платить (однако не забудьте поискать гранты на подобные проекты!). Вероятно, для вас важно время, которое вы могли бы проводить с близкими и друзьями, — в этом случае вам явно не подойдет работа в стартапе, где все трудятся по 12 часов и спят под рабочими столами. И да, подобные места все еще существуют повсеместно.

   Может быть, вам важно некоторое сочетание пользы для общества, самореализации и интеллектуальной реализации. Обязательно сравните имеющиеся варианты с учетом каждого из перечисленных параметров в отдельности. Это вовсе не одно и то же.

   Каковы ваши цели? Чего вы хотите достичь? Хотите ли стать знаменитым, уважаемым или получить общее признание? Вероятно, оптимальный лично для вас вариант окажется некой комбинацией всего вышеперечисленного. Отдаете ли вы себе отчет, насколько важна для вас каждая из целей по сравнению с другими?

2. **Что вас может сдерживать?**

   Могут существовать внешние факторы, которые вы не в силах контролировать, например, вам нужно жить в определенном месте, рядом с семьей. Не забудьте обдумать также ограничения применительно к деньгам и времени, а также политику компании относительно ежегодного отпуска и отпуска по уходу за ребенком. Кроме того, насколько легко будет «продать» себя? Не считайте себя загнанными в угол, а обдумайте способы показать себя с выгодной стороны: образование, сильные и слабые стороны, а также те черты, которые вы можете или не в силах изменить.

Существует множество возможных решений, оптимизирующих важные для вас параметры и учитывающих имеющиеся ограничения. С нашей точки зрения, главное — то, что подходит именно вам, а не какая работа «лучшая» на рынке труда. У разных людей разные ожидания и требования к карьере.

С одной стороны, не забывайте: никакие ваши решения не принимаются навсегда, так что не слишком переживайте. Вы всегда можете заняться чем-нибудь другим позднее — люди часто меняют работу. С другой стороны, жизнь коротка, поэтому всегда старайтесь двигаться в правильном направлении: оптимизируйте то, что для вас важно, и не стойте на месте.

Наконец, если вам кажется, что ваш образ мышления или точка зрения не такие, как у окружающих, — примите и проанализируйте это, возможно, вы «зациклились» или, напротив, в вашем подходе есть рациональное зерно.

Об авторах

Кэти О'Нил (Cathy O'Neil) получила степень PhD по математике в Гарварде, была аспирантом на кафедре математики MIT и профессором в колледже Барнарда, где опубликовала множество научных статей в области арифметической алгебраической геометрии. Позднее она бросила эту сферу деятельности и перешла в частный сектор. Кэти работала специалистом по количественному анализу в хэдж-фонде D. E. Shaw в разгар кредитного кризиса, затем работала на RiskMetrics — компанию, разрабатывающую ПО для оценки рисков авуаров хэдж-фондов и банков. В настоящее время она работает исследователем данных в стартапах Нью-Йорка, ведет блог на сайте mathbabe.org и принимает участие в движении «Захватите Уолл-стрит».

Рэйчел Шатт (Rachel Schutt) — первый вице-президент по исследованию данных в медиахолдинге News Corp. Она получила степень PhD по статистике в Колумбийском университете и несколько лет работала статистиком в исследовательском подразделении компании Google. Сейчас она занимает должность внештатного профессора на кафедре статистики Колумбийского университета и является одним из основателей комитета по образованию Института исследования и инженерии данных в Колумбийском университете. У нее осталось несколько заявок на патенты со времен ее работы в Google, где она участвовала в создании продуктов, предназначенных для конечного пользователя, посредством создания прототипов алгоритмов и построения моделей поведения пользователей. У нее есть степень магистра математики Нью-Йоркского университета, а также степень магистра в области инженерно-экономических систем и исследования операций Стэнфордского университета. Степень бакалавра по математике (курс повышенной сложности) она получила в Мичиганском университете.

Об иллюстрации на обложке

На обложке нашей книги изображен девятипоясный броненосец (Dasypus novemcinctus) — млекопитающее, широко распространенное в обеих Америках. Novemcinctus дословно переводится с латыни как «с девятью поясами» (по числу колец панциря вокруг середины его тела), хотя на самом деле у него может быть от 7 до 11 поясов. Трехпоясный броненосец, который водится в Южной Америке, — единственный из броненосцев, способный сворачиваться в шар для защиты от врагов, у остальных их видов слишком много пластин для этого.

Шкура броненосца, вероятно, самая заметная его особенность. Коричнево-серая и кожистая, она состоит из чешуйчатых пластин, именуемых *щитками* и покрывающих все его тело, кроме живота. У этих животных есть также мощные когти, пригодные для рытья земли, с помощью которых они роют себе по нескольку нор в пределах своей территории, а затем помечают их выделениями пахучих желез. Девятипоясные броненосцы обычно весят от 2,5 до 6,5 кг и размером с крупную домашнюю кошку. Питаются они в основном насекомыми, хотя едят и фрукты, небольших пресмыкающихся и яйца.

Девятипоясные броненосцы могут подпрыгивать в воздух на метр, если их спугнуть. Хотя такая реакция может отпугнуть их естественных врагов, она часто оказывается фатальной, если броненосца напугал приближающийся автомобиль, ведь броненосец сталкивается в таком случае с бампером. Еще одна неудачная взаимосвязь человека с девятипоясными броненосцами состоит в том, что это одни из немногих животных, которые могут быть носителями проказы, — описаны случаи, когда люди заражались этой болезнью, после того как ели или трогали броненосцев.

Изображение на обложке взято из «Зоологии» Джорджа Шоу (George Shaw) и было раскрашено Карен Монтгомери (Karen Montgomery).

Кэти О'Нил, Рэйчел Шатт

Data Science. Инсайдерская информация для новичков. Включая язык R

Перевели с английского *И. Пальти*, *К. Синица*, *С. Черников*

Заведующая редакцией	*Ю. Сергиенко*
Руководитель проекта	*О. Сивченко*
Ведущий редактор	*Н. Гринчик*
Литературный редактор	*Н. Хлебина*
Художественный редактор	*С. Заматевская*
Корректоры	*О. Андриевич, Е. Павлович*
Верстка	*К. Подольцева-Шабович*

Изготовлено в России. Изготовитель: ООО «Прогресс книга».
Место нахождения и фактический адрес: 194044, Россия, г. Санкт-Петербург,
Б. Сампсониевский пр., д. 29А, пом. 52. Тел.: +78127037373.
Дата изготовления: 09.2018. Наименование: книжная продукция.
Срок годности: не ограничен.
Налоговая льгота — общероссийский классификатор продукции ОК 034-2014,
58.11.12 — Книги печатные профессиональные, технические и научные.
Импортер в Беларусь: ООО «ПИТЕР М», 220020, РБ, г. Минск, ул. Тимирязева,
д. 121/3, к. 214, тел./факс: 208 80 01.
Подписано в печать 29.08.18. Формат 70x100/16. Бумага офсетная. Усл. п. л. 29,670.
Тираж 1000. Заказ 0000

КНИГА-ПОЧТОЙ

ЗАКАЗАТЬ КНИГИ ИЗДАТЕЛЬСКОГО ДОМА «ПИТЕР» МОЖНО ЛЮБЫМ УДОБНЫМ ДЛЯ ВАС СПОСОБОМ:

- на нашем сайте: **www.piter.com**
- по электронной почте: **books@piter.com**
- по телефону: **(812) 703-73-74**

ВЫ МОЖЕТЕ ВЫБРАТЬ ЛЮБОЙ УДОБНЫЙ ДЛЯ ВАС СПОСОБ ОПЛАТЫ:

- Наложенным платежом с оплатой при получении в ближайшем почтовом отделении.
- С помощью банковской карты. Во время заказа вы будете перенаправлены на защищенный сервер нашего оператора, где сможете ввести свои данные для оплаты.
- Электронными деньгами. Мы принимаем к оплате Яндекс.Деньги, Webmoney и Kiwi-кошелек.
- В любом банке, распечатав квитанцию, которая формируется автоматически после совершения вами заказа.

ВЫ МОЖЕТЕ ВЫБРАТЬ ЛЮБОЙ УДОБНЫЙ ДЛЯ ВАС СПОСОБ ДОСТАВКИ:

- Посылки отправляются через «Почту России». Отработанная система позволяет нам организовывать доставку ваших покупок максимально быстро. Дату отправления вашей покупки и дату доставки вам сообщат по e-mail.
- Вы можете оформить курьерскую доставку своего заказа (более подробную информацию можно получить на нашем сайте www.piter.com).
- Можно оформить доставку заказа через почтоматы (адреса почтоматов можно узнать на нашем сайте www.piter.com).

ПРИ ОФОРМЛЕНИИ ЗАКАЗА УКАЖИТЕ:

- фамилию, имя, отчество, телефон, e-mail;
- почтовый индекс, регион, район, населенный пункт, улицу, дом, корпус, квартиру;
- название книги, автора, количество заказываемых экземпляров.

БЕСПЛАТНАЯ ДОСТАВКА:

- курьером по Москве и Санкт-Петербургу при заказе на сумму **от 2000 руб.**
- почтой России при предварительной оплате заказа на сумму **от 2000 руб.**

ИЗДАТЕЛЬСКИЙ ДОМ
ПИТЕР®
WWW.PITER.COM

ИЗДАТЕЛЬСКИЙ ДОМ «ПИТЕР» предлагает профессиональную, популярную и детскую развивающую литературу

Заказать книги оптом можно в наших представительствах

РОССИЯ

Санкт-Петербург: м. «Выборгская», Б. Сампсониевский пр., д. 29а
тел./факс: (812) 703-73-83, 703-73-72; e-mail: sales@piter.com

Москва: м. «Электрозаводская», Семеновская наб., д. 2/1, стр. 1, 6 этаж
тел./факс: (495) 234-38-15; e-mail: sales@msk.piter.com

Воронеж: тел.: 8 951 861-72-70; e-mail: hitsenko@piter.com

Екатеринбург: ул. Толедова, д. 43а; тел./факс: (343) 378-98-41, 378-98-42;
e-mail: office@ekat.piter.com; skype: ekat.manager2

Нижний Новгород: тел.: 8 930 712-75-13; e-mail: yashny@yandex.ru; skype: yashny1

Ростов-на-Дону: ул. Ульяновская, д. 26
тел./факс: (863) 269-91-22, 269-91-30; e-mail: piter-ug@rostov.piter.com

Самара: ул. Молодогвардейская, д. 33а, офис 223
тел./факс: (846) 277-89-79, 277-89-66; e-mail: pitvolga@mail.ru,
pitvolga@samara-ttk.ru

БЕЛАРУСЬ

Минск: ул. Розы Люксембург, д. 163; тел./факс: +37 517 208-80-01, 208-81-25;
e-mail: og@minsk.piter.com

Издательский дом «Питер» приглашает к сотрудничеству авторов:
тел./факс: (812) 703-73-72, (495) 234-38-15; e-mail: ivanovaa@piter.com
Подробная информация здесь: http://www.piter.com/page/avtoru

Издательский дом «Питер» приглашает к сотрудничеству зарубежных торговых партнеров или посредников, имеющих выход на зарубежный рынок: *тел./факс: (812) 703-73-73; e-mail: sales@piter.com*

Заказ книг для вузов и библиотек:
тел./факс: (812) 703-73-73, доб. 6243; e-mail: uchebnik@piter.com

Заказ книг по почте: *на сайте www.piter.com; тел.: (812) 703-73-74, доб. 6216; e-mail: books@piter.com*

Вопросы по продаже электронных книг: *тел.: (812) 703-73-74, доб. 6217; e-mail: kuznetsov@piter.com*