

*Памяти любимой матушки,
Клэр Пфеффер.
Да будет память её благословенна.*



ОГЛАВЛЕНИЕ

Предисловие	13
Вступление	15
Благодарности	17
Об этой книге	19
Структура книги	20
О коде и упражнениях	21
Об авторе	21
Автор в сети	22
Об иллюстрации на обложке	22
ЧАСТЬ I	
Введение в вероятностное программирование и систему Figaro	23
Глава 1. О вероятностном программировании в двух словах	24
1.1. Что такое вероятностное программирование?	24
1.1.1. Как мы высказываем субъективное суждение?	25
1.1.2. Системы вероятностных рассуждений помогают принимать решения	26
1.1.3. Система вероятностных рассуждений может рассуждать тремя	
способами	28
1.1.4. Система вероятностного программирования: система вероятностных	
рассуждений, выраженная на языке программирования	32
1.2. Зачем нужно вероятностное программирование?	37
1.2.1. Улучшенные вероятностные рассуждения	37
1.2.2. Улучшенные языки имитационного моделирования	38
1.3. Введение в Figaro, язык вероятностного программирования	40
1.3.1. Figaro и Java: построение простой системы вероятностного	
программирования	43
1.4. Резюме	48
1.5. Упражнения	48
Глава 2. Краткое руководство по языку Figaro	50
2.1. Введение в Figaro	50
2.2. Создание модели и выполнение алгоритма вывода на примере Hello World	52
2.2.1. Построение первой модели	53
2.2.2. Выполнение алгоритма вывода и получение ответа на запрос	54

2.2.3. Построение моделей и задание наблюдений	55
2.2.4. Анатомия построения модели	56
2.2.5. Повторяющиеся элементы: когда они совпадают, а когда различаются?	58
2.3. Базовые строительные блоки: атомарные элементы	59
2.3.1. Дискретные атомарные элементы	60
2.3.2. Непрерывные атомарные элементы	61
2.4. Комбинирование атомарных элементов с помощью составных	64
2.4.1. Элемент If	64
2.4.2. Элемент Dist	65
2.4.3. Составные версии атомарных элементов	66
2.5. Построение более сложных моделей с помощью Apply и Chain	67
2.5.1. Элемент Apply	67
2.5.2. Элемент Chain	70
2.6. Задание фактов с помощью условий и ограничений	73
2.6.1. Наблюдения	73
2.6.2. Условия	74
2.6.3. Ограничения	75
2.7. Резюме	78
2.8. Упражнения	78

Глава 3. Создание приложения вероятностного программирования 80

3.1. Общая картина	80
3.2. Выполнение кода	83
3.3. Архитектура приложения фильтра спама	86
3.3.1. Архитектура аналитического компонента	86
3.3.2. Архитектура компонента обучения	90
3.4. Проектирование модели почтового сообщения	92
3.4.1. Выбор элементов	93
3.4.2. Определение зависимостей	95
3.4.3. Определение функциональных форм	97
3.4.4. Использование числовых параметров	100
3.4.5. Работа с дополнительными знаниями	102
3.5. Разработка аналитического компонента	104
3.6. Разработка компонента обучения	108
3.7. Резюме	112
3.8. Упражнения	113

ЧАСТЬ II

Написание вероятностных программ 115

Глава 4. Вероятностные модели и вероятностные программы 116

4.1. Определение вероятностной модели	117
---	-----

4.1.1. Выражение общих знаний в виде распределения вероятности возможных миров	117
4.1.2. Подробно о распределении вероятности	120
4.2. Использование вероятностной модели для ответа на запросы	121
4.2.1. Применение условий для получения апостериорного распределения вероятности	122
4.2.2. Получение ответов на запросы	124
4.2.3. Применение вероятностного вывода	126
4.3. Составные части вероятностных моделей	127
4.3.1. Переменные	127
4.3.2. Зависимости	129
4.3.3. Функциональные формы	134
4.3.4. Числовые параметры	138
4.4. Порождающие процессы	140
4.5. Модели с непрерывными переменными	144
4.5.1. Бета-биномиальная модель	145
4.5.2. Представление непрерывных переменных	146
4.6. Резюме	150
4.7. Упражнения	150

Глава 5. Моделирование зависимостей с помощью байесовских и марковских сетей. 152

5.1. Моделирование зависимостей	153
5.1.1. Направленные зависимости	153
5.1.2. Ненаправленные зависимости	159
5.1.3. Прямые и косвенные зависимости	162
5.2. Байесовские сети	163
5.2.1. Определение байесовской сети	163
5.2.2. Как байесовская сеть определяет распределение вероятности	166
5.2.3. Рассуждения с применением байесовской сети	166
5.3. Изучение примера байесовской сети	169
5.3.1. Проектирование модели диагностики компьютерной системы	169
5.3.2. Рассуждения с помощью модели диагностики компьютерной системы	174
5.4. Применение вероятностного программирования для обобщения байесовских сетей: предсказание успешности продукта	179
5.4.1. Проектирование модели для предсказания успешности продукта	180
5.4.2. Рассуждения с помощью модели для предсказания успешности продукта	185
5.5. Марковские сети	187
5.5.1. Определение марковской сети	187
5.5.2. Представление марковских сетей и рассуждения с их помощью	191
5.6. Резюме	195
5.7. Упражнения	195

Глава 6. Использование коллекций Scala и Figaro для построения моделей	198
6.1. Работа с коллекциями Scala	199
6.1.1. Моделирование зависимости многих переменных от одной	200
6.1.2. Создание иерархических моделей	203
6.1.3. Моделирование одновременной зависимости от двух переменных	205
6.2. Работа с коллекциями Figaro	208
6.2.1. Почему коллекции Figaro полезны?	208
6.2.2. Иерархическая модель и коллекции Figaro	210
6.2.3. Совместное использование коллекций Scala и Figaro	212
6.3. Моделирование ситуаций с неизвестным числом объектов	215
6.3.1. Открытая вселенная с неизвестным числом объектов	215
6.3.2. Массивы переменной длины	216
6.3.3. Операции над массивами переменной длины	217
6.3.4. Пример: прогнозирование продаж неизвестного числа новых продуктов	218
6.4. Работа с бесконечными процессами	220
6.4.1. Характеристика Process	220
6.4.2. Пример: моделирование состояния здоровья во времени	222
6.4.3. Использование процесса	224
6.5. Резюме	225
6.6. Упражнения	226
Глава 7. Объектно-ориентированное вероятностное моделирование	228
7.1. Объектно-ориентированные вероятностные модели	229
7.1.1. Элементы объектно-ориентированного моделирования	230
7.1.2. Еще раз о модели принтера	232
7.1.3. Рассуждения о нескольких принтерах	236
7.2. Добавление связей в объектно-ориентированные модели	240
7.2.1. Описание общей модели на уровне классов	240
7.2.2. Описание ситуации	243
7.2.3. Представление модели социальной сети на Figaro	246
7.3. Моделирование реляционной неопределенности и неопределенности типа	249
7.3.1. Коллекции элементов и ссылки	249
7.3.2. Модель социальной сети с реляционной неопределенностью	252
7.3.3. Модель принтера с неопределенностью типа	254
7.4. Резюме	257
7.5. Упражнения	257
ГЛАВА 8. Моделирование динамических систем	259
8.1. Динамические вероятностные модели	260
8.2. Типы динамических моделей	261

8.2.1. Марковские цепи	261
8.2.2. Скрытые марковские модели	265
8.2.3. Динамические байесовские сети	268
8.2.4. Модели с нестационарной структурой	272
8.3. Моделирование систем, работающих неопределенно долго	277
8.3.1. Универсумы в Figaro	277
8.3.2. Использование универсумов для моделирования постоянно работающих систем	279
8.3.3. Следящее приложение	281
8.4. Резюме	284
8.5. Упражнения	284

ЧАСТЬ III

Вывод	287
--------------------	------------

Глава 9. Три правила вероятностного вывода	288
---	------------

9.1. Цепное правило: построение совместных распределений по условным распределениям вероятности	290
9.2. Правило полной вероятности: получение ответов на простые запросы из совместного распределения	294
9.3. Правило Байеса: вывод причин из следствий	297
9.3.1. Понимание, причина, следствие и вывод	297
9.3.2. Правило Байеса на практике	299
9.4. Байесовское моделирование	301
9.4.1. Оценивание асимметрии монеты	303
9.4.2. Предсказание результата следующего подбрасывания	307
9.5. Резюме	312
9.6. Упражнения	312

Глава 10. Факторные алгоритмы вывода	314
---	------------

10.1. Факторы	315
10.1.1. Что такое фактор?	315
10.1.2. Факторизация распределения вероятности с помощью цепного правила	318
10.1.3. Задание запросов с факторами с помощью правила полной вероятности	320
10.2. Алгоритм исключения переменных	324
10.2.1. Графическая интерпретация ИП	325
10.2.2. Исключение переменных как алгебраическая операция	329
10.3. Использование алгоритма ИП	332
10.3.1. Особенности ИП в Figaro	332
10.3.2. Проектирование модели, эффективно поддерживающей ИП	334
10.3.3. Приложения алгоритма ИП	338
10.4. Распространение доверия	342

10.4.1. Основные принципы РД	342
10.4.2. Свойства циклического РД.....	343
10.5. Использование алгоритма РД	345
10.5.1. Особенности РД в Figaro	346
10.5.2. Проектирование модели, эффективно поддерживающей РД	347
10.5.3. Приложения алгоритма РД.....	349
10.6. Резюме	350
10.7. Упражнения	350
Глава 11. Выборочные алгоритмы	353
11.1. Принцип работы выборочных алгоритмов	354
11.1.1. Прямая выборка.....	355
11.1.2. Выборка с отклонением	360
11.2. Выборка по значимости	363
11.2.1. Как работает выборка по значимости.....	364
11.2.2. Выборка по значимости в Figaro.....	367
11.2.3. Полезность выборки по значимости.....	368
11.2.4. Приложения алгоритма выборки по значимости	370
11.3. Алгоритм Монте-Карло по схеме марковской цепи	373
11.3.1. Как работает МСМС	374
11.3.2. Алгоритм МСМС в Figaro: алгоритм Метрополиса-Гастингса.....	378
11.4. Настройка алгоритма МГ	382
11.4.1. Специальные схемы предложения	384
11.4.2. Избегание жестких условий	388
11.4.3. Приложения алгоритма МГ	389
11.5. Резюме	391
11.6. Упражнения	392
Глава 12. Решение других задач вывода	394
12.1. Вычисление совместных распределений	395
12.2. Вычисление наиболее вероятного объяснения	397
12.2.1. Вычисление и запрос НВО в Figaro.....	400
12.2.2. Использование алгоритмов для ответа на запросы НВО	402
12.2.3. Приложения алгоритмов НВО	409
12.3. Вычисление вероятности фактов	410
12.3.1. Наблюдение фактов для вычисления вероятности фактов	411
12.3.2. Выполнение алгоритмов вычисления вероятности фактов.....	414
12.4. Резюме	415
12.5. Упражнения	415
Глава 13. Динамические рассуждения и обучение параметров	417
13.1. Мониторинг состояния динамической системы	418
13.1.1. Механизм мониторинга	419
13.1.2. Алгоритм фильтрации частиц.....	421

13.1.3. Применения фильтрации	424
13.2. Обучение параметров модели	425
13.2.1. Байесовское обучение	426
13.2.2. Обучение методом максимального правдоподобия и MAB.....	430
13.3. Дальше вместе с Figaro.....	439
13.4. Резюме	440
13.5. Упражнения	440
Приложение А. Получение и установка Scala и Figaro	443
А.1. Использование sbt.....	443
А.2. Установка и запуск Figaro без sbt	444
А.3. Сборка из исходного кода	445
Приложение В. Краткий обзор систем вероятностного программирования	447
Предметный указатель	450



ПРЕДИСЛОВИЕ

В 1814 году Пьер-Симон Лаплас писал: «по большей части важнейшие жизненные вопросы являются на самом деле лишь задачами теории вероятностей». Спустя сто лет после этих слов на такие вопросы можно было ответить только одним способом (сохраняя верность мнению Лапласа): проанализировать каждую задачу на бумаге, выразить результат в виде формулы и вычислить значение формулы, вручную подставив в нее числа. Наступление эры компьютеров мало что изменило. Просто стало возможно вычислять более сложные формулы, а анализ с помощью «пера и бумаги» теперь может занимать сотни страниц.

Для анализа вероятностной задачи необходимо построить *вероятностную модель*, в которой описывается пространство возможных исходов и каждому из них каким-то образом сопоставляется числовая вероятность. Раньше вероятностные модели формулировались на смеси естественного языка и полужормальной математической нотации. На основе модели с помощью некоторых математических манипуляций выводилась формула или алгоритм для вычисления ответов. Обе стадии были трудоемкими, чреватými ошибками и зависели от конкретной задачи, поэтому применение теории вероятностей на практике сталкивалось с серьезными ограничениями. Вопреки Лапласу, важнейшие жизненные вопросы оставались неразрешенными.

Первым крупным продвижением стала разработка *формальных языков*, в частности байесовских и марковских сетей, для выражения вероятностных моделей. У формального языка имеется точный синтаксис, определяющий, какие выражения допустимы, и точная семантика, определяющая, что означает каждое допустимое выражение (т. е. какая именно вероятностная модель представлена данным выражением). Поэтому появилась возможность описывать вероятностные модели в машиночитаемом виде и разработать единый алгоритм вычисления следствий *любой* выразимой вероятностной модели.

Но в этой бочке меда есть одна ложка дегтя: отсутствие *выразимых* вероятностных моделей. Байесовские и марковские сети как формальные языки обладают очень ограниченными выразительными возможностями. В каком-то смысле их можно назвать вероятностными аналогами булевых схем. Чтобы понять, в чем состоит ограничение, рассмотрим написание программы расчета платежной ведомости для крупной компании. На языке высокого уровня, например Java, она может состоять из десятков тысяч строк кода. А теперь представьте себе реализацию той же функциональности посредством соединения логических вентиляей. Эта задача,

по всей видимости, абсолютно неразрешима. Схема оказалась бы невообразимо огромной, сложной и непонятной, поскольку логическим схемам недостает выразительной силы, соответствующей структуре задачи.

В 1997 году Ави Пфедфер, автор этой книги, тогда еще студент, в соавторстве со своим научным руководителем Дафной Коллер и коллегой Дэвидом Макаллестером, опубликовал пионерскую работу по языкам вероятностного программирования (ЯВП) (probabilistic programming language – PPL), в которой высказана важнейшая идея, связывающая теорию вероятностей с выразительными возможностями языков программирования высокого уровня. Идея заключалась в том, что программу можно рассматривать как вероятностную модель, если ввести некоторые стохастические элементы и определить смысл программы как вероятность каждого возможного пути выполнения. Эта идея вводит полезную связь между двумя важнейшими областями математики, и мы еще только приступаем к исследованию открывающихся на этом пути возможностей.

Книга представляет собой неформальное введение в круг этих идей на примере языка Figaro для иллюстрации основных концепций и их применения. Автор избегает ненужной математики и акцентирует внимание на реальных тщательно подобранных примерах, которые подробно объясняет. Книга доступна любому человеку со стандартной подготовкой в области программирования. Заодно трудолюбивый читатель с меньшими, чем обычно, усилиями освоит принципы и технику байесовского вывода и статистического обучения. Но, пожалуй, еще важнее тот факт, что читатель приобретет навыки моделирования – одно из важнейших умений любого ученого или инженера. Figaro и другие ЯВП позволяют выражать такие модели непосредственно, быстро и точно.

Эта книга – важный шаг на пути вывода вероятностного программирования из научно-исследовательских лабораторий, где оно зародилось, в реальный мир. Без сомнения, такое столкновение с реальностью выявит ограничения имеющихся систем ВП, и у лабораторий появятся новые задачи. С другой стороны, читатели этой книги обязательно откроют неожиданные способы применения Figaro и подобных ему языков к широкому кругу новых задач, о которых авторы даже не подозревали.

Стюарт Рассел

профессор информатики
Калифорнийский университет в Беркли



ВСТУПЛЕНИЕ

Вероятностное программирование – новая захватывающая область исследований, которая привлекает все больший интерес. Постепенно она прокладывает себе дорогу из академических кругов в мир программистов. По сути дела, вероятностное программирование – это новый способ создания вероятностных моделей, позволяющих предсказывать или выводить новые факты, которых нет в результатах наблюдений. Вероятностные рассуждения давно считаются одним из основных подходов к машинному обучению, где модель описывает то, что известно из опыта. Но раньше такие системы были ограничены простыми фиксированными структурами типа байесовских сетей. Вероятностное программирование освобождает системы вероятностных рассуждений от этих оков, предоставляя всю мощь языков программирования для представления моделей. Можно считать, что это аналог перехода от логических схем к языкам высокого уровня.

Я начал заниматься вероятностным программированием с подростковых лет, когда писал футбольный симулятор на BASIC, хотя в то время еще не осознавал этого. В программе встречались предложения вида «GOTO 1730 + RANDOM * 5», призванные выразить случайную последовательность событий. После тщательной настройки эмулятор оказался настолько реалистичным, что я мог развлекаться с ним часами. Разумеется, с тех пор вероятностное программирование далеко ушло от предложений GOTO со случайным адресом.

В 1997 я в соавторстве с Дафной Коллер и Дэвидом Макаллестером написал одну из первых работ по вероятностному программированию. В ней описывался Lisp-подобный язык, но главным новшеством был алгоритм выведения вероятных аспектов программы на основе наблюдаемых результатов ее работы. Это и вывело вероятностное программирование из разряда типичных языков вероятностного моделирования, предоставив средства, позволявшие программе не только продвигаться вперед по одному из возможных путей выполнения, но и строить обратные рассуждения и делать выводы о том, почему получен наблюдаемый результат.

В начале 2000-х годов я разработал IVAL (произносится «айболл») – первую систему вероятностного программирования общего назначения, основанную на функциональном программировании. IVAL обладала высокой выразительной способностью и содержала инновационные алгоритмы вывода, но с годами меня все сильнее раздражали ее ограничения и главное из них – трудность взаимодействия с данными и интеграции с приложениями. Поэтому в 2009 году я начал разрабатывать новую систему вероятностного программирования, которую назвал

Figaro. При проектировании Figaro на первый план выдвигалась практичность, но не жертвуя мощностью вероятностного программирования. Это навело меня на мысль реализовать Figaro в виде библиотеки на языке Scala, что упрощает интеграцию вероятностных моделей с приложениями для виртуальной машины Java. В то же время Figaro предлагала, пожалуй, самый широкий спектр средств представления и алгоритмов выводов из всех известных мне систем вероятностного программирования. В настоящее время Figaro – проект с открытым исходным кодом на GitHub, его текущая версия имеет номер 3.3.

Овладеть вероятностным программированием не так-то просто, потому что необходимы разнообразные навыки и, прежде всего, умение строить вероятностные модели и писать программы. Для многих программистов написание программ – естественный процесс, но вероятностные модели окутаны тайной. Эта книга призвана сорвать покров таинственности, показать, как можно эффективно программировать в процессе создания моделей и помочь в освоении систем вероятностного программирования. Не предполагается, что читатель имеет подготовку в области машинного обучения или вероятностных рассуждений. Знакомство с функциональным программированием и языком Scala будет полезно, но для чтения книги не нужно быть опытным специалистом по Scala. Зато вполне может статься, что после прочтения вы станете программировать на Scala увереннее.

Прочитав книгу, вы сможете проектировать вероятностные модели для различных приложений, извлекающих осмысленную информацию из данных, и степень доктора по машинному обучению для этого не потребуется. Если вы специализируетесь в какой-то предметной области, то книга поможет выразить модели, которые имеются у вас в голове или на бумаге, и сделать их операционными, т. е. допускающими вычисление и анализ вариантов. Если вы – специалист по анализу данных, то, прочитав книгу, сможете разрабатывать более развитые, детальные и потенциально более точные модели, чем позволяют другие средства. Программисту или архитектору, стремящемуся включить в свою систему умение рассуждать в условиях неопределенности, книга поможет не только построить вероятностную модель, но и интегрировать ее с приложением. По какой бы причине вы ни выбрали эту книгу, я надеюсь, что она доставит вам удовольствие и окажется полезной.



БЛАГОДАРНОСТИ

Эта книга потребовала многих лет работы: от первых идей, касающихся вероятностного программирования, через создание систем IVAL и Figaro до замысла, написания и шлифовки книги в сотрудничестве с издательством Manning. Не счесть людей, усилия которых помогли созданию книги.

Своим существованием эта книга в значительной степени обязана трудам моих коллег из компании Charles River Analytics: Джо Гормана (Joe Gorman), Скотта Харрисона (Scott Harrison), Майкла Ховарда (Michael Howard), Ли Келлога (Lee Kellogg), Элисон О'Коннор (Alison O'Connor), Майка Репоза (Mike Repos), Брайна Раттенберга (Brian Ruttenberg) и Гленна Таката (Glenn Takata). Также спасибо Скотту Нилу Рейли (Scott Neal Reilly), который поддерживал Figaro с самого начала.

Большую часть того, что я знаю об искусственном интеллекте и машинном обучении, я почерпнул от Дафны Коллер, моего научного руководителя и коллеги. Стюарт Рассел предоставил мне первую возможность изучать искусственный интеллект и воодушевлял на протяжении всей моей карьеры. А недавно мы начали работать в одном проекте, и он написал предисловие к этой книге. Майкл Стоунбрейкер предоставил мне шанс заняться исследованиями в своем проекте СУБД Postgres, и я много узнал о построении систем, работая в его группе. Алон Халеви (Alon Halevy) пригласил меня провести лето вместе с ним в AT&T Labs, где я имел первые беседы о вероятностном программировании с Дэвидом Макаллестером; в результате родилась на свет статья о вероятностном диалекте Lisp, написанная в соавторстве с Дафной. С Лайзой Гетур (Lise Getoor), делившей со мной кабинет и общую работу, я мог обсуждать идеи, которые вынашивал.

Я глубоко признателен Алексу Ихлеру (Alex Ihler), который любезно проверил книгу на предмет отсутствия технических ошибок. Алекс также оказался великолепным слушателем, на котором последние пару лет я опробовал все идеи, относящиеся к логическому выводу.

Многие люди делились своими замечаниями на различных этапах разработки, в том числе Равишанкар Раджагопалан (Ravishankar Rajagopalan) и Шабеш Балан (Shabeesh Balan), Крис Хенеган (Chris Heneghan), Клеменс Баадер (Clemens Baader), Кристофер Вебер (Cristofer Weber), Эрл Бингэм (Earl Bingham), Джузеппе де Марко (Giuseppe de Marco), Джауме Валлс (Jaume Valls), Хавьер Гуэрра Гиральдес (Javier Guerra Giraldez), Костас Пассадис (Kostas Passadis), Лука Кампобассо (Luca Campobasso), Лукас Наллиндо (Lucas Gallindo), Марк Элстон

(Mark Elston), Марк Миллер (Mark Miller), Нитин Годе (Nitin Gode), Одиссеас Пентаколос (Odysseyas Pentakolos), Петр Рабинович (Peter Rabinovitch), Филлип Брэдфорд (Phillip Bradford), Стивен Уэйкли (Stephen Wakely), Тапош Дутта Рой (Taposh Dutta Roy), Унникришнан Кумар (Unnikrishnan Kumar).

Я благодарен многим замечательным сотрудникам издательства Manning, которые способствовали выходу книги из печати. Отдельное спасибо редактору Дэну Махарри (Dan Maharry), который сделал книгу куда лучше, чем мог бы я сам, и Фрэнку Полманну (Frank Pohlmann), который посоветовал мне написать книгу и помогал на подготовительных этапах.

Спасибо Исследовательской лаборатории ВВС и Управлению перспективных научно-исследовательских проектов Министерства обороны США (DARPA) за финансирование части описанной в этой книге работы в рамках программы РРАМЛ (вероятностное программирование для развития методов машинного обучения). Отдельная благодарность нескольким руководителям программ в DARPA: Бобу Кохоуту (Bob Kohout), Тони Фальконе (Tony Falcone), Кэтлин Фишер (Kathleen Fisher) и Сурешу Джаганнатану (Suresh Jagannathan), которые поверили в вероятностное программирование и прилагали все силы, чтобы оно стало реальностью. Этот материал основан на работе, которая финансировалась ВВС США по контракту № FA8750-14-C-0011. Мнения, открытия и заключения или рекомендации автора, встречающиеся в этом материале, не обязательно отражают точку зрения ВВС США.

И наконец, эта книга не состоялась бы без любви и поддержки со стороны моей семьи. Спасибо моей супруге Дебби Гелбер и моим детям Дине, Номи и Рути за то, что они такие чудесные. И вечная благодарность моей маме Клэр Пфедфер, воспитавшей меня с любовью. Эту книгу я посвящаю твоей памяти.



ОБ ЭТОЙ КНИГЕ

Многие решения, будь то в бизнесе, науке, военном деле или повседневной жизни, принимаются в условиях неопределенности. Если разные факторы толкают в разных направлениях, то как узнать, на что обращать больше внимания? Вероятностные модели дают средство выразить всю информацию, относящуюся к конкретной ситуации. Вероятностные рассуждения позволяют использовать эти модели, чтобы найти вероятности величин, наиболее существенных для принятия решения. С помощью вероятностных рассуждений можно *предсказать* наиболее вероятное развитие событий: хорошо ли будет продаваться ваш продукт по назначенной цене, будет ли пациент отвечать на предложенное лечение, сможет ли кандидат победить на выборах, если займет определенную позицию? Вероятностные рассуждения можно использовать и для *вывода* вероятных причин уже случившихся событий: если продажи провалились, то не потому ли, что была назначена слишком высокая цена?

Вероятностное рассуждение также является одним из основных подходов к машинному обучению. Мы представляем исходные допущения о предметной области в виде вероятностной модели, например общей модели поведения пользователей в ответ на появление продуктов на рынке. Затем, получив обучающие данные, скажем о реакции пользователей на конкретные продукты, мы модифицируем первоначальные представления и получаем новую модель. Эту модель можно уже использовать для предсказания будущих событий, например успеха перспективного продукта, или для объяснения причин наблюдавшихся событий, скажем провала нового продукта.

Раньше для выражения вероятностных рассуждений применялись специализированные языки представления вероятностных моделей. В последние годы мы поняли, что можно использовать обычные языки программирования, и это положило начало вероятностному программированию. У такого подхода три основных достоинства. Во-первых, при построении моделей в нашем распоряжении все средства языка программирования: развитые структуры данных, поток управления и т. д. Во-вторых, вероятностную модель легко интегрировать с другими приложениями. И, в-третьих, для рассуждений о моделях можно использовать универсальные алгоритмы логического вывода.

Задача этой книги – вооружить вас знаниями, необходимыми для использования вероятностного программирования в повседневной деятельности. В частности, объясняется:

- как строить вероятностные модели и выражать их в виде вероятностных программ;
- как устроены вероятностные рассуждения и как они реализованы в различных алгоритмах вывода;
- как с помощью системы вероятностного программирования Figaro создавать практичные вероятностные программы.

Система Figaro реализована в виде библиотеки на языке Scala. Как и в Scala, в ней сочетаются функциональный и объектно-ориентированный стили программирования. Некоторое знакомство с функциональным программированием было бы полезно. Но в этой книге не используются сложные концепции функционального программирования, так что для понимания хватит и скромных познаний. Знание Scala также не помешало бы. Хотя встречающиеся конструкции Scala часто объясняются, эта книга не является введением в Scala. Но экзотические особенности Scala почти не используются, так что поверхностного знакомства будет достаточно.

Структура книги

Часть 1 представляет собой введение в вероятностное программирование и систему Figaro. В начале главы 1 объясняется, что такое вероятностное программирование и почему оно полезно, а затем приводится краткое введение в Figaro. Глава 2 – учебное пособие по работе с Figaro, она позволит вам быстро приступить к написанию вероятностных программ. В главе 3 вы найдете полное приложение вероятностного программирования – фильтр спама – включающее компонент, который рассуждает о том, является данное почтовое сообщение хорошим или спамным, и компонент, который обучает вероятностную модель на данных. Цель главы 3 – дать общую картину взаимосвязей разных частей, перед тем как приступить к детальному изучению приемов моделирования.

Часть 2 целиком посвящена построению вероятностных программ. В главе 4 представлены основные сведения о вероятностных моделях и вероятностных программах, которые необходимы для понимания того, что в действительности стоит за созданием таких программ. В главе 5 описаны две системы моделирования, лежащие в основе вероятностного программирования: байесовские и марковские сети. Главы 6–8 содержат набор полезных приемов программирования, применяемых при создании более сложных программ. В главе 6 речь идет о коллекциях в Scala и Figaro, позволяющих скомпоновать много переменных одного типа. Глава 7 посвящена объектно-ориентированному программированию, которое в функциональных программах не менее полезно, чем в обычных. В главе 8 рассказано о моделировании динамических систем. Динамической называется система, состояние которой изменяется со временем, и в этой главе подробно рассматривается это весьма распространенное и важное применение вероятностных рассуждений.

Из части 3 вы узнаете о вероятностных алгоритмах вывода. Понимать, что такое вывод, необходимо для эффективного применения вероятностного программирования; это позволит выбрать подходящий для решения задачи алгоритм, правиль-

но настроить его и выразить модель так, чтобы она эффективно поддерживала рассуждения. Я старался соблюсти баланс между изложением теории, стоящей за алгоритмами, и практическими примерами их использования. В основополагающей главе 9 сформулированы три правила, в которых заключены основные идеи вероятностного вывода. В главах 10 и 11 описаны два основных семейства алгоритмов вывода. В главе 10 рассматриваются факторные алгоритмы, в том числе введение в факторы и принципы их работы, а также алгоритмы исключения переменных и распространения доверия. В главе 11 обсуждаются выборочные алгоритмы с акцентом на выборку по значимости и методы Монте-Карло по схеме марковской цепи. Если в главах 10 и 11 упор сделан на базовом запросе – вычислении вероятностей величин, представляющих интерес, то в главе 12 показано, как факторные и выборочные алгоритмы можно использовать для вычисления других запросов, например, совместной вероятности нескольких переменных, наиболее вероятных значений переменных и вероятности эмпирических данных. Наконец, в главе 13 обсуждаются две более сложных, но важных задачи вывода: мониторинг динамической системы, изменяющейся во времени, и нахождение числовых параметров вероятностной модели на основе обучающих данных.

Каждая глава сопровождается упражнениями. Это могут быть как простые вычисления, так и задачи на размышление, не предполагающие однозначного ответа.

В книге есть еще два приложения. Приложение А содержит инструкцию по установке Figaro, а приложение В – краткий обзор других систем вероятностного программирования.

О коде и упражнениях

Код набран моноширинным шрифтом, чтобы не путать его с обычным текстом. Многие листинги сопровождаются аннотациями, в которых объяснены важные концепции. В некоторых случаях за листингом следует несколько комментариев, оформленных как отдельные пункты.

Многие примеры кода имеются в сети, найти их можно на сайте книги по адресу www.manning.com/books/practical-probabilistic-programming. Там же есть ответы на избранные упражнения.

Об авторе

Ави Пфэффер – пионер вероятностного программирования, работающий в этой области с момента ее зарождения. Ави – ведущий проектировщик и разработчик Figaro. В компании Charles River Analytics Ави занимается применением Figaro к различным задачам, в том числе к анализу вредоносного ПО, мониторингу исправности транспортных средств, моделированию климата и оценке инженерных систем.

В свободное время Ави поет, сочиняет музыку и продюсирует музыкальные произведения. Ави живет в Кембридже, штат Массачусетс, с женой и тремя детьми.

Автор в сети

Приобретение книги «Практическое вероятностное программирование» открывает бесплатный доступ к закрытому форуму, организованному издательством Manning Publications, где вы можете оставить свои комментарии к книге, задать технические вопросы и получить помощь от автора и других пользователей. Получить доступ к форуму и подписаться на список рассылки можно на странице www.manning.com/books/practical-probabilistic-programming. Там же написано, как зайти на форум после регистрации, на какую помощь можно рассчитывать, и изложены правила поведения в форуме.

Издательство Manning обязуется предоставлять читателям площадку для общения с другими читателями и автором. Однако это не означает, что автор обязан как-то участвовать в обсуждениях; его присутствие на форуме остается чисто добровольным (и не оплачивается). Мы советуем задавать автору какие-нибудь хитроумные вопросы, чтобы его интерес к форуму не угасал!

Форум автора в сети и архивы будут доступны на сайте издательства до тех пор, пока книга не перестанет печататься.

Об иллюстрации на обложке

Рисунок на обложке книги называется «The Venetian», или «Житель Венеции». Он взят из книги Жака Грассе де Сен-Совера «Энциклопедия путешествий», опубликованной в 1796 году. Путешествие ради удовольствия в те времена было новинку, а путеводители, подобные этому, были весьма популярны – они знакомили настоящих туристов и путешественников, не покидающих своего кресла, с жителями других регионов Франции и всего мира.

Разнообразие иллюстраций в «Энциклопедии путешествий» красноречиво свидетельствует об уникальности и индивидуальности городков и провинций, которое можно было наблюдать каких-то 200 лет назад. То было время, когда по манере одеваться, отличающейся в местах, разделенных всего несколькими десятками миль, можно было сказать, откуда человек родом. Этот путеводитель возвращает к жизни ощущение изолированности и удаленности, свойственное тому периоду, да и всем остальным историческим периодам, кроме нашего гиперактивного настоящего.

С тех пор манера одеваться сильно изменилась, и различия между областями, когда-то столь разительные, сгладились. Теперь трудно отличить друг от друга даже выходцев с разных континентов, что уж говорить о странах или областях. Но можно взглянуть на это и с оптимизмом – мы обменяли культурное и визуальное разнообразие на иное устройство личной жизни – основанное на многостороннем и стремительном технологическом и интеллектуальном развитии.

Издательство Manning откликается на новации и инициативы в компьютерной отрасли обложками своих книг, на которых представлено широкое разнообразие местных укладов быта в позапрошлом веке. Мы возвращаем его в том виде, в каком оно запечатлено на рисунках из разных собраний, в том числе из этого путеводителя.



Часть I

ВВЕДЕНИЕ В ВЕРОЯТНОСТНОЕ ПРОГРАММИРОВАНИЕ И СИСТЕМУ FIGARO

Что такое вероятностное программирование? Почему оно полезно? Как им пользоваться? Ответы на эти вопросы составляют основное содержание первой части. В главе 1 излагаются основные идеи вероятностного программирования. Прежде всего, мы познакомимся с концепцией системы вероятностных рассуждений и покажем, как вероятностное программирование соединяет традиционные системы вероятностных рассуждений с технологией языков программирования.

В этой книге мы будем использовать систему вероятностного программирования Figaro. В главе 1 дается краткое введение в Figaro, а глава 2 представляет собой пособие по основным понятиям Figaro, позволяющее быстро приступить к написанию вероятностных программ. Глава 3 содержит законченное вероятностное приложение, из которого вы поймете, как реальная программа собирается из различных частей. Эта глава находится в начале книги, чтобы можно было составить общую картину, но имеет смысл возвращаться к ней и позже, после более глубокого знакомства с различными концепциями.



ГЛАВА 1.

О вероятностном программировании в двух словах

В этой главе.

- Что такое вероятностное программирование?
- Какое мне до него дело? А моему начальнику?
- Как оно работает?
- Figaro – система вероятностного программирования.
- Сравнение вероятностных приложений, написанных с применением и без применения вероятностного программирования.

Из этой главы вы узнаете, как принимать решения с помощью вероятностной модели и алгоритма вывода – двух главных составных частей системы вероятностных рассуждений. Вы также узнаете, как современные языки вероятностного программирования упрощают создание подобных систем по сравнению с такими универсальными языками, как Java или Python. Здесь же вы познакомитесь с *Figaro*, основанным на Scala языком вероятностного программирования, которым мы будем пользоваться в этой книге.

1.1. Что такое вероятностное программирование?

Вероятностное программирование – это способ создания систем, помогающих принимать решения в условиях неопределенности. Многие решения, принимаемые нами ежедневно, подразумевают учет релевантных факторов, которые мы не наблюдаем непосредственно. Исторически одним из способов принять решение в таких условиях неопределенности стали системы вероятностного рассужде-

Конец ознакомительного фрагмента.

Приобрести книгу можно

в интернет-магазине

«Электронный универс»

e-Univers.ru