



# Содержание

<b>От издательства</b> .....	14
<b>Предисловие</b> .....	15
<b>Благодарности</b> .....	16
<b>1 Введение</b> .....	17
1.1. Принятие решений .....	17
1.2. Области применения .....	18
1.2.1. Предотвращение столкновения самолетов .....	19
1.2.2. Автоматизированное вождение .....	19
1.2.3. Скрининг рака молочной железы .....	19
1.2.4. Доля инвестиций и распределение портфеля .....	20
1.2.5. Распределенное наблюдение за лесными пожарами .....	20
1.2.6. Исследование Марса .....	21
1.3. Методы создания агентов .....	21
1.3.1. Явное программирование .....	22
1.3.2. Обучение с учителем .....	22
1.3.3. Оптимизация .....	22
1.3.4. Планирование .....	22
1.3.5. Обучение с подкреплением .....	23
1.4. История автоматизации принятия решений .....	23
1.4.1. Экономика .....	24
1.4.2. Психология .....	25
1.4.3. Нейробиология .....	25
1.4.4. Информатика .....	26
1.4.5. Инженерия .....	26
1.4.6. Математика .....	27
1.4.7. Исследование операций .....	28
1.5. Воздействие на общество .....	28

1.6. Краткий обзор содержания книги .....	30
1.6.1. Вероятностное рассуждение .....	30
1.6.2. Многостадийные задачи.....	30
1.6.3. Неопределенность модели .....	31
1.6.4. Неопределенность состояния.....	31
1.6.5. Мультиагентные системы.....	32

## **Часть I. Вероятностные рассуждения.....33**

### **2 Формальное представление неопределенности .....34**

2.1. Степени доверия и вероятности .....	34
2.2. Распределения вероятностей.....	35
2.2.1. Дискретные распределения вероятностей.....	35
2.2.2. Непрерывные распределения вероятностей .....	36
2.3. Совместные распределения .....	41
2.3.1. Дискретные совместные распределения.....	41
2.3.2. Непрерывное совместное распределение.....	44
2.4. Условные распределения .....	47
2.4.1. Дискретные модели условных распределений .....	48
2.4.2. Условные модели Гаусса.....	49
2.4.3. Линейные модели Гаусса .....	49
2.4.4. Условные линейные модели Гаусса.....	50
2.4.5. Сигмовидные модели .....	50
2.4.6. Детерминированные переменные .....	51
2.5. Байесовские сети.....	51
2.6. Условная независимость.....	54
2.7. Заключение .....	57
2.8. Упражнения .....	57

### **3 Вероятностный вывод .....62**

3.1. Вывод в байесовских сетях .....	62
3.2. Вывод в наивных байесовских моделях .....	67
3.3. Исключение переменной суммированием-перемножением.....	70
3.4. Распространение доверия .....	72
3.5. Вычислительная сложность.....	72
3.6. Прямая выборка .....	73
3.7. Выборка, взвешенная по правдоподобию .....	76
3.8. Выборка Гиббса .....	79
3.9. Вывод в гауссовых моделях.....	81
3.10. Заключение.....	83
3.11. Упражнения .....	84

### **4 Параметрическое обучение .....90**

4.1. Обучение по критерию максимального правдоподобия.....	90
--	----

4.1.1. Оценки максимального правдоподобия для категориальных распределений.....	91
4.1.2. Оценки максимального правдоподобия для распределений Гаусса .....	92
4.1.3. Оценки максимального правдоподобия для байесовских сетей.....	93
4.2. Байесовское параметрическое обучение .....	96
4.2.1. Байесовское обучение для бинарных распределений.....	97
4.2.2. Байесовское обучение для категориальных распределений .....	99
4.2.3. Байесовское обучение для байесовских сетей .....	100
4.3. Непараметрическое обучение.....	101
4.4. Обучение с отсутствующими данными .....	103
4.4.1. Подстановка данных.....	104
4.4.2. Алгоритм ожидания-максимизации .....	107
4.5. Заключение.....	109
4.6. Упражнения .....	110
<b>5 Структурное обучение .....</b>	<b>116</b>
5.1. Оценка байесовской сети .....	116
5.2. Поиск ориентированного графа .....	119
5.3. Марковские классы эквивалентности .....	123
5.4. Поиск частично ориентированного графа .....	124
5.5. Заключение.....	126
5.6. Упражнения .....	126
<b>6 Простые решения.....</b>	<b>129</b>
6.1. Ограничения рациональных предпочтений.....	129
6.2. Функции полезности.....	131
6.3. Выявление полезности .....	132
6.4. Принцип максимальной ожидаемой полезности.....	134
6.5. Сети принятия решений.....	136
6.6. Полезность информации.....	139
6.7. Иррациональность .....	141
6.8. Заключение.....	143
6.9. Упражнения .....	143
<b>Часть II. Задачи последовательного принятия решений .....</b>	<b>148</b>
<b>7 Методы точного решения.....</b>	<b>149</b>
7.1. Марковские процессы принятия решений.....	149
7.2. Оценка стратегии .....	153
7.3. Нахождение стратегии через функцию полезности .....	156
7.4. Итерация по стратегиям .....	157
7.5. Итерация по критерию .....	159

7.6. Асинхронная итерация по критерию.....	162
7.7. Представление задачи в виде линейной программы .....	164
7.8. Линейные системы с квадратичным вознаграждением .....	166
7.9. Заключение .....	170
7.10. Упражнения.....	171

## **8 Приближенное вычисление функции**

<b>полезности</b> .....	179
8.1. Параметрические представления .....	179
8.2. Аппроксимация по ближайшему соседу .....	181
8.3. Ядерное сглаживание.....	183
8.4. Линейная интерполяция .....	185
8.5. Симплексная интерполяция .....	188
8.6. Линейная регрессия.....	191
8.7. Регрессия на основе нейронной сети.....	195
8.8. Заключение.....	196
8.9. Упражнения .....	196

## **9 Онлайн-планирование** ..... 201 |

9.1. Планирование с отступающим горизонтом.....	201
9.2. Стратегия развертывания.....	203
9.3. Прямой поиск .....	204
9.4. Метод ветвей и границ .....	206
9.5. Разреженная выборка .....	207
9.6. Поиск по дереву Монте-Карло .....	209
9.7. Эвристический поиск.....	218
9.8. Эвристический поиск с разметкой .....	219
9.9. Планирование с открытым контуром .....	224
9.9.1. Прогнозирующее управление с детерминированной моделью .....	226
9.9.2. Робастное прогностическое управление .....	228
9.9.3. Многовариантное прогностическое управление.....	229
9.10. Заключение.....	231
9.11. Упражнения .....	231

## **10 Поиск стратегии** ..... 236 |

10.1. Приблизительная оценка стратегии.....	236
10.2. Локальный поиск .....	238
10.3. Генетические алгоритмы .....	241
10.4. Метод перекрестной энтропии .....	242
10.5. Эволюционные стратегии .....	244
10.6. Изотропные эволюционные стратегии .....	248
10.7. Заключение .....	250
10.8. Упражнения .....	251

<b>11</b>	<b>Нахождение градиента стратегии</b> .....	255
11.1.	Конечная разность .....	255
11.2.	Градиент регрессии .....	258
11.3.	Отношение правдоподобия.....	260
11.4.	Предстоящее вознаграждение .....	263
11.5.	Вычитание базисного значения.....	266
11.6.	Заключение.....	270
11.7.	Упражнения.....	270
<b>12</b>	<b>Оптимизация методом градиентного спуска по стратегиям</b> .....	273
12.1.	Обновление стратегии методом градиентного подъема .....	273
12.2.	Ограниченное обновление градиента .....	275
12.3.	Метод натурального градиента.....	277
12.4.	Метод поиска в доверительной области.....	280
12.5.	Зажатие замещенной цели .....	285
12.6.	Заключение.....	288
12.7.	Упражнения.....	289
<b>13</b>	<b>Методы «актор–критик»</b> .....	292
13.1.	Определение актора и критика.....	292
13.2.	Обобщенная оценка преимуществ .....	294
13.3.	Градиент детерминированной стратегии .....	298
13.4.	Метод «актор–критик» с поиском по дереву Монте-Карло .....	301
13.5.	Заключение.....	303
13.6.	Упражнения .....	304
<b>14</b>	<b>Проверка стратегии</b> .....	306
14.1.	Оценка показателей качества стратегии.....	306
14.2.	Моделирование редких событий .....	312
14.3.	Анализ робастности системы .....	315
14.4.	Анализ компромиссов .....	317
14.5.	Состязательный анализ .....	319
14.6.	Заключение.....	322
14.7.	Упражнения.....	322
<b>Часть III. Неопределенность модели</b> .....		325
<b>15</b>	<b>Исследование среды и использование знаний</b> .....	326
15.1.	Задача однорукого бандита.....	326
15.2.	Оценка байесовской модели .....	328
15.3.	Стратегии ненаправленного исследования .....	330
15.4.	Стратегии направленного исследования .....	332

15.5. Оптимальные стратегии исследования .....	336
15.6. Исследование с несколькими состояниями .....	338
15.7. Заключение .....	338
15.8. Упражнения .....	339
<b>16 Методы на основе моделей .....</b>	<b>343</b>
16.1. Модели максимального правдоподобия .....	343
16.2. Схемы обновления модели .....	346
16.2.1. Полное обновление .....	346
16.2.2. Рандомизированное обновление .....	347
16.2.3. Приоритетный механизм обновления .....	347
16.3. Исследование .....	349
16.4. Байесовские методы .....	352
16.5. Адаптивные по Байесу марковские процессы принятия решений .....	355
16.6. Апостериорная выборка .....	356
16.7. Заключение .....	358
16.8. Упражнения .....	359
<b>17 Свободные методы обучения с подкреплением ....</b>	<b>362</b>
17.1. Инкрементное вычисление среднего значения распределения .....	362
17.2. Q-обучение .....	365
17.3. Алгоритм SARSA .....	367
17.4. Следы приемлемости .....	369
17.5. Формирование вознаграждения .....	371
17.6. Аппроксимация функции полезности действия .....	371
17.7. Воспроизведение опыта .....	375
17.8. Заключение .....	378
17.9. Упражнения .....	378
<b>18. Имитационное обучение .....</b>	<b>383</b>
18.1. Поведенческое копирование .....	383
18.2. Агрегация наборов данных .....	386
18.3. Итеративное обучение путем стохастического смешивания .....	389
18.4. Обратное обучение с подкреплением с максимальной разницей .....	392
18.5. Обратное обучение с подкреплением с максимальной энтропией .....	396
18.6. Генеративно-сопоставительное имитационное обучение .....	399
18.7. Заключение .....	400
18.8. Упражнения .....	400
<b>Часть IV. Неопределенность состояния .....</b>	<b>405</b>
<b>19 Убеждения .....</b>	<b>406</b>
19.1. Начальные убеждения .....	406
19.2. Фильтр дискретных состояний .....	407

19.3. Фильтр Калмана .....	412
19.4. Расширенный фильтр Калмана.....	414
19.5. Сигма-точечный фильтр Калмана .....	415
19.6. Парциальный фильтр.....	418
19.7. Внесение частиц .....	422
19.8. Заключение.....	425
19.9. Упражнения .....	426

## **20 Точное планирование с использованием убеждений-состояний.....**

20.1. MDP убеждений-состояний .....	436
20.2. Условные планы.....	437
20.3. Альфа-векторы .....	441
20.4. Сокращение .....	444
20.5. Итерация по полезности.....	447
20.6. Линейные стратегии .....	449
20.7. Заключение .....	451
20.8. Упражнения .....	451

## **21 Офлайн-планирование с использованием убеждений-состояний.....**

21.1. Аппроксимация полностью наблюдаемой полезности.....	455
21.2. Метод быстрой инфограницы .....	458
21.3. Методы быстрой оценки снизу .....	459
21.4. Точечная итерация по полезности.....	461
21.5. Рандомизированная точечная итерация по полезности .....	464
21.6. Пилообразная оценка сверху .....	465
21.7. Выбор точек в наборе убеждений.....	469
21.8. Пилообразный эвристический поиск .....	472
21.9. Триангулированные функции полезности.....	474
21.10. Заключение.....	477
21.11. Упражнения .....	478

## **22 Онлайн-планирование с использованием убеждений-состояний.....**

22.1. Предпросмотр с развертываниями.....	483
22.2. Прямой поиск .....	483
22.3. Метод ветвей и границ .....	486
22.4. Разреженная выборка .....	486
22.5. Поиск по дереву Монте-Карло .....	487
22.6. Поиск по детерминированному разреженному дереву .....	490
22.7. Эвристический поиск на основе разности границ.....	494
22.8. Заключение.....	496



---

22.9. Упражнения .....	497
------------------------	-----

## **23 Понятие контроллера..... 500**

23.1. Контроллеры.....	500
23.2. Итерация по стратегиям.....	504
23.3. Нелинейное программирование.....	509
23.4. Градиентный подъем.....	512
23.5. Заключение.....	518
23.6. Упражнения .....	519

## **Часть V. Многоагентные системы..... 521**

### **24 Логический вывод в многоагентных системах..... 522**

24.1. Простые игры .....	522
24.2. Модели откликов.....	525
24.2.1. Наилучший отклик.....	526
24.2.2. Отклик softmax.....	526
24.3. Равновесие доминирующей стратегии .....	527
24.4. Равновесие Нэша.....	528
24.5. Согласованное равновесие .....	530
24.6. Итеративный поиск лучшего отклика.....	533
24.7. Иерархическая форма модели softmax.....	534
24.8. Фиктивная игра.....	536
24.9. Градиентный подъем.....	539
24.10. Заключение.....	542
24.11. Упражнения .....	542

### **25 Последовательные задачи..... 548**

25.1. Марковские игры.....	548
25.2. Модели отклика.....	550
25.2.1. Наилучший отклик.....	551
25.2.2. Стратегия отклика softmax.....	551
25.3. Равновесие Нэша.....	552
25.4. Фиктивная марковская игра.....	553
25.5. Градиентный подъем.....	557
25.6. Q-обучение Нэша .....	559
25.7. Заключение.....	561
25.8. Упражнения .....	561

### **26 Неопределенность состояния..... 564**

26.1. Частично наблюдаемые марковские игры.....	564
26.2. Оценка стратегии.....	566
26.2.1. Оценка условных планов.....	566
26.2.2. Оценка стохастических контроллеров.....	568

---

26.3. Равновесие Нэша.....	569
26.4. Динамическое программирование.....	571
26.5. Заключение.....	574
26.6. Упражнения .....	575
<b>27 Совместные действия агентов.....</b>	<b>577</b>
27.1. Децентрализованные частично наблюдаемые марковские процессы принятия решений .....	577
27.2. Подклассы .....	578
27.3. Динамическое программирование .....	582
27.4. Итерация по наилучшим откликам .....	582
27.5. Эвристический поиск.....	584
27.6. Нелинейное программирование .....	587
27.7. Заключение .....	588
27.8. Упражнения.....	589
<b>A Основные математические понятия.....</b>	<b>592</b>
<b>B Распределения вероятностей .....</b>	<b>604</b>
<b>C Вычислительная сложность .....</b>	<b>606</b>
<b>D Представление функций в форме нейронных сетей.....</b>	<b>610</b>
<b>E Алгоритмы поиска .....</b>	<b>628</b>
<b>F Задачи принятия решений.....</b>	<b>637</b>
<b>G Язык программирования Julia.....</b>	<b>655</b>
<b>Предметный указатель.....</b>	<b>677</b>

# Предисловие

Эта книга представляет собой развернутый обзор, посвященный алгоритмам принятия решений в условиях неопределенности. Мы постарались охватить максимально широкий спектр тем, связанных с принятием решений, и познакомить читателей с формулировками основных математических задач и алгоритмами их решения. В книге приведены рисунки, примеры и упражнения, которые помогают раскрыть идеи, лежащие в основе различных подходов.

Книга предназначена для студентов старших курсов и аспирантов, а также специалистов в области искусственного интеллекта и систем принятия решений. Чтение этой книги требует определенного уровня математической подготовки и предполагает предварительное знакомство с многомерным исчислением, линейной алгеброй и теорией вероятностей. Некоторые вводные материалы представлены в приложениях. К дисциплинам, в которых книга будет особенно полезна, относятся математика, статистика, информатика, аэрокосмическая промышленность, электротехника и операционные процессы.

В основе этого учебника лежат алгоритмы, реализованные на языке программирования Julia. Мы обнаружили, что этот язык идеально подходит для описания алгоритмов в удобочитаемой форме. Приоритетом при разработке алгоритмических реализаций была простота понимания, а не эффективность кода. Для использования на практике могут понадобиться более оптимальные реализации. Разрешается бесплатно использовать любые фрагменты кода из этой книги с обязательным указанием ссылки на источник кода.

*Микель Кохендерфер*

*Тим Уилер*

*Кайл Рэй*

Стэнфорд, Калифорния,

28 февраля 2022 г.

# 1 Введение

В нашей повседневной жизни многие важные задачи связаны с принятием решений в условиях неопределенности. К ним относятся, например, предотвращение столкновений самолетов, борьба с лесными пожарами и реагирование на стихийные бедствия. При разработке автоматизированных систем принятия решений или систем поддержки принятия решений важно учитывать различные источники неопределенности, тщательно соблюдая баланс между несколькими целями. Мы обсудим эти проблемы с вычислительной точки зрения, стремясь раскрыть теорию, лежащую в основе моделей принятия решений и вычислительных подходов. В этой главе представлена проблема принятия решений в условиях неопределенности, приведены некоторые примеры применения и очерчено пространство вычислительных подходов. Затем мы покажем, как различные дисциплины способствуют нашему пониманию разумного принятия решений, и определим области потенциального взаимодействия систем принятия решений и общества. Главу завершает краткий анонс оставшейся части книги.

## 1.1. Принятие решений

*Агент* – это сущность, которая действует на основе наблюдений за окружающей его средой. Агенты могут быть физическими объектами, такими как люди или роботы, или невещественными объектами, такими как системы поддержки принятия решений, которые полностью реализованы программно. Как показано на рис. 1.1, взаимодействие между агентом и окружающей средой следует циклу «наблюдение–действие», который иногда называют *петлей*.

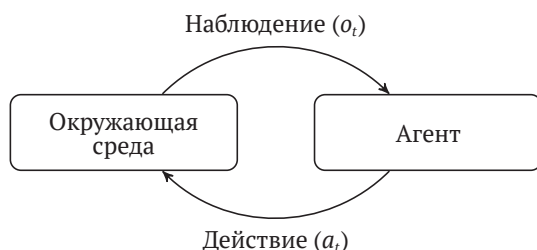


Рис. 1.1. Взаимодействие между агентом и его окружением

Агент в момент времени  $t$  получает *наблюдение* за окружающей средой, обозначенное как  $o_t$ . Наблюдения могут производиться, например, посредством биологических органов чувств, как у людей, или с помощью электронной сенсорной системы, такой как радар в системе управления воздушным движением. Наблюдения часто бывают неполными или зашумленными; например, люди могут не различить приближающийся самолет среди других отметок на экране, или радиолокационная система может пропустить обнаружение из-за электромагнитных помех. Затем агент выбирает *действие*  $a_t$  в процессе принятия решения.

Действие агента, такое как подача сигнала тревоги, может оказывать недетерминированное влияние на среду.

Мы будем говорить об агентах, которые разумно взаимодействуют для достижения своих целей на протяжении определенного времени. Учитывая прошлую последовательность наблюдений  $o_1, \dots, o_t$  и знание среды, агент должен выбрать действие, которое лучше всего способствует достижению его цели при наличии различных источников неопределенности<sup>1</sup>, включая следующие:

- *неопределенность результата*, когда последствия наших действий не определены;
- *неопределенность модели*, когда наша модель задачи не определена;
- *неопределенность состояния*, когда истинное состояние окружающей среды остается неопределенным;
- *неопределенность взаимодействия*, когда поведение других агентов, взаимодействующих в окружающей среде, не определено.

Эта книга построена вокруг упомянутых четырех источников неопределенности. Принятие решений в условиях неопределенности занимает центральное место в области *искусственного интеллекта*<sup>2</sup>, а также во многих других областях, как указано в разделе 1.4. Мы обсудим различные алгоритмы или описания вычислительных процессов для принятия решений, устойчивых к неопределенности.

## 1.2. Области применения

Обобщенную систему принятия решений, упомянутую в предыдущем разделе, можно применить к широкому кругу задач. В этом разделе мы рассмотрим несколько концептуальных примеров реального применения. Приложение F содержит описание других гипотетических задач, которые применяются в этой книге для наглядной демонстрации обсуждаемых алгоритмов.

<sup>1</sup> Здесь мы сосредоточимся на задачах дискретного времени. Задачи с непрерывным временем изучаются в области теории управления. См. D. E. Kirk, *Optimal Control Theory: An Introduction*. Prentice-Hall, 1970.

<sup>2</sup> Всестороннее введение в искусственный интеллект предоставлено в S. Russell, P. Norvig, *Artificial Intelligence: A Modern Approach*, 4th ed. Pearson, 2021.

### 1.2.1. Предотвращение столкновения самолетов

Чтобы помочь авиационным диспетчерам предотвратить столкновение самолетов в воздухе, мы должны разработать систему, которая способна предупреждать пилотов о потенциальных угрозах и указывать им, какие маневры следует совершить<sup>3</sup>. Система взаимодействует с транспондерами самолетов, чтобы определить их положение с определенной степенью точности. Решить, какие указания дать пилотам, очень непросто. Существует неопределенность в том, как быстро пилоты отреагируют и насколько усердно они будут выполнять указания. Кроме того, существует неопределенность в поведении других самолетов. Необходимо, чтобы наша система выдавала предупреждения достаточно рано, оставляя пилотам запас времени для маневра, чтобы избежать столкновений, но, с другой стороны, система не должна выдавать предупреждения слишком рано, так как это приведет ко множеству ненужных маневров, которые сами по себе служат источником дополнительной опасности и неопределенности. Поскольку эта система будет использоваться постоянно во всем мире, нам требуется решение, обеспечивающее исключительно высокий уровень безопасности.

### 1.2.2. Автоматизированное вождение

Инженеры стремятся создать автономный автомобиль, который сможет безопасно передвигаться в обычных городских условиях<sup>4</sup>. Автономное транспортное средство должно принимать безопасные решения, полагаясь на набор собственных датчиков для восприятия окружающей среды. Одним из таких датчиков является *лидар*, который использует отражение лазерного луча для определения расстояний до препятствий и построения объемной картины окружающей обстановки. Другой тип датчика – это камера, которая с помощью алгоритмов компьютерного зрения может обнаруживать пешеходов и другие транспортные средства. Оба этих датчика несовершенны и чувствительны к шуму и помехам. Например, припаркованный грузовик может загораживать пешехода, собравшегося перейти дорогу по пешеходному переходу. Наша система должна предсказывать намерения и ожидаемые траектории движения других транспортных средств, пешеходов и прочих участников дорожного движения на основе их наблюдаемого поведения, чтобы автомобиль безопасно добрался до пункта назначения.

### 1.2.3. Скрининг рака молочной железы

Во всем мире рак молочной железы является наиболее распространенным видом рака у женщин. Раннее обнаружение рака молочной железы может помочь

<sup>3</sup> Это применение обсуждается в главе, написанной М. Кохендерфером для книги *Decision Making Under Uncertainty: Theory and Application*. MIT Press, 2015.

<sup>4</sup> Аналогичное применение представлено в докладе М. Bouton, А. Nakhaei, К. Fujimura, М. J. Kochenderfer, *Safe Reinforcement Learning with Scene Decomposition for Navigating Complex Urban Environments*, in IEEE Intelligent Vehicles Symposium (IV), 2019.

спасти жизни, а маммография (рентгеновское сканирование) является наиболее эффективным доступным методом скрининга. Тем не менее маммография сопряжена с потенциальными рисками, включая ложноположительные результаты, которые могут привести к ненужному и инвазивному последующему диагностическому обследованию. Многолетние исследования ученых из разных стран позволили выработать критерии и методики скрининга населения в зависимости от возраста, позволяющие найти компромисс между пользой и рисками от обследования. Наличие системы, которая может давать рекомендации на основе личных характеристик риска и истории скрининга, может привести к существенному улучшению здоровья определенных групп населения<sup>5</sup>. Успех такой системы можно оценить в сравнении с подходом тотального скрининга, используя критерии общего ожидаемого количества лет жизни с поправкой на качество, количество маммограмм, долю ложноположительных результатов и риска невыявленного рака, требующего оперативного вмешательства.

#### **1.2.4. Доля инвестиций и распределение портфеля**

Предположим, мы хотим построить систему, которая рекомендует людям, какую часть личного дохода они могут потратить по итогам года и сколько должны инвестировать<sup>6</sup>. Инвестиционный портфель может включать в себя акции и облигации с разным уровнем риска и ожидаемой доходности. Рост личных накоплений носит стохастический характер из-за неопределенности как в зарплате, так и в инвестиционном доходе, часто увеличиваясь до тех пор, пока инвестор не достигает пенсионного возраста, а затем неуклонно снижается. Удовольствие, получаемое от потребления условной единицы богатства в течение года, обычно уменьшается с увеличением потребляемой суммы, что приводит к желанию оптимально распределить потребление и инвестирование на протяжении всей жизни человека.

#### **1.2.5. Распределенное наблюдение за лесными пожарами**

При тушении лесных пожаров одной из главных проблем является недостаточная осведомленность о текущей ситуации. Состояние пожара постоянно меняется под влиянием таких факторов, как ветер и распределение горючего материала в окружающей среде. Многие лесные пожары охватывают большие географические регионы. Один из подходов к наблюдению за лесными пожарами заключается в использовании группы дронов, оснащенных датчиками<sup>7</sup>.

<sup>5</sup> Такая концепция предложена в статье T. Ayer, O. Alagoz, and N. K. Stout, *A POMDP Approach to Personalize Mammography Screening Decisions*, *Operations Research*, vol. 60, no. 5, pp. 1019–1034, 2012.

<sup>6</sup> Схожая задача была представлена в R. C. Merton, *Optimum Consumption and Portfolio Rules in a Continuous-Time Model*, *Journal of Economic Theory*, vol. 3, no. 4, pp. 373–413, 1971.

<sup>7</sup> Это применение было детально представлено в K. D. Julian, M. J. Kochenderfer, *Distributed Wildfire Surveillance with Autonomous Aircraft Using Deep Reinforcement Learning*, *AIAA Journal of Guidance, Control, and Dynamics*, vol. 42, no. 8, pp. 1768–1778, 2019.

Дальность действия датчиков отдельных дронов ограничена, но информация от группы дронов может быть объединена для получения единого снимка ситуации и принятия решений о распределении ресурсов. В идеале хотелось бы, чтобы члены группы самостоятельно определяли, как сотрудничать друг с другом для построения наилучшей информационной картины пожара. Эффективный мониторинг требует принятия решения о том, как маневрировать, чтобы охватить области, в которых новая информация может оказаться полезной, – тратить время на сканирование участков, про которые мы точно знаем, есть ли там пожар, было бы расточительно. Выявление предпочтительных областей для исследования требует рассуждений о стохастической эволюции пожара, основанных лишь на неполных знаниях о его текущем состоянии.

### **1.2.6. Исследование Марса**

Марсоходы сделали важные открытия и значительно улучшили наше понимание Марса. Однако основным узким местом в научных исследованиях была связь между марсоходом и операторами на Земле. Для отправки информации датчиков с Марса на Землю и для отправки ответных команд с Земли на Марс может потребоваться до получаса. Кроме того, наведение антенн марсоходов и другие аспекты их работы необходимо планировать заранее, потому что окна радиообмена Земля–Марс ограничены из-за расположения орбитальных аппаратов, служащих ретрансляторами радиосигнала между планетами. Недавние исследования показали, что эффективность научно-исследовательских миссий может быть повышена в пять раз за счет достижения более высоких уровней автономности<sup>8</sup>. Люди-операторы по-прежнему будут обеспечивать руководство высокого уровня в соответствии с целями миссии, но марсоход получит возможность выбирать свои собственные краткосрочные цели, используя самую свежую информацию. Кроме того, желательно, чтобы марсоход адекватно реагировал на различные опасности и системные сбои без вмешательства человека.

## **1.3. Методы создания агентов**

Существует множество методов создания агентов, принимающих решения. В зависимости от назначения агента и предметной области некоторые могут быть более подходящими, чем другие. Они отличаются обязанностями разработчика и задачами, которые подлежат автоматизации. В этой книге основное внимание уделено планированию и обучению с подкреплением, но некоторые методы будут включать в себя элементы обучения с учителем и оптимизации.

---

<sup>8</sup> Эту концепцию представили и оценили D. Gaines, G. Doran, M. Paton, B. Rothrock, J. Russino, R. Mackey, R. Anderson, R. Francis, C. Joswig, H. Justice, K. Kolcio, G. Rabideau, S. Schaffer, J. Sawoniewicz, A. Vasavada, V. Wong, K. Yu, A.-a. Agha-mohammadi, *Self-Reliant Rovers for Increased Mission Productivity, Journal of Field Robotics*, vol. 37, no. 7, pp. 1171–1196, 2020.



### 1.3.1. Явное программирование

Самый прямой метод создания агента, принимающего решения, состоит в том, чтобы предусмотреть все сценарии, в которых может оказаться агент, и явно запрограммировать ответное поведение агента. Подход с явным программированием может хорошо работать для простых задач, но он возлагает на разработчика большую нагрузку и ответственность за полноту выработанной стратегии. Специально для упрощения программирования агентов разработаны различные языки и среды программирования.

### 1.3.2. Обучение с учителем

В некоторых случаях проще показать агенту, что делать, чем писать программу, которую будет выполнять агент. Разработчик предоставляет набор обучающих примеров, и алгоритм автоматического обучения должен обобщить эти примеры, найдя связи и закономерности в обучающих данных. Этот подход известен как *обучение с учителем* и широко применяется для задач классификации. Данный метод иногда называют *поведенческим клонированием*, когда он применяется для обучения сопоставлению наблюдений с действиями. Поведенческое клонирование работает наиболее хорошо, когда опытный разработчик действительно знает наилучший план действий для представленного набора ситуаций. Хотя существует множество различных алгоритмов обучения, в новых ситуациях они, как правило, не могут работать лучше, чем разработчики-люди, предложившие обучающие примеры.

### 1.3.3. Оптимизация

При использовании метода *оптимизации* разработчик задает пространство возможных стратегий принятия решений и показатель качества работы агента, который необходимо максимизировать. Оценка эффективности стратегии принятия решений обычно предусматривает запуск серии прогонов модели. Затем алгоритм оптимизации ищет в этом пространстве оптимальную стратегию. Если пространство относительно невелико, а мера качества не имеет множества локальных оптимумов, то можно использовать различные методы локального или глобального поиска. Хотя обычно предполагают, что для запуска прогонов требуется знание динамической модели, его не используют для направленного поиска.

### 1.3.4. Планирование

*Планирование* – это форма оптимизации, в которой используется модель динамики задачи, помогающая вести поиск. На сегодняшний день в литературе описано множество различных задач планирования. Большинство исследований сосредоточены на детерминированных задачах. Для некоторых задач может быть приемлемо аппроксимировать динамику детерминированной

моделью. Предположение о детерминированной модели позволяет нам использовать методы, которые легче масштабировать для многомерных задач. Для некоторых задач принципиально важно учитывать будущую неопределенность. Эта книга полностью посвящена именно таким задачам.

### **1.3.5. Обучение с подкреплением**

*Обучение с подкреплением* ослабляет предположение о том, что модель известна заранее. Вместо этого стратегия принятия решений изучается «на ходу» – во время взаимодействия агента с окружающей средой. Разработчик должен только предоставить меру качества решений; оптимизацией поведения агента занимается обучающий алгоритм. Одна из интересных сложностей, возникающих при обучении с подкреплением, заключается в том, что выбор действия влияет не только на немедленный успех агента в достижении своих целей, но и на способность агента узнавать об окружающей среде и определять особенности задачи, которыми можно воспользоваться.

## **1.4. История автоматизации принятия решений**

Идея автоматизировать процесс принятия решений уходит своими корнями в мечты ранних философов, ученых, математиков и писателей. Древние греки начали упоминать автоматизацию в мифах и исторических записях еще в 800 г. до н. э. Слово *αὐτόματο* впервые встречается в «Илиаде» Гомера, где содержится упоминание автоматических машин, включая механические треножки, используемые для обслуживания гостей за обедом<sup>9</sup>. В XVII веке философы предложили использовать логические правила для автоматического разрешения разногласий. Их идеи заложили основу для механических систем принятия решений.

Начиная с конца XVIII века изобретатели начали создавать автоматические машины для выполнения рутинной работы. В частности, ряд инноваций в текстильной промышленности привел к разработке автоматических ткацких станков, которые, в свою очередь, заложили основу для первых фабричных роботов<sup>10</sup>. В начале XIX века описания интеллектуальных машин для автоматизации труда стали появляться в научно-фантастических романах. Слово «робот» впервые появилось в пьесе чешского писателя Карела Чапека «R.U.R.» (сокращение от чеш. *Rossumoví univerzální roboti*, «Россумские универсальные роботы») о машинах, которые могут выполнять работу вместо людей. Пьеса вдохновила других писателей-фантастов включить роботов в свои произведения. В середине XX века известный писатель и профессор Айзек Азимов изложил свое видение робототехники в своей знаменитой серии «Робот».

<sup>9</sup> S. Vasileiadou, D. Kalligeropoulos, N. Karcianas, *Systems, Modelling and Control in Ancient Greece: Part 1: Mythical Automata, Measurement and Control*, vol. 36, no. 3, pp. 76–80, 2003.

<sup>10</sup> N. J. Nilsson, *The Quest for Artificial Intelligence*. Cambridge University Press, 2009.

Практические реализации автоматизированного принятия решений хуже всего справляются с неопределенностью. Еще в 1991 г. Джордж Данциг, получивший известность благодаря разработке симплексного алгоритма, заявил:

Оглядываясь назад, интересно отметить, что первоначальная задача, с которой началось мое исследование, до сих пор не решена, а именно задача планирования или составления расписания действий, особенно динамического планирования в условиях неопределенности. Успешное решение такой задачи могло бы (в конечном итоге за счет лучшего планирования) способствовать благополучию и стабильности в мире<sup>11</sup>.

Хотя принятие решений в условиях неопределенности по-прежнему остается областью активных исследований, за последние несколько столетий исследователи и инженеры заметно приблизились к реализации идей древних мечтателей и философов. Современные алгоритмы принятия решений основаны на сближении концепций, возникших в самых разных дисциплинах, включая экономику, психологию, неврологию, информатику, инженерию, математику и исследование операций. Далее мы кратко рассмотрим некоторые основные достижения этих дисциплин. Своеобразное «перекрестное опыление» между дисциплинами уже не раз становилось источником передовых идей в области принятия решений. Вероятно, так будет и впредь.

### 1.4.1. Экономика

Экономика нуждается в моделях, имитирующих принятие решений человеком. Один из подходов к построению таких моделей опирается на теорию полезности, впервые представленную в конце XVIII века<sup>12</sup>. *Теория полезности* предоставляет методику для моделирования и сравнения желательности различных результатов. Например, полезность можно использовать для сравнения желательности денежных количеств. В книге «Основы законодательства» (1887) Иеремия Бентам отметил убывающую полезность богатства:

1. Каждой порции богатства соответствует порция счастья.
2. Из двух людей с неравным состоянием тот, у кого больше всего богатства, имеет больше всего счастья.
3. Избыток счастья богатого не будет столь велик, как избыток его богатства<sup>13</sup>.

Объединив понятие полезности с понятием рационального принятия решений, экономисты середины XX века пришли к формулировке *принципа максимальной ожидаемой полезности*. Этот принцип является ключевой идеей при создании автономных агентов, принимающих решения. Из теории полезности впоследствии развилась *теория игр*, которая пытается описать поведение не-

<sup>11</sup> G. B. Dantzig, *Linear Programming, Operations Research*, vol. 50, no. 1, pp. 42–47, 2002.

<sup>12</sup> G. J. Stigler, *The Development of Utility Theory. I*, *Journal of Political Economy*, vol. 58, no. 4, pp. 307–327, 1950.

<sup>13</sup> J. Bentham, *Theory of Legislation*. Trübner & Company, 1887.

скольких агентов, действующих в присутствии друг друга и стремящихся максимизировать свою выгоду<sup>14</sup>.

### 1.4.2. Психология

Психологи также изучают принятие решений человеком, как правило, с точки зрения человеческого поведения. Психологи с XIX века разрабатывают теории обучения методом проб и ошибок, изучая реакцию животных на раздражители. Исследователи заметили, что животные склонны принимать решения, основываясь на удовлетворении или дискомфорте, которые они испытали в предыдущих подобных ситуациях. Русский психолог Иван Павлов объединил эту идею с концепцией *подкрепления* после наблюдения за характером слюноотделения у собак при кормлении. Психологи обнаружили, что модель поведения можно усилить или ослабить, используя постоянное подкрепление в виде определенного стимула. В середине XX века математик и ученый-компьютерщик Алан Тьюринг высказал мысль о том, что машины могут учиться таким же образом:

Превращение машины в универсальную машину было бы наиболее впечатляющим, если бы механизмы взаимодействия обходились очень ограниченными входными данными. Обучение человеческого ребенка во многом зависит от системы поощрений и наказаний, и это наводит на мысль, что возможно осуществить обучение машины только с двумя взаимоисключающими входами: одним для «удовольствия» или «вознаграждения» (R) и другим для «боли» или «наказания» (P)<sup>15</sup>.

Исследования психологов легли в основу концепции *обучения с подкреплением* – важнейшего метода, применяемого для обучения агентов принятию решений в условиях неопределенности<sup>16</sup>.

### 1.4.3. Нейробиология

Психологи изучают человеческое поведение в том виде, в каком оно наблюдается извне, а нейробиологи сосредоточены на внутренних биологических процессах, формирующих поведение. В конце XIX века ученые обнаружили, что мозг состоит из сети взаимосвязанных нейронов, которая отвечает за его способность воспринимать и осознавать окружающий мир. Пионер искусственного интеллекта Нильс Нильссон описывает значение этого открытия для систем принятия решений следующим образом:

Поскольку именно мозг животного отвечает за преобразование сенсорной информации в действие, вполне очевидно, что в работах нейрофи-

<sup>14</sup> O. Morgenstern, J. von Neumann, *Theory of Games and Economic Behavior*. Princeton University Press, 1953.

<sup>15</sup> M. Turing, *Intelligent Machinery*, National Physical Laboratory, Report, 1948.

<sup>16</sup> R. S. Sutton, A. G. Barto, *Reinforcement Learning: An Introduction*, 2nd ed. MIT Press, 2018.

зиологов и нейроанатомов, изучающих мозг и его основные компоненты, нейроны, можно найти несколько хороших идей<sup>17</sup>.

В 1940-х годах исследователи впервые предложили рассматривать нейроны как отдельные «логические единицы», способные выполнять вычислительные операции при объединении в сеть. Эта идея легла в основу нейронных сетей, которые используются в области искусственного интеллекта для выполнения множества сложных задач.

#### 1.4.4. Информатика

В середине XX века ученые-компьютерщики начали рассматривать проблему разумного принятия решений как задачу организации символических операций с помощью формальной логики. Компьютерная программа Logic Theorist, написанная в середине XX века для выполнения автоматических рассуждений, использовала этот способ формального мышления для доказательства математических теорем. Герберт Саймон, один из ее изобретателей, подчеркнул особую символическую природу программы, связав ее с человеческим разумом:

Мы создали компьютерную программу, способную мыслить нечисловыми категориями, и тем самым решили извечную проблему разума и тела, объяснив, как система, состоящая из материи, может обладать свойствами разума<sup>18</sup>.

Эти символические системы в значительной степени полагались на человеческий опыт. Альтернативный подход к интеллекту, называемый *коннекционизмом*, отчасти вдохновлен достижениями в области нейробиологии и сосредоточен на использовании искусственных нейронных сетей в качестве субстрата для зарождения искусственного интеллекта. Зная, что нейронные сети можно обучить распознаванию образов, коннекционисты пытаются научить машину разумному поведению на основе данных или опыта, а не жестко запрограммированных знаний экспертов. Идея коннекционизма лежит в основе успеха AlphaGo – программы, которая разгромила чемпионов игры в го, – а также большей части интеллектуальной составляющей автономных транспортных средств. Алгоритмы, сочетающие символический и коннекционистский подходы, остаются предметом активных исследований и сегодня.

#### 1.4.5. Инженерия

Специалисты в области инженерии стараются научить физические системы, такие как роботы, принимать разумные решения. Всемирно известный робототехник Себастьян Трун описывает компоненты этих систем следующим образом:

<sup>17</sup> N. J. Nilsson, *The Quest for Artificial Intelligence*. Cambridge University Press, 2009.

<sup>18</sup> Цитируется по J. Agar, *Science in the 20th Century and Beyond*. Polity, 2012.

Общим для систем робототехники является то, что они находятся в физическом мире, воспринимают свою среду с помощью датчиков и манипулируют своей средой с помощью движущихся объектов<sup>19</sup>.

При проектировании робототехнических систем инженеры оперируют такими понятиями, как *восприятие*, *планирование* и *действие*. Физические системы воспринимают мир, используя свои датчики для формирования представления о характерных чертах окружающей их среды. Восприятие как отдельная область исследований сосредоточено на использовании показаний датчиков для построения представления о состоянии мира. Планирование требует рассуждений о способах выполнения задач, для которых предназначена система. Процесс планирования стал возможным благодаря достижениям вычислительной техники на протяжении нескольких последних десятилетий<sup>20</sup>. Как только план сформирован, автономный агент должен выполнять его в реальном мире. Для реализации плана нужны как аппаратные средства (в виде исполнительных механизмов), так и алгоритмы для управления исполнительными механизмами и подавления возмущений. *Теория управления* сосредоточена на стабилизации механических систем посредством управления с обратной связью<sup>21</sup>. Автоматические системы управления широко используются в промышленности, от регулирования температуры в печи до навигации аэрокосмических систем.

### 1.4.6. Математика

Агент должен иметь возможность количественно оценить свою неопределенность, чтобы принимать обоснованные решения. Теория принятия решений в значительной степени зависит от теории вероятностей, рассматриваемой в контексте конкретной задачи. В частности, для этой книги важную роль играет байесовская статистика. В 1763 г. была опубликована посмертная статья Томаса Байеса, содержащая выкладки, которые позже стали известны как *правило Байеса*. Его подход к вероятностному выводу то приобретал, то терял популярность до середины XX века, когда исследователи обнаружили, что байесовские методы дают существенный выигрыш в ряде ситуаций<sup>22</sup>. Математик Бернارد Купман нашел практическое применение теории во время Второй мировой войны:

Каждая операция, связанная с поиском, сопряжена с неопределенностью; ее можно трактовать количественно только с точки зрения [...] вероятности. Сейчас это можно рассматривать как трюизм, но, похоже, развитие оперативного анализа во время Второй мировой войны позволило понять ее практическое значение<sup>23</sup>.

<sup>19</sup> S. Thrun, *Probabilistic Robotics, Communications of the ACM*, vol. 45, no. 3, pp. 52–57, 2002.

<sup>20</sup> G. E. Moore, *Cramming More Components onto Integrated Circuits, Electronics*, vol. 38, no. 8, pp. 114–117, 1965.

<sup>21</sup> D. A. Mindell, *Between Human and Machine: Feedback, Control, and Computing Before Cybernetics*. JHU Press, 2002.

<sup>22</sup> W. M. Bolstad and J. M. Curran, *Introduction to Bayesian Statistics*. Wiley, 2016.

<sup>23</sup> B. O. Koopman, *Search and Screening: General Principles with Historical Applications*. Pergamon Press, 1980.

*Методы выборки* (иногда называемые *методами Монте-Карло*), разработанные в начале XX века для крупномасштабных расчетов в рамках Манхэттенского проекта, сделали возможными некоторые методы логического вывода, которые ранее были недостижимы. Они послужили основой для байесовских сетей, популярность которых значительно возросла в конце XX века благодаря исследованиям в области искусственного интеллекта.

### 1.4.7. Исследование операций

*Исследование операций* (operations research, операционный анализ) связано с поиском оптимальных решений задач, таких как распределение ресурсов, инвестиции в активы и планирование технического обслуживания. В конце XIX века исследователи начали изучать применение математического и научного анализа к производству товаров и услуг. Исследования ускорились во время промышленной революции, когда компании начали подразделять свое управление на отделы, отвечающие за отдельные аспекты общих решений. Во время Второй мировой войны оптимизация решений применялась к распределению ресурсов в армии. Когда война закончилась, предприятия обнаружили, что те же концепции исследования операций, которые ранее использовались для принятия военных решений, могут пригодиться для оптимизации бизнес-решений. Это послужило толчком к развитию науки управления, как описал теоретик организационной деятельности Гарольд Кунц:

Твердое убеждение этой группы состоит в том, что если управление, или организация, или планирование, или принятие решений являются логическим процессом, то его можно выразить в виде математических символов и отношений. Центральным подходом этой школы является модель, поскольку именно с помощью этих приемов проблема выражается в ее основных отношениях и в терминах выбранных целей или задач<sup>24</sup>.

Стремление лучше моделировать и понимать бизнес-решения привело к развитию ряда современных концепций, таких как *линейное программирование*, *динамическое программирование* и *теория очередей*<sup>25</sup>.

## 1.5. Воздействие на общество

Алгоритмические подходы к принятию решений уже изменили общество и, вероятно, продолжают это делать в будущем. Далее мы кратко расскажем о нескольких способах, которыми алгоритмы принятия решений приносят пользу обществу, и о проблемах, которые возникают на этом пути<sup>26</sup>.

<sup>24</sup> H. Koontz, *The Management Theory Jungle*, Academy of Management Journal, vol. 4, no. 3, pp. 174–188, 1961.

<sup>25</sup> F. S. Hillier, *Introduction to Operations Research*. McGraw-Hill, 2012.

<sup>26</sup> Более подробное обсуждение представлено в Z. R. Shi, C. Wang, F. Fang, *Artificial Intelligence for Social Good: A Survey*, 2020. arXiv: 2001.01818v1.



Алгоритмические подходы способствуют сохранению благоприятной экологической обстановки. Например, в контексте энергосбережения байесовская оптимизация применяется к автоматизированным домашним системам управления энергопотреблением. Алгоритмы из области многоагентных систем используются для прогнозирования работы интеллектуальных сетей, управления биржами обмена энергией и прогнозирования выработки электроэнергии солнечными панелями на крышах зданий. Также были разработаны алгоритмы для защиты биоразнообразия. Например, нейронные сети нашли применение в автоматизации учета диких животных, подходы теории игр используются для борьбы с браконьерством в лесах, а методы оптимизации применяются для распределения ресурсов среды обитания.

За последние десятилетия алгоритмы принятия решений получили распространение в медицине. Такие алгоритмы полезны для прикрепления жителей к оптимально расположенным районным больницам и сопоставления доноров органов с нуждающимися пациентами. Первым применением байесовских сетей, которое мы рассмотрим в этой книге, была диагностика заболеваний. С тех пор байесовские сети широко используются в медицине для диагностики и прогнозирования различных заболеваний. Глубокое обучение совершило революцию в области обработки медицинских изображений, и алгоритмические идеи сыграли важную роль в понимании распространения болезней.

Алгоритмы позволили нам понять принципы роста городских территорий и облегчить их проектирование. Алгоритмы, управляемые данными, широко используются для улучшения общественной инфраструктуры. Например, стохастические процессы применяют для прогнозирования отказов в водопроводах, глубокое обучение улучшило управление дорожным движением, а марковские процессы принятия решений и методы Монте-Карло использовались для улучшения реагирования на чрезвычайные ситуации. Благодаря применению децентрализованных мультиагентных алгоритмов удалось оптимизировать маршруты передвижения, а методы планирования пути использовались для оптимизации доставки товаров. Алгоритмы принятия решений широко применяются в автономных автомобилях и для повышения безопасности самолетов.

Алгоритмы оптимизации решений способны усиливать эффект от действий своих пользователей независимо от их намерений. Если целью пользователя этих алгоритмов, например, является распространение дезинформации во время политических выборов, то процессы оптимизации будут способствовать обману избирателей. Однако аналогичные алгоритмы можно использовать для мониторинга и противодействия распространению ложной информации. Иногда реализация алгоритмов принятия решений может привести к последствиям, которые их пользователи не предполагали<sup>27</sup>.

Хотя алгоритмы принятия решений могут принести значительную пользу, существуют и проблемы, связанные с их особенностями и недостатками.

---

<sup>27</sup> Более общие соображения представлены в B. Christian, *The Alignment Problem*. Norton & Company, 2020. См. также D. Amodè, C. Olah, J. Steinhardt, P. Christiano, J. Schulman, D. Mané, *Concrete Problems in AI Safety*, 2016. arXiv: 1606.06565v2.



Алгоритмы, построенные на основе данных, часто страдают от врожденной предвзятости и слепых зон из-за некорректного способа сбора данных. Алгоритмы становятся все более значимой частью нашей жизни. Поэтому важно понимать, как можно снизить риск предвзятости и справедливо распределить преимущества использования алгоритмов. Алгоритмы также могут быть уязвимы для манипулирования злоумышленниками, и очень важно с самого начала разрабатывать алгоритмы, устойчивые к таким атакам. Также важно установить моральные и правовые рамки для предотвращения непреднамеренных последствий и распределения ответственности.

## **1.6. Краткий обзор содержания книги**

Эта книга состоит из пяти частей. В первой части рассматривается проблема рассуждений о неопределенности и целях при принятии одиночных простых решений в конкретный момент времени. Вторая часть расширяет процесс принятия решений до последовательных задач, где мы должны принять последовательность решений в ответ на информацию о результатах наших предыдущих последовательных действий. Третья часть касается *неопределенности модели* – ситуации, когда мы не имеем детерминированной модели и должны выстроить ее через взаимодействие с окружающей средой. Четвертая часть рассматривает *неопределенность состояния*, когда несовершенная информация о восприятии не позволяет нам узнать полное состояние окружающей среды. В заключительной пятой части обсуждаются способы принятия решений с участием нескольких агентов.

### **1.6.1. Вероятностное рассуждение**

Рациональное принятие решений требует рассуждений о нашей неопределенности и целях. Эта часть книги начинается с пояснений о том, как представить неопределенность в виде распределения вероятностей. Реальные задачи требуют понимания распределений по многим переменным. Мы покажем, как строить такие модели распределений, как использовать их для получения выводов и как узнать их параметры и структуру из данных. Затем мы вводим основы *теории полезности* и поясняем, как она формирует основу для рационального принятия решений в условиях неопределенности с помощью принципа максимальной ожидаемой полезности. Затем мы обсудим, как понятия теории полезности могут быть включены в вероятностные графовые модели для формирования так называемых *сетей принятия решений*.

### **1.6.2. Многостадийные задачи**

Многие важные проблемы требуют принятия ряда последовательных решений. Принцип максимальной ожидаемой полезности применим и в этом случае, но оптимальное принятие последовательных решений требует рассуж-

дений о будущих последовательностях действий и наблюдений. В этой части книги будут рассмотрены проблемы последовательного принятия решений в стохастической среде, где результаты наших действий неопределенные. Мы сосредоточимся на общей формулировке задач последовательного принятия решений, предполагая, что модель известна и что окружающая среда полностью наблюдаема. Позже мы ослабим оба этих предположения. Мы начнем со знакомства с *марковским процессом принятия решений* (MDP) – стандартной математической моделью для многостадийных задач принятия решений. Мы обсудим несколько подходов к поиску точных решений таких задач. Поскольку сложные задачи иногда не позволяют находить эффективные точные решения, мы обсудим ряд методов поиска приближенного решения, а также метод, основанный на прямом поиске в пространстве параметрических стратегий принятия решений. Наконец, мы обсудим способы проверки того, что наши стратегии принятия решений будут работать должным образом при развертывании в реальном мире.

### **1.6.3. Неопределенность модели**

В нашем обсуждении задач последовательного принятия решений до этого момента мы предполагали, что модели перехода и вознаграждения известны. Однако во многих задачах динамика и вознаграждение точно неизвестны, и агент должен научиться действовать на основе опыта. Наблюдая за результатами своих действий в виде переходов состояний и вознаграждений, агент должен научиться выбирать действия, которые максимизируют его долгосрочное накопление вознаграждений. Решение задач, в которых присутствует неопределенность модели, является предметом обучения с подкреплением и рассматривается в этой части книги. Мы обсудим несколько проблем, связанных с устранением неопределенности модели. Во-первых, агент должен найти тонкий баланс между поиском новых факторов окружающей среды и использованием знаний, полученных в результате опыта. Во-вторых, вознаграждение может быть получено спустя долгое время после того, как важные решения были приняты, так что более поздние вознаграждения должны быть правильно соотнесены с ранними решениями. В-третьих, агент должен уметь строить обобщения на основе ограниченного опыта. Мы рассмотрим теорию и некоторые из ключевых алгоритмов, соответствующих этим требованиям.

### **1.6.4. Неопределенность состояния**

В этой части книги мы расширим понятие неопределенности, включив в него состояние. Вместо точного наблюдения за состоянием мы получаем наблюдения, имеющие только вероятностную связь с состоянием. Такие задачи можно смоделировать как *частично наблюдаемый марковский процесс принятия решений* (partially observable Markov decision process, POMDP). Общий подход к реализации POMDP включает в себя вывод распределения по убеждениям, исходя из состояния на текущем временном шаге, а затем применение стратегии,

которая сопоставляет убеждения с действиями. Эта часть начинается с обсуждения того, как обновить наше распределение убеждений с учетом прошлой последовательности наблюдений и действий. Затем обсуждаются различные точные и приближенные методы решения POMDP.

### **1.6.5. Мультиагентные системы**

До этого момента в нашей среде за принятие решений отвечал только один агент. В пятой части книги мы расширяем понятия и методы из предыдущих четырех частей на систему из нескольких агентов, обсуждая проблемы, возникающие из-за неопределенности взаимодействия между агентами. Мы начнем с рассмотрения простых игр, в которых группа агентов одновременно выбирает действие. Результатом является индивидуальное вознаграждение для каждого агента на основе комбинированного совместного действия. *Марковская игра* (Markov game, MG) представляет собой обобщение как простых игр для нескольких состояний, так и MDP для нескольких агентов. Следовательно, агенты выбирают действия, которые могут стохастически изменить состояние общей среды. Из-за неопределенности стратегии других агентов алгоритмам MG приходится использовать обучение с подкреплением. *Частично наблюдаемая марковская игра* (partially observable Markov game, POMG) вводит неопределенность состояния, дополнительно обобщая MG и POMDP, поскольку теперь агенты получают только зашумленные локальные наблюдения. *Децентрализованный частично наблюдаемый марковский процесс принятия решений* (Dec-POMDP) фокусирует POMG на совместной многоагентной команде, где агенты получают общее вознаграждение. В этой части книги представлены четыре вышеупомянутые категории задач и рассмотрены точные и приближенные алгоритмы их решения.

# Часть I

## ВЕРОЯТНОСТНЫЕ РАССУЖДЕНИЯ

Рациональное принятие решений требует ясного понимания неопределенности, с которой мы сталкиваемся, и целей, к которым мы стремимся. Неопределенность возникает из-за практических и теоретических ограничений нашей способности предсказывать будущие события. Например, точное предсказание того, как человек-оператор отреагирует на совет системы поддержки принятия решений, потребует, среди прочего, наличия подробной модели работы человеческого мозга. Даже траектории движения спутников бывает трудно предсказать. Хотя ньютоновская физика позволяет предсказывать их с высокой точностью, спонтанные отказы двигателей ориентации могут привести к большим отклонениям от номинального пути, и даже небольшие неточности могут со временем накапливаться. Для достижения своих целей надежная система принятия решений должна учитывать различные источники неопределенности в текущем состоянии мира и будущих событиях. Эта часть книги начинается с рассказа о том, как представляют неопределенность с помощью распределений вероятностей. Для решения реальных задач необходимо знать распределения вероятностей по многим переменным. Мы обсудим, как создавать модели распределения вероятностей, использовать их для выводов и изучать их параметры и структуру на основе данных. Далее мы вводим основы теории полезности и показываем, как она формирует основу для рационального принятия решений в условиях неопределенности. Теорию полезности можно применять в упомянутых ранее вероятностных графовых моделях для формирования так называемых сетей принятия решений. Мы сосредоточимся на одношаговых решениях, оставив обсуждение проблем последовательных решений для следующей части книги.

## 2 **Формальное представление неопределенности**

Вычислительный подход к неопределенности требует формального представления. В этой главе пойдет речь о том, как представлять неопределенность<sup>1</sup>. Мы начнем с понятия *степени доверия* (degree of belief) и покажем, как набор аксиом приводит нас к использованию распределения вероятностей для количественной оценки неопределенности<sup>2</sup>. Мы рассмотрим несколько полезных форм распределений как по дискретным, так и по непрерывным переменным. Поскольку многие важные задачи связаны с распределением вероятностей по большому количеству переменных, мы обсудим способ эффективного представления совместных распределений, основанный на факторе независимости между переменными.

### 2.1. **Степени доверия и вероятности**

В задачах, связанных с неопределенностью, важно уметь сравнивать правдоподобие различных утверждений. Допустим, нам нужно выразить свое представление о том, что утверждение  $A$  более правдоподобно, чем утверждение  $B$ . Если  $A$  представляет собой «отказ исполнительного механизма», а  $B$  представляет «отказ датчика», мы можем написать  $A > B$ . Используя это основное отношение  $>$ , мы можем определить несколько других отношений:

$$A < B \text{ тогда и только тогда, когда } B > A, \quad (2.1)$$

$$A \sim B \text{ тогда и только тогда, когда ни } A > B, \text{ ни } B > A, \quad (2.2)$$

$$A \geq B \text{ тогда и только тогда, когда } A > B \text{ или } A \sim B, \quad (2.3)$$

$$A \leq B \text{ тогда и только тогда, когда } B > A \text{ или } A \sim B. \quad (2.4)$$

Следует сделать некоторые предположения об отношениях, установленных операторами  $>$ ,  $\sim$  и  $<$ . Предположение об *универсальной сравнимости* требу-

---

<sup>1</sup> Детальное описание различных подходов к представлению неопределенности дано в монографии F. Cuzzolin, *The Geometry of Uncertainty*. Springer, 2021.

<sup>2</sup> Для более полного знакомства с темой см. E. T. Jaynes, *Probability Theory: The Logic of Science*. Cambridge University Press, 2003.

ет выполнения ровно одного из следующих условий:  $A > B$ ,  $A \sim B$  или  $A < B$ . Предположение о *транзитивности* требует, чтобы если  $A \geq B$  и  $B \geq C$ , то  $A \geq C$ . Предположения об универсальной сравнимости и транзитивности дают нам возможность представить правдоподобие с помощью функции  $P$ , принимающей вещественные значения и обладающей следующими двумя свойствами<sup>3</sup>:

$$P(A) > P(B) \text{ тогда и только тогда, когда } A > B; \quad (2.5)$$

$$P(A) = P(B) \text{ тогда и только тогда, когда } A \sim B. \quad (2.6)$$

Если мы сделаем ряд дополнительных предположений<sup>4</sup> о форме  $P$ , то сможем показать, что  $P$  должна удовлетворять основным аксиомам вероятности (приложение А.2). Если мы абсолютно уверены в  $A$ , то  $P(A) = 1$ . Если мы считаем, что  $A$  невозможно, то  $P(A) = 0$ . Неуверенность в истинности  $A$  представлена значениями между двумя этими экстремумами. Следовательно, *вероятностная мера* (probability mass) должна лежать между 0 и 1, при этом  $0 \leq P(A) \leq 1$ .

## 2.2. Распределения вероятностей

*Распределение вероятностей* присваивает вероятности различным исходам<sup>5</sup>. Существуют разные способы представления распределений вероятностей в зависимости от того, о каких исходах идет речь – дискретных или непрерывных.

### 2.2.1. Дискретные распределения вероятностей

*Дискретное распределение вероятностей* – это распределение по дискретному набору значений. Мы можем представить такое распределение как *функцию вероятности*, которая присваивает вероятность каждому возможному присвоению своей входной переменной значения. Например, предположим, что у нас есть переменная  $X$ , которая может принимать одно из  $n$  значений:  $1, \dots, n$ , или, используя запись с двоеточием,  $1:n^6$ . Распределение, связанное с  $X$ , определяет  $n$  вероятностей различных присвоений значений этой переменной, в частно-

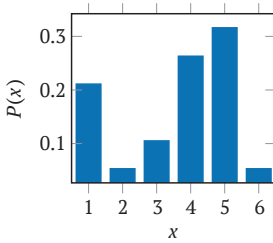
<sup>3</sup> См. E. T. Jaynes, *Probability Theory: The Logic of Science*. Cambridge University Press, 2003.

<sup>4</sup> Аксиоматизация субъективной вероятности дана в работе P. C. Fishburn, *The Axioms of Subjective Probability*, *Statistical Science*, vol. 1, no. 3, pp. 335–345, 1986. Более поздняя аксиоматизация содержится в M. J. Dupré, F. J. Tipler, *New Axioms for Rigorous Bayesian Probability*, *Bayesian Analysis*, vol. 4, нет. 3, стр. 599–606, 2009.

<sup>5</sup> Хорошее введение в теорию вероятностей дано в D. P. Bertsekas, J. N. Tsitsiklis, *Introduction to Probability*. Athena Scientific, 2002.

<sup>6</sup> Мы будем часто использовать это двоеточие для компактности. В других книгах иногда используется обозначение  $[1..n]$  для целых интервалов от 1 до  $n$ . Мы также будем использовать это обозначение с двоеточием для индексов векторов и матриц. Например,  $x_{1:n}$  представляет  $x_1, \dots, x_n$ . Обозначение двоеточия иногда используется в языках программирования, таких как Julia и MATLAB.

сти  $P(X = 1), \dots, P(X = n)$ . На рис. 2.1 показан пример дискретного распределения вероятностей.



**Рис. 2.1.** Функция вероятности для распределения на интервале 1:6

Существуют ограничения на вероятностные меры, связанные с дискретными распределениями. Их значения должны в сумме равняться 1:

$$\sum_{i=1}^n P(X = i) = 1. \quad (2.7)$$

Для удобства записи мы будем использовать строчные буквы и надстрочные индексы в качестве сокращений при обсуждении присвоения значений переменным. Например,  $P(x^3)$  – это сокращенная форма записи  $P(X = 3)$ . Если  $X$  – двоичная переменная, она может принимать значение *true* (истина) или *false* (ложь)<sup>7</sup>. Мы будем использовать 0 для представления *false* и 1 для представления *true*. Например, мы используем запись  $P(x^0)$  для представления вероятности того, что  $X$  ложно.

Параметры распределения определяют вероятности, связанные с различными значениями дискретной переменной. Например, если мы используем  $X$  для представления исхода броска шестигранной кости, то мы получим  $P(x^1) = \theta_1, \dots, P(x^6) = \theta_6$ , где  $\theta_{1:6}$  обозначает шесть параметров распределения. Однако нам достаточно пяти независимых параметров, чтобы однозначно указать распределение по исходам, потому что мы знаем, что сумма распределения должна равняться 1.

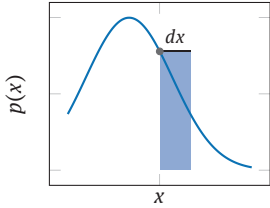
### 2.2.2. Непрерывные распределения вероятностей

Непрерывное распределение вероятностей – это распределение по непрерывному набору значений. Представление распределения по непрерывной переменной выглядит несколько сложнее, чем для дискретной переменной. Например, во многих непрерывных распределениях вероятность того, что переменная примет конкретное значение, бесконечно мала. Одним из способов представления непрерывного распределения вероятностей является использование функции плотности вероятности (рис. 2.2), обозначаемой строчными буквами. Если  $p(x)$  – функция плотности вероятности по  $X$ , то  $p(x)dx$  – это вероятность

<sup>7</sup> Julia, как и многие другие языки программирования, аналогичным образом обрабатывает логические значения как 0 и 1 в числовых операциях.

того, что  $X$  попадает в интервал  $(x, x + dx)$  при  $dx \rightarrow 0$ . Подобно тому, как вероятностные меры, связанные с дискретным распределением, должны в сумме давать 1, интеграл функции плотности вероятности  $p(x)$  должен быть равен 1:

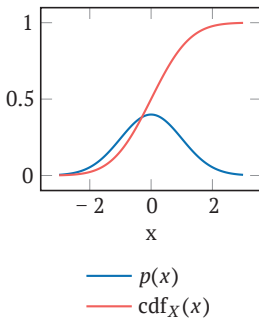
$$\int_{-\infty}^{\infty} p(x)dx = 1. \tag{2.8}$$



**Рис. 2.2.** Функции плотности вероятности используются для представления непрерывных распределений вероятностей. Если  $p(x)$  – плотность вероятности, то  $p(x)dx$ , обозначенная площадью синего прямоугольника, – это вероятность того, что выборка из случайной величины попадает в интервал  $(x, x + dx)$  при  $dx \rightarrow 0$

Другой способ представить непрерывное распределение – использовать *кумулятивную функцию распределения* (рис. 2.3), которая определяет вероятностную меру, связанную со значениями ниже некоторого порога. Если у нас есть кумулятивная функция распределения  $P$ , связанная с переменной  $X$ , то  $P(x)$  представляет вероятностную меру, связанную с  $X$ , принимающей значение, меньшее или равное  $x$ . Кумулятивная функция распределения может быть определена через функцию плотности вероятности  $p$  следующим образом:

$$\text{cdf}_X(x) = P(X \leq x) = \int_{-\infty}^x p(x')dx'. \tag{2.9}$$



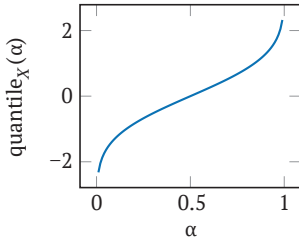
**Рис. 2.3.** Функция плотности вероятности и кумулятивная функция распределения для стандартного распределения Гаусса

С кумулятивной функцией распределения связана *квантильная функция*, также называемая *обратной кумулятивной функцией распределения* (рис. 2.4). Значение  $\text{quantile}_X(\alpha)$  – это значение  $x$ , такое что  $P(X \leq x) = \alpha$ . Другими словами, квантильная функция возвращает минимальное значение  $x$ , кумулятивное значение распределения которого больше или равно  $\alpha$ . Разумеется,  $0 < \alpha < 1$ .

Существует множество различных параметризованных семейств распределений. Мы рассматриваем некоторые из них в приложении В. Простое семейство распределений – это равномерное распределение  $\mathcal{U}(a, b)$ , при котором плотность вероятности равномерно распределена между  $a$  и  $b$  и равна нулю в других местах. Следовательно, функция плотности вероятности равна  $p(x) =$



$1/(b - a)$  для  $x$ , лежащего в интервале  $[a, b]$ . Для представления плотности в точке  $x$  мы можем использовать запись  $\mathcal{U}(x|a, b)$ <sup>8</sup>. *Носителем* (несущим множеством) распределения является набор значений, которым присвоена ненулевая плотность. В случае  $\mathcal{U}(a, b)$  носителем является интервал  $[a, b]$  (пример 2.1).



**Рис. 2.4.** Квантильная функция для стандартного распределения Гаусса

### Пример 2.1. Пример равномерного распределения с нижней границей 0 и верхней границей 10

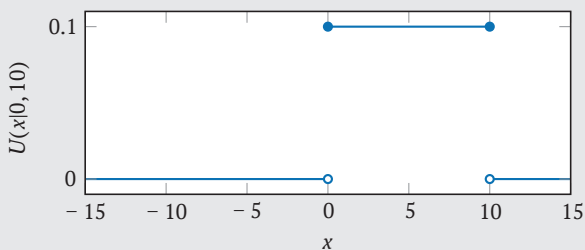
Равномерное распределение  $\mathcal{U}(0, 10)$  присваивает равную вероятность всем значениям в диапазоне  $[0, 10]$  со следующей функцией плотности вероятности:

$$\mathcal{U}(x|0, 10) = \begin{cases} 1/10, & \text{если } 0 \leq x \leq 10 \\ 0 & \text{для остальных значений} \end{cases} \quad (2.10)$$

Вероятность того, что случайная выборка из этого распределения равна константе  $\pi$ , практически нулевая. Однако мы можем определить ненулевые вероятности для выборок, находящихся в пределах некоторого интервала, допустим  $[3, 5]$ . Вероятность того, что выборка находится между 3 и 5, с учетом построенного здесь распределения, равна:

$$\int_3^5 \mathcal{U}(x|0, 10) dx = \frac{5-3}{10} = \frac{1}{5}. \quad (2.11)$$

Носителем этого распределения является интервал  $[0, 10]$ .



<sup>8</sup> В некоторых публикациях для разделения параметров распределения используется точка с запятой. Например, можно также написать  $\mathcal{U}(x; a, b)$ .

Другим распространенным распределением непрерывных переменных является распределение Гаусса (также называемое *нормальным распределением*). Распределение Гаусса характеризуется средним значением  $\mu$  и дисперсией  $\sigma^2$ :

$$p(x) = \mathcal{N}(x|\mu, \sigma^2). \tag{2.12}$$

Здесь  $\sigma$  – *стандартное отклонение*, представляющее собой квадратный корень из дисперсии. Дисперсию также часто обозначают буквой  $v$ . Мы используем запись  $\mathcal{N}(\mu, \sigma^2)$  как обозначение гауссова распределения с параметрами  $\mu$  и  $\sigma^2$  и  $\mathcal{N}(x|\mu, \sigma^2)$ , представляющего плотность вероятности в точке  $x$ :

$$\mathcal{N}(x|\mu, \sigma^2) = \frac{1}{\sigma} \varphi\left(\frac{x-\mu}{\sigma}\right), \tag{2.13}$$

где  $\varphi$  – *функция плотности стандартного нормального распределения вероятностей*:

$$\varphi(x) = \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{x^2}{2}\right). \tag{2.14}$$

В приложении В показаны графики функций плотности нормального распределения вероятностей с различными параметрами.

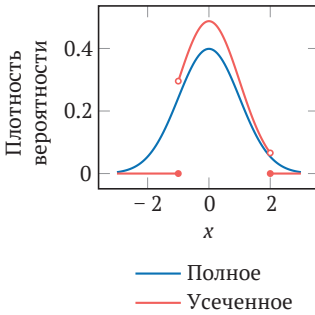
Хотя распределение Гаусса широко распространено и удобно в использовании, поскольку оно определяется только двумя параметрами и упрощает вычисления и вывод, оно имеет некоторые ограничения. В частности, оно присваивает ненулевую вероятность большим положительным и отрицательным значениям, что может не соответствовать реальной величине, которую мы пытаемся смоделировать. Например, очевидно, что не следует присваивать ненулевые вероятности самолетам, летящим под землей или за пределами земной атмосферы. Мы можем использовать *усеченное распределение Гаусса* (рис. 2.5), чтобы ограничить опорный интервал распределения, т. е. диапазон значений, которым присвоены ненулевые вероятности. Функция плотности вероятности определяется выражением

$$\mathcal{N}(x|\mu, \sigma^2, a, b) = \frac{\frac{1}{\sigma} \varphi\left(\frac{x-\mu}{\sigma}\right)}{\Phi\left(\frac{b-\mu}{\sigma}\right) - \Phi\left(\frac{a-\mu}{\sigma}\right)}, \tag{2.15}$$

когда  $x$  находится в интервале  $(a, b)$ .

Функция  $\Phi$  представляет собой *стандартную нормальную кумулятивную функцию распределения*, определяемую выражением

$$\Phi(x) = \int_{-\infty}^{\infty} \varphi(x') dx'. \tag{2.16}$$



**Рис. 2.5.** Функции плотности вероятности для единичного распределения Гаусса и того же распределения, усеченного между  $-1$  и  $2$

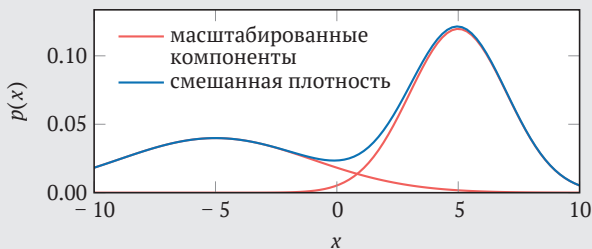
Распределение Гаусса является *унимодальным*, т. е. в распределении есть точка, относительно которой плотность увеличивается с одной стороны и уменьшается с другой. Существуют также различные способы представления непрерывных *мультимодальных* распределений. Один из способов – использовать *смешанную модель* (mixture model), которая представляет собой смесь нескольких распределений. Мы смешиваем набор одномодальных распределений, чтобы получить мультимодальное распределение. *Смешанная модель Гаусса* (Gaussian mixture model) – это модель, которая представляет собой просто средневзвешенное значение различных распределений Гаусса. Параметры такой модели состоят из параметров компонентов гауссова распределения  $\mu_{1:n}$ ,  $\sigma_{1:n}^2$ , а также их веса  $\rho_{1:n}$ . Плотность определяется выражением

$$p(x|\mu_{1:n}, \sigma_{1:n}^2, \rho_{1:n}) = \sum_{i=1}^n \rho_i \mathcal{N}(x|\mu_i, \sigma_i^2), \quad (2.17)$$

где веса должны в сумме равняться 1. В примере 2.2 показана смешанная модель Гаусса с двумя компонентами.

### Пример 2.2. Пример смешанной модели Гаусса

Создадим смешанную модель Гаусса с компонентами  $\mu_1 = 5$ ,  $\sigma_1 = 2$  и  $\mu_2 = -5$ ,  $\sigma_2 = 4$ , с весами  $\rho_1 = 0.6$  и  $\rho_2 = 0.4$  соответственно. На рисунке ниже изображены графики плотности вероятностей двух компонентов модели, масштабированные по их весу:



Другой подход к представлению мультимодальных непрерывных распределений заключается в дискретизации. Например, мы можем представить распределение по непрерывной переменной как *кусочно-равномерную аппроксимацию*. Плотность определяется в границах промежутка, опирающегося на отрезок («кусоч»), и с каждым промежутком связана вероятностная мера. Такое кусочно-равномерное распределение является разновидностью смешанной модели, в которой компоненты представляют собой равномерные распределения.

## 2.3. Совместные распределения

*Совместное распределение* – это распределение вероятностей по нескольким переменным. Распределение по одной переменной называется *одномерным*, а распределение по нескольким переменным называется *многомерным*. Если у нас есть совместное распределение по двум дискретным переменным  $X$  и  $Y$ , то  $P(x, y)$  обозначает вероятность того, что одновременно и  $X = x$ , и  $Y = y$ .

Из совместного распределения мы можем вычислить *маргинальное (частное) распределение* переменной или набора переменных, суммируя все другие переменные в соответствии с так называемым *законом полной вероятности*<sup>9</sup>:

$$P(x) = \sum_y P(x, y). \quad (2.18)$$

Запомните этот прием – мы будем использовать его на протяжении всей книги.

Принятие решений в реальном мире часто требует рассуждений о совместных распределениях, включающих множество переменных. Иногда существуют сложные отношения между переменными, которые важно уметь представить формально. Существуют разные способы представления совместных распределений в зависимости от того, включают ли переменные дискретные или непрерывные значения.

### 2.3.1. Дискретные совместные распределения

Если переменные дискретны, совместное распределение может быть представлено таблицей наподобие табл. 2.1. В ней перечислены все возможные сочетания значений трех двоичных переменных. Каждая переменная может принимать значение только 0 или 1, что дает  $2^3 = 8$  возможных назначений. Как и в случае с другими дискретными распределениями, вероятности в таблице должны в сумме равняться 1. Из этого следует, что хотя в таблице восемь за-

<sup>9</sup> Если наше распределение непрерывно, то при маргинализации мы интегрируем другие переменные. Например:  $p(x) = \int p(x, y) dy$ .

писей, только семь из них независимы. Если  $\theta_i$  представляет вероятность в  $i$ -й строке таблицы, то нам достаточно параметров  $\theta_1, \dots, \theta_7$ , потому что мы знаем, что  $\theta_8 = 1 - (\theta_1 + \dots + \theta_7)$ .

X	Y	Z	$P(X, Y, Z)$
0	0	0	0.08
0	0	1	0.31
0	1	0	0.09
0	1	1	0.37
1	0	0	0.01
1	0	1	0.05
1	1	0	0.02
1	1	1	0.07

**Таблица 2.1.** Пример дискретного совместного распределения с участием бинарных переменных X, Y и Z

Если у нас есть  $n$  бинарных переменных, то нам нужно как минимум  $2^n - 1$  независимых параметров, чтобы задать совместное распределение. Этот экспоненциальный рост числа параметров затрудняет хранение распределения в памяти компьютера. В некоторых случаях мы можем предположить, что наши переменные независимы, т. е. реализация одной переменной не влияет на распределение вероятностей другой. Если X и Y независимы, что иногда записывают как  $X \perp Y$ , то мы знаем, что  $P(x, y) = P(x)P(y)$  для всех  $x$  и  $y$ . Предположим, что у нас есть двоичные переменные  $X_1, \dots, X_n$ , которые не зависят друг от друга, в результате чего  $P(x_{1..n}) = \prod_i P(x_i)$ . Это разложение на множители позволяет нам представить совместное распределение, используя всего  $n$  независимых параметров вместо  $2^n - 1$ , необходимых, когда мы не можем предположить независимость (табл. 2.2). Предположение о независимости переменных может привести к огромной экономии с точки зрения сложности представления, но часто эта предпосылка оказывается ошибочной.

X	$P(X)$	Y	$P(Y)$	Z	$P(Z)$
0	0.85	0	0.45	0	0.20
1	0.15	1	0.55	1	0.80

**Таблица 2.2.** Если мы знаем, что переменные в табл. 2.1 независимы, мы можем представить  $P(x, y, z)$ , используя произведение  $P(x)P(y)P(z)$ . Такое представление требует только одного параметра для каждого из трех одномерных распределений

Мы можем представить совместное распределение в виде факторов. *Фактор*  $\phi$  – это функция, которая ставит в соответствие каждому возможному набору значений некоего множества переменных действительное число (значение фактора). Значения фактора, представляющие распределение вероятностей, должны быть неотрицательными. Фактор с неотрицательными значениями можно нормализовать так, чтобы он представлял собой распределение вероятностей. В алгоритме 2.1 реализованы дискретные факторы, а пример 2.3 демонстрирует, как они работают.

Другой подход к сокращению памяти, необходимой для представления совместных распределений с повторяющимися значениями, заключается в использовании *дерева решений*. Дерево решений, включающее три дискретные

переменные, показано в примере 2.4. Хотя в этом примере экономия по количеству параметров выглядит незначительной, она может стать весьма существенной, когда имеется много переменных и много повторяющихся значений.

---

**Алгоритм 2.1.** Типы и функции, относящиеся к работе с факторами в случае набора дискретных переменных. У переменной есть имя (представленное в виде символа) и целочисленное значение. *Назначение* (assignment) – это сопоставление имен переменных с целыми числами от 1 до  $m$ . Фактор определяется через *факторную таблицу*, которая присваивает значения различным назначениям, включающим множество переменных, и представляет собой сопоставление назначений с действительными значениями. Это сопоставление называется *словарем*. Любые назначения, не содержащиеся в словаре, устанавливаются равными 0. В этот блок кода также включены некоторые служебные функции для возврата имен переменных, связанных с фактором, выбора подмножества назначения, перечисления возможных назначений и нормализации факторов. Как сказано в приложении G.3.3, `product` выполняет декартово произведение набора коллекций. Он импортируется из `Base.Iterators`

```

struct Variable
    name::Symbol
    r::Int # количество возможных значений
end

const Assignment = Dict{Symbol,Int}
const FactorTable = Dict{Assignment,Float64}

struct Factor
    vars::Vector{Variable}
    table::FactorTable
end

variablenames(φ::Factor) = [var.name for var in φ.vars]

select(a::Assignment, varnames::Vector{Symbol}) =
    Assignment(n=>a[n] for n in varnames)

function assignments(vars::AbstractVector{Variable})
    names = [var.name for var in vars]
    return vec([Assignment(n=>v for (n,v) in zip(names, values))
               for values in product((1:v.r for v in vars)...)])
end

function normalize!(φ::Factor)
    z = sum(p for (a,p) in φ.table)
    for (a,p) in φ.table
        φ.table[a] = p/z
    end
    return φ
end
end

```

---