СОДЕРЖАНИЕ

Предисловие 15
Благодарности 16

Глава 1. Введение 17
1.1. Что такое обучение? 17
1.2. Когда необходимо машинное обучение? 19
1.3. Типы обучения 20
1.4. Связи с другими дисциплинами 22
1.5. Как читать эту книгу 23
1.5.1. Варианты построения курса на основе книги 24
1.6. Обозначения 25

ЧАСТЬ I. ОСНОВАНИЯ 28

Глава 2. Малый вперед 29
2.1. Формальная модель – схема статистического обучения 29
2.2. Минимизация эмпирического риска 31
2.2.1. Не все коту масленица – переобучение 31
2.3. Минимизация эмпирического риска с индуктивным смещением 32
2.3.1. Конечные классы гипотез 33
2.4. Упражнения 37

Глава 3. Формальная модель обучения 39
3.1. Вероятно почти корректное обучение 39
3.2. Более общая модель обучения 40
3.2.1. Отказ от предположения о реализуемости – агностическое PAC-обучение 41
3.2.2. Круг моделируемых проблем обучения 43
3.3. Резюме 45
3.4. Библиографические сведения 46
3.5. Упражнения 46

Глава 4. Обучаемость и равномерная сходимость 50
4.1. Равномерная сходимость – достаточное условие обучаемости 50
4.2. Конечные классы допускают агностическое PAC-обучение 51
4.3. Резюме 54
4.4. Библиографические сведения 54
4.5. Упражнения 54

Глава 5. Компромисс между смещением и сложностью 56
5.1. Теорема об отсутствии бесплатных завтраков 57
5.1.1. Теорема о бесплатных завтраках и априорное знание 59
5.2. Разложение ошибки 60
5.3. Резюме 61
5.4. Библиографические сведения 62
5.5. Упражнения 62

Глава 6. VC-размерность 63
6.1. Бесконечные классы могут быть обучаемыми 63
6.2. VC-размерность 64
6.3. Примеры 66
6.3.1. Ступенчатые функции 66
6.3.2. Интервалы 67
6.3.3. Осепараллельные прямоугольники 67
6.3.4. Конечные классы 68
6.3.5. VC-размерность и количество параметров 68
6.4. Фундаментальная теорема PAC-обучения 68
6.5. Доказательство теоремы 6.7 69
6.5.1. Лемма Зауэра и функция роста 70
6.5.2. Равномерная сходимость для классов небольшого эффективного размера 71
6.6. Резюме 74
6.7. Библиографические сведения 74
6.8. Упражнения 75

Глава 7. Неравномерная обучаемость 79
7.1. Неравномерная обучаемость 79
7.1.1. Характеристика неравномерной обучаемости 80
7.2. Структурная минимизация риска 81
7.3. Минимальная длина описания и бритва Оккама 85
7.3.1. Бритва Оккама 87
7.4. Другие концепции обучаемости – согласованность 88
7.5. Обсуждение различных понятий обучаемости 89
7.5.1. Еще раз о теореме об отсутствии бесплатных завтраков 92
7.6. Резюме 92
7.7. Библиографические сведения 93
7.8. Упражнения 93

Глава 8. Время обучения 96
8.1. Вычислительная сложность обучения 97
8.1.1. Формальное определение* 98
8.2. Реализация правила ERM 99
8.2.1. Конечные классы 100
8.2.2. Осепараллельные прямоугольники 100
8.2.3. Булевы конъюнкции 102
8.2.4. Обучение трехчленных ДНФ 103

8.3. Эффективно обучаемый, но не собственный алгоритм ERM 103
8.4. Трудность обучения* 104
8.5. Резюме 106
8.6. Библиографические сведения 106
8.7. Упражнения 106

ЧАСТЬ II. ОТ ТЕОРИИ К АЛГОРИТМАМ 110

Глава 9. Линейные предикторы 111
9.1. Полупространства 112
9.1.1. Линейное программирование для класса полупространств 113
9.1.2. Перцептрон для полупространств 114
9.1.3. VC-размерность класса полупространств 116
9.2. Линейная регрессия 117
9.2.1. Метод наименьших квадратов 118
9.2.2. Линейная регрессия для задач полиномиальной регрессии 119
9.3. Логистическая регрессия 120
9.4. Резюме 121
9.5. Библиографические сведения 122
9.6. Упражнения 122

Глава 10. Усиление 124
10.1. Слабая обучаемость 125
10.1.1. Эффективная реализация ERM для класса решающих пней 127
10.2. Алгоритм AdaBoost 128
10.3. Линейные комбинации базовых гипотез 131
10.3.1. VC-размерность $L(B, T)$ 133
10.4. Применение AdaBoost для распознавания лиц 134
10.5. Резюме 135
10.6. Библиографические сведения 136
10.7. Упражнения 136

Глава 11. Выбор и контроль модели 138
11.1. Выбор модели с помощью SRM 139
11.2. Контроль 140
11.2.1. Зарезервированный набор 140
11.2.2. Контроль при выборе модели 141
11.2.3. Кривая выбора модели 142
11.2.4. k-групповая перекрестная проверка 143
11.2.5. Обучение–контроль–тестирование 144
11.3. Что делать, если обучить не удается 144
11.4. Резюме 147
11.5. Упражнения 148

Глава 12. Выпуклые проблемы обучения 149
12.1. Выпуклость, липшицевость и гладкость 149
12.1.1. Выпуклость 149
12.1.2. Липшицевость 153

12.1.3. Гладкость ... 154
12.2. Выпуклые проблемы обучения ... 156
12.2.1. Обучаемость выпуклых проблем обучения ... 157
12.2.2. Выпуклые-липшицевы/гладкие-ограниченные проблемы обучения ... 158
12.3. Суррогатные функции потерь ... 159
12.4. Резюме ... 161
12.5. Библиографические сведения ... 161
12.6. Упражнения ... 161

Глава 13. Регуляризация и устойчивость ... 163
13.1. Минимизация регуляризированной потери ... 163
13.1.1. Гребневая регрессия ... 164
13.2. Устойчивые правила не подвержены переобучению ... 165
13.3. Регуляризация Тихонова как стабилизатор ... 166
13.3.1. Липшицева потеря ... 168
13.3.2. Гладкая неотрицательная потеря ... 169
13.4. Управление компромиссом между аппроксимацией и устойчивостью ... 170
13.5. Резюме ... 172
13.6. Библиографические сведения ... 172
13.7. Упражнения ... 173

Глава 14. Стохастический градиентный спуск ... 176
14.1. Градиентный спуск ... 177
14.1.1. Анализ метода ГС для выпуклых липшицевых функций ... 178
14.2. Субградиенты ... 180
14.2.1. Вычисление субградиентов ... 181
14.2.2. Субградиенты липшицевых функций ... 182
14.2.3. Субградиентный спуск ... 182
14.3. Стохастический градиентный спуск (СГС) ... 183
14.3.1. Анализ СГС для выпуклых–липшицевых–ограниченных функций ... 183
14.4. Варианты ... 185
14.4.1. Добавление шага проецирования ... 185
14.4.2. Переменный размер шага ... 186
14.4.3. Другие способы усреднения ... 186
14.4.4. Строго выпуклые функции* ... 187
14.5. Обучение с помощью СГС ... 188
14.5.1. Применение СГС для минимизации риска ... 188
14.5.2. Анализ СГС для выпуклых–гладких проблем обучения ... 190
14.5.3. Применение СГС для минимизации регуляризированной потери ... 191
14.6. Резюме ... 192
14.7. Библиографические сведения ... 192
14.8. Упражнения ... 192

Глава 15. Метод опорных векторов ... 194
15.1. Зазор и SVM с жестким зазором ... 194
15.1.1. Однородный случай ... 197

15.1.2. Выборочная сложность правила Hard-SVM 197
15.2. SVM с мягким зазором и регуляризация по норме 198
15.2.1. Выборочная сложность Soft-SVM 200
15.2.2. Сравнение границ, основанных на зазоре и норме, с размерностью 201
15.2.3. Рамповая функция потерь* 201
15.3. Условия оптимальности и «опорные векторы»* 202
15.4. Двойственность* 203
15.5. Реализация Soft-SVM с помощью СГС 204
15.6. Резюме 205
15.7. Библиографические сведения 205
15.8. Упражнения 206

Глава 16. Ядерные методы 207
16.1. Погружение в пространство признаков 207
16.2. Ядерный трюк 209
16.2.1. Ядра как способ выразить априорное знание 213
16.2.2. Характеристика ядерных функций* 214
16.3. Реализация Soft-SVM с ядрами 215
16.4. Резюме 216
16.5. Библиографические сведения 217
16.6. Упражнения 217

Глава 17. Многоклассовая категоризация, ранжирование и сложные проблемы предсказания 219
17.1. Один против всех и все пары 220
17.2. Линейные многоклассовые предикторы 222
17.2.1. Как построить Ψ 222
17.2.2. Стоимостная классификация 224
17.2.3. ERM 224
17.2.4. Обобщенная кусочно-линейная потеря 225
17.2.5. SVM и СГС в многоклассовом случае 226
17.3. Предсказание структурированного выхода 228
17.4. Ранжирование 230
17.4.1. Линейные предикторы для ранжирования 232
17.5. Двудольное ранжирование и многомерные показатели качества 235
17.5.1. Линейные предикторы для двудольного ранжирования 237
17.6. Резюме 239
17.7. Библиографические сведения 239
17.8. Упражнения 240

Глава 18. Решающие деревья 242
18.1. Выборочная сложность 243
18.2. Алгоритмы на решающих деревьях 244
18.2.1. Реализации меры выигрыша 245
18.2.2. Редукция 246
18.2.3. Пороговые правила разбиения для вещественных признаков 247

18.3. Случайные леса 247
18.4. Резюме 248
18.5. Библиографические сведения 248
18.6. Упражнения 248

Глава 19. Ближайшие соседи 250
19.1. Метод ***k*** ближайших соседей 250
19.2. Анализ 251
19.2.1. Граница обобщаемости для правила 1-NN 252
19.2.2. Проклятие размерности 255
19.3. Эффективная реализация* 256
19.4. Резюме 256
19.5. Библиографические сведения 257
19.6. Упражнения 257

Глава 20. Нейронные сети 260
20.1. Нейронные сети прямого распространения 261
20.2. Обучение нейронных сетей 262
20.3. Выразительная способность нейронных сетей 263
20.3.1. Геометрическая интерпретация 265
20.4. Выборочная сложность нейронных сетей 266
20.5. Время обучения нейронных сетей 267
20.6. СГС и обратное распространение 268
20.7. Резюме 272
20.8. Библиографические сведения 272
20.9. Упражнения 273

ЧАСТЬ III. ДОПОЛНИТЕЛЬНЫЕ МОДЕЛИ ОБУЧЕНИЯ 275

Глава 21. Онлайновое обучение 276
21.1. Онлайновая классификация в реализуемом случае 277
21.1.1. Онлайновая обучаемость 279
21.2. Онлайновая классификация в нереализуемом случае 283
21.2.1. Алгоритм взвешенного большинства 284
21.3. Онлайновая выпуклая оптимизация 288
21.4. Алгоритм онлайнового перцептрона 290
21.5. Резюме 293
21.6. Библиографические сведения 293
21.7. Упражнения 294

Глава 22. Кластеризация 296
22.1. Алгоритмы кластеризации на основе связи 299
22.2. Метод k-средних и другие методы кластеризации на основе минимизации стоимости 300
22.2.1. Алгоритм k-средних 302
22.3. Спектральная кластеризация 303
22.3.1. Разрезание графа 304
22.3.2. Лапласиан графа и ослабленные разрезы графа 304

22.3.3. Ненормированная спектральная кластеризация 305
22.4. Метод информационного горлышка* 306
22.5. Общий взгляд на кластеризацию 307
22.6. Резюме 309
22.7. Библиографические сведения 309
22.8. Упражнения 309

Глава 23. Понижение размерности 312
23.1. Метод главных компонент (PCA) 313
23.1.1. Более эффективное решение для случая $d \gg m$ 315
23.1.2. Реализация и демонстрация 315
23.2. Случайные проекции 317
23.3. Сжатое измерение сигнала 319
23.3.1. Доказательства* 321
23.4. PCA или сжатое измерение сигнала? 326
23.5. Резюме 327
23.6. Библиографические сведения 327
23.7. Упражнения 328

Глава 24. Порождающие модели 330
24.1. Оценка максимального правдоподобия 331
24.1.1. Оценка максимального правдоподобия для непрерывных случайных величин 332
24.1.2. Максимальное правдоподобие и минимизация эмпирического риска 333
24.1.3. Анализ обобщаемости 333
24.2. Наивная байесовская классификация 335
24.3. Линейный дискриминантный анализ 335
24.4. Скрытые переменные и EM-алгоритм 336
24.4.1. EM как алгоритм поочередной максимизации 338
24.4.2. EM-алгоритм для смеси нормальных распределений (мягкий алгоритм k-средних) 340
24.5. Байесовское рассуждение 341
24.6. Резюме 343
24.7. Библиографические сведения 343
24.8. Упражнения 343

Глава 25. Отбор и порождение признаков 345
25.1. Отбор признаков 346
25.1.1. Фильтры 347
25.1.2. Подходы на основе жадного отбора 348
25.1.3. Нормы, индуцирующие разреженность 351
25.2. Манипулирование и нормировка признаков 353
25.2.1. Примеры преобразований признаков 355
25.3. Обучение признаков 356
25.3.1. Обучение словаря с помощью автокодировщиков 356
25.4. Резюме 358

25.5. Библиографические сведения 358
25.6. Упражнения 359

ЧАСТЬ IV. ДОПОЛНИТЕЛЬНЫЕ ГЛАВЫ 361

Глава 26. Радемахеровская сложность 362
26.1. Радемахеровская сложность 362
26.1.1. Исчисление Радемахера 367
26.2. Радемахеровская сложность линейных классов 369
26.3. Границы обобщаемости метода SVM 371
26.4. Границы обобщаемости для предикторов с малой нормой ℓ_1 373
26.5. Библиографические сведения 374

Глава 27. Числа покрытия 375
27.1. Покрытие 375
27.1.1. Свойства 375
27.2. От покрытия к радемахеровской сложности через сцепление 376
27.3. Библиографические сведения 378

Глава 28. Доказательство фундаментальной теоремы теории обучения 379
28.1. Верхняя граница для агностического случая 379
28.2. Нижняя граница для агностического случая 380
28.2.1. Доказательство того, что $m(\epsilon, \delta) \geq 0{,}5\log(1/(4\boldsymbol{\delta}))/\epsilon^2$ 381
28.2.2. Доказательство того, что $m(\epsilon, 1/8) \geq 8\mathrm{d}/\epsilon^2$ 382
28.3. Верхняя граница для реализуемого случая 385
28.3.1. От ϵ-сетей к PAC-обучаемости 388

Глава 29. Многоклассовая обучаемость 389
29.1. Размерность Натараджана 389
29.2. Фундаментальная многоклассовая теорема 390
29.2.1. О доказательстве теоремы 29.3 390
29.3. Вычисление размерности Натараджана 391
29.3.1. Метод «один против всех» 391
29.3.2. Сведение многоклассовой категоризации к бинарной классификации в общем случае 392
29.3.3. Линейные многоклассовые предикторы 392
29.4. О хороших и плохих правилах ERM 394
29.5. Библиографические сведения 395
29.6. Упражнения 396

Глава 30. Границы сжатия 397
30.1. Границы сжатия 397
30.2. Примеры 399
30.2.1. Осепараллельные прямоугольники 399
30.2.2. Полупространства 399
30.2.3. Разделение полиномов 401

30.2.4. Разделение с зазором 401
30.3. Библиографические сведения 401

Глава 31. PAC-байесовский подход 402
31.1. PAC-байесовские границы 402
31.2. Библиографические сведения 404
31.3. Упражнения 405

Приложение A. Технические леммы 406

Приложение B. Концентрация меры 409
B.1. Неравенство Маркова 409
B.2. Неравенство Чебышева 410
B.3. Границы Чернова 411
B.4. Неравенство Хёфдинга 412
B.5. Неравенства Беннета и Бернштейна 413
B.5.1. Применение 414
B.6. Неравенство Слада 415
B.7. Концентрация случайных величин χ^2 415

Приложение C. Линейная алгебра 418
C.1. Основные определения 418
C.2. Собственные значения и собственные векторы 419
C.3. Положительно определенные матрицы 419
C.4. Сингулярное разложение 419

Литература 423

Предметный указатель 432

ПРЕДИСЛОВИЕ

Под *машинным обучением* понимается автоматизированное нахождение осмысленных закономерностей – паттернов – в данных. За последние двадцать лет оно стало обычным инструментом для решения почти всех задач, в которых требуется извлекать информацию из больших наборов данных. Нас со всех сторон окружают технологии, основанные на машинном обучении: поисковые системы учатся показывать наиболее полезные результаты (и при этом подсовывать прибыльную рекламу), антиспамные программы учатся фильтровать нашу почту, а операции с кредитными картами защищены программами, которые учатся распознавать мошенничество. Цифровые камеры учатся распознавать лица, а персональные помощники в смартфонах – понимать голосовые команды. Автомобили оснащаются системами предотвращения аварий, в которые встроены алгоритмы машинного обучения. Машинное обучение активно применяется и в научных дисциплинах, в т. ч. в биоинформатике, медицине и астрономии.

Все эти приложения объединяет общая черта – в отличие от традиционного применения компьютеров распознаваемые паттерны настолько сложны, что программист не способен явно сформулировать детальный алгоритм решения таких задач. Разумные существа приобретают или совершенствуют многие свои навыки путем *обучения* на собственном опыте (а не следования явным инструкциям). Задача инструментов машинного обучения – наделить программы способностью «обучаться» и адаптироваться. Первая цель этой книги – предложить строгое и вместе с тем достаточно простое для чтения введение в основные вопросы машинного обучения: что такое обучение; как обучается машина; как количественно оценить ресурсы, необходимые для обучения данной концепции; всегда ли возможно обучение; как узнать, завершился процесс обучения успешно или неудачно.

Вторая цель книги – изложить некоторые важнейшие алгоритмы машинного обучения. Мы выбрали алгоритмы, которые, с одной стороны, успешно применяются на практике, а с другой – представляют широкий спектр технических приемов обучения. Кроме того, мы уделили особое внимание алгоритмам, пригодным для обучения на больших объемах данных (так называемых «больших данных»), поскольку в последние годы наш мир стремительно «оцифровывается», так что объем данных, доступных для обучения невероятно вырос. В результате многие приложения больше не испытывают недостатка в данных, и узким местом становится время вычислений. Поэтому мы явным образом оцениваем как объем данных, так и время, необходимое для обучения данной концепции.

Книга состоит из четырех частей. Задача первой части – дать первоначальные строгие ответы на фундаментальные вопросы обучения. Мы опишем модель обучения, предложенную Валиантом, – вероятно почти корректное (Probably Approximately Correct – PAC), которая стала первым основательным ответом на вопрос «что такое обучение». Мы сформулируем три правила обучения – минимизация

эмпирического риска (Empirical Risk Minimization – ERM), структурная минимизация риска (Structural Risk Minimization – SRM) и минимальная длина описания (Minimum Description Length – MDL), – которые показывают, «как машина может обучаться». Мы количественно оценим объем данных, необходимый для обучения по правилам ERM, SRM и MDL, и, доказав теорему об «отсутствии бесплатных завтраков», покажем, каким образом обучение может завершиться неудачно. Мы также обсудим, сколько времени необходимо для обучения. Во второй части книги мы опишем различные алгоритмы обучения. В некоторых случаях мы сначала представим некий общий принцип обучения, а затем продемонстрируем, как алгоритм следует этому принципу. Если первые две части книги основаны на модели PAC, то в третьей мы расширим свой кругозор, введя в рассмотрение другие модели обучения. И, наконец, в последней части излагается более сложная теория.

Мы старались по возможности сделать книгу независимой. Однако предполагается, что читатель знаком с основами теории вероятностей, линейной алгебры, математического анализа и теории алгоритмов. Первые три части ориентированы на студентов первого года магистратуры по информатике, техническим наукам, математике или статистике. Они доступны также студентам средних курсов с хорошей подготовкой. Главы четвертой части могут быть полезны исследователям, желающим углубить свои теоретические знания.

Благодарности

В основу книги легли курсы «Введение в машинное обучение», прочитанные Шаем Шалев-Шварцем в Еврейском университете и Шаем Бен-Давидом в университете Ватерлоо. Первая черновая редакция появилась на свет из конспектов к курсу, прочитанному в Еврейском университете Шаем Шалев-Шварцем в 2010–2013 гг. Мы высоко ценим помощь Охада Шамира, который был учебным ассистентом на курсе 2010 г., и Алона Гонена, исполнявшего эту роль на курсах 2011–2013 гг. Охад и Алон подготовили конспекты и значительную часть упражнений. Алон, перед которым мы в долгу за помощь на протяжении всей работы над книгой, также подготовил ключ к решениям.

Мы глубоко признательны Дане Рубинштейн за ее ценнейший труд. Дана осуществила научное и литературное редактирование рукописи, превратив ее из разрозненных глав в связный, легко читаемый текст.

Отдельное спасибо Амиту Даниэли, который помогал в тщательной вычитке более сложных разделов книги и написал главу о многоклассовой обучаемости. Мы также благодарны членам иерусалимского клуба читателей, которые внимательно прочли каждую строчку рукописи и высказали конструктивную критику. Это Майя Элрой (Maya Alroy), Йосси Арджевани (Yossi Arjevani), Аарон Бирнбаум (Aharon Birnbaum), Алон Коэн (Alon Cohen), Алон Гонен (Alon Gonen), Рой Ливни (Roi Livni), Офер Месхи (Ofer Meshi), Дэн Розенбаум (Dan Rosenbaum), Дана Рубинштейн (Dana Rubinstein), Шахар Сомин (Shahar Somin), Алон Винников (Alon Vinnikov) и Йоав Вальд (Yoav Wald). Мы также признательны Гэлу Элидану (Gal Elidan), Амиру Глоберсону (Amir Globerson), Нике Нахталаб (Nika Haghtalab), Ши Маннору (Shie Mannor), Амнону Шашуа (Amnon Shashua), Нати Сребро (Nati Srebro) и Рут Эрнер (Ruth Urner) за полезные дискуссии.

Глава 1

ВВЕДЕНИЕ

Тема этой книги – автоматизированное обучение, или, как его чаще называют, машинное обучение (МО). То есть мы хотим запрограммировать компьютер так, чтобы он мог «обучаться» на доступных ему данных. Грубо говоря, обучение – это процесс преобразования эмпирического опыта в знания и умения. На вход алгоритма обучения подаются обучающие данные, представляющие опыт, а на выходе получаются знания, обычно принимающие вид другой компьютерной программы, способной выполнить некоторое задание. Если мы хотим описать эту идею формально-математически, то должны явно определить, что понимается под каждым вышеупомянутым термином. Что представляют собой обучающие данные, к которым будут обращаться наши программы? Как можно автоматизировать процесс обучения? Как оценить успешность этого процесса (т. е. качество результата обучающей программы)?

1.1. Что такое обучение?

Начнем с рассмотрения двух примеров, естественно возникающих при обучении животных. Некоторые из самых фундаментальных проблем МО видны уже в этом контексте, с которым все мы знакомы.

Недоверие к приманке – крысы учатся избегать отравленных приманок. Когда крыса обнаруживает еду с незнакомым видом или запахом, она сначала откусывает очень маленький кусочек, а ее дальнейшее поведение зависит от вкуса пищи и ее физиологических последствий. Если крыса почувствует себя плохо, то свяжет новую еду с недомоганием и не станет ее есть. Очевидно, здесь присутствует какой-то механизм обучения: животное воспользовалось прошлым опытом питания и приобрело умение определять безопасность пищи. Если с прошлым опытом были связаны негативные воспоминания, то животное предсказывает, что и в будущем последствия будут негативными.

Вдохновившись этим примером успешного обучения, продемонстрируем типичную задачу машинного обучения. Пусть требуется запрограммировать машину, которая будет обучаться фильтрации спама. Наивное решение было бы похоже на то, как крыса учится избегать отравленных приманок. Машина просто *запоминает* все предшествующие сообщения, которые были помечены как спам человеком. Когда приходит новое сообщение, машина ищет его среди уже известных спамных и, если находит, то отбрасывает. В противном случае сообщение отправляется в папку «Входящие».

Хотя такое «обучение путем запоминания» иногда полезно, этому подходу недостает важной черты обучающих систем – способности помечать ранее не встречавшиеся почтовые сообщения. Успешный обучаемый должен уметь совершать переход от отдельных примеров к более широкому *обобщению*. Это называется также *индуктивным рассуждением*, или *индуктивным выводом*. В описанном примере недоверия к приманке крыса, встретившая некоторый образчик пищи, распространяет свое отношение к нему на новые, ранее не виданные образчики пищи с похожим вкусом и запахом. Чтобы перейти к обобщению в задаче фильтрации спама, обучаемый может просканировать ранее предъявленные сообщения и выделить набор слов, наличие которых свидетельствует о спамном характере сообщения. Тогда, получив новое сообщение, машина сможет проверить, встречаются ли в нем подозрительные слова, и, соответственно, предсказать метку. Такая система потенциально могла бы правильно предсказывать метки ранее не предъявлявшихся сообщений.

Однако индуктивное рассуждение может приводить к ложным заключениям. Проиллюстрируем это еще одним примером обучения животных.

Суеверное поведение голубей. В эксперименте психолога Б.Ф. Скиннера группу голодных голубей помещали в клетку. Клетка была оборудована автоматическим механизмом, который через одинаковые интервалы выдавал голубям пищу вне зависимости от их поведения. Голодные голуби бродили по клетке, и в момент первой доставки пищи каждая птица чем-то занималась (долбила клювом пол, поворачивала голову и т. д.). Появление пищи подкрепляло действие каждой птицы, и впоследствии она чаще повторяла это действие, чем другие. Это, в свою очередь, увеличивало вероятность, что при следующем кормлении голуби также будут совершать это действие. В результате формируется цепочка событий, которая подкрепляет у голубей ассоциацию между появлением пищи и теми действиями, которые они совершали при первом кормлении. В дальнейшем они совершают эти действия намеренно[1].

Чем отличаются механизмы, приводящие к суеверию и к полезному обучению? Этот вопрос имеет первостепенное значение для разработки автоматизированных систем обучения. Человек может полагаться на здравый смысл, отбрасывая случайные бессмысленные заключения, но, переходя к обучению машины, мы должны сформулировать четкие, не допускающие двоякого толкования принципы, которые не дадут программе прийти к бессмысленным или бесполезным выводам. Разработка таких принципов и есть главная цель теории машинного обучения.

Так в чем же все-таки причина того, что обучение крыс оказалось успешнее, чем обучение голубей? В качестве первого шага к ответу на этот вопрос рассмотрим более пристально феномен недоверия к приманке.

Еще раз о недоверии к приманке: у крыс не формируется условно-рефлекторная связь между пищей и ударом электрическим током или между звуком и тошнотой. Механизм недоверия к приманке у крыс оказывается сложнее, чем могло бы показаться. В экспериментах Гарсия (Garcia & Koelling, 1996) было продемонстрировано, что если неприятный стимул, сопутствующий поеданию пищи, заменить, например, ударом электрическим током (а не тошнотой), то условного

[1] См. http://psychclassics.yorku.ca/Skinner/Pigeon.

рефлекса не возникает. Даже после нескольких испытаний, в которых поедание определенной пищи сопровождалось неприятным ударом током, крысы не отказывались от этой пищи. Условный рефлекс не формировался и тогда, когда свойство пищи, вызывающее тошноту (например, вкус или запах), заменялось звуковым сигналом. У крыс, похоже, имеется априорное «встроенное» знание о том, что временная связь между пищей и тошнотой может быть причинно-следственной, а наличие причинно-следственной связи между поеданием пищи и ударом током или между звуками и тошнотой маловероятно.

Мы приходим к выводу, что одним из отличий между недоверием к приманке и суеверному поведению голубей является наличие *априорного знания*, которое модифицирует механизм обучения. Его еще называют *индуктивным смещением* (inductive bias). Голуби в эксперименте готовы принять *любое* объяснение появления еды. Но крысы «знают», что еда не может быть причиной удара электрическим током и что сопровождение еды шумом вряд ли может оказать влияние на питательные качества этой еды. Процесс обучения крыс смещен в сторону обнаружения определенных паттернов, тогда как другие временные корреляции между событиями игнорируются.

Выясняется, что включение априорного знания, смещающего процесс обучения, является обязательным условием успешности алгоритмов обучения (это утверждение, строго сформулированное и доказанное в главе 5, получило название «теоремы об отсутствии бесплатных завтраков»). Разработка инструментов для выражения знания о предметной области, трансляции их в смещение алгоритма обучения и количественной оценки влияния этого смещения на успех обучения – центральная тема теории машинного обучения. Грубо говоря, чем сильнее априорные знания (или априорные предположения), с которыми мы начинаем процесс обучения, тем легче идет обучение на последующих примерах. Однако чем сильнее априорные предположения, тем менее гибким будет обучение – оно ограничено необходимостью соблюдать эти предположения. Мы будем явно обсуждать эти вопросы в главе 5.

1.2. Когда необходимо машинное обучение?

Когда возникает необходимость прибегнуть к машинному обучению, вместо того чтобы непосредственно запрограммировать компьютер на решение стоящей задачи? Два свойства задачи наводят на мысль воспользоваться программами, которые могут обучаться и совершенствоваться на основе своего «опыта»: сложность и требование адаптивности.

Задачи, слишком сложные для программирования

- *Задачи, решаемые животными или людьми*. Есть немало задач, которые человек решает естественно и привычно, и тем не менее мы недостаточно хорошо понимаем, как мы это делаем, чтобы оформить это в виде программы. Вот несколько примеров: вождение автомобиля, распознавание речи, понимание того, что изображено на картинке. Во всех этих случаях современные программы машинного обучения, «учащиеся на своем опыте», позволяют

добиться вполне удовлетворительных результатов, если им предъявить достаточно много обучающих примеров.

- *Задачи, выходящие за пределы человеческих возможностей*. Еще один класс задач, для которого машинное обучение дает очевидный выигрыш, связан с анализом очень больших и сложных наборов данных: астрономические данные, преобразование архивов медицинских записей в медицинские знания, прогнозирование погоды, анализ генома, поисковые системы в вебе, электронная торговля. По мере накопления цифровых данных становится очевидно, что в архивах скрыты информационные сокровища, вот только для человеческого разума они слишком велики и сложны. Обучение машин выявлять осмысленные паттерны в больших и сложных наборах данных – это многообещающая область, в которой сочетание обучающихся программ с почти безграничной памятью и постоянно растущим быстродействием компьютеров открывает новые горизонты.

Адаптивность

Одно из ограничений программируемых инструментов – отсутствие гибкости: после того, как программа написана и установлена, она уже не изменяется. С другой стороны, многие задачи изменяются со временем или в зависимости от конкретного пользователя. Средства машинного обучения – программы, поведение которых адаптируется к входным данным, – предлагают решение таких проблем; они по природе своей приспосабливаются к изменениям среды, с которой взаимодействуют. Типичные примеры успешного применения машинного обучения к такого рода задачам – программы распознавания рукописного текста (одна и та же программа способна адаптироваться к почерку разных пользователей), программы обнаружения спама (автоматически адаптируются к изменениям в характере спамных сообщений) и программы распознавания речи.

1.3. Типы обучения

Конечно, обучение – чрезвычайно обширная область. Поэтому в машинном обучении выделяют несколько дисциплин, различающихся типами обучения. Мы дадим приблизительную таксономию парадигм обучения, чтобы читатель понимал, какое место эта книга занимает в обширном массиве исследований по машинному обучению.

Мы будем классифицировать парадигмы обучения по четырем параметрам.

- **С учителем и без учителя.** Поскольку обучение подразумевает взаимодействие между обучаемым и окружением, задачи обучения можно различать по характеру этого взаимодействия. Прежде всего, отметим разницу между обучением с учителем и без учителя. В качестве примера сравним задачи о распознавании спама и об обнаружении аномалий. В случае задачи о распознавании спама будем считать, что обучаемый получает образцы сообщений с метками «спам – не спам». По результатам такого обучения обучаемый должен выработать правило пометки новых сообщений. С другой стороны, в задаче об обнаружении аномалий обучаемый полу-

чает только тело сообщения (без каких бы то ни было меток) и должен выявить «необычные» сообщения.

Если рассматривать обучение более абстрактно – как процесс «использования эмпирического опыта для получения знаний», то обучению с учителем соответствует случай, когда «опыт» – обучающий пример – содержит существенную информацию (например, метки «спам – не спам»), отсутствующую в еще не виденных «тестовых примерах», к которым будут применены обретенные в ходе обучения знания. При таком подходе цель приобретения знаний – предсказать отсутствующую в тестовых данных информацию. Мы можем считать окружение учителем, который «наставляет» обучаемого, предоставляя ему дополнительную информацию (метки). А в случае обучения без учителя между обучающими и тестовыми данными нет никакого различия. Обучаемый обрабатывает входные данные, имея целью составить на их основе некоторый дайджест, или сжатое представление. Типичный пример такой задачи – кластеризация набора данных, т. е. разбиение его на подмножества похожих объектов.

Существует также промежуточный тип обучения, когда обучающие примеры содержат больше информации, чем тестовые, а обучаемый должен предсказать для тестовых примеров еще больше информации. Например, можно попытаться обучить функцию, которая для каждой позиции на шахматной доске оценивает, насколько позиция белых лучше, чем у черных. При этом единственная информация, доступная на этапе обучения на позициях фактически сыгранных партий, – метка, сообщающая, кто в итоге выиграл. Такие подходы обычно изучаются в дисциплине, называемой *обучение с подкреплением*.

- **Активный и пассивный обучаемый.** Парадигмы обучения могут классифицироваться по роли обучаемого. Мы различаем «активного» и «пассивного» обучаемого. Активный обучаемый взаимодействует с окружением на этапе обучения, например, задавая вопросы или ставя эксперименты, тогда как пассивный только наблюдает за информацией, поставляемой окружением (или учителем), не оказывая на нее никакого влияния и не направляя процесс обучения. Отметим, что обучаемый фильтр спама обычно пассивен: он ждет, пока пользователи пометят входящие сообщения. Но можно было бы представить и активную конфигурацию, когда обучаемый сам выбирает или даже составляет сообщения и просит пользователей их пометить, тем самым улучшая свое понимание того, что такое спам.
- **Полезность учителя.** В процессе обучения человека, будь то ребенок дома или ученик в школе, обычно присутствует готовый прийти на помощь учитель, который пытается дать обучаемому информацию, наиболее полезную для достижения цели обучения. Напротив, когда ученый исследует природу, окружающую среду, роль учителя можно в лучшем случае считать пассивной: яблоко падает, солнце светит, дождь льет, не обращая никакого внимания на потребности обучаемого. Для моделирования таких типов обучения мы постулируем, что обучающие данные (опыт обучаемого) порождаются некоторым случайным процессом. Это основная идея «статистического обучения». Наконец, обучение имеет место и тогда, когда входные данные генерирует некий противодействующий «учитель». Так может

быть и при обучении фильтра спама (когда спамер стремится сбить с толку проектировщика фильтра), и при обучении обнаружению мошеннических действий. Модель противодействующего учителя – так называемое состязательное обучение – применяется также в худшем случае, когда нельзя безопасно предполагать более мягких условий. Если система может обучиться в условиях, когда учитель активно противодействует обучению, то она гарантированно добьется успеха при взаимодействии с любым сколь угодно странным учителем.

- **Онлайновое и пакетное обучение.** И напоследок упомянем различие между ситуацией, в которой обучаемый должен отвечать в режиме онлайн на протяжении всего процесса обучения, и ситуацией, когда разрешается применить обретенные знания уже после обработки значительного объема данных. Например, биржевой брокер должен ежедневно принимать решения на основе полученного к этому моменту опыта. Со временем он, возможно, станет экспертом, но в процессе обучения может допускать дорогостоящие ошибки. С другой стороны, во многих задачах добычи данных обучаемый – добытчик – располагает большим объемом обучающих данных, с которыми он может экспериментировать, прежде чем от него потребуют конкретных выводов.

В этой книге мы будем обсуждать только подмножество возможных парадигм обучения. Основное внимание мы уделим статистическому пакетному обучению с учителем и пассивным обучаемым (например, мы попытаемся выдавать прогноз течения болезни на основе больших архивов медицинских записей, которые были собраны независимо и уже помечены информацией о судьбе пациента). Мы также кратко остановимся на онлайновом обучении и пакетном обучении без учителя (в частности, кластеризации).

1.4. Связи с другими дисциплинами

Располагаясь на стыке различных дисциплин, машинное обучение связано многочисленными нитями с математической статистикой, теорией информации, теорией игр и оптимизацией. Естественно, оно является частью информатики, поскольку наша цель – программирование машин, способных обучаться. В некотором смысле машинное обучение можно рассматривать как отрасль искусственного интеллекта (ИИ), поскольку умение обращать опыт в знания и обнаруживать осмысленные паттерны в сложной сенсорной информации – это краеугольный камень разума человека (и животного). Однако следует отметить, что, в отличие от традиционного ИИ, машинное обучение стремится не столько добиться автоматизированной имитации разумного поведения, сколько использовать сильные стороны компьютеров и присущие только им возможности, чтобы дополнить человеческий разум, и на этом пути зачастую решает задачи, выходящие за рамки возможностей человека. Например, способность просматривать и обрабатывать гигантские базы данных позволяет программам машинного обучения обнаруживать паттерны за пределами человеческого восприятия.

То, что мы называем в машинном обучении опытом, часто относится к случайно сгенерированным данным. Задача обучаемого – на основе обработки слу-

чайных примеров прийти к выводам, справедливым для окружения, из которого эти примеры выбирались. Такое описание машинного обучения раскрывает его тесную связь со статистикой. И действительно, между этими двумя дисциплинами много общего – в части как целей, так и применяемых методов. Но есть, однако, и существенные различия в постановке задачи: если врач выдвигает гипотезу о наличии корреляции между курением и болезнями сердца, то задача статистика – обработать выборку пациентов и проверить достоверность этой гипотезы (это типичная статистическая задача о проверке гипотез). Напротив, цель машинного обучения – использовать данные, собранные о пациентах, и предложить описание возможных причин болезней сердца. При этом мы надеемся, что автоматизированные методы смогут выявить осмысленные паттерны (или гипотезы), ускользнувшие от внимания человека.

В отличие от традиционной статистики, в машинном обучении вообще и в этой книге в частности важную роль играют алгоритмические соображения. Суть машинного обучения заключается в обучении компьютеров, поэтому от алгоритмов никуда не деться. Мы разрабатываем алгоритмы для решения задач обучения, и их вычислительная эффективность нам отнюдь не безразлична. Еще одно отличие состоит в том, что статистику часто интересует асимптотическое поведение (например, сходимость выборочных статистических оценок, когда размер выборки стремится к бесконечности), тогда как теория машинного обучения акцентирует внимание на конечных выборках. Точнее, теория пытается оценить, какой верности предсказаний можно ожидать от обучаемого при данном размере доступных выборок.

Существуют и другие различия между обеими дисциплинами, но здесь мы упомянем только одно из них. В статистике, вообще говоря, имеется некоторое априорное предположение о модели данных (например, о нормальности порождающих данные распределений или о линейности функциональных зависимостей), тогда как в машинном обучении упор делается на работу в условиях «неизвестного распределения», когда обучаемый старается не делать никаких предположений о природе распределения данных, а обучающему алгоритму предоставляется возможность определить, какая модель наилучшим образом аппроксимирует процесс порождения данных. Для строгого обсуждения этого вопроса необходима математическая подготовка, но мы вернемся к нему позже и, в частности, в главе 5.

1.5. Как читать эту книгу

В первой части книги излагаются базовые теоретические принципы, лежащие в основе машинного обучения (МО). В каком-то смысле это фундамент, на котором возведено здание книги. На базе этой части можно прочитать мини-курс по теоретическим основаниям МО.

Во второй части книги дается введение в наиболее распространенные алгоритмические подходы к машинному обучению с учителем. Подмножество этих глав можно использовать как введение в машинное обучение в рамках общего курса ИИ для студентов, изучающих математику, информатику или технические дисциплины.

В третьей части рамки обсуждения расширяются: мы переходим от статистической классификации к другим моделям обучения. Рассматривается онлайновое обучение, обучение без учителя, понижение размерности, порождающие модели и обучение признаков.

Четвертая часть книги ориентирована на читателей с исследовательской жилкой. Здесь излагаются более сложные математические методы, необходимые для анализа и дальнейшего развития теоретического машинного обучения.

В приложениях описаны некоторые технические средства, используемые в книге. В частности, приводятся основные результаты из теории концентрации меры и линейной алгебры.

Разделы, помеченные звездочкой, адресованы более подготовленным студентам. Каждая глава завершается списком упражнений. Ключ к решениям имеется на сайте курса.

1.5.1. Варианты построения курса на основе книги

14-недельный вводный курс для студентов магистратуры

1. Главы 2–4.
2. Глава 9 (без вычисления VC).
3. Главы 5–6 (без доказательств).
4. Глава 10.
5. Главы 7, 11 (без доказательств).
6. Главы 12, 13 (с некоторыми простыми доказательствами).
7. Глава 14 (с некоторыми простыми доказательствами).
8. Глава 15.
9. Глава 16.
10. Глава 18.
11. Глава 22.
12. Глава 23 (без доказательств для сжатого измерения сигнала).
13. Глава 24.
14. Глава 25.

14-недельный повышенный курс для студентов магистратуры

1. Главы 26, 27.
2. (продолжение)
3. Главы 6, 28.
4. Глава 7.
5. Глава 31.
6. Глава 30.
7. Главы 12, 13.
8. Глава 14.
9. Глава 8.
10. Глава 17.
11. Глава 29.
12. Глава 19.
13. Глава 20.
14. Глава 21.

1.6. Обозначения

По большей части обозначения, встречающиеся в книге, либо стандартны, либо определяются на месте. В этом разделе мы опишем основные применяемые соглашения и приведем сводную таблицу обозначений (табл. 1.1). Читатель может пропустить этот раздел и возвращаться к нему по ходу чтения книги, встретив непонятное обозначение.

Скаляры и абстрактные объекты обозначаются строчными буквами (например, x и λ). Если мы хотим подчеркнуть, что объект является вектором, то употребляем полужирный шрифт (например, $\mathbf{x}$ и $\boldsymbol{\lambda}$). i-й элемент вектора $\mathbf{x}$ обозначается x_i. Заглавными буквами обозначаются матрицы, множества и последовательности. О чем конкретно идет речь, всегда понятно из контекста. Как мы скоро увидим, на вход алгоритма обучения подается последовательность обучающих примеров. Мы обозначаем z абстрактный пример, а $S = z_1, \ldots, z_m$ – последовательность m примеров. Исторически сложилось, что S называют обучающим *набором*, или *множеством*, однако мы всегда предполагаем, что S – *последовательность*, а не множество. Последовательность m векторов обозначается $\mathbf{x}_1, \ldots, \mathbf{x}_m$, а i-й элемент вектора $\mathbf{x}_t$ – $x_{t,i}$.

В книге повсеместно используются понятия из теории вероятностей. Мы обозначаем $\mathcal{D}$ распределение на некотором множестве[1], например, Z. Запись $z \sim \mathcal{D}$ означает, что z выбрано из распределения $\mathcal{D}$. Если $f\colon Z \to \mathbb{R}$ – случайная величина, то ее математическое ожидание обозначается $\mathbb{E}_{z\sim\mathcal{D}}[f(z)]$. Иногда мы сокращаем эту запись до $\mathbb{E}[f]$, если зависимость от z очевидна из контекста. В случае, когда $f\colon Z \to \{\text{true, false}\}$, для обозначения $\mathcal{D}(\{z\colon f(z) = \text{true}\})$ используется также нотация $\mathbb{P}_{z\sim\mathcal{D}}[f(z)]$. В следующей главе мы введем также нотацию $\mathcal{D}^m$ для обозначения вероятности на Z^m, индуцированной выборкой $(z_1, \ldots, z_m)$, где каждая точка z_i выбирается из распределения $\mathcal{D}$ независимо от других точек.

Таблица 1.1. Сводка обозначений

Символ	Значение
$\mathbb{R}$	множество вещественных чисел
$\mathbb{R}^d$	множество d-мерных векторов над $\mathbb{R}$
$\mathbb{R}_+$	множество неотрицательных вещественных чисел
$\mathbb{N}$	множество натуральных чисел
$O, o, \Theta, \omega, \Omega, \tilde{O}$	асимптотическая нотация (см. в тексте)
$1_{[\text{булево выражение}]}$	индикаторная функция (равна 1, если выражение истинно, и 0 в противном случае)
$[a]_+$	$= \max(0, a)$
$[n]$	множество $\{1, \ldots, n\}$ (для $n \in \mathbb{N}$)
$\mathbf{x}, \mathbf{v}, \mathbf{w}$	векторы-столбцы
x_i, v_i, w_i	i-й элемент вектора
$\langle \mathbf{x}, \mathbf{v} \rangle$	$= \sum_{i=1}^{d} x_i v_i$ (скалярное произведение)
$\|\|\mathbf{x}\|\|_2$ или $\|\|\mathbf{x}\|\|$	$= \sqrt{\langle \mathbf{x}, \mathbf{x} \rangle}$ (ℓ_2 – норма вектора $\mathbf{x}$)

[1] Математически правильнее было бы определять $\mathcal{D}$ на σ-алгебре подмножеств Z. Читатели, не знакомые с теорией меры, могут пропустить несколько сносок и замечаний, относящихся к формальным определениям и предположением об измеримости.

Окончание табл. 1.1

Символ	Значение
$\|\|\mathbf{x}\|\|_1$	$= \sum_{i=1}^{d} \|x_i\|$ (ℓ_1 – норма вектора $\mathbf{x}$)
$\|\|\mathbf{x}\|\|_\infty$	$= \max_i \|x_i\|$ (ℓ_∞ – норма вектора $\mathbf{x}$)
$\|\|\mathbf{x}\|\|_0$	количество ненулевых элементов $\mathbf{x}$
$A \in \mathbb{R}^{d,k}$	матрица $d \times k$ над $\mathbb{R}$
$A^\top$	матрица, транспонированная к A
$A_{i,j}$	элемент (i, j) матрицы A
$\mathbf{x}\mathbf{x}^\top$	матрица A размера $d \times d$ такая, что $A_{i,j} = x_i x_j$ (где $\mathbf{x} \in \mathbb{R}^d$)
$\mathbf{x}_1, \ldots, \mathbf{x}_m$	последовательность m векторов
$x_{i,j}$	j-й элемент i-го вектора последовательности
$\mathbf{w}^{(1)}, \ldots, \mathbf{w}^{(T)}$	значения вектора $\mathbf{w}$ в итеративном алгоритме
$w_i^{(t)}$	i-й элемент вектора $\mathbf{w}^{(t)}$
$\mathcal{X}$	множество образцов
$\mathcal{Y}$	множество меток
Z	множество примеров
$\mathcal{H}$	класс гипотез (множество)
$\ell : \mathcal{H} \times Z \to \mathbb{R}_+$	функция потерь
$\mathcal{D}$	распределение на некотором множестве (обычно на Z или на $\mathcal{X}$)
$\mathcal{D}(A)$	вероятность множества $A \subseteq Z$ при распределении $\mathcal{D}$
$z \sim \mathcal{D}$	выборка z из распределения $\mathcal{D}$
$S = z_1, \ldots, z_m$	последовательность m примеров
$S \sim \mathcal{D}^m$	выборка $S = z_1, \ldots, z_m$ независимых и одинаково распределенных (с распределением $\mathcal{D}$) случайных величин
$P, \mathbb{E}$	вероятность и математическое ожидание случайной величины
$\mathbb{P}_{z\sim\mathcal{D}}[f(z)]$	$= \mathcal{D}(\{z: f(z) = \text{true}\})$ для $f : Z \to \{\text{true}, \text{false}\}$
$\mathbb{E}_{z\sim\mathcal{D}}[f(z)]$	математическое ожидание случайной величины $f : Z \to \mathbb{R}$
$N(\mu, C)$	нормальное распределение с математическим ожиданием μ и ковариацией C
$f'(x)$	производная функции $f : \mathbb{R} \to \mathbb{R}$ в точке x
$f''(x)$	вторая производная функции $f : \mathbb{R} \to \mathbb{R}$ в точке x
$\dfrac{\partial f(\mathbf{w})}{\partial w_i}$	частная производная функции $f: \mathbb{R}^d \to \mathbb{R}$ в точке $\mathbf{w}$ по w_i
$\nabla f(\mathbf{w})$	градиент функции $f : \mathbb{R}^d \to \mathbb{R}$ в точке $\mathbf{w}$
$\partial f(\mathbf{w})$	множество дифференциалов функции $f : \mathbb{R}^d \to \mathbb{R}$ в точке $\mathbf{w}$
$\min_{x\in C} f(x)$	$= \min\{f(x) : x \in C\}$ (минимальное значение f на множестве C)
$\max_{x\in C} f(x)$	$= \max\{f(x) : x \in C\}$ (максимальное значение f на множестве C)
$\operatorname{argmin}_{x\in C} f(x)$	множество$\{x \in C : f(x) = \min_{z\in C} f(z)\}$
$\operatorname{argmax}_{x\in C} f(x)$	множество$\{x \in C : f(x) = \max_{z\in C} f(z)\}$
log	натуральный логарифм

Вообще говоря, мы старались избегать асимптотической нотации. Но иногда все же используем ее, чтобы прояснить главные результаты. В частности, если имеются функции $f : \mathbb{R} \to \mathbb{R}_+$ и $g : \mathbb{R} \to \mathbb{R}_+$, то мы пишем $f = O(g)$, если существуют x_0 и $\alpha \in \mathbb{R}_+$ такие, что для любого $x > x_0$ имеет место неравенство $f(x) \leq \alpha g(x)$.

Мы пишем $f = o(g)$, если для любого $\alpha > 0$ существует x_0 такое, что для всех $x > x_0$ имеет место неравенство $f(x) \leq \alpha g(x)$. Мы пишем $f = \Omega(g)$, если существуют такие x_0 и $\alpha \in \mathbb{R}_+$, что для всех $x > x_0$ имеет место неравенство $f(x) \geq \alpha g(x)$. Нотация $f = \omega(g)$ определяется аналогично. Нотация $f = \Theta(g)$ означает, что $f = O(g)$ и $g = O(f)$. Наконец, нотация $f = \tilde{O}(g)$ означает, что существует $k \in \mathbb{N}$ такое, что $f(x) = O(g(x) \log^k(g(x)))$.

Скалярное произведение векторов $\mathbf{x}$ и $\mathbf{w}$ обозначается $\langle \mathbf{x}, \mathbf{w} \rangle$. Если явно не сказано, о каком векторном пространстве идет речь, то предполагается d-мерное евклидово пространство, и тогда $\langle \mathbf{x}, \mathbf{w} \rangle = \sum_{i=1}^{d} x_i w_i$. Евклидова (или ℓ_2) норма вектора $\mathbf{w}$ равна $\|\mathbf{w}\|_2 = \sqrt{\langle \mathbf{w}, \mathbf{w} \rangle}$. Мы опускаем нижний индекс в ℓ_2, если вид нормы понятен из контекста. Мы используем также другие нормы ℓ_p, $\|\mathbf{w}\|_p = (\sum_i |w_i|^p)^{1/p}$, и, в частности, $\|\mathbf{w}\|_1 = \sum_i |w_i|$ и $\|\mathbf{w}\|_\infty = \max_i |w_i|$.

Мы используем нотацию $\min_{x \in C} f(x)$ для обозначения минимального значения в множестве $\{f(x) : x \in C\}$. С точки зрения математики, следовало бы использовать $\inf_{x \in C} f(x)$ в случае, когда минимум не достигается. Но в контексте этой книги различие между минимумом и нижней гранью редко представляет интерес. Поэтому, чтобы не усложнять изложение, мы иногда употребляем нотацию min даже в тех случаях, когда inf было бы правильнее. Аналогичное замечание относится к max и sup.