

# 1

## Введение

### 1.1. Что такое машинное обучение

Машинное обучение — это раздел информатики, посвященный созданию алгоритмов, опирающихся на набор данных о каком-либо явлении. Эти данные могут быть получены из естественной среды, созданы вручную или сгенерированы другим алгоритмом.

Машинное обучение также можно определить как процесс решения практической задачи путем 1) формирования набора данных и 2) алгоритмического построения статистической модели на его основе. Предполагается, что эта статистическая модель будет каким-то образом использоваться для решения практической задачи.

Чтобы сэкономить на нажатиях клавиш, я буду использовать термины «обучение» и «машинное обучение» как взаимозаменяемые.

### 1.2. Типы обучения

Обучение может быть с учителем, без учителя и с подкреплением.

#### 1.2.1. Обучение с учителем

В обучении с учителем набор данных организован как коллекция **размеченных образцов**  $\{(\mathbf{x}_i, y_i)\}_{i=1}^N$ . Каждый элемент  $\mathbf{x}_i$  из  $N$  называется **вектором признаков**. Вектор признаков — это вектор, в котором каждое измерение  $j = 1, \dots, D$  содержит значение, описывающее некоторую характеристику образца. Это значение называется **признаком** и обозначается как  $x^{(j)}$ . Например, если каждый образец  $\mathbf{x}$  в нашей

коллекции представляет человека, тогда первый признак,  $x^{(1)}$ , мог бы описывать его рост в сантиметрах, второй признак,  $x^{(2)}$ , мог бы описывать его вес в килограммах,  $x^{(3)}$  — пол, и т. д. Для всех данных в наборе данных признак в позиции  $j$  в векторе признаков всегда описывает одну и ту же характеристику. Это означает, что если  $x_i^{(2)}$  описывает вес некоторого образца  $\mathbf{x}$ , в килограммах, тогда  $x_k^{(2)}$  также будет описывать вес в килограммах каждого образца  $x_k$ ,  $k = 1, \dots, N$ . **Метка**  $y_i$  может быть элементом конечного множества **классов**  $\{1, 2, \dots, C\}$ , вещественным числом или более сложной структурой, такой как вектор, матрица, дерево или граф. В этой книге, если явно не оговаривается иное, под  $y_i$  будет подразумеваться элемент конечного множества классов или вещественное число<sup>1</sup>. Класс можно представить как категорию, которой принадлежит образец. Например, если роль данных играют электронные письма и вы решаете задачу определения спама, тогда вы могли бы определить два класса:  $\{\text{спам}, \text{не\_спам}\}$ .

Цель **алгоритма обучения с учителем** — на основе набора данных создать **модель**, которая принимает вектор признаков  $\mathbf{x}$  на входе и возвращает информацию, которая позволяет определить метку для этого вектора признаков. Например, модель, созданная с использованием набора данных людей, могла бы принимать вектор признаков, описывающих человека, и возвращать вероятность, что этот человек болен раком.

## 1.2.2. Обучение без учителя

В **обучении без учителя** набор данных представлен коллекцией **неразмеченных образцов**  $\{\mathbf{x}_i\}_{i=1}^N$ . И снова,  $\mathbf{x}$  — это вектор признаков, а цель **алгоритма обучения без учителя** — создать **модель**, которая принимает вектор признаков  $\mathbf{x}$  на входе и преобразует его в другой вектор или в значение, которое можно использовать для решения практической задачи. Например, в задачах **кластеризации** модель возвращает идентификатор кластера для каждого вектора признаков в наборе данных. В задачах **уменьшения размерности** модель возвращает вектор признаков, который имеет меньше элементов, чем входной вектор  $\mathbf{x}$ . В задачах **выявления аномалий** возвращается действительное число, которое указывает, насколько  $\mathbf{x}$  отличается от «типичного» образца в наборе данных.

## 1.2.3. Обучение с частичным привлечением учителя

В обучении с частичным привлечением учителя (semi-supervised learning) набор данных содержит как размеченные, так и неразмеченные образцы. Обычно неразмеченных образцов намного больше, чем размеченных. **Алгоритм обучения**

<sup>1</sup> Вещественное число — это величина, которую можно представить как расстояние на числовой прямой. Примеры вещественных чисел: 0, -256.34, 1000, 1000.2.

**с частичным привлечением учителя** преследует ту же цель, что и алгоритм обучения с учителем. В данном случае предполагается, что использование множества неразмеченных данных поможет алгоритму обучения найти (можно также сказать «произвести» или «вычислить») лучшую модель.

Предположение о выигрыше от добавления большего количества неразмеченных данных может показаться нелогичным. На первый взгляд кажется, что мы, наоборот, добавляем больше неопределенности. Однако добавляя неразмеченные данные, вы вносите больше информации о задаче: большая выборка лучше отражает распределение вероятностей в данных, откуда взяты размеченные образцы. Теоретически, алгоритм обучения должен уметь воспользоваться этой дополнительной информацией.

### 1.2.4. Обучение с подкреплением

**Обучение с подкреплением** — это раздел машинного обучения, где предполагается, что машина «живет» в определенном окружении и способна воспринимать **состояние** этого окружения как вектор характеристик. Машина может выполнять некоторые **действия** в каждом состоянии. Разные действия приносят разные **вознаграждения**, а также могут перевести машину в другое состояние окружения. Цель алгоритма обучения с подкреплением — выучить **линию поведения**.



Линия поведения — это стратегия (похожая на модель в обучении с учителем), которая принимает вектор признаков, описывающий состояние, и возвращает оптимальное действие для выполнения в этом состоянии. Действие является оптимальным, если приводит к максимальному **ожидаемому среднему вознаграждению**.

Обучение с подкреплением решает особый класс задач, когда решения принимаются последовательно, а цель является долгосрочной, например игра в видеоигру, роботизация производства, управление ресурсами или логистика. В этой книге основное внимание будет уделяться принятию одновременных решений, когда исходные данные не зависят друг от друга, и решений, принятых в прошлом. Обучение с подкреплением я оставляю за рамками этой книги.

## 1.3. Как работает обучение с учителем

В этом разделе я кратко объясню, как работает обучение с учителем, чтобы дать вам общую картину процесса, прежде чем углубиться в детали. В качестве примера

я выбрал обучение с учителем, так как этот тип машинного обучения чаще других используется на практике.

Процесс обучения с учителем начинается со сбора данных. Данные для этого вида обучения — это коллекция пар (вход, выход). Входными данными может быть все что угодно, например электронные письма, изображения или замеры, полученные от датчика. Выходными данными обычно являются действительные числа или метки (например, «спам», «не\_спам», «кошка», «собака», «мышь» и т. д.). В некоторых случаях выходные данные могут быть представлены векторами (например, четырьмя координатами углов прямоугольника вокруг человека на картинке), последовательностями (например, [«прилагательное», «прилагательное», «существительное»] для входа «большая красивая машина») или иметь какую-то другую структуру.

Допустим, вы решили использовать обучение с учителем для решения задачи определения спама. Вы собираете данные, например, 10 000 электронных писем, каждое из которых снабжается меткой «спам» или «не\_спам» (вы можете добавить эти метки вручную или отдать эту работу на аутсорсинг). Затем вы должны преобразовать каждое электронное письмо в вектор признаков.

На основе своего опыта специалист по анализу данных решает, как преобразовать сущность реального мира, такую как электронное письмо, в вектор признаков. Часто для преобразования текста в вектор признаков используется метод, называемый **мешком слов**. Его суть заключается в том, чтобы взять словарь (допустим, что он содержит 20 000 слов, отсортированных по алфавиту) и условиться, что в векторе признаков:

- первый признак равен 1, если электронное письмо содержит слово «а» (союз), и 0 в другом случае;
- второй признак равен 1, если электронное письмо содержит слово «аарон» (имя), и 0 в другом случае;
- ...;
- признак в позиции 20 000 равен 1, если электронное письмо содержит слово «ящур» (болезнь), и 0 в другом случае.

Вы повторяете описанную процедуру для каждого электронного письма в вашей коллекции и получаете 10 000 векторов признаков (каждый вектор имеет размерность 20 000) и меток («спам»/«не\_спам»).

Теперь у вас есть машиночитаемые входные данные, но выходные метки по-прежнему имеют вид простого текста. Некоторые алгоритмы обучения требуют преобразования меток в числа. Например, некоторые алгоритмы требуют исполь-

зовать числа 0 (для представления метки «не\_спам») и 1 (для представления метки «спам»). Модель машинного обучения, которую я использую для иллюстрации обучения с учителем, называется **методом опорных векторов (Support Vector Machine, SVM)**. Она требует, чтобы положительная метка (в данном случае «спам») имела числовое значение +1 (один), а отрицательная метка («не\_спам») имела значение -1 (минус один).

Теперь у вас есть **набор данных** и **алгоритм обучения** и вы готовы применить алгоритм обучения к набору данных, чтобы получить **модель**.

SVM рассматривает каждый вектор признаков как точку в многомерном пространстве (в данном случае пространство имеет 20 000 измерений). Алгоритм помещает все векторы признаков на воображаемый 20 000-мерный график и рисует воображаемую 19 999-мерную линию (*гиперплоскость*), которая отделяет данные с положительными метками от данных с отрицательными метками. Граница, разделяющая данные разных классов, в машинном обучении называется **границей принятия решения**.

Уравнение гиперплоскости задается двумя **параметрами**: вещественным вектором  $\mathbf{w}$  той же размерности, что и входной вектор признаков  $\mathbf{x}$ , и действительным числом  $b$ , например:

$$\mathbf{w}\mathbf{x} - b = 0,$$

где выражение  $\mathbf{w}\mathbf{x}$  означает  $w^{(1)}x^{(1)} + w^{(2)}x^{(2)} + \dots + w^{(D)}x^{(D)}$ , а  $D$  — число измерений в векторе признаков  $\mathbf{x}$ .

(Сейчас некоторые уравнения могут показаться вам сложными, но в главе 2 мы рассмотрим необходимые математические и статистические понятия. А пока просто попробуйте понять происходящее в меру своих знаний. Многое прояснится, когда вы прочитаете следующую главу.)

Теперь прогнозируемую метку для некоторого входного вектора признаков  $\mathbf{x}$  можно выразить так:

$$y = \text{sign}(\mathbf{w}\mathbf{x} - b),$$

где  $\text{sign}$  — это математический оператор, принимающий произвольное значение и возвращающий +1, если входное значение является положительным числом, и -1, если входное значение является отрицательным числом.

Цель алгоритма обучения, в данном случае SVM, используя набор данных, найти оптимальные значения  $\mathbf{w}^*$  и  $b^*$  для параметров  $\mathbf{w}$  и  $b$ . После того как

алгоритм обучения найдет эти оптимальные значения, модель  $f(\mathbf{x})$  будет определяться как:

$$f(\mathbf{x}) = \text{sign}(\mathbf{w}^* \mathbf{x} - b^*).$$

Чтобы с помощью модели SVM предсказать, является ли электронное письмо спамом или нет, вы должны взять текст письма, преобразовать его в вектор признаков, затем умножить этот вектор на  $\mathbf{w}^*$ , вычесть  $b^*$  и взять знак результата. Это даст нам прогноз (+1 означает «спам», а -1 означает «не\_спам»).

Но как машина находит  $\mathbf{w}^*$  и  $b^*$ ? Она решает задачу оптимизации. Машины хорошо справляются с оптимизацией функций в условиях ограничений.

Итак, какие ограничения должны удовлетворяться здесь? Прежде всего, модель должна правильно предсказывать метки имеющихся 10 000 данных. Напомню, что каждый образец  $i = 1, \dots, 10\,000$  задается парой  $(\mathbf{x}_i, y_i)$ , где  $\mathbf{x}_i$  — вектор признаков  $i$ -го образца, а  $y_i$  — его метка, которая принимает значение -1 или +1. Ограничения выглядят следующим образом:

$$\begin{aligned} \mathbf{w}\mathbf{x}_i - b &\geq +1, \text{ если } y_i = +1, \\ \mathbf{w}\mathbf{x}_i - b &\leq -1, \text{ если } y_i = -1. \end{aligned}$$

Желательно также, чтобы гиперплоскость отделяла положительные данные от отрицательных с максимальным **зазором**. **Зазор** — это расстояние между ближайшими образцами двух классов, отделяемых границей принятия решения. Большой зазор способствует лучшему **обобщению**, то есть тому, насколько хорошо модель будет классифицировать новые данные. Для максимизации зазора нужно минимизировать евклидову норму  $\mathbf{w}$ , которая обозначается как  $\|\mathbf{w}\|$  и определяется выражением  $\sqrt{\sum_{j=1}^D (w^{(j)})^2}$ .

Машина должна решить задачу оптимизации, которая формулируется так:

*Минимизировать  $\|\mathbf{w}\|$  с учетом  $y_i(\mathbf{w}\mathbf{x}_i - b) \geq 1$  для  $i = 1, \dots, N$ .* Выражение  $y_i(\mathbf{w}\mathbf{x}_i - b) \geq 1$  — это всего лишь компактная запись двух ограничений выше.

Решение этой задачи оптимизации, обозначенной параметрами  $\mathbf{w}^*$  и  $b^*$ , называется **статистической моделью**, или просто **моделью**. Процесс построения модели называется **обучением**.

Для двумерных векторов задачу и решение можно представить визуально, как показано на рис. 1.1. Синие и оранжевые точки представляют положительные и отрицательные образцы соответственно, а линия, заданная уравнением  $\mathbf{w}\mathbf{x} - b = 0$ , представляет границу принятия решения.