Содержание

Предисловие

Первое издание этой книги вышло в свет в 2009 г., т. е. к моменту написания второго издания прошло более 10 лет. Поскольку за эти годы область машинного обучения существенно продвинулась вперед, мы решили подготовить второе издание. Мы постарались включить в него наиболее важные достижения, чтобы новая версия книги содержала актуальную информацию об этой области и была полезна исследователям, аспирантам и прикладным специалистам, работающим в этой области.

Каковы основные изменения? Прежде всего если вы просто сравните количество глав двух изданий, то заметите, что их стало вдвое больше. Примерно так же увеличилось количество страниц.

Отметим, что на момент написания первого издания термина *автоматизированное машинное обучение* (automated machine learning, AutoML) даже не существовало. Очевидно, что мы должны были описать это новаторское направление в новом издании, а также прояснить его связь с метаобучением. Кроме того, автоматизация методов проектирования цепочек операций – в настоящее время называемых *конвейерами* (pipeline) или *рабочими процессами* (workflow) – находилась в зачаточном состоянии. Разумеется, мы осознавали необходимость обновить существующий материал, чтобы не отставать от этого развития.

В последние годы исследования в области AutoML и метаобучения привлекают большое внимание не только исследователей, но и многих компаний, занимающихся искусственным интеллектом, включая, например, Google и IBM. Как можно использовать метаобучение для улучшения систем AutoML – это один из важнейших вопросов, на который в настоящее время пытаются ответить многие исследователи.

Эта книга нацелена в будущее. Как это обычно случается, чем лучше исследователи разбираются в какой-то области, тем больше перед ними возникает новых вопросов. Мы позаботились о том, чтобы включить некоторые из них в соответствующие главы.

Авторами первого издания были Павел Браздил, Кристоф Жиро-Каррье, Карлос Соарес и Рикардо Вилальта. Учитывая масштабные нововведения в этой области, мы решили укрепить команду, пригласив Хоакина Ваншорена и Яна ван Рейна присоединиться к проекту. К сожалению, Кристоф и Рикардо не смогли принять участие в работе над новым изданием. Тем не менее все авторы второго издания очень благодарны за их вклад в начало проекта.

Как устроена эта книга

Эта книга состоит из трех частей. В части I (главы 2–7) рассмотрены основные концепции и архитектура систем метаобучения и AutoML, а в части II (главы 8–15) обсуждаются различные расширения. Часть III (главы 16–18) описывает способы хранения и управления метаданными (например, хранилища метаданных) и заканчивается заключительными замечаниями.

Часть I. Основные понятия и архитектура

Глава 1 начинается с объяснения основных понятий, используемых в этой книге, таких как машинное обучение, метаобучение, автоматизированное машинное обучение и др. Затем она продолжается обзором базовой архитектуры системы метаобучения и служит введением к остальной части книги. Над этой главой работали все соавторы книги.

Глава 2 посвящена методам ранжирования на основе метаданных, поскольку их относительно легко реализовать, но это не умаляет их полезности в практических приложениях. Эта глава была написана П. Браздилом и Я. ван Рейном[1]. Глава 3, написанная теми же авторами, посвящена теме оценки метаобучения и систем AutoML. В главе 4 обсуждаются различные показатели наборов данных, которые играют важную роль в качестве метапризнаков в системах метаобучения. Эта глава, как и следующая, также написана П. Браздилом и Я. ван Рейном. Главу 5 можно рассматривать как продолжение главы 2. В ней обсуждаются различные подходы к метаобучению, включая, например, попарные сравнения, которые применялись в прошлом. В главе 6 обсуждается оптимизация гиперпараметров. Она охватывает как базовые методы поиска, так и более продвинутые, применяемые в области AutoML. Эту главу написали три автора – П. Браздил, Я. ван Рейн и Х. Ваншорен. В главе 7 обсуждается вопрос автоматизации построения рабочих процессов или конвейеров, представляющих собой последовательности операций. Эта глава написана П. Браздилом, но в ней повторно использованы некоторые материалы из первого издания, подготовленного К. Жиро-Каррье.

Часть II. Передовые технологии и методы

Часть 2 (главы 8–15) продолжает темы части I, но охватывает различные расширения базовой методологии. Глава 8, написанная П. Браздилом и Я. ван Рейном, посвящена теме построения пространств конфигураций и планированию экспериментов. В двух последующих главах обсуждается конкретная тема ансамблей моделей. Глава 9, написанная Ш. Жиро-Каррье, дополняет материал этой книги. Она описывает различные способы организации набора алгоритмов базового уровня в ансамбли. Авторы второго издания не видели необходимости изменять эту главу, поэтому она сохранена в том виде, в каком появилась в первом издании.

Глава 10 продолжает тему ансамблей и показывает, как метаобучение можно использовать при построении ансамблей (ансамблевом обучении).

---

[1] Части глав 2 и 3 первого издания, написанные К. Соаресом и П. Браздилом, были повторно использованы и адаптированы для этой главы.

Эта глава была написана К. Соаресом и П. Браздилом. Последующие главы посвящены более конкретным темам. Глава 11, написанная Я. ван Рейном, описывает применение метаобучения для предоставления рекомендаций по выбору алгоритма потоковой обработки данных. Глава 12, написанная Р. Вилалтой и М. Месхи, посвящена переносу метамоделей и представляет собой вторую дополняющую главу этой книги. Это существенно обновленная версия аналогичной главы из первого издания, написанная Р. Вилалтой. Глава 13, написанная М. Хьюисманом, Я. ван Рейном и А. Плаатом, обсуждает метаобучение в глубоких нейронных сетях и представляет собой третью дополняющую главу этой книги. Глава 14 посвящена относительно новой теме автоматизации науки о данных. Эта глава была составлена П. Браздилом и содержит обзор различных идей и предложений его соавторов. Цель главы состоит в том, чтобы обсудить различные операции, обычно выполняемые в науке о данных, и рассмотреть вопрос о том, возможна ли здесь автоматизация и можно ли использовать в этом процессе метазнания. Цель главы 15, написанной П. Браздилом, также состоит в том, чтобы заглянуть в будущее и рассмотреть возможность автоматизации проектирования более сложных решений. В их число могут входить не только конвейеры операций, но и более сложные структуры управления (например, итерации) и автоматические изменения в базовом представлении.

Часть III. Организация и использование метаданных

Часть III охватывает некоторые практические вопросы и содержит последние три главы (16–18). В главе 16, написанной Х. Ваншореном и Я. ван Рейном, обсуждаются репозитории метаданных и, в частности, репозиторий, известный под названием OpenML. Этот репозиторий содержит данные о многих экспериментах по машинному обучению, проведенных в прошлом, и их соответствующие результаты. В главе 17, написанной Я. ван Рейном и Х. Ваншореном, показано, как можно изучать метаданные, чтобы получить более глубокое представление об исследованиях машинного обучения и метаобучения и, как следствие, сконструировать новые эффективные прикладные системы. Глава 18 завершает книгу краткими заключительными замечаниями о роли метазнания, а также представляет некоторые перспективные направления исследований. В основном глава была написана П. Браздилом, но содержит вклад других соавторов, в частности Я. ван Рейна и К. Соареса.

Благодарности

Мы выражаем благодарность всем тем, кто помог осуществить этот проект.

Мы признательны за грант 612.001.206 от Голландского исследовательского совета (NWO), направленный на финансирование издания этой книги.

Павел Браздил выражает благодарность Университету Порту, экономическому факультету, научно-исследовательскому институту INESC TEC и одному из его научных центров, а именно Лаборатории искусственного интеллекта и поддержки принятия решений (LIAAD), за их постоянную поддержку. Выполненная работа была частично поддержана национальными фонда-

ми через португальское агентство FCT Fundação para a Cincia e a Tecnologia в рамках проекта UIDB/50014/2020.

Ян ван Рейн выражает благодарность Фрайбургскому университету, Институту обработки и анализа данных (DSI) Колумбийского университета и Лейденскому институту передовых компьютерных наук (LIACS) Лейденского университета за их поддержку на протяжении всего проекта.

Карлос Соарес выражает благодарность Университету Порту и инженерному факультету за их поддержку.

Хоакин Ваншорен выражает благодарность Технологическому университету Эйндховена и Группе интеллектуального анализа данных за их поддержку.

Мы в большом долгу перед многими нашими коллегами за полезные идеи и предложения, отраженные в этой книге, это Салису Абдулрахман, Бернд Бишль, Хендрик Блокил, Изабель Гайон, Хольгер Хоос, Джеффри Холмс, Франк Хуттер, Жоао Гама, Руи Лейте, Андреас Мюллер, Бернхард Пфарингер, Ашвин Шринивасан и Мартин Вистуба.

Мы также отмечаем вклад многих других исследователей, сделанный ими в результате различных встреч и дискуссий, лично или по электронной почте, это Херке ван Хоф, Петер Флах, Павел Кордик, Том Митчелл, Катарина Морик, Сергей Муравьев, Аске Плаат, Рикардо Пруденсио, Люк Де Рэдт, Мишель Себаг и Кейт СмитМайлз.

Мы также хотим поблагодарить многих других коллег, с которыми сотрудничали: Педро Абреу, Митру Баратчи, Бильге Челика, Виктора Черкейру, Андре Коррейю, Афонсу Кошту, Тиаго Кунья, Катарину Эггенспергер, Матиаса Фойрера, Питера Гийсберса, Карлоса Гомеса, Тачиану Гомес, Хендрика Хугебума, Майка Хьюсмана, Матиаса Кенига, Ларса Котхоффа, Хорхе Канда, Аарона Кляйна, Уолтера Костерса, Мариуса Линдауэра, Марту Мерсье, Перикла Миранда, Феликса Мора, Мохаммада Нозари, Сильвию Нуньес, Катарину Оливейра, Маркоса де Паула Буэно, Флориана Пфистерера, Фабио Пинто, Питера ван дер Путтена, Сахи Рави, Адриано Риволли, Андре Росси, Клаудио Са, Прабханта Сингха, Артура Соуза, Бруно Соуза, Фрэнка Берета и Джонатана Виса. Мы также благодарны сообществу OpenML за их усилия по обеспечению воспроизводимости исследований в области машинного обучения.

Мы также благодарны нашему редактору Ронану Ньюджент из издательства Springer за его терпение и поддержку на протяжении всего проекта.

Мы хотим выразить благодарность Мануэлю Карамело за его тщательную корректуру рукописи всей книги и за многочисленные исправления.

Павел Браздил
Ян ван Рейн
Карлос Соарес
Хоакин Ваншорен

Порту, Эйндховен, Лейден
Март 2021 г.

Часть I

ОСНОВНЫЕ КОНЦЕПЦИИ И АРХИТЕКТУРА

1

Введение

КРАТКОЕ СОДЕРЖАНИЕ ГЛАВЫ Эта глава начинается с описания структуры книги, состоящей из трех частей. В части I обсуждаются некоторые основные концепции, в том числе, например, что такое метаобучение и как оно связано с *автоматизированным машинным* обучением (automated machine learning, AutoML). Далее следуют представление базовой архитектуры систем метаобучения/AutoML и обсуждение систем, использующих выбор алгоритма с использованием априорных метаданных, методологии, используемой при их оценке, и различных типов моделей метауровня, при этом упоминаются соответствующие главы, в которых можно получить более подробную информацию. Эта часть также содержит обсуждение методов, используемых для оптимизации гиперпараметров и разработки рабочего процесса. Часть II включает в себя обсуждение более продвинутых технологий и методов настройки конфигурационных пространств и проведения экспериментов. В последующих главах обсуждаются различные типы ансамблей, метаобучение в ансамблевых методах, алгоритмы, используемые для потоков данных, и перенос метамоделей между задачами. Одна глава посвящена метаобучению для глубоких нейронных сетей. В последних двух главах обсуждаются проблемы автоматизации различных задач обработки данных и попытки проектирования более сложных систем. Часть III относительно короткая. В ней обсуждаются репозитории метаданных (включая результаты экспериментов) и приводятся примеры информации, которую можно извлечь из этих метаданных. В последней главе представлены заключительные замечания.

1.1. Структура книги

Эта книга состоит из трех частей. В части I (главы 2–7) мы обрисуем основные концепции и архитектуру систем метаобучения, уделяя особое внимание тому, какие метазнания можно собрать, наблюдая за работой различных моделей при выполнении предшествующих задач, и как можно использовать их в метаобучении для более эффективного освоения новых задач. Поскольку

метаобучение тесно связано с AutoML, мы подробно рассмотрим и эту тему, но с особым акцентом на то, как мы можем улучшить AutoML с помощью метаобучения.

Часть II (главы 8–15) охватывает проекцию этих основных идей на более конкретные задачи. Сначала мы обсудим методы, которые можно применять при проектировании пространств конфигураций, влияющих на поиск систем метаобучения и AutoML. Затем мы покажем, как метаобучение используют для создания наилучших ансамблей и рекомендации алгоритмов для потоковой передачи данных. Далее мы обсудим перенос информации из ранее изученных моделей на новые задачи, используя перенос обучения и обучение на нескольких примерах в нейронных сетях. Последние две главы посвящены проблемам автоматизации обработки данных и проектирования сложных систем.

Часть III (главы 16–18) содержит практические советы о том, как организовать метаданные в репозиториях и использовать их в исследованиях по машинному обучению. Последняя глава включает в себя наши заключительные замечания и обзор перспективных направлений исследований.

1.2. Основные концепции и архитектура (часть I)

1.2.1. Основные понятия

Роль машинного обучения

Мы окружены данными. Ежедневно мы сталкиваемся с ними в разнообразных формах. Компании пытаются продавать свою продукцию с помощью рекламы в виде рекламных баннеров и видеороликов. Обширные сенсорные сети и чувствительные телескопы наблюдают за сложными процессами, происходящими вокруг нас на Земле и во всей Вселенной. Фармацевтические компании исследуют взаимодействия между разными типами молекул в поисках новых лекарств, а медики и биологи продолжают открывать новые болезни.

Все эти данные ценны тем, что позволяют нам охарактеризовать различные ситуации, научиться разделять их на разные группы и выстраивать из них систему, помогающую нам принимать решения. Именно благодаря стройной системе данных мы можем выявлять мошеннические транзакции в банковской сфере, разрабатывать новые медицинские препараты на основе клинических данных или строить предположения об эволюции небесных тел во Вселенной. Построение системы данных обязательно включает в себя обучение.

Научное сообщество разработало множество методов анализа и обработки данных. Традиционной научной задачей является *моделирование* – упро-

щенное описание сложного явления с целью извлечения из него какой-либо полезной информации. С этой целью было разработано множество *методов моделирования данных*, основанных на различных интуитивных предположениях и догадках. Эта область исследований называется *машинным* обучением (machine learning, ML).

Роль метаобучения

Как известно, мы не можем рассчитывать на существование единого алгоритма, который работает для всех видов данных, поскольку каждый алгоритм охватывает лишь свою область знаний. Выбор правильного алгоритма для определенной задачи и набора данных является ключом к получению адекватной модели. Выбор алгоритма сам по себе можно рассматривать как задачу обучения.

Этот процесс обучения с переносом знаний между задачами обычно называют метаобучением. Однако за последние десятилетия различные исследователи машинного обучения толковали этот термин очень широко, охватывая такие понятия, как *метамоделирование*, обучение обучению, а также *непрерывное* и *ансамблевое* обучение и *перенос* обучения. Столь обширная и непрерывно растущая область применения убедительно говорит о том, что метаобучение может сделать машинное обучение значительно более эффективным, простым и надежным.

В области метаобучения ведутся очень активные исследования. Появляется много новых и интересных научных направлений, которые по-новому решают общую задачу. В этой книге мы постарались дать краткий обзор наиболее авторитетных исследований на сегодняшний день. Поскольку за десятилетие, прошедшее с момента первого издания этой книги, область метаобучения сильно разрослась, материал второго издания пришлось организовать по-новому и структурировать в отдельные блоки. Например, характеристики наборов данных обсуждаются в отдельной главе (глава 4), хотя они играют важную роль во многих других главах.

Определение метаобучения

Давайте начнем с определения метаобучения, как оно рассматривается в контексте этой книги:

> Метаобучение (metalearning) – это наука о принципиальных методах, использующих метазнание для получения эффективных моделей и решений путем адаптации процессов машинного обучения.

Упомянутые выше *метазнания* (metaknowledge) обычно включают в себя любую информацию, полученную из предыдущих задач, например описания предыдущих задач, использованных конвейеров и нейронных архитектур или готовых моделей. Во многих случаях сюда также входят знания, полученные в ходе поиска наилучшей модели для новой задачи, которые можно использовать для поиска лучших моделей обучения. Лемке и др. (Lemke et al., 2015) описывают метазнания с системной точки зрения:

> «Система метаобучения должна включать подсистему обучения, которая адаптируется с опытом. Опыт приобретается за счет использования метазнаний, извлеченных: а) в предыдущем эпизоде обучения на одном наборе данных и/или б) из разных областей или задач».

В настоящее время в большинстве случаев целью метаобучения является использование метаданных, полученных как из прошлых задач, так и текущего набора данных.

Метаобучение или автоматизированное машинное обучение?

Нам часто задают вопрос: в чем разница между системой метаобучения и системой AutoML? Хотя это довольно субъективный вопрос, на который могут быть даны разные ответы, здесь мы представляем определение AutoML, которое дали Гийон и др. (Guyon et al., 2015):

> «AutoML охватывает все аспекты автоматизации процесса машинного обучения, помимо выбора модели, оптимизации гиперпараметров и поиска модели...»

Многие системы AutoML используют опыт, извлеченный из ранее просмотренных наборов данных. Таким образом, многие системы AutoML, согласно приведенному выше определению, одновременно являются системами метаобучения. В этой книге мы сосредоточимся на методах, включающих метаобучение, а также на системах AutoML, которые часто используют метаобучение.

Происхождение термина «метаобучение»

В этой главе мы еще рассмотрим подробнее новаторскую работу Райса (Rice, 1976). Эта работа опередила свое время и стала широко известна в сообществе машинного обучения гораздо позже. В 1980-х гг. Ларри Ренделл опубликовал ряд статей по *управлению предубеждениями* (bias management, эта тема обсуждается в главе 8). Одна из его статей (Rendell et al., 1987) содержит следующий абзац:

> «VBMS [Система управления переменными предубеждениями] может выполнять метаобучение. В отличие от большинства других систем обучения, VBMS обучается на разных уровнях. В процессе изучения понятия система также приобретет знания о задачах индукции, предубеждениях и отношениях между ними. Таким образом, система будет не только изучать понятия, но и узнавать о взаимосвязи между задачами и методами решения задач».

Павел Браздил впервые столкнулся с термином *метаинтерпретатор* (meta-interpreter) в связи с работой Ковальски (Kowalski, 1979) в Эдинбургском университете в конце 70-х гг. В 1988 г. он организовал семинар по машинно-

му обучению, метавыводу и логике (*Machine Learning, Meta-Reasoning and Logics*, Brazdil, Konolige, 1990). Введение в эту книгу содержит следующий абзац:

> «Некоторые метазнания представляют собой знания, которые говорят о других знаниях (объектного уровня). Назначение такого метазнания в основном состоит в том, чтобы контролировать умозаключение. Согласно другой точке зрения метазнание играет несколько иную роль: оно используется для управления процессом приобретения и переформулирования знаний (обучения)».

Исследованию метаобучения был также посвящен проект StatLog (1990–93) (Michie et al., 1994).

1.2.2. Основные типы задач

В научной литературе обычно выделяют ряд типовых задач, многие из которых будут упоминаться на протяжении всей книги. Общая цель систем метаобучения состоит в том, чтобы извлечь уроки из использования предыдущих моделей (как они были построены и насколько хорошо они работали) для построения качественной модели целевого набора данных. Если задачей базового уровня является классификация, это означает, что система метаобучения может предсказывать значение целевой переменной, т. е. в данном случае значение класса. В идеале она должна делать это лучше или эффективнее экспертной модели (как минимум не хуже) за счет использования информации, помимо самих обучающих данных.

Выбор алгоритма (algorithm selection, AS): исходя из известного перечня алгоритмов и имеющегося целевого набора данных, необходимо определить, какой алгоритм лучше всего подходит для моделирования этого набора.

Оптимизация гиперпараметров (hyperparameter optimization, HPO): исходя из имеющегося алгоритма с определенными гиперпараметрами и целевого набора данных, необходимо определить наилучшее сочетание значений гиперпараметров для моделирования этого набора.

Комбинированный выбор алгоритма и оптимальных гиперпараметров (combined algorithm selection and hyperparameter optimization, CASH): пусть имеется ряд алгоритмов, каждый со своим набором гиперпараметров, и целевой набор данных; необходимо определить, какой алгоритм использовать и как настроить его гиперпараметры для моделирования целевого набора данных. Некоторые системы CASH также решают более сложную задачу синтеза конвейера, обсуждаемую далее.

Синтез рабочего процесса/конвейера (workflow/pipeline synthesis): пусть имеется ряд алгоритмов, каждый со своим набором гиперпараметров, и целевой набор данных; необходимо спроектировать рабочий процесс (конвейер), состоящий из одного или нескольких алгоритмов для моделирования целевого набора данных. Добавление определенного алгоритма и настроек его гиперпараметров в рабочий процесс можно рассматривать как отдельную задачу CASH.

Поиск и/или синтез архитектуры (architecture search and/or synthesis): задачи этого типа можно рассматривать как обобщение предыдущей задачи. Но в данном случае отдельные компоненты не обязательно должны быть организованы в последовательности, как это делается в конвейерах. Архитектура может содержать, например, частично упорядоченные или древовидные структуры. К этой категории задач можно также отнести проектирование архитектуры нейронной сети.

Обучение на нескольких примерах (few-shot learning): располагая целевым набором данных, который состоит из небольшого количества записей, и различными наборами данных, которые очень похожи, но содержат много записей, необходимо построить модель, обученную на предыдущих наборах данных (примерах), и настроить ее так, чтобы она хорошо работала с целевым набором данных.

Отметим, что задачи выбора алгоритма определяются на *дискретном* наборе алгоритмов, в то время как задачи HPO и CASH обычно определяются в непрерывных пространствах конфигураций или гетерогенных пространствах как с дискретными, так и с непрерывными переменными. Методы выбора алгоритма также могут быть легко применены к дискретным версиям последнего.

В этой книге мы следуем соглашению о терминах, повсеместно принятому сообществом машинного обучения. *Гиперпараметр* – это определяемый пользователем параметр, определяющий поведение конкретного алгоритма машинного обучения; например, коэффициент отсечения в дереве решений или скорость обучения в нейронных сетях являются гиперпараметрами. *Параметр* (применительно к модели) – это значение, полученное на основе обучающих данных. Например, веса нейронной сети являются параметрами модели.

1.2.3. Базовая архитектура систем метаобучения и AutoML

Задача выбора алгоритма была впервые сформулирована Райсом (Rice, 1976). Он заметил, что производительность[1] алгоритмов можно связать с *характеристиками/признаками набора данных*. Другими словами, признаки набора данных часто являются неплохими предикторами производительности алгоритмов. Их можно использовать при выборе наиболее эффективного алгоритма для имеющегося целевого набора данных. С тех пор данный подход нашел применение во многих областях, в том числе за пределами машинного обучения (Smith-Miles, 2008).

---

[1] Термин *performance*, который в IT-литературе обычно переводят как «производительность», на самом деле очень многозначен. В машинном обучении в зависимости от контекста он может означать точность прогноза модели, ее быстродействие, стабильность работы, а иногда все одновременно, т. е. совокупный уровень качества. В этой книге мы тоже используем перевод «производительность», подразумевая под ним свойства алгоритма или модели, зависящие от контекста. – *Прим. перев.*

Обобщенная архитектура систем метаобучения, решающих задачу выбора алгоритма, показана на рис. 1.1. Сначала собирают *метаданные*, которые содержат информацию о предыдущих эпизодах обучения. Сюда входят описания задач, которые мы решали ранее. Например, это могут быть задачи классификации для заданного набора данных или задачи обучения с подкреплением, определенные различными средами обучения. *Характеристики* этих задач часто очень полезны для рассуждений о том, как новые задачи могут быть связаны с предыдущими. Метаданные также включают *алгоритмы* (например, конвейеры машинного обучения или нейронные архитектуры), которые ранее использовались для изучения этих задач, и оценочную информацию о производительности, показывающую, насколько хорошо сработали эти решения. В некоторых случаях мы также можем хранить обученные модели или измеримые свойства этих моделей. Такие метаданные из многих предыдущих (или текущих) задач могут быть объединены в базу данных – своего рода «память» о предыдущем опыте, на который мы должны опираться. Эти метаданные можно использовать разными способами, например непосредственно внутри алгоритмов метаобучения или для обучения модели метауровня. Такую метамодель можно включить в состав системы метаобучения, как показано на рис. 1.2. Основываясь на характеристиках новой целевой задачи, метамодель может создавать или рекомендовать новые алгоритмы для опробования, а затем использовать оценку производительности для итеративного обновления текущего алгоритма или выдачи рекомендаций до тех пор, пока не будет выполнено какое-либо условие остановки (обычно ограничение по времени работы или достижение заданной производительности). В некоторых случаях характеристики задачи недоступны, и система метаобучения вынуждена учиться на предыдущем опыте и наблюдениях только за новой задачей.

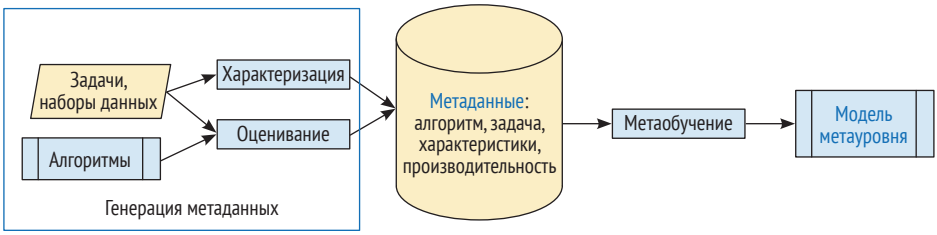

Рис. 1.1 ❖ Построение модели на метауровне

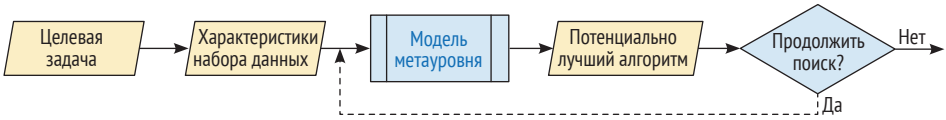

Рис. 1.2 ❖ Использование модели метауровня для прогнозирования наилучшего алгоритма

1.2.4. Выбор алгоритма с использованием метаданных из предыдущих задач (главы 2,5)

В главах 2 и 5 детально обсуждаются методы, использующие метаданные производительности алгоритмов из предыдущих задач, чтобы рекомендовать наиболее подходящие алгоритмы для целевого набора данных. Эти рекомендации могут быть выданы в форме ранжированного списка алгоритмов-кандидатов (глава 2) или метамоделей, которые предсказывают пригодность алгоритмов для новых задач (глава 5).

В главе 2 мы описываем относительно простой подход – *метод ранжирования по среднему* (average ranking method), который часто используется в качестве базового метода. Алгоритм ранжирования по среднему использует метазнания, полученные при решении предыдущих задач, для определения потенциально лучших алгоритмов базового уровня применительно к текущей задаче.

Этот подход требует, чтобы соответствующий показатель оценки, например точность, был выбран заранее. В этой главе мы также описываем метод, который строит рейтинг на основе сочетания точности и времени выполнения, обеспечивая тем самым оптимальную производительность при различных условиях.

В главе 5 обсуждаются более продвинутые методы, но в обоих случаях они предназначены для работы с дискретными задачами.

1.2.5. Оценка и сравнение различных систем (глава 3)

При работе с определенной системой метаобучения важно знать, можем ли мы доверять ее рекомендациям и какова ее эффективность по сравнению с другими конкурирующими подходами. В главе 3 обсуждается типичный способ оценки систем метаобучения и проведения сравнений.

Чтобы получить достоверную оценку производительности системы метаобучения, ее необходимо оценить на множестве наборов данных. Поскольку производительность алгоритмов может существенно различаться в зависимости от конкретного набора данных, многие системы сначала нормализуют значения производительности, чтобы сделать сравнения значимыми. В разделе 3.1 мы рассмотрим некоторые из наиболее распространенных методов нормализации, используемых на практике.

Предполагая, что система метаобучения выдает последовательность алгоритмов для тестирования, мы можем изучить, насколько эта последовательность далека от идеальной. Для этого можно измерить степень корреляции между двумя последовательностями, о чем пойдет речь в разделе 3.2.

Отметим, что описанный выше подход сравнивает прогнозы, сделанные моделью метауровня, с метацелью (т. е. с правильным порядком алгоритмов). Недостаток этого подхода заключается в том, что он не показывает напрямую эффект применения алгоритма с точки зрения производительности на уровне целевой задачи. Этой проблемы можно избежать, рассмат-

ривая соответствующую базовую производительность различных систем метаобучения и наблюдая, как она меняется со временем. Если известна идеальная производительность, можно рассчитать величину *потери производительности* (performance loss), которая представляет собой разницу между фактической производительностью и идеальным значением. Кривая потерь показывает, как потери меняются со временем, например при подборе алгоритмов. В некоторых системах заранее указывают максимально доступное время (т. е. бюджет времени). Затем можно сопоставить различные системы и их варианты, сравнив кривые потерь. Более подробная информация представлена в разделе 3.3.

В разделе 3.4 также представлены некоторые полезные критерии, такие как *свободная точность* (loose accuracy) и *дисконтированный совокупный прирост* (discounted cumulative gain), которые часто используются при сравнении последовательностей. Завершает главу раздел 3.5, где описана методология, которую обычно применяют при сравнении нескольких систем метаобучения/AutoML.

1.2.6. Роль характеристик/метапризнаков набора данных (глава 4)

Отметим, что в упомянутой ранее публикации Райса (1976) характеристики набора данных играют решающую роль. С тех пор их используют во многих системах метаобучения. Как правило, они помогают определить рамки поиска потенциально лучшего алгоритма. Если характеристики недоступны или их трудно конкретизировать для данной предметной области, поиск все равно возможен. Основные подходы, основанные на ранжировании или попарных сравнениях, которые обсуждались в главах 2 и 5, можно использовать и без каких-либо характеристик наборов данных. Это важное преимущество, поскольку во многих предметных областях действительно трудно придумать много информативных характеристик, позволяющих различать большое количество очень похожих конфигураций алгоритмов.

Одну из наиболее важных групп представляют *характеристики, основанные на производительности*. В эту группу входят, например, *выборочные ориентиры* (sampling landmarker), представляющие эффективность определенных алгоритмов на выборках данных. Их можно получить практически во всех областях.

Несомненно, некоторые характеристики имеют принципиальное значение. Возьмем, например, основную характеристику целевой переменной – ее тип. Если это числовая переменная, то предполагается, что следует использовать подходящий алгоритм регрессии, а если категориальная, то алгоритм классификации. Аналогичные рассуждения справедливы, когда мы сталкиваемся со сбалансированными или несбалансированными данными. Эта характеристика тоже обусловливает выбор правильного метода. В главе 4 обсуждаются различные характеристики набора данных, организованные по типам задач, таким как классификация, регрессия или временные ряды.

В различных главах этой книги показано, как характеристики наборов данных эффективно используются в различных подходах к метаобучению (например, главы 2 и 5).

1.2.7. Различные типы моделей метауровня (глава 5)

В последнее время исследователи рассматривают несколько типов моделей метауровня:

- *регрессионные*;
- *классифицирующие*;
- *относительной производительности* (relative performance).

В главе 5 такие модели рассмотрены более детально. Они используются в различных подходах, обсуждаемых на протяжении всей книги.

В регрессионной модели используется подходящий алгоритм регрессии, который обучается на метаданных; затем полученная модель применяется для прогнозирования производительности заданного набора алгоритмов базового уровня. Прогнозы можно использовать для упорядочивания этих алгоритмов и, следовательно, определения лучшего из них.

Регрессионные модели также играют важную роль в поиске наилучшей конфигурации гиперпараметров, особенно если они числовые и непрерывные. Например, в методе, называемом *последовательной оптимизацией на основе моделей* (sequential model-based optimization), описанном в главе 6, алгоритм регрессии на метауровне используется для моделирования функции потерь и определения перспективных настроек гиперпараметров.

Классифицирующая модель определяет, какие из алгоритмов базового уровня *применимы* к целевой задаче классификации. Это означает, что такие алгоритмы, скорее всего, продемонстрируют относительно хорошую производительность при выполнении целевой задачи. Отметим, что эта задача метаобучения применяется к дискретной области.

Если бы мы использовали на метауровне вероятностные классификаторы, которые, помимо класса (например, «применимый» или «неприменимый»), предоставляют также числовые значения, связанные с вероятностью классификации, то эти значения можно было бы использовать для определения потенциально наилучшего алгоритма базового уровня или для изучения ранжирования в дальнейшем поиске.

Модель относительной производительности основана на предположении, что нет необходимости выяснять подробности фактической производительности алгоритмов, если целью является выявление наиболее эффективных алгоритмов. Все, что необходимо, – это информация об их относительной производительности. Модели относительной производительности могут использовать либо ранжирование, либо попарные сравнения. Во всех этих сценариях можно использовать классические алгоритмы поиска для определения потенциально лучшего алгоритма, ориентированного на целевой набор данных.

1.2.8. Оптимизация гиперпараметров (глава 6)

В главе 6 описаны различные подходы к оптимизации гиперпараметров, а также комбинированные задачи выбора алгоритма и оптимизации гиперпараметров.

Эта глава отличается от глав 2 и 5 одним важным аспектом: в ней обсуждаются методы, использующие метаданные производительности, полученные в основном на целевом наборе данных. Метаданные применяются для создания относительно простых и быстро тестируемых моделей конфигурации целевого алгоритма (алгоритма с соответствующими настройками гиперпараметров), к которым можно обращаться с запросами. Целью этих запросов является определение наилучшей конфигурации на тестовом наборе, т. е. конфигурации с наивысшей оценкой производительности (например, точности). Этот тип поиска называется *поиском на основе модели* (model-based search).

Впрочем, здесь ситуация не совсем однозначна. Как показал ряд исследований, метаданные, собранные в предыдущих наборах данных, также могут быть полезны и способны улучшать производительность поиска на основе моделей.

1.2.9. Автоматические методы формирования конвейера (глава 7)

Многие задачи требуют решения, в котором используется не один алгоритм базового уровня, а последовательность из нескольких алгоритмов. Для обозначения таких последовательностей часто используют термины *рабочий процесс* (workflow) или *конвейер* (pipeline). В общем случае последовательность может быть упорядочена только частично.

При проектировании конвейеров количество конфигураций может резко возрасти. Это связано с тем, что каждый элемент конвейера в принципе может быть заменен соответствующей операцией базового уровня, и таких операций может быть несколько. Проблема усугубляется тем фактом, что последовательность из двух или более операторов вообще может выполняться в любом порядке, если явно не указано обратное. Это создает проблему, так как для N операторов существует $N!$ возможных упорядоченных комбинаций. Следовательно, если набор операторов должен выполняться в определенном порядке, для этого должны быть даны явные инструкции. Если порядок не имеет значения, системе также следует запретить экспериментировать с альтернативными порядками. Все альтернативные рабочие процессы и их конфигурации (включая все возможные настройки гиперпараметров) составляют так называемое *пространство конфигураций* (configuration space).

В главе 7 обсуждаются различные средства, которые применяются для ограничения количества вариантов проектирования и, таким образом, уменьшения размера пространства конфигураций. К ним относятся, напри-

мер, онтологии и контекстно-независимые грамматики. Каждый из этих подходов имеет свои достоинства и недостатки.

Многие платформы прибегают к системам планирования, использующим набор операторов. Они могут быть разработаны в соответствии с заданными онтологиями или грамматиками. Эта тема обсуждается в разделе 7.3.

Поскольку пространство поиска может быть довольно большим, важно использовать предыдущий опыт. Эта тема рассматривается в разделе 7.4, в котором обсуждается *ранжирование планов* (rankings of plans), которые оказались полезными в прошлом. Конвейеры, доказавшие свою эффективность в прошлом, можно извлечь и использовать в качестве планов будущих задач. Таким образом, можно одновременно задействовать два фундаментальных подхода – планирование и метаобучение.

1.3. Передовые технологии и методы (часть II)

1.3.1. Настройка пространств конфигураций и экспериментов (глава 8)

Одна из проблем, с которыми в настоящее время сталкиваются исследования метаобучения и AutoML, заключается в том, что количество алгоритмов (в общем случае конвейеров) и их конфигураций настолько велико, что поиск приемлемого решения в этом пространстве может оказаться очень затруднительным. Кроме того, становится невозможно иметь полный набор экспериментальных результатов (полные метаданные). Отсюда вытекает несколько вопросов.

1. Достаточно ли пространства конфигураций для интересующего нас набора задач? Этот вопрос рассматривается в разделе 8.3.
2. Какие части пространства конфигураций важны больше, а какие меньше? Этот вопрос рассматривается в разделе 8.4.
3. Можем ли мы уменьшить пространство конфигураций, чтобы сделать метаобучение более эффективным? Этот вопрос рассматривается в разделе 8.5.

С точки зрения методики выбора алгоритма эти вопросы связаны с пространством алгоритмов.

Для успешного обучения также важны некоторые аспекты пространства задач. Мы акцентируем внимание на следующих вопросах.

1. Какие наборы данных нам нужны, чтобы иметь возможность перенести знания на новые наборы данных? Этот вопрос рассматривается в разделе 8.7.
2. Нужны ли нам полные метаданные или достаточно неполных метаданных? Этот вопрос уже частично рассмотрен в главе 2 и более подробно рассматривается в разделе 8.8.

3. Какие эксперименты необходимо запланировать в первую очередь, чтобы получить адекватные метаданные? Этот вопрос рассматривается в разделе 8.9.

1.3.2. Автоматические методы для ансамблей и потоков

Объединение базовых учеников в ансамбли (глава 9)

Ансамбли классифицирующих и регрессионных моделей представляют собой важную область машинного обучения. Причина их популярности в том, что они достигают более высокой производительности по сравнению с одиночными моделями. Вот почему мы посвящаем этой теме отдельную главу. Мы начнем с изучения понятия ансамбля и представим обзор некоторых из его наиболее известных методов, включая среди прочего бэггинг, бустинг, стекинг и каскадное обобщение.

Метаобучение ансамблевыми методами (глава 10)

Растет число подходов, объединяющих методы метаобучения – в том смысле, в котором термин используется в этой книге, – в ансамблевые способы обучения[1] [1]. В главе 10 мы обсудим некоторые из этих подходов. Мы начнем с общего обзора, а затем подробно проанализируем их в отношении используемого ансамбля, способа метаобучения и, наконец, задействованных метаданных.

Мы покажем, что ансамблевое обучение предоставляет много возможностей для исследования метаобучения в контексте очень интересных проблем, а именно с точки зрения размера пространства конфигураций, определения областей компетенции моделей и зависимости между ними. Поскольку ансамблевые системы обучения структурно очень сложны, отдельной проблемой является применение метаобучения для понимания и объяснения их поведения.

Рекомендации по выбору алгоритма для потоковых данных (глава 11)

Анализ потоков данных в режиме реального времени является ключевой областью исследований интеллектуального анализа данных. Многие данные, собранные в реальном мире, на самом деле представляют собой поток, в котором наблюдения поступают одно за другим, и алгоритмы их обработки часто имеют ограничения по времени и памяти. Примерами таких данных

[1] В литературе по ансамблевому обучению термин *метаобучение* используется для обозначения определенных подходов к ансамблевому обучению (Chan and Stolfo, 1993) и имеет несколько иное значение, чем то, которое используется в этой книге.

являются колебания стоимости акций, значения показателей человеческого тела и другие измерения, поступающие с датчиков. Природа данных может меняться со временем, фактически делая устаревшими модели, которые мы создали раньше.

Эта проблема хорошо известна научному сообществу, и поэтому многие алгоритмы машинного обучения были адаптированы или специально разработаны для работы с потоками данных. Примерами потоковых алгоритмов являются *деревья Хёффдинга*, *онлайн-бустинг* и *усиленный бэггинг*. Исследователи предложили так называемые *детекторы изменений* (drift detector) – механизмы, определяющие, когда созданная модель больше не применима к данным. В этом случае мы снова сталкиваемся с проблемой выбора алгоритма, которую можно решить с помощью метаобучения.

В главе 11 мы обсудим три подхода к тому, как методы, описанные в этой книге, использовались для решения упомянутой проблемы. Во-первых, мы рассматриваем подходы метаобучения, которые делят потоки на отдельные части, вычисляют метапризнаки этих частей и используют метамодель для каждой части потока, чтобы выбрать, какой классификатор использовать. Во-вторых, мы обсуждаем ансамблевые методы, которые используют производительность моделей на последних данных, чтобы определить, какие члены ансамбля все еще актуальны. В некотором смысле эти методы гораздо проще применять на практике, поскольку они не опираются на основу метаобучения и неизменно превосходят подходы метаобучения. В-третьих, мы обсуждаем подходы, основанные на идее повторяемости. Действительно, разумно предположить некоторую сезонность в данных, и модели, которые устарели сейчас, могут снова стать актуальными в какой-то момент позже. В этой части главы описываются системы, работающие с такими данными. Наконец, глава завершается анализом нерешенных вопросов и направлений будущих исследований.

1.3.3. Перенос метамоделей между задачами (глава 12)

Многие исследователи придерживаются мнения, что обучение не следует рассматривать как изолированный процесс, который начинается с нуля при каждой новой задаче. Вместо этого алгоритм обучения должен обладать способностью использовать результаты предыдущих процессов обучения для решения новых задач. Эту область часто называют *переносом знаний между задачами* или просто *переносом* обучения[1]. Иногда в этом контексте используют немного неуклюжий термин *обучение обучению* (learning to learn).

Глава 12 посвящена переносу обучения, целью которого является улучшение обучения путем обнаружения, извлечения и использования определенной информации в разных задачах. Эту главу написали приглашенные авторы Рикардо Вилалта и Михаил Месхи с целью дополнить материал этой книги.

[1] Transfer learning, иногда используют кальку «трансферное обучение». – *Прим. перев.*

Авторы обсуждают различные подходы к переносу знаний между задачами, а именно репрезентативный и функциональный. Термин *репрезентативный перенос* (representational transfer) используется для обозначения случаев, когда целевая и исходная модели обучаются в разное время и перенос происходит после того, как одна или несколько исходных моделей уже обучены. В этом случае имеется явная форма знаний, т. е. *осмысленные представления*, передаваемые непосредственно в целевую модель или в метамодель, которая извлекает нужную ей часть знаний, полученных в прошлых эпизодах обучения.

Термин *функциональный перенос* (functional transfer) используется для обозначения случаев, когда две или более моделей обучаются одновременно. Эту ситуацию иногда называют *многозадачным* обучением (multi-task learning). В этом случае модели имеют (частично или даже полностью) общую внутреннюю структуру во время обучения. Подробнее об этом можно узнать в разделе 12.2.

Авторы обращаются к вопросу о том, что именно может быть передано между задачами, и различают перенос обучения на основе экземпляров, признаков и параметров (раздел 12.3). Перенос обучения на основе параметров описывает случай, когда параметры, найденные в исходной предметной области, можно использовать для инициализации поиска в целевой предметной области. Отметим, что этот вид стратегии также обсуждается в главе 6 (раздел 6.7).

Поскольку нейронные сети играют важную роль в области искусственного интеллекта (ИИ), отдельный раздел 12.3 посвящен проблеме передачи данных в нейронных сетях. Так, например, один из подходов предполагает перенос части сетевой структуры. В этом разделе также описывается *архитектура двойного цикла* (double loop architecture), в которой базовый ученик выполняет итерации по обучающему набору во внутреннем цикле, а метаученик выполняет итерации по различным задачам для изучения метапараметров во внешнем цикле. В этом разделе также рассмотрен перенос в ядерных методах и в параметрических байесовских моделях. В завершающем разделе 12.4 описаны теоретические основы.

1.3.4. Метаобучение глубоких нейронных сетей (глава 13)

Методы глубокого обучения в последнее время привлекают большое внимание из-за их успехов в различных областях применения, таких как распознавание изображений или речи. Поскольку процесс обучения, как правило, протекает медленно и требует большого количества данных, метаобучение может предложить решение этой проблемы. Метаобучение может помочь определить наилучшие настройки гиперпараметров, а также параметров, связанных, например, с весами нейросетевой модели.

Как уже отмечалось в главе 12, большинство методов метаобучения использует процесс обучения на двух уровнях. На *внутреннем уровне* системе

предъявляется новая задача, и она пытается усвоить понятия, связанные с этой задачей. Этой адаптации способствуют накопленные знания, которые агент извлек из других задач на *внешнем уровне*.

Основываясь на предыдущих работах, авторы делят эту область метаобучения на три группы: методы, основанные на метрике, модели и оптимизации. После введения обозначений и предоставления справочной информации в этой главе описываются ключевые методы каждой категории, а также определяются основные проблемы и нерешенные вопросы. Расширенная версия этого обзора также доступна отдельно от этой книги (Huisman et al., 2021).

1.3.5. Автоматизация обработки данных и проектирование сложных систем

Автоматизация науки о данных (глава 14)

Исследователи отмечают, что в науке о данных большая часть усилий обычно уходит на различные подготовительные этапы, предшествующие построению модели. Этап фактического построения модели обычно требует меньше усилий. Это побудило исследователей изучить автоматизацию подготовительных этапов и привело к созданию методики, известной под названием *CRISPDM* (Shearer, 2000).

Основными этапами этой методики являются изучение проблемы и постановка текущей задачи, получение данных, предварительная обработка и различные преобразования данных, построение модели, ее оценка и автоматическое формирование отчетов. Некоторые из этих этапов могут быть инкапсулированы в конвейер, поэтому цель фактически состоит в том, чтобы разработать конвейер с максимальной потенциальной производительностью.

Этап построения модели, включая оптимизацию гиперпараметров, обсуждается в главе 6. Более сложные модели в виде конвейеров обсуждаются в главе 7. Цель главы 14 состоит в том, чтобы сосредоточиться на других шагах, которые не рассматриваются в этих главах.

Область, связанная с определением текущей проблемы (задачи), включает в себя различные этапы. В разделе 14.1 мы утверждаем, что понимание проблемы экспертом предметной области должно быть преобразовано в описание, которое может обработать система метаобучения. Последующие шаги могут быть выполнены с помощью автоматизированных методов. В число этих шагов входит создание дескрипторов задач (например, ключевых слов), которые помогают определить тип задачи, область и цели. Это, в свою очередь, позволяет нам искать и извлекать специфичные для предметной области знания, подходящие для поставленной задачи. Этот вопрос обсуждается в разделе 14.2.

Автоматизация процесса получения данных может быть нетривиальной задачей, так как необходимо определить, существуют ли уже данные или нет. В последнем случае необходимо разработать план относительно того, как их получить. Иногда необходимо объединить данные из разных источников

(базы данных, OLAP-куб[1] и т. д.). В разделе 14.3 эти вопросы обсуждаются более подробно.

Область предварительной обработки и преобразования данных больше всего интересует сообщество *автоматизированной науки о данных* (automatized data science, AutoDS). Разработаны методы выбора экземпляров и/или устранения выбросов, дискретизации и различных других видов преобразований. Эту область иногда называют *очисткой данных* (data wrangling) или *первичной обработкой*. Первичным преобразованиям можно научиться, используя существующие методы машинного обучения (например, обучение посредством демонстрации). Более подробную информацию можно найти в разделе 14.3.

Еще одна важная область науки о данных рассматривает решения относительно подходящего уровня детализации, который будет использоваться в приложении. Как известно, данные можно суммировать с помощью соответствующих операций агрегирования, таких как операции понижения/повышения детализации (drill down/up) в заданном кубе OLAP. Категориальные данные также могут быть преобразованы путем введения новых признаков более высокого уровня. Этот процесс включает в себя определение правильного уровня детализации. В принципе, можно автоматизировать и этот этап, но здесь еще нужно проделать большую работу, прежде чем удастся предложить практические решения пользователям. Подробнее об этой задаче можно узнать в разделе 14.4.

Автоматизация проектирования сложных систем (глава 15)

В этой книге мы рассмотрели проблему автоматизации проектирования конвейеров и других задач обработки данных. Возникает вопрос, можно ли распространить методы на несколько более сложные задачи. Эти вопросы обсуждаются в главе 15, но в этой книге основное внимание уделяется символическим подходам.

Нам хорошо известно, что многие успешные приложения в настоящее время, особенно в зрении и *обработке естественного языка* (natural language processing, NLP), используют *глубокие нейронные сети* (deep neural network, DNN), *сверточные нейронные сети* (convolutional neural network, CNN) и *рекуррентные нейронные сети* (recurrent neural network, RNN). Несмотря на это, мы считаем, что символические подходы сохраняют свою актуальность. Мы думаем, что это так по следующим причинам:

- для правильной работы DNN обычно требуются большие обучающие данные. В некоторых областях доступно мало обучающих примеров (например, случаи редких заболеваний). Кроме того, всякий раз, когда источником примеров служит человек (как, например, в обработке данных, обсуждаемой в главе 14), нам нужно, чтобы система была способна вызвать правильное преобразование на основе небольшого

[1] OLAP – это аббревиатура от «online analytical processing», т. е. оперативный анализ данных. OLAP-куб – это многомерный массив, применяемый для хранения и визуализации OLAP-данных. – *Прим. перев.*

количества примеров. Многие системы в этой области используют символические представления (например, правила), поскольку в них легко включить фоновые знания, которые часто также имеют форму правил;

- похоже, что всякий раз, когда системам ИИ необходимо общаться с людьми, выгодно прибегать к символическим понятиям, которые могут быть легко переданы между человеком и системой;
- поскольку человеческое мышление включает в себя как символическую, так и субсимволическую части, можно предположить, что будущие системы ИИ также будут придерживаться этого принципа. По нашему мнению, две системы рассуждений будут сосуществовать в своего рода функциональном симбиозе. Действительно, одна из современных тенденций связана с так называемым *объяснимым ИИ*.

Структура этой главы следующая. В разделе 15.1 обсуждаются более сложные операторы, которые могут потребоваться при поиске решения более сложных задач. Сюда входят, например, условные операторы и операторы итерационной обработки.

В разделе 15.2 обсуждаются изменения степени детализации путем введения новых понятий. В этом разделе представлены различные подходы, изученные в прошлом, такие как конструктивная индукция, пропозиционализация, переформулировка правил и др. Мы обращаем внимание читателей на новые достижения в этой области, такие как построение признаков в DNN.

Есть задачи, для которых невозможно обучить модель за один подход; в таких случаях требуется разделение на подзадачи, составление плана изучения составных частей и соединение их вместе. Эта методология обсуждается в разделе 15.3. Некоторые задачи нуждаются в итеративном процессе обучения. Подробнее об этом можно узнать в разделе 15.4. Есть задачи, цели которых взаимозависимы. Одна такая задача анализируется в разделе 15.5.

1.4. Хранилища результатов экспериментов (часть III)

1.4.1. Хранилища метаданных (глава 16)

В этой книге мы обсуждаем преимущества использования знаний о прошлых наборах данных, классификаторах и экспериментах. По всему миру ежедневно проводятся тысячи экспериментов по машинному обучению, генерируя постоянный поток эмпирической информации о методах машинного обучения. Свободный доступ к деталям экспериментов очень важен, так как это позволяет воспроизвести эксперименты и проверить правильность выводов, а также использовать эти знания в дальнейшей работе. Открытый обмен результатами экспериментов способствует ускорению прогресса науки.

Эта глава начинается с обзора сетевых репозиториев, где исследователи могут обмениваться данными, кодом и экспериментами. В частности,

она описывает OpenML, онлайн-платформу для автоматического обмена и организации данных машинного обучения. OpenML содержит тысячи наборов данных и алгоритмов, а также миллионы результатов экспериментов с ними. В этой главе мы описываем идеологию открытого репозитория и его основные компоненты: наборы данных, задачи, потоки, настройки, прогоны и наборы тестов. OpenML имеет привязки API к различным языкам программирования, что упрощает пользователям взаимодействие с API на их родном языке. Одной из отличительных особенностей OpenML является интеграция в различные наборы инструментов машинного обучения, такие как Scikit-learn, Weka и mlR. Пользователи этих наборов инструментов могут автоматически загружать все доступные данные, что открывает перед ними двери огромного хранилища результатов экспериментов.

1.4.2. Обучение на метаданных в репозиториях (глава 17)

Наличие обширного и общедоступного набора экспериментов, собранных и организованных в структурированном виде, позволяет нам проводить различные виды экспериментальных исследований. Частично опираясь на предыдущую работу (Vanschoren et al., 2012), мы представляем три типа экспериментов, которые изучают метаданные OpenML для решения определенных задач: эксперименты с одним набором данных, с несколькими наборами данных и эксперименты, направленные на работу с характеристиками определенного набора данных или алгоритма.

Что касается экспериментов с одним набором данных, в разделе 17.1 показано, как можно использовать метаданные OpenML для простого сравнительного анализа и, в частности, для оценки влияния настроек конкретного гиперпараметра. Применительно к экспериментам с несколькими наборами данных в разделе 17.2 показано, как можно использовать метаданные OpenML для оценки преимуществ оптимизации гиперпараметров, а также различий в прогнозах между алгоритмами. Наконец, для экспериментов с конкретными характеристиками в разделе 17.3 рассмотрено использование метаданных OpenML для исследования и ответа на определенные научные гипотезы, например для каких типов наборов данных подходят линейные модели и для каких типов наборов данных полезен выбор признаков. Кроме того, мы представляем исследования, целью которых является установление относительной важности гиперпараметров в наборах данных.

1.4.3. Заключительные замечания (глава 18)

В последней главе книги (глава 18) представлены заключительные замечания по отношению ко всей книге. Она состоит из двух разделов. Поскольку метазнание играет центральную роль во многих подходах, обсуждаемых в этой книге, здесь мы проанализируем этот вопрос более подробно. В частности,

нас волнует вопрос о том, какие метазнания используются в различных задачах метаобучения/AutoML, таких как выбор алгоритма, оптимизация гиперпараметров и генерация конвейера. Мы обращаем внимание читателей на то, что одни метазнания *извлекаются* (усваиваются) системами, а другие *назначаются* (например, разные аспекты заданного пространства конфигураций). Подробнее об этом можно узнать в разделе 18.1.

В разделе 18.2 обсуждаются задачи, которые еще предстоит решить, такие как лучшая интеграция подходов метаобучения и AutoML или понимания того, насколько детальное руководство доступно для задачи настройки систем метаобучения/AutoML на новые параметры. Эта задача включает (полу) автоматическое сокращение пространства конфигураций, чтобы сделать поиск более эффективным. В последней части этой главы обсуждаются различные проблемы, с которыми мы сталкиваемся, пытаясь автоматизировать различные этапы обработки данных.

1.5. Литература

Brazdil, P. and Konolige, K. (1990). *Machine Learning, Meta-Reasoning and Logics*. Kluwer Academic Publishers.

Chan, P. and Stolfo, S. (1993). *Toward parallel and distributed learning by metalearning*. In Working Notes of the AAAI-93 Workshop on Knowledge Discovery in Databases, pp. 227–240.

Guyon, I., Bennett,K.,Cawley,G.,Escalante,H.J.,Escalera,S.,Ho,T.K.,Macia,N., Ray, B., Saeed, M., Statnikov, A., et al. (2015). *Design of the 2015 ChaLearn AutoML challenge*. In 2015 International Joint Conference on Neural Networks (IJCNN), pp. 1–8. IEEE.

Huisman, M., van Rijn, J. N., and Plaat, A. (2021). *A survey of deep meta-learning*. Artificial Intelligence Review.

Kowalski, R. (1979). *Logic for Problem Solving*. North-Holland.

Lemke, C., Budka, M., and Gabrys, B. (2015). *Metalearning: a survey of trends and technologies*. Artificial Intelligence Review, 44(1):117–130.

Michie, D., Spiegelhalter, D. J., and Taylor, C. C. (1994). *Machine Learning, Neural and Statistical Classification*. Ellis Horwood.

Rendell, L., Seshu, R., and Tcheng, D. (1987). *More robust concept learning using dynamically-variable bias*. In Proceedings of the Fourth International Workshop on Machine Learning, pp. 66–78. Morgan Kaufmann Publishers, Inc.

Rice, J. R. (1976). *The algorithm selection problem*. Advances in Computers, 15:65–118. Shearer, C. (2000). The CRISP-DM model: the new blueprint for data mining. J Data Warehousing, 5:13–22.

Smith-Miles, K. A. (2008). *Cross-disciplinary perspectives on meta-learning for algorithm selection*. ACM Computing Surveys (CSUR), 41(1):6:1–6:25.

Vanschoren, J., Blockeel, H., Pfahringer, B., and Holmes, G. (2012). *Experiment databases: a new way to share, organize and learn from experiments*. Machine Learning, 87(2):127–158.