Потапов Алексей Сергеевич

Распознавание образов и машинное восприятие: Общий подход на основе принципа минимальной длины описания

Санкт-Петербург
2006

УДК 004.855
ББК 32.973.26
П64

Р е ц е н з е н т ы: член-корреспондент РАН, доктор технических наук, профессор М. М. Мирошников и кафедра компьютерной фотоники Санкт-Петербургского государственного университета информационных технологий, механики и оптики

Потапов А. С.

П64 Распознавание образов и машинное восприятие: Общий подход на основе принципа минимальной длины описания. — СПб.: Политехника, 2007. — 548 с.: ил.

ISBN 5-7325-0881-3

В книге подробно рассмотрен принцип минимальной длины описания, являющийся следствием теоретико-информационного подхода к построению моделей и выбору гипотез. Этот принцип становится все более популярным при решении сложных задач автоматического анализа данных, традиционно относившихся к области искусственного интеллекта. Рассмотрены задачи распознавания образов, машинного восприятия и грамматического и логического выводов, для которых использование принципа минимальной длины описания уже позволило получить более эффективные решения. На конкретных примерах показана возможность разработки унифицированного подхода к решению указанных задач.

Книга предназначена для широкого круга читателей: студентов, молодых ученых и специалистов, интересующихся компьютерными науками и, в частности, искусственным интеллектом.

УДК 004.855
ББК 32.973.26

ISBN 5-7325-0881-3

ОГЛАВЛЕНИЕ

ПРЕДИСЛОВИЕ

Задача поиска закономерностей в некотором наборе исходных данных возникает на всех уровнях работы мозга, начиная с сенсорного восприятия и заканчивая рассудочной деятельностью. В науке также существует проблема анализа эмпирических данных, которая является одной из центральных и обычно обозначается как проблема индуктивного вывода. Индуктивный вывод часто формулируется как проблема выбора модели, наилучшим образом описывающей или объясняющей имеющиеся данные. И здесь возникает вопрос: что же является критерием качества модели? Именно этот вопрос является центральным для всей книги.

При автоматизации различных сфер человеческой деятельности проблема индуктивного вывода оказывается одной из главных: построение машинной системы, которая бы проявляла свойства адаптивности и обучаемости, невозможно без ее решения. Осуществление прогнозирования также невозможно без анализа эмпирических данных.

В зависимости от природы исходных данных задача их анализа может превращаться в задачу анализа изображений, акустического сигнала или сигналов сенсоров других типов, лингвистического анализа, распознавания образов или машинного обучения вообще. Для каждой из соответствующих (а также многих других) предметных областей разработано множество методов автоматического анализа. Как правило, эти методы опираются на частные эвристики, работающие лишь при определенных ограничениях на исходные данные. Однако на практике эти ограничения могут не соблюдаться, что приводит к ухудшению результатов анализа. Наиболее явно эта проблема проявляется в использовании эвристических критериев качества модели. Такие критерии, хотя и выглядят вполне корректными, могут приводить к не вполне адекватному выбору модели. Чаще всего эта неадекватность проявляется в проблеме переобучения, или чрезмерно близкой подгонки, о которой будет подробно говориться в книге. В связи с этим для каждой предметной области существует необходимость разработки таких методов анализа, которые бы имели теоретическую основу и, в частности, использовали строго обоснованный критерий качества модели.

Хотя исходные данные могут очень сильно различаться в разных предметных областях, сама задача анализа дан-

ных сохраняется практически без изменений. Достаточно часто бывает так, что методы, разработанные для одного типа данных, оказывались применимыми для другого типа. Создание общей методики, которая бы формально описывала процесс разработки алгоритмов автоматического анализа данных независимо от их типа, является, безусловно, весьма актуальным.

В этой книге рассказывается о теоретико-информационном подходе к проблеме индуктивного вывода. Основные его идеи были сформулированы в 1960-х годах. Центральная концепция этого подхода выражается в принципе минимальной длины описания (МДО): среди всех моделей следует выбрать ту, которая позволяет описать данные наиболее коротко (с учетом длины описания самой модели), причем длины описаний определялись через алгоритмическую сложность. Этот подход давал строгое теоретическое разрешение проблемы критерия качества модели в индуктивном выводе, однако исходно не мог быть применен на практике из-за ряда трудностей. Предпосылки возникновения данного подхода, его теоретическая основа, а также трудности, связанные с практическим применением, описаны в гл. 1.

Принцип МДО получил широкое распространение при решении задач автоматического анализа данных только в 1990-х годах. В различных областях информатики этот принцип неоднократно переизобретался в более частных и конкретных формах, однако с сохранением общей идеи. Обширная литература, посвященная использованию принципа МДО, показывает его явные преимущества в получении более эффективных решений по сравнению с другими методами. При этом принцип МДО не отвергает существующие методы, а позволяет уточнить использующиеся в них критерии качества модели, что приводит к улучшению этих методов. В то же время далеко не всегда является очевидным, как именно следует применять этот принцип.

Автором предпринимается попытка обобщить существующий мировой опыт по применению принципа МДО в задачах анализа данных разных типов. Проблемам распознавания образов, регрессии и сегментации и их решению на основе принципа МДО посвящена гл. 2 книги. В гл. 3 описывается применение этого принципа для решения задач анализа изображений и речи. Проблемы теоретико-информационного подхода в задачах восстановления грамматик и деревьев решений рассматриваются в гл. 4. В книге при-

водятся некоторые рекомендации по построению автоматических методов анализа, которые являются обобщением опыта, накопленного исследователями в данной области.

Автор надеется, что этот материал даст читателю достаточное представление о сущности теоретико-информационного подхода и поможет успешно применить его на практике при решении задач из своей предметной области. Таким образом, книга может оказаться полезной для разработчиков прикладных систем автоматического анализа данных. В книге также затрагиваются теоретические вопросы индуктивного вывода, которые могут оказаться интересными научным работникам, специализирующимся на некоторых разделах искусственного интеллекта. Обсуждение возможности применения принципа МДО в распознавании образов, анализе изображений и речи, лингвистическом анализе и т. д. сопровождается достаточно подробным описанием классических методов, используемых в этих областях.

Автор выражает благодарность С. А. Родионову, Н. И. Потаповой, В. А. Ляховецкому и И. Г. Гарипову за то, что они взяли на себя труд ознакомиться с рукописью и высказали ряд конструктивных замечаний. Автор признателен А. Плахову, с которым планировалось совместное написание книги, чему, к сожалению, помешали жизненные обстоятельства. Автор также благодарен коллективу лаборатории автоматических методов обработки изображений Государственного Оптического Института им. С. И. Вавилова, опыт работы в котором сыграл существенную роль при написании данной книги.

Все замечания и пожелания автор просит направлять по адресу: 191023, Санкт-Петербург, Инженерная ул., 6, издательство «Политехника» или на электронный адрес автора: apotapov@mail.wplus.net

Глава 1

ИНДУКТИВНЫЙ ВЫВОД

1.1. ПРОБЛЕМА ВЫБОРА ГИПОТЕЗ В ИНДУКТИВНОМ ВЫВОДЕ

1.1.1. Что такое индуктивный вывод? Неформальное рассмотрение

Все рациональные рассуждения традиционно делятся на дедуктивные и индуктивные [1, с. 141]. Принято считать, что индукция — это умозаключение от частных фактов к некоторому общему гипотетическому утверждению, в то время как дедукция — это способ рассуждения, при котором осуществляется переход от общего знания или фактов к частным следствиям. Однако индуктивному выводу придается и более широкий смысл, если рассмотрение не ограничивается формальной логикой. Наиболее широко индуктивный вывод можно определить как проблему выбора модели из некоторого множества моделей, которая наилучшим образом «объясняет» исходные данные [2, с. 1]. Здесь под частными фактами понимается набор данных, а под общим утверждением — модель, описывающая эти данные (содержащиеся в них закономерности).

Это означает, что индуктивным выводом будет являться и проведение некоторой интерполяционной кривой по заданному набору точек, и составление словесного описания изображения. Более того, многие виды реальных рассуждений, традиционно относимых к дедуктивным, могут также быть причислены и к индуктивным. Так, классический пример дедуктивного рассуждения [1, с. 143]: «Все люди смертны. Сократ — человек, следовательно, Сократ смертен» опирается на две посылки, по крайней мере, первая из которых («Все люди смертны») не является с необходимостью истинной, а является результатом обобщения данных опыта. Тогда и консеквенту (заключению) этого вывода («Сократ смертен») можно присвоить лишь некоторую вероятность, отличную от единицы, а значит, в этом выводе производится выбор более достоверной гипотезы из двух возможных. Именно недостоверность результата и является признаком индуктивного вывода, отличающим его от де-

дуктивного вывода, в котором следствие с необходимостью получается из посылок и истинность посылок переносится на следствие.

Индуктивные рассуждения являются неотъемлемой частью естествознания, а недостоверность индуктивного вывода тесно связана с проблемой обоснования научного знания, которая наиболее отчетливо проявилась в философии Нового времени. В связи с этим изучение индукции изначально проводилось в философии науки. Попытки разработать адекватную логическую теорию индуктивного вывода (или индуктивную логику) проводились со времен Фрэнсиса Бэкона. Однако классическое понимание индукции как простого обобщения эмпирических фактов (результатов наблюдений, физических измерений или экспериментов над объектами внешнего мира) приводит к непреодолимым трудностям. Еще Д. Беркли заметил, что на основе самого по себе индуктивного подхода идеализм, в частности субъективный, неопровержим, поскольку невозможно установить, над чем именно осуществляется исходное наблюдение [3, с. 360]. И действительно, невозможно отличить феномен, даваемый нам в виде каких-то ощущений, от феномена, который совпадает с самими этими ощущениями.

Более детально эта проблема была рассмотрена Дэвидом Юмом, который впервые подверг глубокому исследованию понятие причинности. Так, в феноменологическом эмпиризме Юма как причинно-следственная связь, так и общие понятия являются не более чем психологической привычкой к ассоциативному связыванию идей («копий» с первоначальных впечатлений). Связывание идей оказывается возможным лишь как результат деятельности мышления и не зависит от наличия объективного аналога итога такого связывания. Выявленные слабости существовавшего к тому времени понимания индуктивной логики вообще поставили под сомнение ее право называться логикой. В результате Юмом была в общем виде сформулирована следующая задача [3, с. 363]: дать строгое, точное и объективное обоснование и оправдание индуктивной логики. Эта задача до сих пор не решена.

Поскольку чисто эмпирический подход к индуктивной логике, т. е. ее рассмотрение как простое обобщение результатов наблюдений и экспериментов, потерпел неудачу, то стало ясно, что она должна базироваться на некотором более прочном фундаменте. Возможно, именно поэтому Кант пытался построить метафизическую теорию, чтобы найти для

строго научного познания сущности изучаемых феноменов априорные основания, лежащие вне сферы чувственного опыта [3, с. 364]. Проблемы, родственные кантовскому вопросу о том, как возможны априорные синтетические суждения, сейчас заново встают при разработке систем машинного обучения — области знаний, тесно связанной с индуктивным выводом и являющейся разделом искусственного интеллекта (часто также называемого «практической гносеологией»).

Другой родственной областью исследований является автоматизация научных исследований, в которой исследуются проблемы выдвижения и проверки гипотез (см., например, [4]). Эта область знаний лежит на стыке философии науки и искусственного интеллекта (как научной дисциплины), в котором проблема индуктивного вывода играет существенную роль. Поэтому часто вопрос «Может ли машина мыслить?» конкретизируется как «Может ли машина формулировать и проверять гипотезы?» [5, с. 11].

Именно в искусственном интеллекте проблема индуктивного вывода получила свое дальнейшее развитие, поскольку эта область предоставляет наиболее сложные и интересные задачи, с одной стороны, и требует явного задания правил вывода (многие из которых осуществляются человеком неосознанно) — с другой.

И наконец, еще одно направление исследований, непосредственно связанное с индуктивным выводом, — статистический анализ, в котором также производится построение модели данных. Таким образом, есть несколько проблем, близких по содержанию к индуктивному выводу, — это машинное обучение, автоматическое выдвижение научных гипотез и статистический анализ данных. Все они могут быть охарактеризованы как вычислительный индуктивный вывод, который и будет в центре нашего внимания. Поскольку указанные три направления относятся к разным областям науки и философии, используемая в них терминология несколько различается. В связи с этим основные понятия требуют определенного уточнения.

1.1.2. Основные понятия индуктивного вывода

Рассмотрим некоторые важнейшие понятия индуктивного вывода, которые будут являться центральными для всего дальнейшего изложения. При этом будем учитывать, что

для каждого из этих понятий существует набор эквивалентных терминов, свойственных различным областям знаний.

• Исходная информация, на основе которой осуществляется вывод, может обозначаться такими терминами, как «частные факты», «набор исходных данных» («данные наблюдений»), «выборка измеренных значений случайной величины», «реализация случайного процесса». Здесь будет использоваться в основном термин «данные», а термин «выборка» будет фигурировать преимущественно в контексте рассмотрения статистических методов. Исходный набор данных в дальнейшем будет обозначаться символом D.

• Для обозначения результатов индуктивного вывода могут использоваться следующие понятия: «гипотеза» (реже — «теория»), «общее правило», «модель» и «оцененные параметры». Термины «гипотеза» и «модель» мы будем употреблять наравне, привнося в них в качестве отличия лишь слабый смысловой оттенок следующего содержания. В определенных контекстах под гипотезой будет пониматься атомарный объект, принадлежащий некоторому множеству таких же неделимых объектов, а под моделью — объект, обладающий (возможно, сложной) внутренней структурой. Таким образом, в то время как гипотезы выбираются, модели могут конструироваться. Некоторая гипотеза будет обозначаться с помощью символа h, если сущность гипотезы не уточняется (например, если не говорится, что гипотезой является программа для машины Тьюринга или класс образов).

• Все возможные результаты вывода объединяются в пространство (или множество) гипотез либо образуют класс моделей. Наряду с этими терминами может также использоваться и такое понятие, как «язык представления» (или просто «представление»). Это понятие особенно удобно использовать, когда модель, описывающая исходные данные, задается в виде цепочки символов. Изредка может использоваться также и такой термин, как «метамодель», который указывает, что само пространство гипотез (или язык представления) может варьироваться, являясь результатом индуктивного вывода следующего уровня, имеющего дело с целой предметной областью. Пространство гипотез будет обозначаться с помощью символа H.

• Одним из наиболее важных элементов индуктивного вывода является критерий, с помощью которого производится сравнение альтернативных гипотез. Он также называ-

ется критерием рациональности выводов и может уточняться либо как точность предсказания, даваемая моделью, либо как близость данной модели к «истинной» модели. На возможных вариантах задания этого критерия мы чуть подробнее остановимся ниже. Качество некоторой гипотезы h при условии, что есть исходные данные D, будет обозначаться как $r(h \mid D)$.

• Для наименования самого процесса вывода могут использоваться различные термины, конкретизирующие привлекаемый метод, например «статистический вывод» или, еще более узко, «байесовский вывод». Более общие названия этого процесса: индуктивный вывод, оценивание параметров, выбор или поиск модели. Выбор лучшей гипотезы можно описать следующим образом:

$$h^* = \arg\max_{h \in \mathrm{H}} r(h \mid D). \qquad (1.1)$$

Здесь не указывается, к каким именно предметным областям относятся те или иные термины. Такую информацию можно найти, например, в работе [2, табл. 1.1–1.3].

Помимо терминологических расхождений в различных подобластях индуктивного вывода дополнительная путаница может возникать из-за существования определенных отличий индуктивного вывода от близких проблем анализа данных, в которых ставятся другие цели, такие как предсказание или принятие решений (о вопросе разделения индуктивного вывода и теории принятия решений см., например, [1, гл. 13–14]). Для нашего изложения эти различия зачастую будут несущественными, но при необходимости мы будем на них указывать.

1.1.3. Критерии сравнения гипотез

Сформулировав задачу индуктивного вывода как выбор из некоторого множества модели, наилучшим образом объясняющей исходные данные, приходим к первичной проблеме, заключающейся в установлении приемлемого критерия для выбора лучшей модели. Нахождение такого критерия — это центральный вопрос, общий для таких областей, как статистический анализ, машинное обучение и философия науки [2, с. 3]. Отметим, что здесь идет речь именно об универсальном критерии, который можно было бы

использовать при решении любой задачи, представляемой в виде индуктивного вывода. Частные же критерии, которые придумываются человеком для решения конкретных задач, обычно оказываются неприменимы в новых задачах, поэтому для решения последних требуется творческое участие человека.

Было бы естественно предположить, что лучшая модель — это та, которая наиболее близка к истинной модели. Но тогда нужно было бы не только иметь возможность задать метрику в пространстве моделей, но и заранее знать истинную модель, а это доступно лишь в исключительных случаях. В некоторых задачах статистического анализа вводятся частные критерии близости данной модели к истинной, такие, как, например, среднеквадратичное отклонение. Однако подобные критерии, хотя и могут казаться интуитивно очевидными, перестают давать адекватный результат как только нарушаются заложенные в них (явные или неявные) априорные предположения. Такие примеры мы еще подробно разберем.

Принципиально другой подход к обсуждаемой проблеме заключается в выборе той модели, которая дает наибольшую точность предсказания. Классическим приемом для получения объективной оценки точности предсказания является разделение выборки на обучающую и тестовую части. Существуют также различные методы перекрестной проверки. Однако, во-первых, использование тестовой выборки приводит к уменьшению объема данных, по которым строится модель, а значит, понижается и точность модели. Во-вторых, не во всех задачах индуктивного вывода можно численно выразить точность предсказания. Поэтому ее желательно оценивать косвенно, привлекая некий другой критерий.

В философии науки используются такие критерии, как простота гипотезы (часто, особенно в зарубежной литературе, этот критерий связывается с принципом бритвы Оккама) и ее фальсифицируемость (отождествляемая с содержательной емкостью) [1, с. 231]. Принцип фальсифицируемости, введенный К. Р. Поппером, гласит, что выбирать нужно ту гипотезу, которая раньше других опровергалась бы новыми данными, полученными в результате наблюдений или эксперимента, если была бы ложной. Возможно, что понятия простоты и фальсифицируемости по смыслу достаточно близки [1, с. 233]. К сожалению, эти критерии ос-

таются бесполезными для вычислительного индуктивного вывода до тех пор, пока не являются вполне формализованными.

Именно понятие простоты используется в байесовских методах для определения априорных вероятностей моделей [5, с. 715]. В этих методах (как и в ряде других статистических методов) лучшей считается наиболее вероятная модель. Вообще, понятие вероятности неразрывно связано с индуктивным выводом: «...не только при анализе статистических выводов, но и при обсуждении, на первый взгляд, чисто качественных проблем индукции исчисление вероятностей играет центральную роль. Более того, хотя статистические выводы можно считать всего лишь частными и нетипичными образцами индуктивных выводов, нельзя сколько-нибудь обоснованно отказать им в принадлежности к области индуктивной логики» [1, с. 6]. А поскольку за байесовским выводом закрепилась репутация оптимального вывода, то следующая глава будет посвящена его рассмотрению.

Проблемой, сопутствующей установлению критерия рациональности гипотез, является выбор пространства гипотез, размер которого может заметно варьироваться в зависимости от задачи. Так, в статистическом выводе могут рассматриваться однопараметрические классы моделей, а могут — и гораздо большей размерности. Но в целом статистический анализ характеризуется наиболее ограниченными пространствами гипотез. В машинном обучении существуют проблемы, тесно примыкающие к статистическому выводу и также вовлекающие пространства гипотез «обозримого» размера. Наименее ограниченные пространства гипотез рассматриваются, пожалуй, в индуктивном выводе, изучаемом философией (что хорошо видно по возникающим здесь парадоксам, на которых мы еще остановимся позднее), и при разработке универсальных систем машинного обучения.

Ограничение, накладываемое на пространство гипотез, можно трактовать как априорно принятое решение об отказе проводить сравнение качества всех гипотез, не вошедших в выбранное пространство. Таким образом, задание пространства гипотез и определение критерия их сравнения — это разные стороны одной и той же проблемы, что более явно будет показано позднее. Поэтому неудивительно, что наибольшие сложности в установлении приемлемого критерия сравнения моделей возникают именно в фило-

софии науки и при разработке «сильного» искусственного интеллекта. Хотя эти теоретические сложности и находят свое отражение в конкретных практических проблемах, но последние менее ярко выражены. Принцип бритвы Оккама, привлекаемый в первой из этих областей, нашел во второй области свое формальное численное воплощение, которое позволяет разрешить эти сложности, по крайней мере частично. Оно и будет составлять основной предмет данной книги.

1.1.4. Бритва Оккама и принцип минимальной длины описания

Простота гипотезы — это один из наиболее часто применяемых критериев в индуктивном выводе (см., например [1, гл. 12]). Однако сама по себе простота гипотезы не может являться критерием при выборе модели, поскольку самая простая гипотеза — это просто отсутствие какой-либо регулярной модели, выявляющей внутренние закономерности в данных. Так, на любой наблюдаемый факт мы можем сказать: «Такова божья воля». Другими словами, простейшая гипотеза гласит, что данные абсолютно случайны, что, естественно, допускает и произвольную экстраполяцию, а значит, никак не может помочь в прогнозировании.

С другой стороны, точность, с которой модель описывает данные, тоже не является подходящим самостоятельным критерием. И действительно, существуют так называемые *гипотезы ad hoc*, которые просто повторяют имеющиеся данные, производят описание данных без их объяснения. Подобные гипотезы ad hoc также не могут помочь в прогнозировании. При этом они абсолютно точно описывают данные, но обладают значительной по сравнению с простейшей гипотезой сложностью.

В связи с этим принято считать, что лучшая гипотеза, дающая наибольшую точность предсказания, — это компромисс между простотой гипотезы и тем, насколько хорошо она удовлетворяет данным наблюдений [5, с. 715]. В философии науки такое положение часто связывается с принципом бритвы Оккама [2, с. 10]. Этот принцип гласит: «То, что можно объяснить посредством меньшего, не следует выражать посредством большего» (Frustra fit per plura quod potest fieri per pauciora), или «Без необходимости не следу-

ет утверждать многое» (Pluralitas non est ponenda sine necessitate). Чаще приводится другая формулировка: «Сущностей не следует умножать без необходимости» (Entia non sunt multiplicanda sine necessitate), но она, по-видимому, в произведениях Уильяма Оккама не встречается.

Как уже было замечено, в байесовских методах простота гипотезы используется для задания априорных вероятностей гипотез, и такие методы также называют формализацией бритвы Оккама [5, с. 715; 6, п. 1.1]. Однако более интересным вариантом формализации понятия простоты, с нашей точки зрения, оказываются подходы, основанные на теории информации. Здесь сложность (как противоположность простоты) получает конкретное измерение числом бит. Для корректного вычисления количества информации оказывается необходимым привлекать алгоритмическую теорию информации.

В машинном обучении (и в вычислительном индуктивном выводе вообще) такой подход был воплощен в нескольких концепциях. Среди этих концепций присутствуют такие, как *алгоритмическая вероятность* (АЛВ; Algorithmic Probability, ALP), разработанная Р. Соломоновым в 1964 г. [7]; принцип *минимальной длины сообщения* (МДС; Minimum Message Length, MML), предложенный К. Уалласом и Д. Болтоном в 1968 г. [8] (более поздние разработки по МДС можно найти, например, в работах [9, 10]), принцип *минимальной длины описания* (МДО; Minimum Description Length, MDL), описанный в 1978 г. Ж. Риссаненом [11] и несколько пересмотренный им позднее [12]; концепция идеальной МДО (Ideal MDL), предложенная М. Ли и П. Витани в 1989 г. [13] (см. также [14]), и *принцип аппроксимации сложности* (ПАС; Complexity Approximation Principle, CAP), введенный в работе [15]. Чуть ли не каждый из этих методов разными авторами связывается с формализацией бритвы Оккама [16, 17, 18, с. 10]

Хотя эти принципы несколько отличаются в деталях, основная идея у них одна, и ее можно сформулировать следующим образом (см., например, [19]). *Среди множества моделей следует выбрать ту, которая позволяет минимизировать сумму:* 1) длины описания модели (в битах); 2) длины данных, описанных посредством этой модели (в битах).

Общее правило, сформулированное таким образом, мы будем называть принципом минимальной длины описания (МДО). В случаях, когда речь будет идти о формальной те-

ории Риссанена, название которой и было выбрано для обозначения общего принципа, это будет оговариваться особо, чтобы не вызывать терминологической путаницы.

Длина описания модели определяет ее сложность, а длина описания данных с использованием модели — то, насколько хорошо модель удовлетворяет данным. Таким образом, компромисс между простотой и точностью модели приобретает объективный численный характер.

Прежде чем мы перейдем к более строгому рассмотрению вопроса сравнения гипотез с целью обоснования принципа МДО, а также к демонстрации различных его приложений, рассмотрим неформальное обоснование критерия простоты.

1.1.5. Бритва Оккама в научной эстетике и биологических системах

На удивление, два таких, казалось бы, разных вопроса, как «Что должно служить критерием истины?» и «Что такое красота?», оказываются тесно связанными через понятие простоты. Как уже отмечалось, простоту как критерий истинности связывают с принципом бритвы Оккама. Но и в эстетике (по крайней мере, научной, хотя есть основания думать, что не только в ней — см., например, работу [20] о применении понятия простоты в искусстве) критерий простоты находит свое применение. Так, всемирно известный математик Джордж Дейвид Биркгоф в 30-х годах XX века ввел уравнение красоты (см. [21–23], а также ряд других его работ, их переиздания и сборники) как

$$B = O/C,$$

где O — присутствующий в некотором объекте порядок (order); C — его сложность (complexity).

Примерно в это же время вышла в свет книга искусствоведа и драматурга В. М. Волькенштейна «Опыт современной эстетики», в которой, в частности, обосновывается идея того, что некоторый научный результат эстетичен, если с его помощью осуществляется сведение видимой сложности явлений к лежащей в их основе простоте. При этом также указывается, что, по замечаниям великих ученых, такое низведение сопряжено с сильными эстетическими переживаниями. И именно красота теории (естественно, при ее

согласованности с наблюдательными данными) часто служит критерием ее истинности, а слова «красивый», «прекрасный» нередко встречаются в трудах таких ученых, как Эйнштейн, Дарвин, Дирак, Больцман и многих других. Некоторые же из них, например Эйнштейн и Дирак, открыто говорили, что эстетический критерий является одним из важнейших в научном творчестве.

Несложно проследить, что выбор наиболее красивой теории в науке обычно оказывался и более успешным. Так, в астрономии развитие представлений о законах движения планет по небесной сфере, в частности переход от сложной системы эпициклов Птолемея к простой гелиоцентрической системе Коперника, можно охарактеризовать как выбор более красивых моделей. В химии таблица Д. И. Менделеева, безусловно, явилась красивым объяснением многообразия химических элементов. То же можно сказать и о дарвиновских законах развития в живой природе, и об открытии ДНК. В то же время все эти теории не только красивы, но и просты. Простота как значимый критерий красоты отмечается и в более поздних работах, например [24; 25].

Таким образом, успешность красивых теорий в науке может служить неформальным обоснованием критерия простоты, коль скоро простота и красота связаны. В связи с этим такие избитые выражения, как «краткость — сестра таланта» или «все гениальное просто», переосмысливаются и приобретают гораздо более глубокое значение.

Естественно, пока не существовало строгого определения понятия сложности, идеи Биркгофа и Волькенштейна (и, вероятно, многих других, здесь не упомянутых, людей) были не более чем уделом рефлексирующих ученых. Но в результате развития понятия алгоритмической сложности эти идеи нашли новый отклик [20, 26, 27], появилась также возможность их непосредственного применения в науке (см., например, [28, 29]), особенно в термодинамике и теории хаоса [30–32]. Интересно, что в рамках алгоритмической сложности понятие красоты естественным образом оказывается субъективным (основанным на личном опыте и на человеческой природе) [33].

Мы не можем более детально останавливаться на вопросе красоты в науке, так как не занимаемся здесь рассмотрением проблем ни научной методологии, ни эстетики, но не упомянуть о существовании тесной связи с принципом МДО было нельзя.

Другим источником неформального обоснования принципа минимальной длины описания являются данные, полученные в результате исследования принципов функционирования естественных нейронных сетей. Естественно, невозможно достаточно обоснованно утверждать, что работа человеческого мозга происходит согласно принципу МДО — об устройстве мозга еще слишком мало известно.

Тем не менее встречаются высказывания, согласно которым «очевидно, что биологические нейронные сети решили проблему бритвы Оккама» [6, с. 6], а сам принцип МДО используется в качестве критерия при выборе модели когнитивных процессов [34]. Это вызвано обоснованностью принципа МДО как подходящего критерия в индуктивном выводе, а обработка ощущений животными и человеком как раз и является индуктивным выводом, поэтому эта обработка с необходимостью должна следовать принципу МДО, чтобы быть корректной. Однако сейчас мы рассматриваем как раз обратный вопрос: существуют ли данные, подтверждающие, что естественные нейронные сети действительно следуют этому принципу?

Хотя мы не можем сделать такое заключение обо всем мозге в целом, есть данные, показывающие, что это верно, по крайней мере, для некоторых его подсистем. Эти результаты относятся преимущественно к системам первичной обработки сенсорной информации. Так, согласно Х. Барлоу [35–38], Д. Филду [39, 40] и ряду других авторов [41–44], важнейшей характеристикой обработки сенсорной информации в мозге является уменьшение ее избыточности или, иными словами, сжатие, что подтверждается различными исследованиями. Естественно, сжатие является не самоцелью, а результатом выделения из входного потока статистически независимых компонентов, порожденных различными источниками. Было бы интересно провести интерпретацию некоторых нейрофизиологических данных на основе принципа МДО, но, к сожалению, и этот вопрос выходит за рамки данной книги.

Помимо нейрофизиологических данных, подтверждающих сжатие информации на уровне отдельных групп нейронов, существуют и психологические исследования, показывающие значимость количественных информационных показателей для когнитивных процессов. Например, в работе [45] устанавливается тот факт, что скорость изучения человеком нового понятия строго зависит от алгоритмиче-

ской сложности этого понятия. К сожалению, как указано в работе [45], идея привлечения принципа МДО при исследовании человеческих когнитивных способностей лишь недавно попала в поле зрения ученых (см., например, [46–49]).

Таким образом, стремление к простоте с минимальной потерей информативности проявляется на разных уровнях: начиная от функционирования отдельных нейронов и заканчивая наукой в целом. Есть основания думать, что действие принципа МДО прослеживается и в других процессах. Например, ДНК, будучи своего рода моделью среды обитания, хотя и не может рассматриваться как минимальная программа, но определенная взаимосвязь между геномом и принципом минимальной длины описания в некоторых экспериментах прослеживается [50, 51]. Все это служит весьма сильным свидетельством в пользу данного принципа и заставляет искать причины такой универсальности. Далее в этой части книги мы попытаемся проследить логическую историю этого поиска, начиная с теоремы Байеса, а также обозначить все еще не решенные проблемы.

1.2. БАЙЕСОВСКИЕ МЕТОДЫ В ИНДУКТИВНОМ ВЫВОДЕ И МАШИННОМ ОБУЧЕНИИ

1.2.1. Теорема Байеса для выбора модели

Введем для начала некоторые определения, которые понадобятся нам для дальнейшего изложения.

Через выражение $\Pr(S)$ обозначим *вероятность* наступления некоторого события S в результате проведения испытания. В качестве такого события может выступать, например, «выпадение “решки”», а в качестве испытания — подбрасывание монетки.

Пусть задана *случайная величина* X, которая может принимать значения из некоторого множества $\mathrm{X} = \{x_1, x_2, \ldots, x_n, \ldots\}$. Например, X — это выпадающая в результате очередного подбрасывания сторона монетки; $\mathrm{X} = \{$«орел», «решка»$\}$. В результате единичного испытания случайная величина принимает одно и только одно значение.

Тогда *распределением вероятностей* случайной величины X называют отображение $P : \mathrm{X} \to [0, 1]$ такое, что $P(x_i) = \Pr(X = x_i)$, где выражение $\Pr(X = x_i)$ обозначает ве-

роятность реализации события, соответствующего принятию случайной величиной значения x_i в проведенном испытании. Чтобы подчеркнуть, что данное распределение относится к случайной величине X, пишут $P_X(x)$. Мы будем обычно опускать этот индекс, поскольку из контекста ясно, о какой именно случайной величине идет речь. Напомним, что для распределения вероятностей должно выполняться условие нормировки: $\sum_{x \in \mathrm{X}} P(x) = 1$.

Теперь пусть заданы две случайные величины: X и Y. Множество значений случайной величины Y обозначим через $\mathrm{Y} = \{y_1, y_2, \ldots, y_n, \ldots\}$. Тогда величина $P(x, y) = \Pr(X = x \,\&\, Y = y)$ задает *совместное распределение вероятностей*. Здесь величина $\Pr(S_1 \,\&\, S_2)$ означает вероятность одновременного (при проведении одного испытания) наступления событий S_1 и S_2, а условие нормировки принимает следующий вид:

$$(\forall x \in \mathrm{X}) \left(\sum_{y \in \mathrm{Y}} P(x, y) = P(x) \right). \qquad (1.2)$$

Еще одним используемым понятием будет *условная вероятность* $\Pr(S_1 \mid S_2)$, которая определяет вероятность наступления события S_1 при условии того, что наступило событие S_2. Тогда можно определить *условное распределение* $P(x \mid y) = \Pr(X = x \mid Y = y)$.

Для произвольных случайных величин выполняется следующее соотношение: $P(x, y) = P(x \mid y)P(y)$. Часто это соотношение дается в качестве определения условной вероятности. Можно также заметить, что в случае $P(y) = 0$ вероятность $P(x \mid y)$ оказывается неопределенной. Если верно равенство $P(x \mid y) = P(x)$, то случайные величины называются *статистически независимыми*. Нетрудно убедиться, что для статистически независимых случайных величин выполняется также и равенство $P(x, y) = P(x)P(y)$.

Теперь несложно получить правило Байеса. Пусть у нас имеются две случайные величины — X и Y. Рассмотрим уравнение, определяющее вероятность того, что $X = x$ при условии, что $Y = y$: $P(x, y) = P(x \mid y)P(y)$. Аналогично можно записать: $P(x, y) = P(y \mid x)P(x)$, следовательно, $P(y \mid x)P(x) = P(x \mid y)P(y)$. Тогда при условии, что $P(y) \neq 0$, получаем *теорему* (правило) *Байеса*:

$$P(x \mid y) = P(x)\frac{P(y \mid x)}{P(y)}. \qquad (1.3)$$

Несмотря на тот факт, что правило Байеса — это просто переписанное определение условной вероятности и ничего более, именно его интерпретация и приложения имеют наиболее фундаментальный характер и вызывают очень резкие дебаты в течение последних двух веков [51].

Чтобы раскрыть смысл этой теоремы и связать ее с проблемой выбора гипотез, перепишем правило Байеса в новых обозначениях:

$$P(h_i \mid D) = \frac{P(h_i)P(D \mid h_i)}{P(D)},\ h_i \in \mathrm{H}, \qquad (1.4)$$

где h_i — i-я гипотеза из N альтернатив (из которых одна, и только одна, гипотеза верна); $\mathrm{H} = \{h_1, h_2, \ldots, h_N\}$ — пространство гипотез (в общем случае может быть бесконечным); D — данные наблюдений или свидетельство (см. п. 1.1.2). Тогда $P(h_i \mid D)$ — вероятность того, что гипотеза h_i верна при условии, что имеются данные D, т. е. это *апостериорная вероятность* гипотезы; $P(h_i)$ — *априорная вероятность* гипотезы; $P(D \mid h_i)$ — вероятность получить данные D при условии, что верна гипотеза h_i, т. е. эта величина описывает *правдоподобие* данных наблюдений D исходя из гипотезы h_i. Эти величины являются ключевыми для данной главы. Вероятность $P(D)$ обычно не вовлекается, поскольку она одинакова для всех гипотез.

Таким образом, при байесовском подходе к индуктивному выводу критерием качества гипотезы служит ее апостериорная вероятность $r(h \mid D) = P(h \mid D)$, для вычисления которой «надо знать лишь априорные вероятности всех конкурирующих с ней альтернатив (включая ее саму), при условии, что данное свидетельство совместимо с интересующей нас гипотезой (подсчет условной вероятности свидетельства при данной гипотезе $P(D \mid h_i)$ не вызывает затруднения)» [1, с. 33]. Проблемы машинного обучения или индуктивной логики при этом сводятся к вероятностному выводу.

Для пояснения рассмотрим пример, относящийся к проблеме классификации, которая заключается в отнесении оптимальным образом некоторого объекта, описанного набором признаков, к одному из нескольких классов.

Пусть на автоматизированном производстве выполняется обзор производственного помещения в целях обнаруже-

ния посторонних объектов в сфере действия робота. При этом требуется определить, может ли появившийся посторонний предмет повредить роботу или нет. Если может, то робот должен быть остановлен, в противном случае работу необходимо продолжить. Поскольку этот анализ должен выполняться автоматически и невозможно заранее описать все возможные посторонние объекты, то отнесение объекта к классу опасных или безопасных выполняется на основе таких общих характеристик, как, например, размеры s, скорость перемещения v, высота над уровнем пола h, извлеченных из стереоизображений помещения. Значения этих характеристик и будут данными наблюдений $D = (s, v, h)$, а пространство гипотез будет H = {«опасный», «безопасный»}.

Величины P («опасный») и P («безопасный») — это априорные вероятности соответствующих гипотез, т. е. частота опасных и безопасных посторонних объектов, появляющихся внутри данного помещения. Вероятность $P(s, v, h \,||$«опасный») — доля всех опасных объектов, имеющих размеры s, скорость перемещения v и высоту h; аналогично $P(s, v, h \mid$ «опасный») — доля всех безопасных объектов с такими характеристиками. Эти условные вероятности определяют правдоподобие того, что объект с такой скоростью перемещения, размерами и положением над полом может быть причиной поломки робота в случае столкновения.

Оценки как априорных вероятностей, так и величин правдоподобия могут быть получены из обучающей выборки. Пусть, например, за время испытания системы обзора производственного помещения наблюдалось сто посторонних объектов, для которых были измерены характеристики s, v, h, а человеком была оценена степень опасности этих объектов. Тогда P («опасный») — это число наблюдавшихся опасных объектов, деленное на сто, а $P(s, v, h \,|$«опасный») — это число повстречавшихся опасных объектов со скоростью v, размером s и высотой h, отнесенное к общему числу повстречавшихся опасных объектов.

Теперь в процессе эксплуатации системы после обнаружения постороннего объекта и измерения его характеристик необходимо сравнить $P(s, v, h \,|$«опасный») $\times$ P («опасный») и $P(s, v, h \,|$«безопасный») $\times$ P («безопасный»), которые уже не представляет сложности вычислить. Чем больше появляется опасных объектов по сравнению с безопасными, тем

более вероятно, что данный объект является опасным, и наоборот. Чем более характерны значения скорости перемещения, размера и положения над полом для данного класса, тем более вероятно, что обнаруженный объект относится именно к нему. Далеко не всегда можно будет однозначно заключить, к какому классу относится объект, поскольку их возможные характеристики перекрываются (к примеру, объекты с одинаковыми размерами могут представлять разную угрозу в зависимости от своей массы, которую, однако, на основе изображения объекта оценить нельзя), но правило Байеса позволяет минимизировать среднее число ошибок.

Более подробно вопросы распознавания образов будут освещены в гл. 2 книги.

1.2.2. Принятие решений и предсказание на основе правила Байеса

Мы уже упоминали о возможных различиях между индуктивным выводом и такими проблемами, как предсказание и принятие решений. Здесь мы коснемся этого вопроса чуть подробнее применительно к правилу Байеса.

В распознавании образов выбор класса, к которому с наибольшей вероятностью принадлежит данный объект, часто рассматривается в качестве конечной цели. Однако такой подход в ряде случаев может приводить к неожиданным (на первый взгляд) результатам. Типичным примером [52, с. 83] является установление медицинского диагноза. В случае, если диагностику проходят люди, среди которых здоровых гораздо больше, чем больных, а значит, априорная вероятность того, что данный человек здоров, существенно выше априорной вероятности того, что он болен, то бóльшая часть больных будет классифицирована как здоровые.

При «неоптимальной» классификации долю неверно классифицированных больных можно заметно уменьшить, но при этом гораздо сильнее возрастет доля неверно классифицированных здоровых людей. А это и означает, что число ошибок классификации увеличится по сравнению с байесовским подходом, но, тем не менее, «неоптимальный» результат оказывается предпочтительнее. Означает ли это, что следует отказаться от правила Байеса? Нет. Выходом здесь является назначение величин потерь, связываемых с той или иной неверной классификацией. Если вернуть-

ся к примеру с обзором производственных помещений, то потери здесь будут иметь конкретное денежное выражение: стоимость ремонта робота в случае пропуска опасного объекта и потери от простоя производства в случае ложной тревоги. Таким образом, минимизироваться будет не число ошибок, допущенных в ходе классификации, а потери, вызванные этими ошибками.

Возвращаясь от проблемы распознавания образов к более общей задаче, устанавливаем, что каждой гипотезе h_i нужно присвоить некоторый вес w_i, определяющий ценность этой гипотезы. Тогда лучшую гипотезу нужно выбирать согласно величине $w_i P(h_i) P(D \mid h_i)$.

Однако может возникнуть и другая задача, в которой лучшую гипотезу выбирать не нужно, а необходимо сделать предсказание. Предположим, например, что каждой гипотезе h_i соответствует конкретное значение w_i некоторой величины W. Тогда оценку значения этой величины с учетом апостериорных вероятностей гипотез можно сделать следующим образом:

$$w_0 = \sum_{i=1}^{N} w_i P(h_i \mid D) = \sum_{i=1}^{N} w_i \frac{P(h_i) P(D \mid h_i)}{P(D)}. \qquad (1.5)$$

Однако такое рассмотрение имеет смысл только в том случае, если значения w_i не являются дискретными, например, если величины w_i — это некоторые потери, связанные с реализацией конкретной гипотезы (здесь то, какая именно гипотеза реализуется, не зависит от нашей воли, поэтому эта задача несколько отличается от проблемы принятия решения). В противном случае есть опасность получить такой ответ w_0 в качестве наиболее вероятного, как, например, два с половиной человека, если у нас есть две равновероятные гипотезы, которые утверждают, что ответы — два и три человека соответственно. Поэтому чаще с каждой гипотезой связывают некоторое распределение вероятностей $P(w_j \mid h_i)$, а саму величину W трактуют как случайную. Аналогично получаем:

$$P(w_j \mid D) = \sum_{i=1}^{N} P(w_j \mid h_i, D) P(h_i \mid D). \qquad (1.6)$$

Заметим, что здесь мы продолжаем считать, что $\sum_{i=1}^{N} P(h_i \mid D) = 1$, а точнее, что имеет место одна, и только одна,

гипотеза. Наряду с уравнением (1.4) уравнение (1.6) часто рассматривают как основу байесовского вывода. Уравнение (1.6) показывает, что лучшие предсказания — это средневзвешенные значения по предсказаниям отдельных гипотез. Таким образом, при байесовском подходе предсказание строится при использовании всех гипотез, вместо того, чтобы использовать единственную «лучшую» (апостериорно наиболее вероятную).

Уравнение (1.6) можно преобразовать к виду

$$P(w_j \mid D) = \sum_{i=1}^{N} P(w_j \mid h_i, D)P(h_i \mid D) = \sum_{i=1}^{N} \frac{P(w_j \mid h_i, D)P(h_i, D)}{P(D)} =$$

$$= \sum_{i=1}^{N} \frac{P(w_j, h_i, D)}{P(D)} = \frac{P(w_j, D)}{P(D)} = \frac{P(D \mid w_j)P(w_j)}{P(D)}. \qquad (1.7)$$

Видно, что при использовании уравнения (1.6) суть байесовского подхода не изменяется, а просто вводится дополнительный уровень вывода. Это оказывается полезным, когда вероятности $P(w_j \mid D)$ не могут быть вычислены напрямую из данных, и приходится привлекать какие-то модели. Например, мы не сможем предсказать наиболее вероятное положение некоторой планеты на небе по ее предыдущим положениям, если не рассмотрим модели ее движения. Более подробно на задаче предсказания мы останавливаться не будем, а перейдем к тем проблемам, которые являются общими для всех байесовских методов.

1.2.3. Методы максимума апостериорной вероятности и максимального правдоподобия

В реальных задачах пространство гипотез может быть очень большим или бесконечным. Примером такой задачи может служить интерполяция данного набора точек полиномами произвольной степени. Для таких случаев суммирование в уравнении (1.6) неосуществимо, поэтому оптимального предсказания получить не удается. Также нельзя выбрать и лучшую гипотезу в уравнении (1.4), если только не вводить дополнительных ограничений. В связи с этим возникает потребность использовать некоторые аппрокси-

мации и упрощения, позволяющие получить методы, реализуемые на практике.

Одной из таких аппроксимаций является метод нахождения *максимума апостериорной вероятности* (MAB; Maximum a posteriori, MAP). Упрощение, которое применяется в этом методе, заключается в том, что в целях предсказания мы рассматриваем лишь гипотезу h_{MAP}, обладающую максимальной апостериорной вероятностью (MAB-гипотезу), и приравниваем: $P(w \mid D) \approx P(w \mid h_{MAP})$. Чем больше доступно данных, тем MAB и байесовское предсказание становятся ближе, потому что гипотезы, конкурирующие с MAB-гипотезой, становятся все менее и менее вероятными, а апостериорная вероятность MAB-гипотезы, напротив, стремится к единице. Поэтому в случае больших выборок данных использование MAB-метода вполне оправданно. При этом нахождение MAB-гипотезы часто существенно проще, чем байесовское обучение, так как оно требует решения проблемы оптимизации (при этом удается избежать просмотра каждой гипотезы) вместо суммирования или интегрирования по большому пространству гипотез.

Но сложности могут возникнуть не только на этапе предсказания при использовании уравнения (1.6), а раньше — при подсчете самих апостериорных вероятностей гипотез. Основная сложность здесь возникает из-за того, что априорные вероятности гипотез могут быть неизвестны. Самым простым решением этой проблемы, окончательно упрощающим методы, основанные на правиле Байеса, является введение предположения о равенстве априорных вероятностей в данном пространстве гипотез, т. е. $\left(\forall h_i, h_j \in \text{H}\right)\left(P(h_i) = P(h_j)\right)$. В этом случае MAB-подход сводится к выбору гипотезы, которая максимизирует соответствующее ей значение правдоподобия:

$$h_{ML} = \arg\max_{h \in \text{H}} P(D \mid h). \qquad (1.8)$$

Данный метод называется *методом максимального правдоподобия* (МП; maximum likelihood, ML).

МП-метод широко используется при решении конкретных практических задач в различных областях человеческой деятельности, где требуется статистический анализ данных. Этот подход является вполне разумным, когда нет причин предпочесть одну гипотезу другой априори, т. е., когда пространство гипотез является действительно однород-

ным. Более того, если исходная выборка данных достаточно велика, то выбор априорных вероятностей (при условии, что априорная вероятность истинной гипотезы не была взята равной нулю) практически не будет влиять на выбор истинной гипотезы. Это связано с тем, что правдоподобие данных большого объема для ложных гипотез будет очень незначительным, что компенсирует грубый выбор априорных вероятностей. Таким образом, МП-метод при определенных ограничениях представляет хорошую аппроксимацию байесовского или MAB-подхода. Однако результаты его применения могут оказаться некорректными в случае выборок данных малых объемов или существенно неоднородного пространства гипотез.

Упрощения, применяемые в МП-методе, к сожалению, часто оказываются неправомерными. В общем случае проблема выбора априорных вероятностей оказывается краеугольным камнем как в индуктивном выводе, так и в машинном обучении, порождая ряд парадоксов в философии и существенные сложности при разработке сколько-нибудь общих систем машинного обучения.

1.2.4. Проблема априорных вероятностей

Одной из основных характеристик методов, основанных на теореме Байеса, принято считать то, что в них верная гипотеза в результате накопления данных наблюдений всегда рано или поздно начинает доминировать [5, с. 714]. Это обосновывается тем, что для любых фиксированных априорных вероятностей $P(h_i)$, для которых вероятность верной гипотезы отлична от нуля, апостериорные вероятности $P(D \mid h_i)$ ложных гипотез будут стремиться к нулю при неограниченном увеличении выборки данных.

На первый взгляд, данное мнение кажется абсолютно справедливым. И действительно, давайте рассмотрим следующий пример. Пусть есть такая особенная монетка, которая в 90 % случаев выпадает «орлом» и только в 10 % случаев — «решкой», но априорно вероятности выпадения «орла» и «решки» неизвестны. Обозначим через h_p такую гипотезу, которая гласит, что вероятность выпадения «орла» равна $p \in [0,1]$. Хотя таких гипотез бесконечно много, мы для простоты рассмотрим случай выбора из трех гипотез: $\mathrm{H} = \{h_{0,5}, h_{0,8}, h_{0,9}\}$. При этом присвоим следующие

априорные вероятности $P(h_{0,5}) = 0,9999;\ P(h_{0,8}) = 0,00008$ и $P(h_{0,9}) = 0,00002,$ не отказывая таким странным монеткам в праве на существование, но предполагая, что гораздо более характерным является равновероятное выпадение «орла» и «решки» (для данного примера несущественно то замечание, что в природе не существует «идеальных» монеток с абсолютно равными вероятностями выпадения обеих сторон).

Пусть данные D гласят, что было проведено N опытов, в которых «орел» выпал m раз. Несложно заметить, что для гипотезы h_p правдоподобие данных будет равно $P(D \mid h_p) = C_N^m p^m (1-p)^{N-m}$, где C_N^m — это число сочетаний из N по m элементов. Поскольку как C_N^m, так и $P(D)$ зависят только от N и m, но не зависят от рассматриваемой гипотезы, то мы их подсчитывать не будем и обозначим $c(D) = C_N^m / P(D)$.

Подсчитаем численные значения апостериорных вероятностей для следующих двух значений N и m.

1. $D: N = 30;\ m = 27$

$$P(h_{0,5} \mid D) = c(D) \cdot 0,9999 \cdot 0,5^3 \cdot 0,5^{27} \approx 9,3 \cdot 10^{-10} c(D);$$

$$P(h_{0,8} \mid D) = c(D) \cdot 8 \cdot 10^{-5} \cdot 0,2^3 \cdot 0,8^{27} \approx 1,5 \cdot 10^{-9} c(D);$$

$$P(h_{0,9} \mid D) = c(D) \cdot 2 \cdot 10^{-5} \cdot 0,1^3 \cdot 0,9^{27} \approx 1,2 \cdot 10^{-9} c(D).$$

Видно, что все гипотезы оказываются примерно равновероятными уже при тридцати испытаниях. Хотя истинная гипотеза еще не доминирует, но гипотеза о равновероятном исходе подбрасывания уже имеет меньшую апостериорную вероятность, несмотря на большую априорную вероятность. Теперь посмотрим, что будет при дальнейшем увеличении числа испытаний.

2. $D: N = 100;\ m = 90$

$$P(h_{0,5} \mid D) = c(D) \cdot 0,9999 \cdot 0,5^{10} \cdot 0,5^{90} \approx 7,9 \cdot 10^{-31} c(D);$$

$$P(h_{0,8} \mid D) = c(D) \cdot 8 \cdot 10^{-5} \cdot 0,2^{10} 0,8^{90} \approx 1,6 \cdot 10^{-20} c(D);$$

$$P(h_{0,9} \mid D) = c(D) \cdot 2 \cdot 10^{-5} \cdot 0,1^3 \cdot 0,9^{27} \approx 1,5 \cdot 10^{-19} c(D).$$

Гипотеза $h_{0,5}$ имеет уже исчезающе малую апостериорную вероятность по сравнению с двумя другими гипотезами, а ги-

потеза $h_{0,8}$ на порядок менее вероятна, чем $h_{0,9}$. Несложно понять, что будет при дальнейшем увеличении объема данных опытов. Несмотря на то, что отношение $m/N = 0{,}9$ в опытах будет нарушаться (это лишь наиболее вероятное значение), и несмотря на то, что априорная вероятность гипотезы $h_{0,9}$ может быть взята еще более низкой (но только не нулевой!), она рано или поздно начнет доминировать.

В случае, если гипотезе $h_{0,5}$ априорно назначена вероятность, равная единице, а остальным гипотезам — нулю, то эта гипотеза будет выбираться независимо от полученных данных. Тогда на вопрос «Какая сторона выпадет при следующем подбрасывании монетки?» — последует ответ: «С равной вероятностью выпадет “орел” или “решка”, поскольку последующее испытание статистически независимо от предыдущих», даже если при предыдущих ста испытаниях решка не выпала ни разу. В абстрактной математической задаче, в которой априорная вероятность задается условием этой задачи, такой результат не является парадоксальным. Однако он часто переносится на наблюдение реальных объектов. При этом оценка данных опыта подгоняется под гипотезу с априорной вероятностью, равной единице, а не наоборот — осуществляется выбор гипотезы на основе данных наблюдений. Аргументация здесь, например, может быть такой. Любая последовательность выпадений «орлов» и «решек» ничем не лучше любой другой, поэтому в последовательности из ста «орлов» нет ничего выделенного — ведь также можно было бы удивляться выпадению любой другой последовательности, имеющей ту же вероятность (в этом месте как раз неявно предполагается верность гипотезы $h_{0,5}$). Но интуитивно кажется, что эти вероятности не равны. Оставим этот вопрос до того момента, пока не рассмотрим понятие алгоритмической сложности бинарной строки, а сейчас вернемся к проблеме априорных вероятностей.

Казалось бы, рассмотренный пример подтверждает, что при увеличении объема данных верная гипотеза начинает доминировать. Это делает проблему априорных вероятностей надуманной, а байесовский критерий сравнения гипотез самодостаточным (не требующим задания априорных вероятностей каким-либо другим методом).

К сожалению, это оказывается неверным. Первое возражение, носящее, скорее, практический характер, заключается в том, что при неадекватных априорных вероятностях тре-

буемый объем исходных данных увеличивается, что делает байесовский метод неоптимальным: на тех же данных можно будет получить лучшую точность предсказания, задав «хорошие» априорные вероятности. По мере возрастания сложности задачи эта неоптимальность будет расти, что может сделать данный подход, игнорирующий проблему априорных вероятностей, неприемлемым. Но даже если отмахнуться от этого «практического» возражения, можно указать на сложности, носящие более фундаментальный характер и возникающие в задачах, в которых пространство гипотез чрезмерно избыточно.

Для пояснения рассмотрим следующий пример. Пусть набор точек $\{x_i, y_i\}_{i=1}^{N}$ — данные наблюдений, а пространство гипотез — полиномы произвольной степени. Для простоты рассмотрим детерминистический случай: координаты точек заданы без погрешности. В частности, это означает, что при условии, что выбран некоторый полином, по набору абсцисс однозначно восстанавливаются ординаты, а значит, правдоподобие может принимать только значения «1» и «0» (в случае, если полином проходит или не проходит через все указанные точки).

Если предположить, что все гипотезы равновероятны (точнее, если обратиться к методу максимального правдоподобия), то для любого (конечного) набора точек будет существовать бесконечно много гипотез, абсолютно точно удовлетворяющих данным, а значит, соответствующие им правдоподобия будут равны 1. Это говорит о том, что мы ни при каком конечном наборе данных не сможем выбрать правильную гипотезу. С другой стороны, задав какие-либо априорные вероятности, выбор единственной гипотезы возможен, но выбранная гипотеза будет зависеть от того, как именно заданы эти априорные вероятности. Это со всей отчетливостью показывает важность данной проблемы.

Чтобы ее решить, можно эвристически присвоить априорные вероятности гипотезам тем меньше, чем больше степень многочлена, соответствующего данной гипотезе. Посмотрим, что при этом получится. Пусть истинная гипотеза — это некоторый многочлен степени M. Как только число точек N становится больше степени M истинного полинома, то этот полином будет являться единственным полиномом, удовлетворяющим данному набору точек и имеющим степень, не превосходящую M. Все остальные подходящие гипотезы будут соответствовать многочленам степени, не

меньшей $N > M$, а значит, присвоенные этим гипотезам априорные вероятности будут меньше, чем у истинной гипотезы, которая и будет выбрана. В этом случае тезис о том, что при накоплении данных верная гипотеза будет доминировать, соответствует истине, хотя нам и потребовалось эвристически задать априорные вероятности, которые не всегда так очевидны, как в примере с многочленами: что, например, более вероятно — синус или тангенс?

К сожалению, и для этого решения сразу можно привести контрпример. Рассмотрим следующий набор точек: $\left\{\left(x_i, y_i = e^{x_i}\right)\right\}_{i=1}^{N}$. Истинной гипотезой для него является «многочлен» бесконечной степени, а именно ряд $y = \sum_{n=0}^{\infty} \frac{1}{n!} x^n$, поэтому ее априорная вероятность, заданная нами эвристически, будет равна нулю. Только в пределе удастся получить эту истинную гипотезу, что, естественно, на практике принципиально неосуществимо.

Чтобы показать, что речь не идет о получении некоторой «абсолютной истины», а о практической применимости байесовских методов, рассмотрим еще один набор точек $\left\{\left(x_i, y_i = 1/x_i\right)\right\}_{i=1}^{N}$, который также представим в виде степенного ряда. Пусть нас интересуют значения y для $x \in [-1, 1]$, не вошедших в выборку. Условие дифференцируемости $y(x)$ в нуле нарушается, так что эта функция не может быть представлена сходящимся степенным рядом. Об этом, однако, априори не известно. Полиномиальная аппроксимация не позволит получать предсказание новых значений $y(x)$ для такого набора данных. В расширенном пространстве моделей, содержащем функции вида $y(x) = x^{-n}$, по конечному набору точек можно добиться достаточно точного предсказания. Видно, что точность предсказания связана не столько с алгоритмом аппроксимации, сколько с языком представления (аналогичная ситуация имеет место и, например, для такой задачи, как изучение понятий: способность выучить понятие, в первую очередь, зависит от представимости этого понятия в выбранном пространстве моделей, и лишь затем — от алгоритмов обучения).

Все это говорит о том, что, априорно предположив, что истинной гипотезой является многочлен конечной степени, мы отбросили бесчисленное множество других гипотез.

Расширив же пространство гипотез функциями других типов, мы опять вернемся к обсуждаемой проблеме — как задавать априорные вероятности. И тем не менее, в истории науки человеком неоднократно решались аналогичные задачи; один из таких примеров мы уже упоминали — это переход от эпициклов Птолемея к уравнениям Кеплера.

Здесь же обратимся еще к одному примеру, являющемуся классическим философским парадоксом, возникающим в индуктивном выводе (в частности, при применении теоремы Байеса). Это парадокс «зелубых» изумрудов или аналогичных по смыслу предикатов, рассмотренных впервые Гудманом [53] и названных им «непредсказуемыми». Предикат «зелубой (x)» означает, что x является зеленым некоторое время вплоть до 3000 года (в оригинальном варианте фигурировала уже наступившая дата) и голубым некоторое время после наступления 3000 года. Все имеющиеся свидетельства о каком-либо конкретном изумруде одинаково хорошо подтверждают обе гипотезы: и то, что он является зеленым, и то, что он является зелубым. Этот пример показывает, что не все гипотетические модели данных обладают одинаковыми априорными вероятностями и что у нас нет адекватного способа присваивать бóльшие априорные вероятности одним гипотезам и меньшие — другим [33].

Интуитивно кажется, что гипотеза, гласящая, что изумруды являются зелеными, является проще, чем та, которая говорит о том, что они «зелубые». Связывая интуитивное понятие простоты и априорные вероятности гипотез, можно заключить следующее. Человек рождается с некоторыми врожденными априорными вероятностями (здесь вовсе не имеется в виду то, что они хранятся в мозгу в явном виде), на основе которых формирует расширенный набор априорных вероятностей как обобщение своего жизненного опыта. Поскольку субъективные априорные вероятности различны у разных людей, имеющих разный жизненный опыт, то это объясняет, почему на основе одних и тех же данных разные люди могут делать разные выводы. Врожденные же априорные распределения вероятностей были получены в ходе органической эволюции, так как лучше выживали те организмы, которые делали лучшие предсказания на основе присущих им априорных вероятностей. Это стиль рассуждений Хомского для объяснения того, откуда взялись наши субъективные распределения вероятностей, как пишет Р. Соломонов [17].

Однако ссылки на интуицию и здравый смысл являются неудовлетворительными при исследовании вычислительного индуктивного вывода (эти ссылки являются также неудовлетворительными и с точки зрения философии, см. [1, с. 257–262]), поскольку не говорят нам, как именно должны выглядеть эти вероятности. Существует еще ряд парадоксов индуктивного вывода, связанных с заданием априорных вероятностей (см., например, [1, 33, 51]), которые здесь разбираться не будут. Нам лишь важно было продемонстрировать тот факт, что необходимость выбора априорных вероятностей (особенно, если этот выбор необходимо производить автоматически) порождает глубокие теоретические трудности, которые необходимо решить для создания достаточно универсальной системы машинного обучения.

Однако эти трудности находят отражение и в конкретных практических задачах, вызывая так называемый эффект *переобучения* (или чрезмерно близкой подгонки). Для пояснения вернемся к задаче интерполяции заданного набора точек полиномами произвольной степени. Но пусть теперь точки задаются с некоторой погрешностью. Несложно заметить, что истинный многочлен будет давать некоторую ошибку при описании данного набора точек, в то время как будут существовать многочлены больших степеней, которым будет соответствовать меньшая ошибка. Для многочлена достаточно большой степени ошибка будет равна нулю. Таким образом, выбор многочлена лишь на основе того, насколько он хорошо удовлетворяет имеющимся данным, будет приводить к предпочтению неверных многочленов избыточной сложности. Аналогичный эффект переобучения обнаруживается и в искусственных нейронных сетях и в ряде других приложений. Возможность устранения этого эффекта принципиально необходима для решения многих практических задач, таких как распознавание речи или обнаружение и распознавание объектов по их изображениям в присутствии шумов.

Итак, переобучение, как правило, возникает в случаях, когда пространство гипотез слишком избыточно, т. е. для любого набора данных можно подобрать множество гипотез, точно (или достаточно хорошо) описывающих эти данные. Избежать переобучения можно в результате адекватного выбора априорных вероятностей, штрафующих «плохие» гипотезы. В качестве такого эвристического критерия для штрафа гипотезы часто выступает ее сложность, кото-

рая интуитивно определяется человеком (например, по числу параметров в гипотезе), что дает в результате субъективные априорные вероятности гипотез. Обычно приводят следующие неформальные аргументы в пользу критерия сложности (см., например [5, с. 715]): во-первых, число сложных гипотез больше, чем число простых гипотез; во-вторых, более сложные гипотезы имеют больше возможностей («степеней свободы»), чтобы удовлетворить данным. Хотя сами аргументы весьма разумны, из-за неформального характера их применение далеко не универсально и требует определенного вмешательства человека.

Подведем некоторый итог. Предсказание на основе правила Байеса принято считать оптимальным, является ли набор данных большим или маленьким. Существует «немало точек зрения, согласно которым теорема Байеса является главным инструментом научных или индуктивных выводов или выводов из данных опыта вообще» [1, с. 33], а также утверждений, что при данных априорных вероятностях любое другое предсказание будет реже являться верным [5, с. 713; 6, с. 3]. Приняв этот тезис как верный, заметим, однако, что байесовские методы нуждаются в задании извне априорных вероятностей гипотез, а в этом и заключается *основная* проблема индуктивного вывода и машинного обучения.

Процитируем Р. Соломонова: «При проведении индуктивного вывода имеется два типа информации: первый — это сами данные и второй — априорные данные — информация, которая имеется до наблюдения данных. Можно делать предсказания без данных, но нельзя осуществлять предсказания без априорной информации» [33].

И действительно, Байесовские методы действуют в сильно ограниченных пространствах гипотез; всем гипотезам, которые не вошли в это пространство, по сути, априори присвоена нулевая вероятность, а таких гипотез несравненно больше, чем оставшихся. Если бы не это, байесовские методы вообще не могли бы давать хоть немного разумные результаты, что хорошо видно из парадокса зелубых изумрудов и других приведенных примеров. Но и в этих ограниченных пространствах необходимо задавать априорные вероятности для оставшихся гипотез, чтобы избежать проблемы переобучения и получать качественное предсказание на малых объемах обучающих выборок данных.

Иными словами, Байесов подход не может рассматриваться в качестве самостоятельного метода (не может претен-

довать на роль «парадигмы»), а является лишь полезным инструментом и нуждается в построении некоторой метатеории. Привлечение понятия сложности гипотезы без его формализации оказывается тоже недостаточно надежным, хотя и помогает в частных случаях. Таким образом, можно считать, что первой задачей является именно формализация понятия сложности, а для этого естественно попробовать привлечь теорию информации и понятие сложности отождествить с количеством информации. Однако, как мы в дальнейшем увидим, классической (разработанной К. Шенноном) теории информации для этого оказывается недостаточно, и понятие сложности находит адекватную формализацию в алгоритмической теории информации, развитой А. Н. Колмогоровым.

1.3. ОСНОВНЫЕ ПОЛОЖЕНИЯ ТЕОРИИ ИНФОРМАЦИИ

1.3.1. Теория информации Шеннона: историческая справка

В предыдущей главе мы решили рассматривать сложность гипотезы как количество информации, содержащейся в ней. Поскольку исходные данные также несут некоторый объем информации, то интуитивно кажется, что более сложная гипотеза обладает большей неопределенностью, и на ее выбор «расходуется» больше информации, содержащейся в этих данных, чем на выбор более простой гипотезы. Сложность гипотезы становится той степенью неопределенности, измеренной в битах, которую нужно устранить для ее выбора, а значит, нам нужно обратиться к формальному определению количества информации. В связи с этим мы начнем рассмотрение с классической теории информации, которая базируется на теории вероятностей.

Момент возникновения теории информации связывают [54, с. 7] с публикацией в 1948 г. двух фундаментальных статей Клода Шеннона: «Математическая теория связи» и «Связь при наличии шума» [55], хотя существуют и более ранние работы [56, 57]. Изначально теория информации сформировалась из прикладных задач телеграфии и радиосвязи как ветвь статистической теории связи, основы которой были заложены классическими работами К. Шенно-

на, А. Н. Колмогорова, В. А. Котельникова и Н. Винера [54, с. 6]. Наиболее важная идея теории информации заключалась в том, что информация — это нечто, что может быть измерено количественно, и что количество информации тесно связано с ее вероятностью [17].

Основными вопросами, исследуемыми в «шенноновской» теории информации, являются вопросы построения методов оптимального кодирования сообщений в двух целях: для минимизации затрат на передачу сообщения в зависимости от их источника и для увеличения надежности передачи сообщений по каналам связи с шумом. Именно поэтому теория информации оказала столь большое влияние на создание эффективных и надежных систем коммуникации [54, 58], но она также воздействовала и на некоторые научные дисциплины, например, на статистику [59, 60] и статистическую механику [59, 61]. Сейчас основные теоретико-информационные методы и критерии находят применение и во многих других областях: прикладной математике, распознавании образов, искусственном интеллекте и др. Так что, несмотря на свое практическое происхождение, теория информации является глубокой математической теорией, связанной с самой сутью коммуникационного процесса [58].

Теория информации исходит из статистических моделей источников сообщений и каналов связи. Из статистических же соображений вводится количественная мера для информации (о способе задания количества информации без привлечения понятия вероятности будет сказано позднее). При этом количество информации определяется только вероятностными свойствами сообщений. В задачах машинного обучения и индуктивного вывода источником сообщений является внешний мир, модель которого неизвестна, и ее требуется определить. Поэтому эти задачи можно сформулировать как задачи, обратные проблеме оптимального кодирования: лучшей моделью источника сообщения следует считать такую модель, которая позволяет осуществлять оптимальное кодирование. Именно идея обращения задачи оптимального кодирования в целях выбора модели привела Р. Соломонова [17] к разработке той концепции индуктивного вывода [7], которая стала первой в целой серии близких подходов.

Однако, предположив верность некоторой модели, нам в этих подходах все еще нужно определить соответствующее ей количество информации (ее длину описания), для чего мож-

но попытаться привлечь классическую теорию информации. Таким образом, поскольку энтропия и количество информации по Шеннону выступают элементами целевой функции, определяющей качество модели, рассмотрим эти понятия.

1.3.2. Энтропия дискретной случайной величины

Пусть есть случайная величина X, которая может принимать любое значение из множества $\mathrm{X} = \{x_1, x_2, \ldots, x_n, \ldots\}$. И пусть для этой случайной величины задано распределение вероятностей $P(x_i) = \Pr(X = x_i)$. Прежде чем перейти к определению количества информации, сделаем небольшое терминологическое замечание.

В теории информации, и особенно в ее приложениях, используются такие понятия, как «*сообщение*» (соответствующее некоторому значению x_i случайной величины) и «ансамбль» (соответствующий множеству X). Поскольку теория информации является математической теорией, опирающейся на понятия и методы теории вероятностей, то без ущерба для теории при изложении можно пользоваться терминологией теории вероятностей [59], что представляется удобным для единообразного совместного изложения теоретико-информационных и статистических подходов. Однако при введении некоторых определений удобнее пользоваться понятием «сообщение».

Количеством (собственной) информации, содержащейся в сообщении $x \in \mathrm{X}$, называется величина $I(x) = -\log_2 P(x)$, измеряемая в битах.

К примеру, количество собственной информации, содержащейся в сообщении о выпадении «решки» в результате подбрасывания монетки, равно I («решка») $= -\log_2 0{,}5 = 1$ бит, причем предполагается, что модель монетки известна априори. Если бы модель была другой (например, вероятность выпадения «решки» была бы 0,75, а «орла» — 0,25), то и количества информации, содержащиеся в соответствующих сообщениях, были бы другие: I («решка») $= = -\log_2 0{,}75 \approx$ $\approx 0{,}42$ и I («орел») $= \log_2 0{,}25 = 2$ бит соответственно. Здесь видно также, что количество информации в сообщении тем больше, чем это сообщение менее вероятно. Например, количество информации в сообщении о вытаскивании какой-то конкретной карты из колоды в 36 карт будет равно $-\log_2 1/36 = 5{,}2$ бит.

Заметим, что количество информации $I(X)$ случайной величины X также является случайной величиной, так как оно зависит от того, какое значение данная случайная величина принимает в испытании. Математическое ожидание $E_X[I(X)]$ количества информации $I(X)$ называется *энтропией* $H(X)$ данной случайной величины:

$$H(X) = E_X\left[I(X)\right] = \sum_{x\in \mathrm{X}} I(x)P(x) = -\sum_{x\in \mathrm{X}} P(x)\log_2 P(x). \quad (1.9)$$

Энтропия некоторой случайной величины выражает неопределенность нашего знания о том, какое именно значение примет эта случайная величина при очередном испытании. Она соответствует количеству информации, в среднем необходимому для устранения этой неопределенности. Несложно заметить, что энтропия, вычисленная по формуле (1.9), является неотрицательной, поскольку $(\forall x \in \mathrm{X})\left(-\log_2 P(x) \geq 0\right)$.

В простейшем случае равновероятных значений случайной величины энтропия будет равна

$$H(X) = -\sum_{i=1}^{N} P(x_i)\log_2 P(x_i) = -\sum_{i=1}^{N} \frac{1}{N}\log_2 \frac{1}{N} = \log_2 N, \quad (1.10)$$

где N — это общее число возможных альтернатив (чем больше значение N, тем больше степень неопределенности).

Таким образом, для такого простейшего случая энтропия всего ансамбля как среднее количество собственной информации в этом ансамбле совпадает с количеством собственной информации каждого сообщения. Такое определение количества информации для равновероятных исходов было дано Р. Хартли еще в 1928 г. [57].

Интересно посмотреть, как ведет себя энтропия для различных распределений вероятностей, заданных на одном и том же множестве сообщений. Так, для примера с монеткой в первом случае (исходы равновероятны) энтропия равна одному биту, в то время как для второго случая (вероятности для «орла» и «решки» отличаются в три раза) энтропия равна $-0,75\log_2 0,75 - 0,25\log_2 0,25 \approx 0,81$ бит. Несложно показать, что и в общем случае максимальной энтропией будет обладать та случайная величина, которая с равной вероятностью принимает значения из данного множества, а значит, верно следующее неравенство:

$$H(X) \leq \log_2 N. \quad (1.11)$$

Если задана вторая случайная величина Y, принимающая значения из множества $\mathrm{Y} = \{y_1, y_2, \ldots, y_n, \ldots\}$, тогда, используя совместное распределение вероятностей $P(x, y)$, можно рассчитать *совместную энтропию* двух случайных величин:

$$H(X,Y) = E_{XY}\left[I(X,Y)\right] = -\sum_{y\in \mathrm{Y}}\sum_{x\in \mathrm{X}} P(x,y)\log_2 P(x,y). \tag{1.12}$$

Здесь X и Y могут быть, в частности, компонентами некоторого двумерного случайного вектора. Формулу (1.12) не составляет труда обобщить и на случайный вектор произвольной размерности.

Условная энтропия случайной величины X относительно случайной величины Y, опирающаяся на понятие условной вероятности, вычисляется следующим образом.

$$H(X \mid Y) = -\sum_{y\in \mathrm{Y}}\left(\sum_{x\in \mathrm{X}} P(x \mid y)\log_2 P(x \mid y)\right)P(y) =$$
$$= -\sum_{y\in \mathrm{Y}}\sum_{x\in \mathrm{X}} P(x,y)\log_2 P(x \mid y). \tag{1.13}$$

При статистической независимости двух случайных величин верно $P(x \mid y) = P(x)$, а значит, $H(X \mid Y) = H(X)$.

Несложно заметить, что выполняется следующее условие *аддитивности энтропии*:

$$H(X,Y) = H(X \mid Y) + H(Y). \tag{1.14}$$

Покажем это (в дальнейшем на доказательстве подобных фактов мы останавливаться не будем в целях экономии места):

$$H(X,Y) - H(X \mid Y) = -\sum_{y\in \mathrm{Y}}\sum_{x\in \mathrm{X}} P(x,y)\log_2 P(x,y) +$$
$$+\sum_{y\in \mathrm{Y}}\sum_{x\in \mathrm{X}} P(x,y)\log_2 P(x \mid y) = -\sum_{y\in \mathrm{Y}}\sum_{x\in \mathrm{X}} P(x,y)\log_2 \frac{P(x,y)}{P(x \mid y)} =$$
$$= -\sum_{y\in \mathrm{Y}}\sum_{x\in \mathrm{X}} P(x,y)\log_2 P(y) = -\sum_{y\in \mathrm{Y}}\left[\sum_{x\in \mathrm{X}} P(x,y)\right]\log_2 P(y) =$$
$$= -\sum_{y\in \mathrm{Y}} P(y)\log_2 P(y) = H(Y).$$

Очевидно, что в случае независимых случайных величин совместная энтропия будет равна

$$H(X,Y) = H(X) + H(Y). \qquad (1.15)$$

Это соотношение может служить *критерием статистической независимости*. С другой стороны, если $Y = \varphi(X)$, т. е. связано с X функционально, то $H(X, Y) = H(X)$.

Поясним рассмотренные понятия на следующих примерах. Пусть случайная величина X соответствует числу, выпавшему при подбрасывании игральной кости, а Y соответствует четности этого числа. Очевидно, что случайный вектор, составленный из этих двух случайных величин, может принимать лишь значения: (1, «нечетно»), (2, «четно»), (3, «нечетно»), (4, «четно»), (5, «нечетно»), (6, «четно»). Вероятности же таких значений, как, например, (2, «нечетно») или (5, «четно»), равны нулю. Поскольку все перечисленные допустимые значения случайного вектора равновероятны, то $H(X,Y) = \log_2 6 = H(X)$. Таким образом, сообщение о четности выпавшего значения не несет дополнительной информации, если известно, какое именно значение выпало. Но при этом $H(Y) = \log_2 2$, что означает следующие условные энтропии: $H(Y \mid X) = 0$ и $H(X \mid Y) = \log_2 6 - \log_2 2 = \log_2 3$. Последнее означает, что, зная четность выпавшего числа, у нас все еще остается неопределенность в выборе из трех альтернатив. Рассмотрим теперь на примере, что будет в случае независимых величин.

Пусть осуществляется вытаскивание карты из колоды. Случайной величиной X будет являться масть выбранной карты, а величиной Y — ее старшинство («туз», «король» и т. д.). Совместно значения этих случайных величин однозначно определяют карту, полностью компенсируя исходную неопределенность. Их совместная энтропия будет равна $H(X,Y) = \log_2 36$. В то же время $H(X) = \log_2 4$, $H(Y) = \log_2 9$ и $H(X) + H(Y) = \log_2 4 + \log_2 9 = \log_2 36 = H(X,Y)$, что и говорит об их статистической независимости.

Хотя многие примеры для наглядности рассматривались на случаях равновероятных исходов тех или иных событий, соответствующие результаты имеют силу и при произвольных распределениях вероятностей.

Введем следующее определение. *Количеством информации в сообщении* $x \in X$ *о сообщении* $y \in Y$ *называется величина*

$$I(x,y) = I(y) - I(y \mid x). \qquad (1.16)$$

К примеру, x — это сообщение о том, что при подбрасывании кубика выпало значение, большее трех (четыре, пять или шесть), а y — это сообщение о том, что это число нечетно. Поскольку $P(y|x) = 1/3$ (сообщение x оставляет возможными три числа, из которых одно число нечетно) и $P(y|x) = 1/2$, то $I(x, y) = 1 - \log_2 3$. Интересно, что в данном случае сообщение x несет отрицательное количество информации о сообщении y. Логически истинное («правдивое») высказывание может, тем не менее, убавлять количество информации о некотором предмете, «вводить в заблуждение».

Нетрудно заметить, что $I(x,y) = -\log_2 P(y) + \log_2 P(y \mid x) = \log_2 \dfrac{P(x,y)}{P(x)P(y)}$. Поскольку эта величина симметрична, то количество информации в паре сообщений друг о друге будет совпадать и поэтому данную величину называют *количеством взаимной информации* между сообщениями x и y. Среднее количество взаимной информации или просто *средняя взаимная информация* будет равна:

$$\mathrm{I}(X,Y) = E_{XY}\left[I(x,y)\right] = \sum_{x\in \mathrm{X}} \sum_{y\in \mathrm{Y}} P(x,y) \log_2 \frac{P(x,y)}{P(x)P(y)}. \quad (1.17)$$

Нетрудно показать, что верны следующие соотношения:

$$\mathrm{I}(X,Y) = H(X) - H(X \mid Y) = H(Y) + H(X) - H(X,Y) = H(Y) - H(Y \mid X) = \mathrm{I}(Y,X). \quad (1.18)$$

Средняя взаимная информация обращается в ноль в случае статистически независимых случайных величин и принимает максимальное значение, когда случайные величины связаны друг с другом функционально, а не статистически. К этому замечательному ее свойству мы еще вернемся в п. 1.4.4.

1.3.3. Энтропия непрерывной случайной величины

Выше предполагалось, что множество значений X случайной величины X конечно либо счетно. В то же время многие данные физических измерений (или сенсорной информации) могут принимать значения из непрерывного пространства, например, пространства вещественных чисел.

И хотя при использовании цифровых компьютеров множество различных значений, которые можно получить в результате некоторого измерения, всегда конечно (так как под результат измерения отводится ячейка памяти конечного или даже фиксированного размера), при вычислении энтропии величин с плавающей запятой возникают те же проблемы, что и при вычислении энтропии непрерывной случайной величины. Поэтому кратко остановимся на этом вопросе.

Случайной величине с непрерывной областью значений X приписывается некоторая *плотность распределения вероятностей* $p : X \to [0, 1]$, которая для произвольного объема $V \subseteq \mathrm{X}$ определяется следующим образом: $\int_V p(x)dx = \Pr(X \in V)$. Часто для конкретизации случайной величины, к которой относится данная плотность вероятности, пишут $p_X(x)$.

По аналогии с (1.9) энтропию непрерывной случайной величины можно было бы определить следующим образом:

$$H(X) = -\int_{\mathrm{X}} p(x) \log_2 p(x)dx. \qquad (1.19)$$

Такое определение энтропии обладает следующими недостатками. Во-первых, в отличие от энтропии дискретной случайной величины она может изменяться при некотором невырожденном преобразовании значений случайной величины $Y = f(X)$. Действительно, пусть $\mathrm{X} = [0, 1]$ и $p(x) = 1$. И пусть $Y = 0{,}5\,X$, т. е. $Y = [0, 0{,}5]$ и $p(y) = 2$, поскольку $\int_{\mathrm{Y}} p(y)dy = 1$. Тогда по формуле (1.19) получаем:

$$H(X) = -\int_0^1 1 \cdot \log_2 1 dx = 0 \quad \text{и} \quad H(Y) = -\int_0^{0{,}5} 2 \cdot \log_2 2 dy = -1.$$

Принимаемые энтропией значения сами по себе являются странными (степень неопределенности принимает нулевое и отрицательное значения), но более негативным фактом является различие $H(X)$ и $H(Y)$ уже при линейном преобразовании между случайными величинами.

Во-вторых, неясно, как определять энтропию в том случае, когда случайная величина может принимать как непрерывные значения из некоторой области, так и набор дискретных значений. Посмотрим, что будет, если плотность распределения вероятностей дискретной случайной вели-

чины задать в виде взвешенной суммы дельта-функций. Пусть

$$p(x) = \sum_{i=1}^{N} P(x_i)\delta(x - x_i). \qquad (1.20)$$

Тогда, подставляя эту плотность вероятности в уравнение (1.19), по определению дельта-функции получаем

$$H(X) = -\int_X \left[\sum_{i=1}^{N} P(x_i)\delta(x - x_i)\right]\left[\log_2 \sum_{i=1}^{N} P(x_i)\delta(x - x_i)\right] dx =$$

$$= \sum_{i=1}^{N} P(x_i)\log_2\left(P(x_i)\delta(0)\right) = \sum_{i=1}^{N} P(x_i)\log_2 P(x_i) + \log_2 \delta(0), \qquad (1.21)$$

что отличается от уравнения (1.9) для энтропии дискретной случайной величины на бессмысленную, на первый взгляд, добавку $\log_2 \delta(0)$.

Одним из способов решения этой проблемы является введение некоторой опорной плотности $\nu_0(x)$, предполагающейся заданной. Энтропия при этом вычисляется по формуле

$$H(X) = -\int_X p(x)\log_2 \frac{p(x)}{\nu_0(x)}\, dx. \qquad (1.22)$$

Обоснование корректности такого подхода можно найти, например, в работе [59]. В частности, при преобразовании случайной величины вместе с ней таким же образом должна преобразовываться опорная плотность, что снимает первую проблему, упомянутую для энтропии, определенной по формуле (1.19).

Оказывается также, что и вторую проблему возможно решить. При задании плотности вероятности дискретной случайной величины опорная плотность должна принимать бесконечные значения в тех же точках, что и плотность вероятности $p(x)$, т. е.

$$\nu_0(x) = \sum_{i=1}^{N} \delta(x - x_i), \qquad (1.23)$$

тогда несложно убедиться, что энтропия, вычисленная по формулам (1.9) и (1.22), будет совпадать. В комбинирован-

ном случае непрерывной и дискретной случайных величин к этой плотности также нужно будет добавить непрерывную опорную плотность, характеризующую непрерывную случайную величину.

Чтобы смысл этой опорной плотности стал более понятен, рассмотрим частный случай непрерывной случайной величины при равенстве опорной плотности некоторой константе $\nu_0(x) = 1/\varepsilon_0$. Тогда уравнение (1.22) преобразуется к виду

$$H(X) = -\int_X p(x) \log_2 \left(\varepsilon_0 p(x)\right) dx = \\ = -\int_X p(x) \log_2 p(x) dx + \log_2 \frac{1}{\varepsilon_0}. \qquad (1.24)$$

Иными словами, энтропия непрерывной случайной величины задана с точностью до некоторой константы (для простейшей опорной плотности). Этому можно дать следующую интерпретацию. Пусть есть некоторое аналоговое устройство, производящее измерение параметра некоторой физической системы. И пусть поступающие с устройства данные оцифровываются, что означает использование определенного количества разрядов. Поскольку результирующая случайная величина дискретна, то можно рассчитать ее энтропию, исходя из уравнения (1.9). Однако мы можем отвести под результаты измерений большее количество бит, тогда число состояний, имеющих ненулевую вероятность, этой дискретной величины увеличится, а энтропия возрастет. Можно считать, что после использования некоторого количества бит для записи результата при последующем увеличении объема отводимой ячейки памяти младшие биты будут неинформативны, т. е. будут содержать лишь некоррелированный шум. Если мы не знаем априорно точности измерений, то мы не можем сказать, сколько именно бит нужно отводить под результаты измерений (а число отведенных бит может быть любым, при этом их содержимое будет произвольным и будет изменяться от измерения к измерению). Таким образом, введенная величина ε_0 задает точность измерений или является неопределенным параметром, если эта точность априори неизвестна.

В случае дискретных случайных величин такой проблемы не возникает, так как при расширении множества значений случайной величины (при добавлении дополнительных бит) вероятности этих значений оказываются равны-

ми нулю. Можно заметить, что добавка $\log_2 \delta(0)$ в уравнении (1.21) соответствует ошибке измерений, равной нулю. Данная проблема характерна при работе с любыми нецелочисленными переменными, а в ряде случаев она возникает и при оценивании энтропии целочисленных величин (например, интенсивности пикселей изображения), поэтому при использовании теоретико-информационного подхода ее часто приходится решать.

При определении совместной энтропии двух непрерывных случайных величин, а также их условной энтропии и средней взаимной информации возникает необходимость установления способа комбинирования разных опорных плотностей. Этот вопрос мы, однако, рассматривать не будем, а интересующемуся читателю рекомендуем обратиться к литературе (см., например, [59, с. 37–42]).

1.3.4. Префиксное кодирование

Одной из основных задач теории информации является решение задачи оптимального кодирования. Эта задача заключается в том, чтобы каждому возможному сообщению из данного ансамбля поставить в соответствие последовательность символов таким образом, чтобы набор сообщений в среднем кодировался как можно более короткой цепочкой символов. Такие задачи, как, например, построение помехоустойчивых кодов, здесь не рассматриваются. Для более формального описания задачи оптимального кодирования нам потребуется ряд дополнительных определений.

Обозначим через A некоторое множество, состоящее из d элементов ($d > 1$): $\mathrm{A} = \{a_1, a_2, \ldots, a_d\}$. Назовем его *алфавитом кода источника*. Элементы a_i множества A будем называть *кодовыми символами*. Последовательность кодовых символов $\alpha = a'_1 a'_2, \ldots, a'_N, a'_i \in \mathrm{A}$ называется *кодовым словом*. *Длина кодового слова* — это количество символов в нем (N). Под обозначением A* будем понимать множество всех возможных кодовых слов α, составленных из символов данного алфавита. Произвольное семейство кодовых слов $\Gamma \subseteq \mathrm{A}^*$ называется *кодом над алфавитом* A.

Примером может служить алфавит русского языка, состоящий из 33 символов (d = 33), если не различать строчные и прописные буквы. Кодами над этим алфавитом являются как множество всех осмысленных слов русского

языка, так и множество произвольных последовательностей букв (а также любое его подмножество). Последовательность символов «слово» является кодовым словом длины 5.

Другим примером кода является множество бинарных строк произвольной длины $\{0,1\}^*$, заданное над алфавитом $\{0,1\}$. Бинарная строка «0010011101» является кодовым словом длины 10. Но более интересны случаи, когда максимальная длина кодового слова ограничена.

Код называется *равномерным*, если все его слова имеют одинаковую длину m, это число называется *длиной кода*. Несложно заметить, что количество различных слов равномерного кода длины m не превосходит d^m — числа различных d-ичных последовательностей длины m. Если хотя бы два кодовых слова имеют различные длины, то код называют *неравномерным*.

Примером равномерного кода над алфавитом $\{0,1\}$ является следующее множество кодовых слов: $\{00,01,10,11\}$. Длина этого кода равна двум. Неравномерным кодом будет являться, например, такой: $\{0,10,110,111\}$.

Кодированием сообщений ансамбля X (или кодированием множества значений X случайной величины X) *посредством кода* Г называется отображение $\varphi: X \to Г$ множества сообщений в множество кодовых слов. Мы будем рассматривать взаимно однозначные отображения, хотя в общем случае условие взаимной однозначности между значениями случайной величины и кодовыми словами не обязательно выполняется.

Заметим, что можно образовать случайную величину $G = \varphi(X)$, принимающую значения из множества кодовых слов Г. Причем, если кодирующее отображение φ является взаимно однозначным, то $H(G) = H(X)$.

Пусть, например, X — масть выбранной наугад карты. Эта случайная величина может принимать четыре значения: x_1, ..., x_4. И пусть $Г = \{00, 01,10,11\}$ — равномерный код длины 2 над алфавитом $\{0,1\}$. Тогда мы можем сопоставить сообщению о том, что выбранная карта имеет масть «пики», кодовое слово «00» и т. д. Заметим также, что длина равномерного кода в данном случае совпадает с энтропией кодируемой случайной величины. Другим примером кодирования может служить отображение букв русского алфавита (рассматриваемых теперь как значения случайной величины) в последовательности точек и тире, принятых в азбуке Морзе.

Обозначим через m_i длину слова, кодирующего значение $x_i \in X$ случайной величины X. Пусть $P(x_i)$ — соответствующая вероятность. Тогда

$$\bar{m}(X) = \sum_{x_i \in X} m_i P(x_i) \qquad (1.25)$$

будет средней длиной кодовых слов (или *средней длиной кода*), кодирующих значения, принятых X при проведении бесконечного множества испытаний. Целью задачи оптимального кодирования является построение кодов, обладающих минимальной средней длиной. Чтобы перейти к описанию таких кодов, нам понадобятся следующие определения.

Коды, в которых ни одно слово не является началом другого, называются *префиксными*.

Коды, в которых любая последовательность кодовых слов допускает однозначное разбиение на кодовые слова, называются *кодами со свойством однозначного декодирования*.

Префиксные коды всегда являются кодами со свойством однозначного декодирования, но не наоборот. Все равномерные коды являются префиксными. Префиксным является такой код, как, например, уже приводимый нами {0,10,110,111}. А вот код {0,10,100,111} префиксным не является, так как строка «10» является началом строки «100».

Для любого кода со свойством однозначного декодирования оказывается верным следующее неравенство (приводимое здесь без доказательства), которое связывает среднюю длину кода и энтропию случайной величины:

$$\bar{m}(X) \geq \frac{H(X)}{\log_2 d}. \qquad (1.26)$$

Для наглядности перепишем это соотношение следующим образом:

$$\sum_{x_i \in X} \left(m_i + \log_d P(x_i)\right) P(x_i) \geq 0. \qquad (1.26\text{а})$$

Теперь несложно убедиться, что это неравенство обращается в точное равенство, если $P(x_i) = d^{-m_i}$ для любого $x_i \in X$.

В дальнейшем для сокращения будем писать $\frac{H(X)}{\log_2 d} = H_d(X)$, $H_d(X) = -\sum_{x \in X} P(x) \log_d P(x)$. Например, $H_2(X) = H(X)$ — эн-

тропия, выраженная в битах; $H_8(X)$ — энтропия, выраженная в байтах.

Итак, если вероятности сообщений являются целыми отрицательными степенями числа d, то соответствующий d-ичный код будет иметь среднюю длину, в точности равную $H_d(X)$, а средняя длина любого кода не может быть менее этого значения. Но оказывается также, что существует и ограничение сверху на минимальную среднюю длину кода. А именно: всегда существует d-ичный код со свойством однозначного декодирования, для которого

$$\bar{m}(X) < H_d(X) + 1. \qquad (1.27)$$

Классические способы построения таких кодов мы приведем чуть позже.

Рассмотрим теперь кодирование последовательности сообщений. Последовательности из n сообщений соответствует случайный вектор X^n. Для него существует код, длина которого удовлетворяет следующим ограничениям:

$$H_d(X^n) \le \bar{m}(X^n) < H_d(X^n) + 1. \qquad (1.28)$$

Здесь величина $\bar{m}(X^n)$ — среднее число символов, кодирующих совокупность из n значений случайной величины X. Тогда на кодирование одного значения будет в среднем приходиться $\bar{m}(X^n)/n$ символов:

$$\frac{1}{n} H_d(X^n) \le \frac{\bar{m}(X^n)}{n} < \frac{1}{n} H_d(X^n) + \frac{1}{n}. \qquad (1.29)$$

Рассмотрение кодирования последовательности сообщений имеет два важных для нас следствия.

1. Пусть последовательность измерений является статистически независимой. Тогда $H_d(X^n) = nH_d(X)$, а значит, верно

$$H_d(X) \le \frac{\bar{m}(X^n)}{n} < H_d(X) + \frac{1}{n}. \qquad (1.30)$$

Это означает, что среднюю длину наикратчайшего кода для величины X можно с хорошей точностью считать равной энтропии $H_d(X)$, так как добавку $1/n$ можно сделать сколь угодно малой, кодируя одновременно по несколько значений случайной величины.

Поясним это на следующем примере. Пусть случайная величина X принимает два значения: «0» и «1» с вероят-

ностями $\Pr(X=0)=0,8$ и $\Pr(X=1)=0,2$. Ее энтропия будет $H=-0,2\log_2 0,2-0,8\log_2 0,8\approx 0,722$. Кодируя отдельно каждое значение (соответствующими кодовыми словами 0 и 1), мы получим среднюю длину кода, равную одному символу. Это наименьшая возможная средняя длина, поскольку нельзя кодировать сообщение меньше чем одним символом. Но посмотрим, что будет, если мы будем кодировать сразу тройки значений (x_1, x_2, x_3). Всего возможно восемь различных исходов серии из трех испытаний, которые обладают следующими вероятностями и которым мы присвоим следующие коды:

(0,0,0) $\Pr\left(X^3=(0,0,0)\right)=0,8^3=0,512$ код «0»

(0,0,1) $\Pr\left(X^3=(0,0,1)\right)=0,8^2\cdot 0,2=0,128$ код «100»

(0,1,0) $\Pr\left(X^3=(0,1,0)\right)=0,128$ код «101»

(1,0,0) $\Pr\left(X^3=(1,0,0)\right)=0,128$ код «110»

(0,1,1) $\Pr\left(X^3=(0,1,1)\right)=0,8\cdot 0,2^2=0,032$ код «11100»

(1,0,1) $\Pr\left(X^3=(1,0,1)\right)=0,032$ код «11101»

(1,1,0) $\Pr\left(X^3=(1,1,0)\right)=0,032$ код «11110»

(1,1,1) $\Pr\left(X^3=(1,1,1)\right)=0,2^3=0,008$ код «11111»

Несложно подсчитать среднюю длину кода:

$$\bar{m}(X^3)=0,512+3\cdot 3\cdot 0,128+3\cdot 5\cdot 0,032+5\cdot 0,008=2,184$$

символа на три значения случайной величины, что соответствует 2,184/3 = 0,728 символам, приходящимся на кодирование одного значения, а это уже весьма близко к значению энтропии. Если бы мы кодировали сразу, скажем, по шесть исходов испытаний, то получили бы еще более близкое к энтропии значение.

2. Если последовательность испытаний не является статистически независимой, то $H(X^n)<nH(X)$. И действитель-

но, равенство $H(X,Y) = H(X) + H(Y)$ выполняется, только если X и Y статистически независимы. В противном случае $H(X,Y) = H(X \mid Y) + H(Y)$, а $H(X \mid Y) < H(X)$. Это означает, что для последовательности сообщений существует более короткий код, чем код со средней длиной слова $H(X)$, приходящейся на одно сообщение. Таким образом, чтобы посчитать количество информации, содержащейся в последовательности статистически зависимых сообщений, нужно считать совместную энтропию $H(X^n)$ для всех этих сообщений.

Обычно рассматривается случайный процесс, порождающий последовательность сообщений и являющийся моделью источника сообщений. Эта модель задает вероятности появления различных сообщений. Вероятности могут зависеть от ранее появившихся сообщений или некоторых дополнительных параметров (например, от текущего момента времени), поэтому энтропия такого источника должна вычисляться как энтропия описывающего его случайного процесса.

Мы так подробно остановились на этом вполне очевидном моменте, поскольку он часто является источником заблуждений людей, недостаточно хорошо знакомых с теорией информации. Особенно это касается подсчета количества информации в тексте на естественном языке. Поскольку слова в предложении и предложения в тексте не являются статистически независимыми, то мы не можем посчитать количество информации в тексте, просто как «побуквенную» энтропию, умноженную на число букв, или как энтропию, посчитанную через вероятности появления тех или иных слов, считая слова статистически независимыми отсчетами некоторой случайной величины. А именно различие количеств информации, посчитанных при предположении о статистической независимости слов или букв, для двух фраз, в разных словах выражающих одну и ту же мысль, иногда неправомерно приводится в качестве недостатка теории информации. Здесь реализацией случайной величины является целый текст, вероятности появления которого мы, однако, не знаем и не можем посчитать эмпирически, перебрав все имеющиеся тексты. Таким образом, для корректного оценивания количества информации в тексте необходима модель источника сообщений, т. е. человека (причем человека, составившего этот текст). Построение такой модели на данный момент, естественно, является недоступным,

поэтому адекватное оценивание количества информации в тексте произвести нельзя (можно лишь сделать грубую оценку, руководствуясь сильными упрощающими предложениями), но эта проблема не относится непосредственно к самой теории информации.

В классической теории информации рассмотрены некоторые классы моделей источников сообщений, и для них найдены алгоритмы оптимального кодирования. Нам для дальнейшего изложения они не понадобятся, поэтому мы остановимся лишь на алгоритмах оптимального кодирования для дискретного источника без памяти, т. е. формирующего статистически независимые сообщения.

Оптимальным называется код, средняя длина кодовых слов которого равна минимально возможной. Как мы уже упоминали, если для любого $x_i \in X$ существует некоторое целое m_i, такое, что $P(x_i) = d^{-m_i}$, то существует оптимальный d-ичный однозначно декодируемый код, для которого верно $\bar{m}(X) = H_d(X)$. В таком коде сообщению x_i соответствует слово длины $-\log_d P(x_i) = m_i$. Существует следующий метод построения такого кода (метод Шеннона—Фано):

1) множество сообщений X разделяется на d подмножеств таким образом, чтобы вероятности каждого из подмножеств были одинаковыми; каждому подмножеству назначается символ $a_i \in A$, отличный от символов, назначенных другим подмножествам; символ, назначенный данному подмножеству, будет выступать первым символом в кодовых словах, соответствующих всем сообщениям, вошедшим в это подмножество;

2) каждое из подмножеств рассматривается отдельно в качестве самостоятельного множества сообщений и для него выполняется шаг 1.

Мы не будем приводить пример построения кода Шеннона—Фано, а рассмотрим построение методом Хаффмана [62] префиксных кодов для общего случая, когда сообщения в ансамбле имеют произвольные вероятности. Любой набор длин кодовых слов однозначно декодируемого кода может быть получен и на префиксном коде.

Для простоты изложения рассмотрим случай $d = 2$, обобщение на случай алфавита произвольного размера не представляет сложности. Не умаляя общности, можно также считать, что значения случайной величины упорядочены по их вероятностям: $P(x_1) \geq P(x_2) \geq \ldots \geq P(x_N)$. Здесь в отличие

от предыдущего метода будет производиться не разделение всего множества сообщений на подмножества, а постепенное объединение сообщений. На каждом шаге метода определяется пара наименее вероятных сообщений. Одному из сообщений в паре приписывается код «0», а другому — «1». Затем эти сообщения объединяются и формируют новое сообщение, вероятность которого равна сумме вероятностей обоих сообщений. Считая исходные сообщения листьями некоторого дерева, производные сообщения — узлами, а конечное сообщение, имеющее вероятность 1, — корнем, можно построить дерево. Движение от корня по этому дереву к одному из листьев соответствует формированию кодового слова: когда мы идем по левой ветви, то к кодовому слову приписываем 0, а по правой — 1. Любой код, который можно представить в виде такого дерева, является префиксным.

Алгоритм гораздо понятнее становится на примере. Рассмотрим семь сообщений с вероятностями 0,03, 0,05, 0,09, 0,10, 0,12, 0,23 и 0,38 (рис. 1.1). Объединяя первые два сообщения, получаем новое сообщение с вероятностью 0,08. Теперь объединяем сообщения с вероятностями 0,08 и 0,09, затем — с вероятностями 0,10 и 0,12 и т. д. Построив таким методом дерево, мы теперь можем получить кодовые слова для каждого из исходных сообщений, например, для сообщения с вероятностью 0,10 кодовое слово будет «010».

Сформулировав основные необходимые нам факты из теории информации, мы можем вернуться к проблеме сравнения моделей. Поскольку здесь мы не занимались систе-

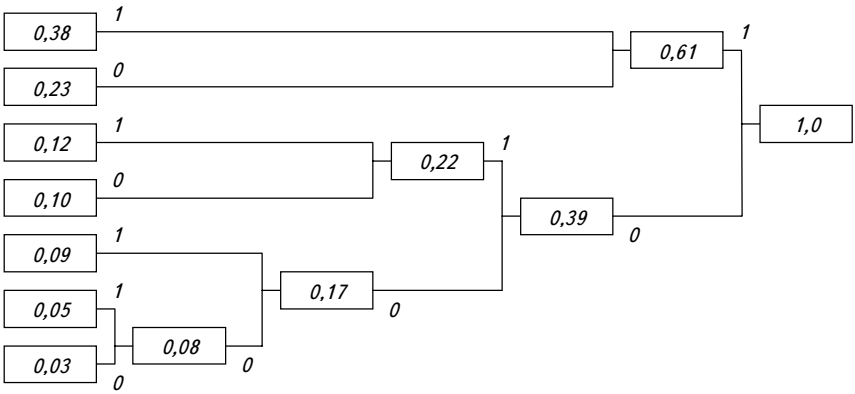

Рис. 1.1. Построение оптимального кода методом Хаффмана

матическим изложением основ теории информации, а лишь напомнили те ее факты, которые понадобятся нам для дальнейшего изложения, читателю, не знакомому с теорией информации или желающему более детально ознакомиться с какими-то ее моментами, рекомендуем обратиться к литературе, посвященной именно этим вопросам (см., например, [54, 59]).

1.4. ИНФОРМАЦИОННАЯ МЕРА ПРИ ВЫБОРЕ МОДЕЛИ

1.4.1. Теоретико-информационная интерпретация правила Байеса

Приведем еще раз формулу Байеса (1.4), дающую выражение для апостериорной вероятности гипотезы h_i при данных D:

$$P(h_i \mid D) = \frac{P(h_i)P(D \mid h_i)}{P(D)}.$$

После обращения к теории информации, устанавливающей связь между вероятностью и количеством информации, несложно догадаться взять отрицательный логарифм от обеих частей этого уравнения. Сейчас этот прием является уже классическим, и сложно определить, кто первый его использовал. Результатом такой операции будет выражение

$$-\log_2 P(h_i \mid D) = -\log_2 P(h_i) - \log_2 P(D \mid h_i) + \log_2 P(D). \tag{1.31}$$

По определению собственного количества информации получаем:

$$I(h_i \mid D) = I(h_i) + I(D \mid h_i) - I(D). \tag{1.32}$$

Таким образом, максимизация апостериорной вероятности $P(h_i \mid D)$ соответствует минимизации количества информации $I(h_i \mid D)$, которая характеризует степень неопределенности гипотезы h_i при данных D.

Мы обратились к теории информации, поскольку хотели определить способ задания априорных вероятностей гипотез. Но пока количество информации вычисляется через

эти же вероятности, поэтому проблема априорных вероятностей не решена. Прежде чем перейти к вопросу нахождения априорных вероятностей путем обращения задачи оптимального кодирования, полезно будет дать теоретико-информационную интерпретацию соответствующих членов уравнения (1.32).

Величина $I(h_i)$ — это собственное количество информации гипотезы h_i. Если у нас есть пространство гипотез $H = \{h_i\}$ с заданным на нем априорным распределением вероятностей $P(h_i)$, то существует оптимальный код, который каждой гипотезе h_i сопоставляет бинарное слово длины, примерно соответствующей $I(h_i)$. Таким образом, $I(h_i)$ — это *длина описания гипотезы* (или ее сложность).

Величина $I(D)$ — это собственное количество информации в данных. Обычно эта величина неизвестна, так как неизвестна модель источника данных. Более того, значение $I(D)$ не влияет на выбор гипотезы, поэтому часто не рассматривается.

Величина $I(D \mid h_i)$ — это длина описания данных D с помощью гипотезы h_i. Эта величина равна нулю, если данные полностью укладываются в гипотезу, и больше, если либо гипотеза неадекватна, либо в сообщении D содержатся какие-то данные, нерелевантные по отношению к гипотезе.

Например, пусть D — набор точек $\{x_i, y_i\}$; гипотезы — это некоторые функциональные зависимости вида $y = f(x)$. Если для какой-то гипотезы $y_i - f(x_i) \neq 0$, то можно считать, что эта гипотеза плохо удовлетворяет данным. Это выражается в необходимости хранения величин невязок в дополнение к описанию самой модели. В то же время часть данных, описывающих абсциссы x_i, является нерелевантной по отношению к гипотезам, так как эти значения придется хранить отдельно при любой гипотезе. Условно говоря, можно записать: $I(D \mid f) = I(\{x_i\}) + I(\{y_i - f(x_i)\})$. В этом примере описание данных с помощью функциональной зависимости разделяется на описание абсцисс (эта добавка не зависит от гипотезы функциональной зависимости, т. е. является нерелевантной по отношению к ней) и на описание величин невязок. В свою очередь, символьное описание этой функциональной зависимости в рамках некоторого языка представления определит ее длину описания $I(f)$. Например, запись $f(x) = x + 2$ более короткая, чем запись $f(x) = 0,5\,(x-2)^2 + \frac{4}{15}\left(2^x - 1\right)$, поэтому она обладает большей априорной вероятностью,

и при выборе лучшей гипотезы для данных $D = \{(0, 2), (4, 6)\}$ она будет предпочтительнее.

Видно, что, выбирая разные языки представления, мы будем получать различные длины описаний для одной и той же гипотезы. Поэтому мы лишь перевели проблему выбора априорных вероятностей в проблему выбора наилучшего языка представления. Но и это само по себе немаловажно, так как несложно убедиться, насколько проще задать некий язык описания (к примеру, для приведенной выше задачи интерполяции), чем назначать каждой гипотезе вероятность, руководствуясь эвристическими правилами. Можно также надеяться, что используемые человеком языки представления достаточно близки к оптимальным в этом смысле и дают неплохие приближения к истинным априорным вероятностям.

Теперь рассмотрим все слагаемые в уравнении (1.32) вместе. Заметим, что сумма $I(D \mid h_i) + I(h_i) = I(D, h_i)$ является длиной совместного описания данных и гипотезы. Тогда $I(h_i \mid D) = I(D, h_i) - I(D)$, т. е. это недостаток информации, содержащейся в данных D, о гипотезе h_i. «Ноль» достигается тогда, когда имеем полную информацию: $I(D, h_i) = I(D)$ — описание гипотезы содержится в данных: присовокупив к данным описание гипотезы, мы не увеличили количество информации. Также очевидно, что

$$0 \le I(h_i \mid D) \le I(h_i). \qquad (1.33)$$

Сделаем следующее наблюдение. Пусть h_i — истинная гипотеза; D — некоторая выборка данных; $I(D) < I(h_i)$. Тогда

$$I(h_i \mid D) = I(h_i) + I(D \mid h_i) - I(D) \ge I(h_i) - I(D) > 0. \qquad (1.34)$$

Другими словами, данные не содержат достаточного количества информации, чтобы достоверно выбрать правильную гипотезу.

Например, нам нужно определить, какая карта была извлечена из колоды, состоящей из 32 карт (без «шестерок»), если известно, что эта карта — «картинка» «пиковой» масти. Здесь мы имеем 1 + 2 = 3 бита информации, а любая гипотеза должна описываться пятью битами. Это значит, что $I(D) < I(h_i)$. Или если нам нужно провести параболу по двум точкам, то нам также *не хватает информации* (в совершенно определенном количественном смысле) для ре-

шения этой задачи, что является естественной формулировкой, используемой человеком в подобных ситуациях. Такие примеры не являются впечатляющими, так как аналогичные результаты можно получить и на основе более простых комбинаторных соображений. Однако такой анализ можно применить и для более сложных случаев, когда вычисление количества информации не является столь тривиальным.

По сути же, требование того, чтобы выполнялось соотношение $I(D) > I(h_i)$, означает, что не должны выбираться гипотезы, более сложные (или более содержательные), чем имеющиеся данные. Вспомним формулировку бритвы Оккама: «Без необходимости не следует утверждать многое».

Если отбросить слагаемое $I(D)$, то для выбора лучшей гипотезы нужно минимизировать следующую сумму длин описаний:

$$L = I(h_i) + I(D \mid h_i), \tag{1.35}$$

а оптимальная гипотеза будет:

$$h = \arg\min_{h \in \mathrm{H}} \left[I(h) + I(D \mid h) \right]. \tag{1.36}$$

Получаем, что лучшая гипотеза — это компромисс между ее сложностью $I(h_i)$ и тем, насколько хорошо (коротко) с ее помощью описываются данные: $I(D \mid h_i)$. Чаще именно это правило соотносится с бритвой Оккама в ее другой формулировке: «То, что можно объяснить посредством меньшего, не следует выражать посредством большего».

Здесь описание данных D представляется в двух частях: описание модели и описание данных посредством модели, т. е. производится поиск наиболее короткого описания данных. Но длина этого совместного описания будет не меньше (а обычно больше), чем $I(D)$. С чем это связано? Дело в том, что минимальную длину описания $I(D)$ можно достичь только тогда, когда модель данных известна априори, т. е. известно распределение вероятностей по возможным реализациям исходных данных.

Поясним это положение на примере с оптимальными кодами Хаффмана. Код Хаффмана позволяет минимизировать среднюю длину кода по всем возможным реализациям данных. При этом каждому набору данных будет соответствовать кодовое слово, длина которого будет примерно равна $I(D)$. Но помимо самих закодированных данных нам

нужно еще и хранить таблицу перекодировки, иначе мы не сможем восстановить исходные данные. Вот длина этой таблицы перекодировки и отвечает величине $I(h_i)$. Это простейший случай. В более сложном случае пространство моделей может быть более широким, а исходная неопределенность в выборе модели — больше.

Таким образом, сумма $L = I(h_i) + I(D \mid h_i)$ показывает полную длину описания данных, а сам выбор лучшей гипотезы по формуле (1.36) можно описать так: «следует выбирать ту гипотезу, которая позволяет описать данные наиболее коротко». Это и есть *принцип Минимальной Длины Описания* в его общей словесной формулировке. Покажем, что этот принцип обобщает некоторые широко распространенные критерии качества моделей.

1.4.2. Методы второго порядка и предположение нормальности

При построении моделей, описывающих случайные величины, принимающие численные значения (мы здесь ограничимся вещественными числами), наиболее широкое распространение получили такие критерии, как дисперсия (или ее корень — среднеквадратичное отклонение, СКО)

$$D_X = E_X\left[\left(X - \overline{X}\right)^2\right] = \int\limits_{x \in \mathrm{X}} p(x)\left(x - \overline{X}\right)^2 dx, \;\; \sigma_X = \sqrt{D_X} \tag{1.37}$$

и корреляция

$$C_{XY} = E_{XY}\left[(X - \overline{X})(Y - \overline{Y})\right] \tag{1.38}$$

или коэффициент корреляции (принимающий значения в диапазоне $[-1, 1]$)

$$\rho_{XY} = \frac{C_{XY}}{\sigma_X \sigma_Y}. \tag{1.39}$$

Дисперсия является одной из характеристик плотности вероятности одной случайной величины. Она призвана выразить меру рассеяния этой случайной величины, поэтому чаще всего используется для определения степени отклонения исходных данных от некоторой модели. Следова-

тельно, дисперсия остатков, получившихся после вычитания модели из исходных данных, служит критерием качества модели.

Понятие корреляции призвано определить наличие стохастической (а не строго функциональной) зависимости между величинами, когда одна из величин зависит не только от второй величины, но также и от ряда неизвестных факторов, либо когда обе величины нестрого зависят от некоторого общего фактора. Коэффициент корреляции так же, как и дисперсия, может выступать в качестве критерия качества модели, но здесь модель будет описывать взаимосвязь между наборами данных (или между случайными величинами).

При использовании коэффициента корреляции и дисперсии в качестве критериев качества моделей в силу автоматически вступают определенные (весьма сильные) предположения о свойствах исходных данных. Если эти предположения не соответствуют действительности, то результат выбора модели может оказаться неадекватным. Часто налагаемые ограничения не осознаются, что приводит к некоторой неправомерной абсолютизации данных критериев. Покажем, что это за ограничения.

Сначала заметим, что плотность распределения вероятностей некоторой случайной величины X можно представить в виде совокупности ее моментов:

$$\mu_n[X] = \int_{-\infty}^{+\infty} x^n p(x) dx. \qquad (1.40)$$

Теперь несложно увидеть, что как коэффициент корреляции, так и квадратичное отклонение, полностью определяются моментами второго порядка (если ограничиться центрированными случайными величинами), т. е. вовлекают лишь немногую информацию о распределении вероятностей случайных величин.

Также несложно убедиться, что гауссово (нормальное) распределение

$$p(x|a,\sigma) = \frac{1}{\sqrt{2\pi}\sigma} \mathbf{e}^{-\frac{(x-a)^2}{2\sigma^2}} \qquad (1.41)$$

задается двумя параметрами: a и σ. При этом математическое ожидание X равно a, а дисперсия равна σ^2.

Это распределение имеет множество интересных свойств и является в некотором смысле выделенным. Уникальность его заключается в том, что при «смешивании» случайных величин их плотность вероятности стремится к распределению Гаусса. В частности, при многократном суммировании случайной величины с самой собой $X^{(n)} = \sum_{i=1}^{n} X$ распределение результирующей случайной величины $X^{(n)}$ будет приближаться к нормальному. В то же время при сложении двух случайных величин X и Y, имеющих нормальные распределения с σ_X и σ_Y соответственно, получится случайная величина, также имеющая нормальное распределение, но с квадратичным отклонением, равным $\sigma_{X+Y} = \sqrt{\sigma_X^2 + \sigma_Y^2}$ и математическим ожиданием, равным $a = a_X + a_Y$. Поэтому шумы как совокупное действие многих независимых факторов часто оказываются распределенными по нормальному закону.

Нормальное распределение имеет еще одно очень интересное свойство: оно обладает максимальной энтропией при заданной дисперсии. Это, в частности, означает, что при «смешивании» случайных величин (например, при сложении) энтропия относительно логарифма дисперсии возрастает или остается постоянной (например, для двух случайных величин, распределенных нормально). Для сравнения, равномерное распределение обладает максимальной энтропией при фиксированных минимальном и максимальном значениях случайной величины.

Именно в связи с особенными свойствами нормального распределения и тем, что оно зависит только от математического ожидания и дисперсии, говорят, что дисперсионные и корреляционные методы (а также все прочие методы второго порядка) работают при предположении нормальности распределений случайных величин. И хотя эти методы корректно работают также при других классах функций распределения вероятностей, зависящих только от моментов первого и второго порядков (при условии, что класс функций распределения известен и нужно лишь оценить параметры конкретной плотности распределения вероятностей), мы также будем для удобства говорить о предположении нормальности распределений.

Но если эти критерии имеют дело только со случайными величинами, распределенными по нормальному или близкому закону, то существует ли более общий критерий?

Оказывается, что понятие энтропии расширяет критерий среднеквадратичного отклонения, а средняя взаимная информация — коэффициент корреляции на случай отклонения распределений вероятности от нормального распределения. Это вызвало заметный интерес в статистике к методам, основанным на энтропии, и изучение подобного рода методов является отдельным направлением исследований. В то же время критерии, базирующиеся на энтропии и средней взаимной информации, можно рассматривать как частные случаи критерия, основанного на длине описания. Рассмотрим эти положения более подробно, привлекая конкретные примеры.

1.4.3. Среднеквадратичное отклонение и энтропия

Сначала установим связь между дисперсией и энтропией. Для этого рассчитаем энтропию нормального распределения:

$$H(X) = -E_X\left[\log_2 p(x)\right] = \int_{-\infty}^{+\infty}\left(\frac{1}{\sqrt{2\pi}\sigma}\,\mathbf{e}^{-\frac{(x-a)^2}{2\sigma^2}}\right)\times$$

$$\times\left(\log_2\left(\sqrt{2\pi}\sigma\right)+\frac{(x-a)^2}{2\sigma^2\ln 2}\right)dx =$$

$$=\log_2\left(\sqrt{2\pi}\sigma\right)\int_{-\infty}^{+\infty}\left(\frac{1}{\sqrt{2\pi}\sigma}\,\mathbf{e}^{-\frac{(x-a)^2}{2\sigma^2}}\right)dx +$$

$$+\int_{-\infty}^{+\infty}\frac{1}{\sqrt{2\pi}\sigma}\,\frac{(x-a)^2}{2\sigma^2\ln 2}\,\mathbf{e}^{-\frac{(x-a)^2}{2\sigma^2}}\,dx.$$

Первый интеграл равен единице, так как это интеграл от плотности распределения вероятностей, а второй интеграл — константа, не зависящая ни от a, ни от σ, что становится очевидно при замене $y = (x - a)/\sigma$. Таким образом, эн-

тропия случайной величины с нормальной плотностью вероятности будет равна

$$H(X) = \log_2 \sigma + c. \qquad (1.42)$$

Мы здесь опустили опорную плотность, определяющую погрешность (или точность) задания численных значений. Константа c ее поглотит. Полученное уравнение показывает, что изменение среднеквадратичного отклонения в два раза изменит энтропию на один бит. Таким образом, минимизация среднеквадратичного отклонения равносильна минимизации энтропии в случае нормального распределения.

В связи с этим обычно говорят, что энтропия обобщает критерий среднеквадратичного отклонения на случай произвольной плотности распределения вероятностей, так как последнее основывается только на центральных моментах второго порядка [63, с. 81]. Также можно утверждать, что моменты второго порядка «работают» в предположении о том, что метрика пространства, в котором задан случайный вектор, евклидова. Если быть точнее, то в случае многомерного пространства нужно говорить о расстоянии Махаланобиса; вопрос о связи метрики пространства и виде плотности распределения вероятностей несколько подробнее будет рассмотрен в п. 2.3.

Если это условие выполняется, то критерии, основанные на энтропии и дисперсии, дают одинаковый результат: минимум энтропии соответствует минимуму среднеквадратичного отклонения (рис. 1.2, 1.3). Естественно, на практике это соответствие может выполняться лишь приближенно.

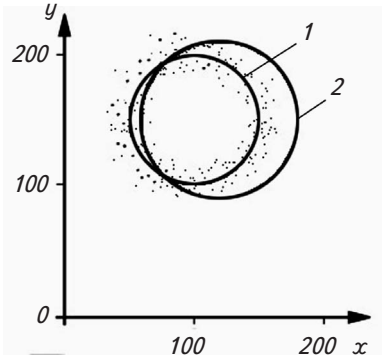

Рис. 1.2. Набор точек, сформированных как точки окружности со случайным отклонением, распределенным по нормальному закону. Окружность *1* соответствует истинной модели расположения точек, окружность *2* немного сдвинута и имеет больший радиус. Для обеих моделей были оценены энтропия и среднеквадратичное отклонение для X-координат: $H(\Delta X) \approx \approx 4{,}45$ и $\sigma_{\Delta X} \approx 8{,}1$ для истинной модели (*1*) и $H(\Delta X) \approx 5{,}26$ и $\sigma_{\Delta X} \approx 15{,}8$ для неверной модели (2)

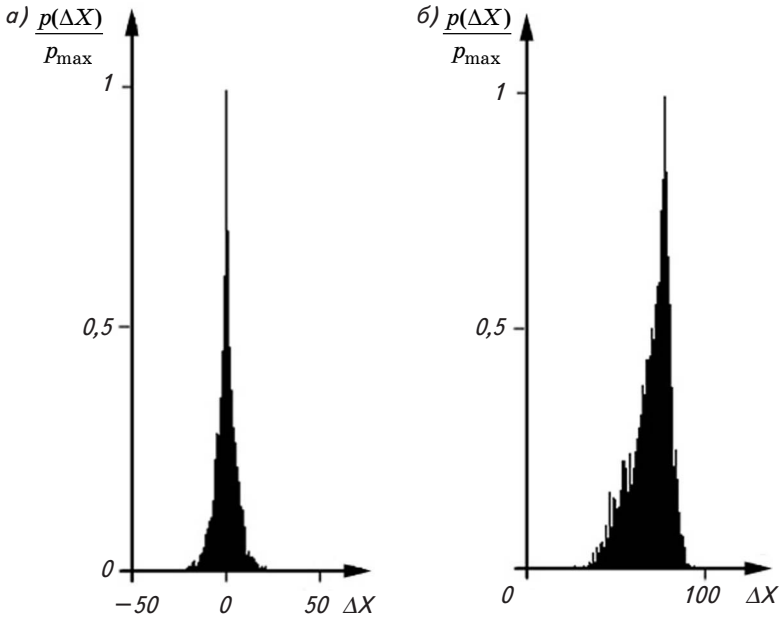

Рис. 1.3. Гистограммы отклонений точек выборки: *а* — от истинной модели *1* (см. рис. 1.2); *б* — от неверной модели *2*

Интересно посмотреть, насколько разные модели будут выбираться на основе этих критериев, когда ошибка не описывается нормальным законом. Простейший пример такой ситуации — наличие выбросов. Оказывается (см., например, [63]), что энтропия гораздо менее чувствительна к выбросам, чем среднеквадратичное отклонение. Например, если есть набор точек, образующих прямую линию, а одна точка расположена в стороне (выброс), то при выборе лучшей прямой, описывающей данный набор точек, при использовании среднеквадратичного отклонения выбранная прямая будет иметь тенденцию отклоняться в сторону отстоящей точки. Причем это отклонение будет тем сильнее, чем дальше находится эта точка. Если точка расположена достаточно далеко, то прямая будет проходить близко от нее и от центра масс остальных точек. Для решения подобных проблем разрабатываются достаточно изощренные методики исключения выбросов (которые также можно обобщить на основе информационных критериев).

При использовании же энтропии лучшая прямая не будет отклоняться в сторону такого выброса независимо от

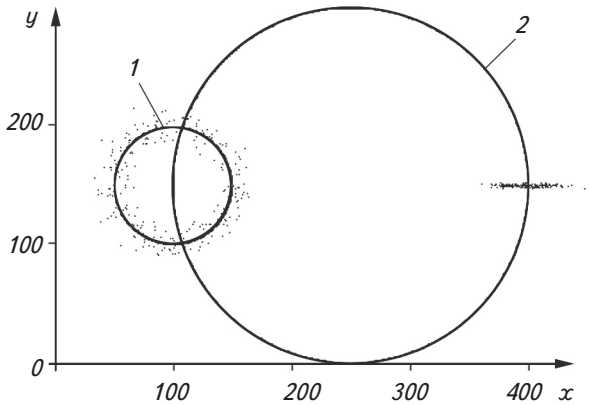

Рис. 1.4. Набор точек и две модели, выбранные на основе энтропийного критерия (окружность *1*) и на основе среднеквадратичного отклонения (окружность *2*). Набор точек состоит из набора, представленного на рис. 1.2, дополненного таким же количеством точек, расположенных справа. Для обеих моделей были оценены энтропия и среднеквадратичное отклонение для X-координат: $H(\Delta X) \approx 6{,}25$ и $\sigma_{\Delta X} \approx 125{,}7$— для модели *1* и $H(\Delta X) \approx 6{,}78$ и $\sigma_{\Delta X} \approx 28{,}2$ — для модели 2

расстояния до него. Это хорошо иллюстрирует несколько утрированный пример, представленный на рис. 1.4–1.5, из которых видно, почему критерии на основе энтропии оказываются менее чувствительными к выбросам, чем на основе дисперсии: в них входит не расстояние между точками, а только совпадение этих расстояний.

Следует, однако, заметить, что энтропия тоже не является универсальным критерием (не говоря уже о том, что ею гораздо сложнее пользоваться на практике). Естественно, энтропия может оказаться одинаковой для совершенно различных распределений вероятностей. А приведенный выше пример не вполне корректен, поскольку привлекаемый класс моделей не соответствует имеющимся данным (расположение точек на рис. 1.4 нельзя объяснить с помощью лишь одной окружности).

Но, несмотря на это, энтропия обладает положительным свойством: минимизация энтропии соответствует разделению модели на максимально статистически независимые (в рамках данного представления) компоненты, что также видно из рис. 1.4. Это полезно, например, в распознавании образов при снижении размерности пространства призна-

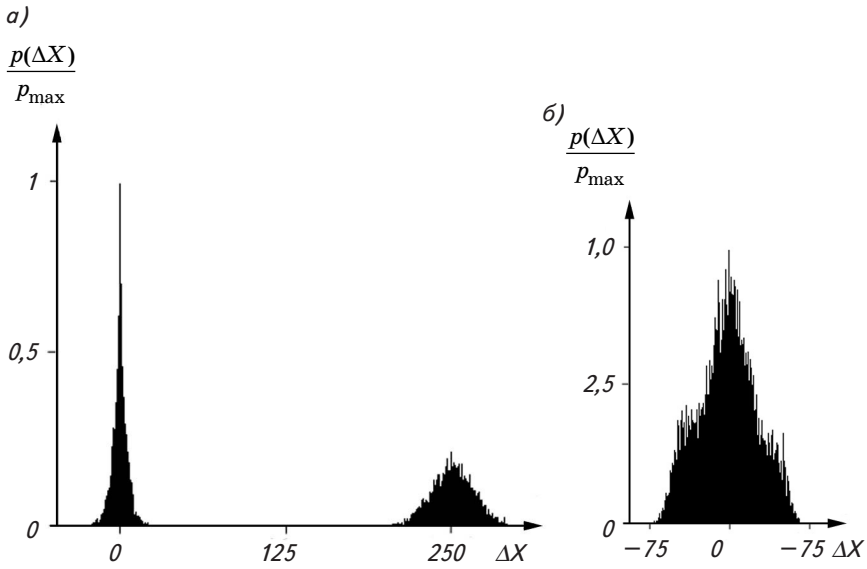

Рис. 1.5. Гистограмма невязок по X-координате: *а* — для окружности *1* (см. рис. 1.4); *б* — для окружности *2*

ков, которое, кстати, может быть существенно неевклидовым. К вопросу о применении энтропии в задачах такого типа мы еще вернемся в п. 2.5.

Минимизация энтропии соответствует минимизации длины описания невязок, что отвечает отрицательному логарифму от правдоподобия данных при конкретной модели. И действительно, вероятность получить данный набор D из N точек при некоторой выбранной модели h и соответствующим ей невязкам Δx_i равна (при условии статистической независимости величин невязок)

$$p(D \mid h) = \prod_{i=1}^{N} \Pr(\Delta X = \Delta x_i) \Rightarrow -\log_2 p(D \mid h) = \\ = -\sum_{i=1}^{N} \log_2 \Pr(\Delta X = \Delta x_i). \qquad (1.43)$$

Математическое ожидание количества слагаемых с одинаковым значением Δx_i равно $N \Pr(\Delta X = \Delta x_i)$. Отсюда получаем, что минус логарифм правдоподобия равен:

$$-\log_2 p(D \mid h) = NH(\Delta X). \qquad (1.44)$$

В общем случае нужно учитывать и второе слагаемое: $-\log_2 p(h)$, соответствующее длине описания самой гипотезы, о чем было много сказано при обсуждении правила Байеса в п. 1.4.1.

Таким образом, критерии, основанные на энтропии, сводятся к методу максимального правдоподобия. Они, хотя и расширяют такие критерии, как среднеквадратичное отклонение, должны рассматриваться как частный случай методов, основанных на принципе МДО.

1.4.4. Коэффициент корреляции и средняя взаимная информация

Как уже отмечалось, корреляция двух величин необходима для определения статистической зависимости между ними. Отсутствие корреляции между двумя случайными величинами X и Y означает, что $\rho_{XY} = 0$. К сожалению, это условие оказывается гораздо более слабым, чем статистическая независимость X и Y, выражающаяся через соотношение (плотностей) распределений вероятностей $p_{XY}(x, y) = p_X(x)p_Y(y)$.

В то время как отсутствие корреляции подразумевает

$$E_{XY}[XY] - E_X[X]E_Y[Y] = 0, \qquad (1.45)$$

для статистически независимых случайных величин верно соотношение

$$E_{XY}[\varphi_1(X)\varphi_2(Y)] - E_X[\varphi_1(X)]E_Y[\varphi_2(Y)] = 0 \qquad (1.46)$$

для любых (определенных всюду на множествах X и Y соответственно) функций φ_1 и φ_2. Несложно установить, что при выполнении равенства (1.45) выполняется также и равенство (1.46), но только при линейных функциях φ_1 и φ_2. Аналогично и значение коэффициента корреляции инвариантно (с точностью до знака) к линейным преобразованиям случайных величин.

Как было замечено в п. 1.3.2, для статистической независимости X и Y необходимо и достаточно, чтобы их совместная энтропия была равна $H(X, Y) = H(X) + H(Y)$, что равносильно равенству нулю средней взаимной информации $I(X, Y) = 0$. В случае нормального распределения случайных величин так же, как минимум энтропии соответствует минимуму среднеквадратичного отклонения, равенство нулю

средней взаимной информации совпадает с равенством нулю коэффициента корреляции.

Равенство нулю средней взаимной информации означает выполнение уравнения (1.46) при любых функциях φ_1 и φ_2, а равенство нулю коэффициента корреляции — только при линейных. Поскольку критерии совпадают в случае нормально распределенных случайных величин, функциональная связь между которыми может быть только линейной, то говорят, что средняя взаимная информация обобщает коэффициент корреляции на случай произвольных распределений (а не наоборот).

Покажем, к чему приводит нарушение ограничения линейности на следующем примере. Пусть $\{x_t\}_{t=1}^{N}$ — отсчеты (результаты измерений) некоторой случайной величины, а $\{y_{t+\Delta t} = \varphi(x_t) + n_t\}_{t=1}^{N}$ — отсчеты другой случайной величины, связанной с первой величиной функциональной зависимостью $\varphi(x)$ со случайными отклонениями n_i, являющимися отсчетами некоррелированного гауссова шума (Δt — неизвестное смещение между двумя выборками). Можно рассчитать значение коэффициента корреляции и взаимную информацию между выборками при различных значениях смещения для нахождения последнего. Корректный критерий сходства должен давать максимальное значение при истинном смещении.

На рис. 1.6 представлены три такие последовательности отсчетов. Первая из них рассматривается как опорная, сдвиг относительно которой следует определить для двух других. Как видно из рис. 1.7 и 1.8, корреляционная функция дает верный результат, когда функциональная зависимость отсутствует (такой же результат будет и при линейной зависимости), но уже при квадратичной зависимости между величинами максимум корреляции смещается от истинного значения сдвига и не может служить адекватной мерой сходства.

В отличие от коэффициента корреляции, средняя взаимная информация позволяет найти истинное смещение не только для линейно преобразованных величин, но также и при произвольной функциональной зависимости между ними (рис. 1.9, 1.10). Данный результат, хоть и является очевидным с теоретической точки зрения, тем не менее оказывается неожиданным и весьма полезным (см., например,

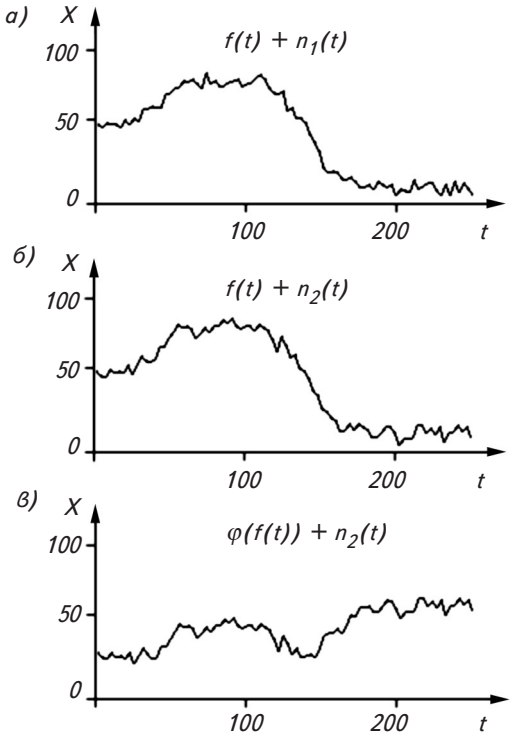

Рис. 1.6. Функции, образованные из некоторой регулярной функции $f(t)$ следующим образом: *а*, *б* — добавлением некоррелированного гауссова шума $n_1(t)$ и $n_2(t)$ соответственно; *в* — преобразованием с помощью регулярной функции $\varphi(f) = af^2 + bf + c$ (a, b, c — некоторые коэффициенты) с добавлением такого же шума $n_2(t)$

[63]) при применении на практике: ведь между представленными на рис. 1.6, *а* и *в* функциями даже на глаз сложно заметить сходство.

При максимизации средней взаимной информации производится поиск такого сдвига между двумя наборами данных, при котором эти данные можно совместно описать наиболее эффективно. Иными словами, этот подход можно также рассматривать с точки зрения принципа МДО. Как и в случае критериев, основанных на энтропии, данный принцип позволяет расширить подходы, основанные на средней взаимной информации. Такое расширение возможно по нескольким направлениям.

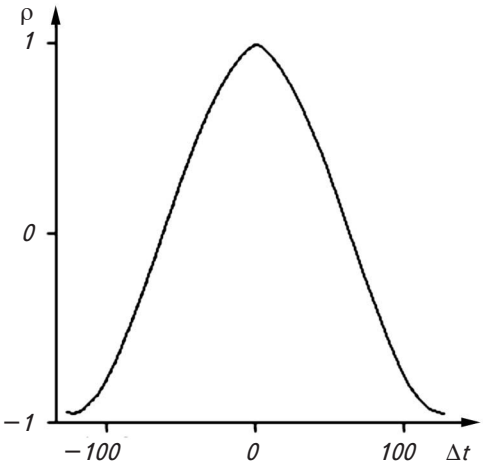

Рис. 1.7. Корреляционная функция (зависимость коэффициента корреляции от смещения Δt) для последовательностей измерений, представленных на рис. 1.6, *а*, *б* (корреляционный максимум точно соответствует истинному (нулевому) смещению; при проведении вычислений последовательности рассматривались как циклические)

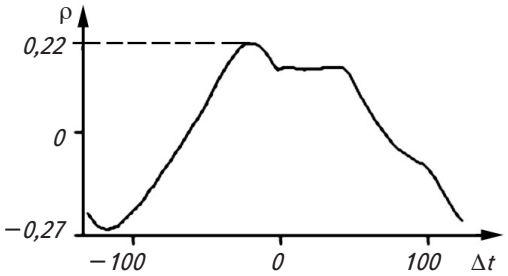

Рис. 1.8. Корреляционная функция для последовательностей измерений, представленных на рис. 1.6, *а*, *в* (при нелинейной зависимости между случайными величинами происходит разрушение корреляции: сам коэффициент корреляции близок к нулю при любых сдвигах, а его максимум приходится на случайное, отличное от истинного, смещение)

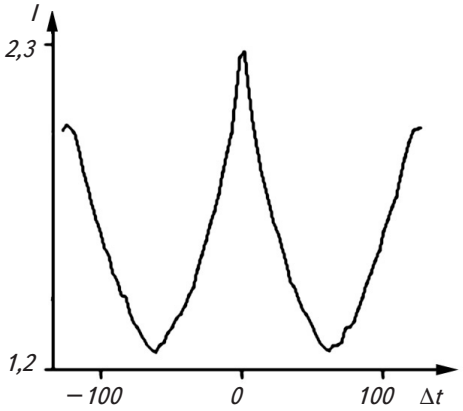

Рис. 1.9. Зависимость средней взаимной информации от сдвига между последовательностями измерений, представленных на рис. 1.6, *а*, *б* (положение максимума соответствует истинному смещению)

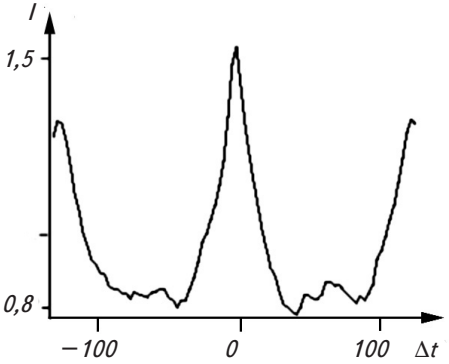

Рис. 1.10. Зависимость средней взаимной информации от сдвига между последовательностями измерений, представленных на рис. 1.6, *а*, *в* (положение максимума соответствует истинному смещению)

Во-первых, можно не ограничиваться лишь сдвигом между наборами данных, а применять более сложные преобразования координат. Приложением, в котором это необходимо, является, например, задача совмещения изображений. В этой задаче изображения могут быть преобразованы друг относительно друга не только посредством сдвига, но и посред-

ством поворота, а в более общем случае пространственное преобразование может быть аффинным или проективным. Более сложное преобразование всегда будет давать бóльшую среднюю взаимную информацию, чем более простое. В связи с этим необходимо учитывать также и длину описания модели в дополнение к средней взаимной информации.

Во-вторых, если руководствоваться принципом МДО, то при совместном описании двух наборов данных помимо установления соответствия между ними необходимо также минимизировать длину описания каждого из них. Таким образом, необходимо строить модель, описывающую каждый из наборов данных, и согласовывать эти модели между собой для достижения минимальной длины описания. По сути, это задача интеграции (или смешения) данных от различных источников, которая является очень важной и сложной, и, как видно, принцип МДО также может оказаться здесь полезным.

1.4.5. Проблема информативности модели

Теория информации связывает понятия вероятности и количества информации. Последнее оказывается гораздо удобнее и естественнее для определения априорных вероятностей гипотез. Априорные вероятности теперь выражаются через сложность гипотез (через количество содержащейся в них информации). Запись правила Байеса в теоретико-информационных терминах дает принцип Минимальной Длины Описания.

С другой стороны, принцип МДО можно считать расширением классических критериев сравнения гипотез. Это отчетливо видно по приведенным примерам и по популярности более частных информационных критериев, таких как энтропия и средняя взаимная информация.

Однако здесь нам все еще не удается решить основную проблему, которая была указана для правила Байеса. Как было замечено, для корректного определения количества информации в рамках классической теории информации требуется знать модель источника сообщений. Но именно она и не известна в задачах индуктивного вывода. Перебор всех моделей и выбор той из них, которая дает минимальное количество информации, казалось бы, должен дать искомый результат. Однако всегда есть возможность запаковать име-

ющийся набор данных в 1 бит, выбрав надлежащую модель источника [17]. Эта модель будет гласить, что имеющийся набор данных — это единственный набор, который можно получить от источника, а сам набор данных будет обозначен, скажем, однобитовой строкой «1». Естественно, такое решение неприемлемо и, согласно принципу МДО, необходимо также учитывать длину описания (сложность) модели.

И действительно, в данном примере «моделью» является таблица перекодировки, содержащая весь исходный набор данных, который тоже необходимо сохранить вместе с закодированными данными, чтобы была возможность восстановить их обратно. Интуитивно понятно, что сжатия тут происходить не будет. Таким образом, энтропия будет являться корректным способом оценивания количества информации, только если можно надежно определить модель источника (например, распределение вероятностей сообщений). Оценивание сложности модели не может быть выполнено в рамках классической теории информации, поскольку не известны вероятности появления тех или иных моделей.

Единственное, что можно сделать, это задать некоторый язык представления, в котором каждой модели будет конструктивно назначаться определенная длина. Это хотя и упрощает задание априорных вероятностей на практике, но, по сути, не отличается от него. Другими словами, проблема априорных вероятностей остается, превращаясь в проблему задания адекватного языка описания.

Интересно, что при неизвестной модели источника количество информации в сообщении оказывается зависимым от имеющейся априорной информации. Значит, информативность сообщения естественным образом оказывается субъективной.

Одной из не рассмотренных здесь нами, но актуальных проблем при классическом подходе к определению количества информации является задача оценивания плотности распределения вероятностей по имеющимся отсчетам случайной величины с целью определения ее энтропии (обсуждение этой проблемы см., например, в работах [63, 64]). Еще более острой она становится при вычислении совместных распределений (для вычисления, например, средней взаимной информации), так как при этом размерность пространства увеличивается в два раза, а также, когда размер выборки мал.

Оценивание плотностей вероятности является одной из центральных проблем в статистических методах обучения

и возникает во многих задачах. Мы ее кратко коснемся при обсуждении методов распознавания образов (см. пп. 2.3.4 и 2.3.5). Пока лишь заметим, что задача определения плотности распределения вероятностей, служащая для решения проблемы выбора модели, сама превращается в такую же проблему.

Способом преодоления основной проблемы, заключающейся в определении длины описания модели, может являться привлечение понятия количества информации, основанного не на теории вероятностей, а на комбинаторной основе, то есть на количестве символов в строке. Поскольку модель также можно записать в некоторой символьной форме, как и перекодированные с ее помощью данные, т. е. позволяет корректно вычислить количество информации.

По утверждению А. Н. Колмогорова, реальная сущность энтропии держится на чисто комбинаторных предположениях, которые несравненно более слабые, чем привлеченные К. Шенноном вероятностные предположения. Основная идея А. Н. Колмогорова [65] заключалась в том, что теория информации должна предшествовать теории вероятностей, а не основываться на ней, поскольку в отличие от последней основания теории информации по самой своей сути должны иметь конечный комбинаторный характер.

Одним из способов такого определения количества информации является ее определение на основе формального понятия алгоритма. Тогда количество информации, содержащейся в некотором наборе данных, понимается как программа, обладающая минимальной длиной, которая способна этот набор данных породить. В этом случае, однако, априорно задается модель устройства, выполняющего соответствующие программы, что также вносит некоторую субъективность. Но прежде чем рассмотреть более подробно этот вопрос, обратимся к понятию алгоритма.

1.5. МАШИНА ТЬЮРИНГА И АЛГОРИТМИЧЕСКАЯ СЛОЖНОСТЬ

1.5.1. Понятие алгоритма

Для математиков было естественным задуматься не только над решением определенных математических задач, но и над рассмотрением самого процесса решения с матема-

тической точки зрения. Попытка анализа процесса решения задач как математического объекта приводит к необходимости его формализации. Под алгоритмом можно понимать заданную в явном виде последовательность действий, выполнив которую при некоторых исходных данных, можно решить задачу. Понятие алгоритма в его общем виде, как и понятие информации, принадлежит к числу основных первичных понятий математики, не допускающих полного определения через более простые понятия. Любое уточнение этого понятия приводит к его сужению.

Но для ответа на вопрос, для любой ли строго поставленной математической задачи существует алгоритм ее решения, понятие алгоритма необходимо уточнить. Как показал в 1931 г. Курт Гёдель [66], для некоторого формально заданного класса алгоритмов существуют алгоритмически неразрешимые математические проблемы. Это вызвало необходимость установления наиболее общей формализации понятия алгоритма, чтобы проверить, будут ли верны результаты Гёделя и в рамках этого формализма.

Такие уточнения понятия алгоритма предложили в 1936 г. Эмиль Пост и Алан Тьюринг [67, 68]. Известны также другие формальные модели алгоритма, например предложенное А. А. Марковым понятие нормального алгорифма [69], лямбда-исчисление Алонзо Чёрча [70], формализм рекурсивных функций, основные идеи которого были предложены Эрбраном и развиты Гёделем [71], и теория нормальных систем Э. Поста [72]. Все эти формализмы оказываются эквивалентными в том смысле, что проблемы, обладающие решениями в рамках одного формализма, также обладают решениями и в рамках остальных формализмов. Доказательство эквивалентности некоторых из перечисленных определений понятия алгоритма можно найти, например, в работе [73, гл. 5]. Однако определение алгоритма как программы для некоторого гипотетического автоматического устройства, за которым сейчас закрепилось название «машина Тьюринга», является, похоже, наиболее удобным для реализации (но не обязательно для абстрактных математических доказательств), так как алгоритм, по сути, и есть механический процесс обработки информации.

Вероятно, поскольку понятие алгоритма связано, в первую очередь, с формализацией процесса решения именно математических проблем, машина Тьюринга представляет собой модель устройства, имитирующего элементарные опе-

рации человека над строками символов при проведении им вычислений. В общем виде машина Тьюринга — это устройство, которое может находиться в конечном числе внутренних состояний, а также обладает лентой, реализующей бесконечную внешнюю память. В каждой клетке ленты, на которые она разделена, может быть записана любая из букв некоторого (фиксированного для данной машины Тьюринга) алфавита. Также машина Тьюринга снабжена головкой, движущейся по ленте и способной читать и записывать символы в клетки ленты. В процессе развития теории алгоритмов было предложено множество модификаций машины Тьюринга. Рассмотрим одно из них.

1.5.2. Формализм машины Тьюринга

Машиной Тьюринга (МТ) называется пятерка $T = (Q, \mathrm{A}, \delta, q_0, F)$. Здесь Q — конечное множество внутренних состояний, содержащее в себе начальное состояние $q_0 \in Q$ и множество заключительных состояний $F \subseteq Q$. Множество A — это алфавит, состоящий из конечного множества ленточных символов, один из которых — пустой символ Λ. Пусть направление движения головки по рабочей ленте задается одним из двух значений: L — влево и R — вправо. Тогда отображение

$$\delta : Q \times \mathrm{A} \to Q \times (\mathrm{A} \setminus \{\Lambda\}) \times \{L, R\},$$

которое может быть не определено для некоторых аргументов, задает таблицу переходов МТ, определяющих правила ее функционирования. Это отображение, однозначно описывающее конкретную машину T в алфавите A, обычно называется схемой или *программой данной машины Тьюринга* и часто отождествляется с самой МТ.

Обозначим через $\Gamma = (\mathrm{A} \setminus \{\Lambda\})^*$ множество всех возможных строк, составленных из символов данного алфавита A, не включающих пустого символа Λ.

Полное описание текущего состояния машины T называется ситуацией, или *конфигурацией машины Тьюринга* и определяется тройкой (q, α, n), где $q \in Q$ — текущее внутреннее состояние; строка $\alpha \in \Gamma$ (или $\alpha = a_1 a_2 \ldots a_N$, $a_i \in \mathrm{A} \setminus \{\Lambda\}$) описывает содержимое непустой части ленты; $1 \le n \le N + 1$ — номер текущей клетки, на которую указывает головка; N — число символов в непустой части ленты (или ее дли-

на). Тройка (q, α, n) называется *заключительной конфигурацией*, если $q \in F$.

Пусть $(q, a_1 a_2 \ldots a_N, n)$ — конфигурация машины T. Если в программе этой МТ есть *команда* $\delta(q, a_n) = (p, A, R)$, машина T переходит в новую конфигурацию:

$$(q, a_1 a_2 \ldots a_N, n) \underset{T}{\mapsto} (p, a_1 a_2 \ldots a_{n-1} A a_{n+1} \ldots a_N, n+1). \quad (1.47)$$

Таким образом, выполнение данной команды соответствует изменению внутреннего состояния со значения q на значение p, записи в n-ю ячейку символа A и перемещения головки на одну клетку вправо. Если перед выполнением команды положение головки было $n = N + 1$, то после ее выполнения длина непустой части ленты также увеличится на единицу. Несложно установить, как будет изменена конфигурация МТ и в случае команды $\delta(q, a_n) = (p, A, L)$. Отдельно лишь нужно оговорить ситуацию, при которой головка перемещается влево и $n = 1$, для МТ с лентой, бесконечной в одну сторону. Как правило, полагают, что такая ситуация соответствует заключительной конфигурации. Если отображение δ является недоопределенным, то конфигурации $(q, a_1 a_2 \ldots a_N, n)$, для которых нет соответствующих команд $\delta(q, a_n)$, также приводят к останову МТ, т. е. являются заключительными.

Таким образом, работа МТ состоит в последовательном (по шагам) преобразовании исходной конфигурации в последующие конфигурации в соответствии с программой машины. Последующая конфигурация однозначно определяется текущей конфигурацией, т. е. текущим внутренним состоянием и символом, записанным в клетке, на которую указывает головка. Смена конфигураций происходит до тех пор, пока не будет достигнута какая-либо заключительная конфигурация, т. е. пока не произойдет останов машины. Эта конфигурация и считается результатом работы МТ. Естественно, не всякая МТ останавливается при любой исходной конфигурации.

В зависимости от того, какая составляющая заключительной конфигурации (q, α, n) считается более существенной (α или q), машина Тьюринга, в частности, может рассматриваться либо как автомат-преобразователь цепочек символов (α — результат преобразования), либо как автомат-распознаватель цепочек (q — номер класса, к которому принадлежит входная цепочка).

Существует много модификаций МТ. Например, часто в определение МТ добавляют подмножество $\Sigma \subset A$, являющееся множеством входных символов, не содержащим символа Λ и, возможно, некоторых других символов. Могут рассматриваться машины Тьюринга, в которых перемещение головки в результате выполнения команды не ограничивается лишь сдвигами влево или вправо на одну клетку, но может принимать и другие значения, например, оставаться на месте или смещаться на произвольное число клеток. Иногда оказывается удобным рассматривать МТ с лентой, являющейся бесконечной в обе стороны, а также многоленточные МТ с несколькими головками для чтения и записи. Все эти модификации МТ являются эквивалентными и обозначаются одним и тем же названием — машина Тьюринга, — которое объединяет в себе целый класс абстрактных вычислительных машин. В связи с этим использование данного термина без уточнения конкретного типа МТ может приводить к путанице.

Модификации МТ, выделяющиеся по названиям в отдельные классы, — это вероятностные и недетерминированные машины Тьюринга. Для обоих этих типов машин программа может содержать различные команды с одинаковыми левыми частями, и вместо одной последовательности конфигураций рассматриваются всевозможные последовательности конфигурации, согласованные с программой. Для вероятностных МТ при этом также считается вероятность каждой конфигурации на основе вероятности, назначенной для каждой команды, в то время как для недетерминированной машины Тьюринга проверяется сама возможность получить какую-либо конфигурацию в результате применения различных комбинаций допустимых команд.

1.5.3. Универсальная машина Тьюринга

Как было замечено еще Аланом Тьюрингом, отображение δ, задающее программу или таблицу переходов конкретной МТ, можно закодировать в виде конечной строки символов. Тогда можно представить себе такую машину Тьюринга, которая бы воспринимала в качестве входа строку-описание таблицы переходов некоторой МТ (а также ее начальное состояние), за которой следовала бы сама входная строка, а затем вычисляла выходную строку, какую бы по-

лучила описанная в начале ленты МТ. Как было показано Тьюрингом, такая МТ действительно возможна, и поскольку она может эмулировать любую другую МТ, она получила название *универсальной машины Тьюринга* (УМТ).

Пусть $\alpha \in \Gamma$ — вход (содержимое непустой части ленты) для машины Тьюринга T. Обозначим через $T(\alpha) \in \Gamma$ результат работы машины T при данном входе, а через $d[T]$ — символьное описание этой машины. Тогда, если U — это УМТ, то

$$U(d[T], \alpha) = T(\alpha) \qquad (1.48)$$

для любой машины T в алфавите A. Здесь $d[T]$ можно считать программой для машины U, поскольку это описание отображения δ, называемого программой для машины T, строку α — данными. Для удобства также можно считать, что они подаются на двух различных лентах.

Помимо того, что такая универсальная машина существует, оказывается также, что минимальное число шагов, необходимых машине U для имитации любого шага машины T, является полиномом от числа символов в описании $d[T]$. То, что эта зависимость не является экспоненциальной, важно при рассмотрении вопросов, касающихся сложности вычисления.

Известны также следующие результаты. Если язык порождается неограниченной грамматикой (о формальных грамматиках разговор пойдет в п. 4.2), то он допускается некоторой машиной Тьюринга. И наоборот, если язык допускается некоторой машиной Тьюринга, то он порождается неограниченной грамматикой. А значит, любой неограниченный язык допускается УМТ. Также известно, что УМТ может вычислить любую рекурсивную функцию (поэтому часто действие некоторой МТ рассматривают как вычисление соответствующей функции). Эти и ряд других результатов об эквивалентности различных формальных определений алгоритмов позволили тезису Чёрча—Тьюринга стать общепризнанным.

Согласно этому тезису, проблемы, разрешимые универсальной машиной Тьюринга, — это в точности те проблемы, которые вообще имеют какое-либо алгоритмическое решение. Другими словами, интуитивное понятие алгоритма предлагается отождествить с УМТ или какой-либо другой эквивалентной моделью в силу того, что не удается найти противоречащие этому тезису примеры, и предполагается, что любое разумное определение алгоритма, какое только можно

предложить, будет эквивалентно уже имеющимся. В терминах теории рекурсивных функций это означает, что рекурсивные функции совпадают с вычислимыми функциями.

Часто расширяют понятие универсальной машины Тьюринга на машины Тьюринга, эквивалентные УМТ по возможности вычислять любую функцию, но не эмулирующие напрямую другие МТ. В связи с этим естественного разделения на программу $d[T]$ и данные α для них нет, поэтому в дальнейшем мы будем преимущественно использовать запись $U(\alpha)$, подразумевая, что строка α содержит как программу, так и ее входные данные. Такие УМТ могут быть весьма простыми. В табл. 1.1 приведены параметры наиболее известных простых УМТ.

Поскольку УМТ — это частный случай машины Тьюринга, то одна универсальная машина Тьюринга может эмулировать другую УМТ. В связи с этим обычно не рассматривается конкретная машина Тьюринга, а под программой для УМТ подразумевается программа, написанная на некотором языке программирования высокого уровня. Поскольку каждый (неспециализированный) компьютер является физической моделью УМТ (наиболее существенное отличие заключается в конечности памяти компьютера), то все такие компьютеры эквивалентны, с точки зрения их принципиальной способности выполнять произвольные алгоритмы. Интересно, что формализм искусственных нейронных сетей также может рассматриваться как эквивалент универсальной машины Тьюринга [74, с. 448; 75, с. 34].

Можно также заметить, что человек способен эмулировать действия любой МТ по ее описанию и входной строке. Тезис Чёрча—Тьюринга можно интерпретировать как утверждение, что человек является универсальной машиной Тьюринга. Из этого, в частности, следует, что человеческое мышление можно будет эмулировать с помощью другой УМТ, коль скоро у нас появится его исчерпывающее описание, которое можно подать на вход машины Тьюрин-

Таблица 1.1. **Параметры наиболее простых известных универсальных машин Тьюринга**

Параметр	Значение						
Число символов в алфавите	2	3	4	5	6	10	18
Число внутренних состояний	22	10	7	5	4	3	2

га. Мы не будем обсуждать этот скользкий вопрос, а заметим лишь, что если человека рассматривать как машину Тьюринга, то нужно учесть, что «входной и выходной лентой» для него является окружающий мир, который, например, меняется не только (и не столько) под воздействием человека, но и независимо от него. Рассмотрение же всей Вселенной в качестве УМТ, в которой таблица переходов задается физическими законами, также является далеко не бесспорным.

Еще один, безусловно, важный вопрос, которого мы касаться практически не будем, — это вопрос об алгоритмически неразрешимых проблемах, т. е. таких проблемах, для которых не существует алгоритма их решения. Классическим примером является проблема останова, которая заключается в том, что необходимо определить, остановится ли когда-нибудь данная машина Тьюринга при данном содержимом входной ленты. Смысл алгоритмической неразрешимости этой проблемы заключается в том, что не существует такой программы для МТ, которая бы для любой другой программы (например, для самой себя) за конечное время определяла, остановится ли она когда-нибудь или будет работать бесконечно.

По этим, а также другим незатронутым здесь вопросам теории алгоритмов заинтересованному читателю рекомендуем обратиться к специальной литературе (см., например [76]). Нас же интересует строгое определение понятия сложности, к рассмотрению которого мы и переходим.

1.5.4. Понятие алгоритмической сложности

Сложность алгоритма зависит от того, какое именно формальное определение алгоритма привлекается. Поскольку различные понятия алгоритма являются эквивалентными, часто рассматривают только один из вариантов алгоритмической сложности. Исторически первое определение алгоритмической сложности, которое также является и наиболее используемым в задачах индуктивного вывода, было сформулировано в 1960-е годы и основывается на определении понятия алгоритма с помощью машины Тьюринга. Это понятие алгоритмической сложности было независимо и с различной мотивацией разработано А. Н. Колмогоровым [77, 78], Р. Соломоновым [7] и Г. Чайтином [79; 80].

Для простоты обычно рассматривается алфавит, состоящий из двух символов (не считая пустого): $A = \{0,1\} \cup \{\Lambda\}$, т. е. рассматриваются бинарные строки $\alpha \in \{0,1\}^*$.

Тогда *алгоритмической сложностью* или сложностью по Колмогорову (см., например, [77, 81, 82]), $C(\beta)$ некоторой строки $\beta \in \{0,1\}^*$ по отношению к данной универсальной машине Тьюринга U называют длину наиболее короткой программы α, которая на выходе печатает строку β, а затем останавливается: $C(\beta) = \min\limits_{\alpha}\left[l(\alpha) \mid U(\alpha) = \beta\right]$, где $l(\alpha)$ — число символов в строке α.

Первоначальная модель алгоритма, предложенная Тьюрингом, включает специальный символ Λ, отделяющий входную строку или помечающий незадействованные ячейки. В классической теории информации такие коды называются *кодами с разделительными знаками*. Использование этого формализма и дает описанную алгоритмическую сложность $C(\beta)$. Однако чаще полезнее бывает рассматривать *префиксную алгоритмическую сложность* $K(\beta)$. Ее отличие от простой алгоритмической сложности заключается в том, что строки-программы α должны являться префиксными кодами (т. е. ни одна программа не может являться началом другой программы). Благодаря этому возникает возможность использовать *коды без разделительных знаков*, которые при этом обладают свойством однозначного декодирования.

Рассмотрим частный вариант префиксных кодов — так называемые *коды с саморазграничением* (см., например, [19]), которые определяются следующим образом. Сначала вводится отображение множества натуральных чисел в множество бинарных строк: $1 \to 0$, $2 \to 1$, $3 \to 00$, $4 \to 01$ и т. д. (также числу «0» ставится в соответствие пустая строка ε). Таким образом, мы сможем закодировать длину бинарной строки $l(\alpha)$ также в виде некоторой бинарной строки.

Пусть есть некоторая строка $\chi = x_1 x_2 \ldots x_N$, соответствующая числу, обозначающему длину бинарной строки $l(\alpha)$. Тогда ей соответствует код с саморазграничением $\underline{\chi} = x_1 x_1 x_2 x_2 \ldots x_N \neg x_N$. Здесь $\neg x_N = 0$, если $x_N = 1$ и $\neg x_N = 1$, если $x_N = 0$. Естественно, считывая такую строку, мы можем определить, где она заканчивается.

Теперь для произвольной строки α можно ввести следующий стандартный код с саморазграничением: $\alpha' = \underline{l(\alpha)}\alpha$ (т. е. перед строкой α записывается ее длина таким образом, чтобы можно было сначала корректно считать эту дли-

ну, а затем, зная длину строки, считать уже саму строку). Просто_проверить, что выполняются следующие соотношения: $l(\alpha) = 2N$ и $l(\alpha') = N + 2\lceil \log_2 N \rceil$. Здесь $[\log_2 N]$ обозначает ближайшее целое число, большее, чем $\log_2 N$.

Таким образом, численное различие между двумя типами алгоритмической сложности заключается в логарифмическом слагаемом, которое определяет число дополнительных бит, необходимых для ограничения входной строки. Префиксная алгоритмическая сложность обладает рядом привлекательных свойств, которые полезны при определении длины описания, в частности, она допускает более корректное определение вероятности. Индивидуальные преимущества и недостатки обоих подходов к определению сложности описаны в работе [81], но какую из них использовать — вопрос договоренности.

Введем следующее понятие. *Условной* (префиксной) *алгоритмической сложностью* строки β относительно строки χ называется длина наикратчайшей программы, которая при исполнении на универсальной машине Тьюринга U печатает ее на выходе, если строка χ дана в качестве дополнительного входа: $K(\beta \mid \chi) = \min\limits_{\alpha} \left[l(\alpha) \mid U(\alpha, \chi) = \beta \right]$. Здесь строка α принадлежит множеству префиксных кодов. Формально можно записать $K(\beta) = K(\beta \mid \varepsilon)$, где ε — пустая строка. Строки α и χ могут либо подаваться на разных лентах, либо в качестве одной строки, полученной в результате их конкатенации, что допустимо, если это префиксные коды.

Существует ряд расширений понятия алгоритмической сложности. Примером может служить алгоритмическая сложность множества, которая позволяет корректно определять понятие сложности в случае, когда строка α является неточным описанием строки β, а именно: строка α определяется как наикратчайшая программа, печатающая некоторую строку из множества строк S, содержащего строку β. Нам эти расширения не понадобятся, и мы не будем их рассматривать, а перейдем к описанию некоторых свойств алгоритмической сложности.

1.5.5. Индивидуальная случайность бинарной строки

При обсуждении теоремы Байеса приводился пример с подбрасыванием монетки. Напомним, что проблема заключалась в том, что последовательность из тысячи выпавших «решек»

обладает той же вероятностью при использовании модели равновероятного выпадения «орла» и «решки», что и любая другая последовательность. В то же время она кажется более упорядоченной. Это может служить примером общей проблемы собственной случайности единичного объекта. Один из мотивов разработки понятия алгоритмической сложности и заключался в решении этой проблемы.

Сначала отметим следующее. Пусть некоторое число N — это длина строки, а $N-n$ — длина некоторой программы. Всего имеется 2^N строк длины N и, соответственно, $2^{N-n+1}-1$ программ длины $N-n$ и менее (если снять ограничение на то, что программа записывается в виде префиксного кода). Разные строки не могут порождаться одной и той же программой (обратное утверждение неверно).

Таким образом, среди 2^N строк длины N существует не более $2^{N-n+1}-1$ строк, алгоритмическая сложность которых не более $N-n$. В действительности, доля таких строк еще меньше, так как среди строк с длиной, большей N, могут также присутствовать строки, сложность которых не более $N-n$. Итак, доля строк, сложность которых на n меньше их длины, не превосходит примерно 2^{-n+1}, т. е. экспоненциально падает с ростом n (см., например, [83]).

Это позволяет считать строки, алгоритмическая сложность которых примерно соответствует их длине, случайными последовательностями символов. Таким образом, можно ввести понятие собственной случайности индивидуального объекта, что в принципе не допускается классической теорией вероятностей. Существуют и другие определения случайности, например, Мартин-Лёф [84] разработал набор «тестов на случайность» (см. также [82; 85–87]), которым, однако, удовлетворяет и определение случайности, основанной на алгоритмической сложности.

Алгоритмический подход к определению степени случайности бинарной строки дает интуитивно правдоподобные результаты. Так, возвращаясь к примеру с монеткой, можно утверждать, что программа, печатающая последовательность из тысячи нулей («решек»), будет существенно короче, чем программа, воспроизводящая действительно случайную последовательность нулей и единиц, которую можно получить, подбрасывая монетку. Для того чтобы это стало совершенно очевидно, можно сравнивать длины соответствующих программ на каком-либо языке программирования: для печати результатов подбрасывания «нормальной» монет-

ки придется эту последовательность указывать в программе в явном виде, в то время как для такой последовательности, как, например, «01010101...», потребуется лишь один простой цикл.

Выбранные из большого числа случайных строк немногие строки, алгоритмическая сложность которых заметно меньше их исходной длины, оказываются именно теми, какие бы и хотелось получить в качестве строк, содержащих регулярности. Так, если мы возьмем запись рационального числа в виде бесконечной периодической дроби и возьмем из этой записи несколько (повторяющихся) периодов в качестве тестируемой строки, то, очевидно, эту строку можно сжать до достаточно короткой программы.

Далее возьмем отрезок двоичной записи какого-нибудь иррационального числа (например, корня из двух). Если этот отрезок будет достаточно длинный, он будет идентифицирован как регулярный, так как существует достаточно короткая программа генерации представления числа $\sqrt{2}$. Из всего множества иррациональных чисел, обладающих бесконечной записью, мы можем представить себе лишь те из них, которые обладают конечной алгоритмической сложностью.

Рассмотрим еще более сложные числа. Для какого-нибудь известного трансцендентного числа, например π, может потребоваться еще более длинный отрезок, чтобы его алгоритмическая сложность оказалась меньше длины этого отрезка. Но этот отрезок будет конечным, несмотря на бесконечность числа знаков в двоичной (или десятичной) записи числа π. На самом деле, программы, написанные на языке Си и печатающие знаки чисел $\sqrt{2}$ и π, имеют приблизительно одинаковый размер. Примером программы, печатающей 2400 знаков числа π, может служить программа Д. Т. Винтера, состоящая из 158 символов (стремление уменьшить количество символов, конечно, сказалось и на оформлении программы):

```
int a=10000,b,c=8400,d,e,f[8401],g;
main(){for(;b-c;)f[b++]=a/5;
for(;d=0,g=c*2;c-=14,printf("%.4d",e+d/a),e=d%a)
for(b=c;d+=f[b]*a,f[b]=d%--g,d/=g--,--b;d*=b);}
```

Существуют и чуть более короткие программы. Для знакомства с ними и с принципом их построения можно обратить-

ся к Интернет-ресурсу http://numbers.computation.free.fr/, который был доступен на момент написания книги.

Любое вещественное число представимо в виде конечной или бесконечной бинарной строки. Множество конечных бинарных строк является счетным, в то время как множество бесконечных бинарных строк имеет мощность континуума, т. е. несравнимо больше первого множества. Можно заметить, что из бесконечного числа действительных чисел бесконечно малая доля обладает конечной алгоритмической сложностью (является вычислимой) и в этом смысле не является случайной. Все представимые человеком числа, часть из которых имеет бесконечную десятичную запись, являются именно такими числами, т. е. числами, имеющими конечную сложность! Можно придумать некое невычислимое число (например, число, у которого в i-м разряде стоит «1», если i-я программа для УТМ останавливается, и «0», если не останавливается), но потребность в таких числах в естественных науках сомнительна.

Понятие алгоритмической сложности можно перенести и на такие объекты, как функции (например, функции действительной переменной). Вернемся к примеру аппроксимации множества точек некоторой функцией. Основная трудность при использовании правила Байеса заключалась в том, что если мы априори ограничиваем класс функций, то в ряде случаев истинная модель выбрана быть не может ни при каком конечном наборе исходных данных. Например, это имеет место при аппроксимации набора точек $\left\{\left(x_i, y_i = \mathbf{e}^{x_i}\right)\right\}_{i=1}^{N}$ с помощью полиномов. Если же множество привлекаемых функций никак не ограничивать, то задача не может быть решена из-за чрезмерной избыточности пространства моделей: каждому набору точек соответствует бесконечно много моделей, априорные вероятности которых определить не представляется возможным.

Алгоритмическая сложность позволяет (по крайней мере, в теории) решить эту проблему. Практически любая функция, используемая человеком, может быть реализована с помощью программы конечной длины, а значит, обладает ненулевой вероятностью. Это касается и экспоненциальной зависимости в только что приведенном примере. Иными словами, программы для машины Тьюринга задают такое пространство моделей, которое может быть использо-

вано для аппроксимации функций заранее неизвестного вида, представленных конечным набором точек. Такого рода полнота является очень привлекательной чертой алгоритмической сложности.

К сожалению, алгоритмическая сложность оказывается невычислимой (см., например, [81]). Это тесно связано с неразрешимостью проблемы останова. Алгоритмическая сложность определяется в результате перебора программ, которые на выходе должны давать искомую строку. Определить, действительно ли произвольная программа даст на выходе эту строку, можно только выполнив ее, но она может никогда не остановиться, в чем и заключается невычислимость алгоритмической сложности. Существуют, однако, вычислимые приближения к алгоритмической сложности [8, 11].

С понятием индивидуальной случайности связан еще один очень интересный (но, по сути, риторический) вопрос [88]: почему наш мир является столь сильно сжимаемым в информационном смысле? Если, грубо говоря, представить описание мира в виде бинарной строки, то ее алгоритмическая сложность будет существенно меньше ее длины. Это говорит о том, что данная бинарная строка, а вместе с ней и мир, далеко не случайна и маловероятна в указанном выше смысле. А вспоминая о связи сложности и красоты, можно также заключить, что мир красив, гармоничен. К сожалению, обсуждение данного вопроса увело бы нас очень далеко от проблемы индуктивного вывода, поэтому заинтересовавшемуся читателю мы рекомендуем обратиться к литературе (например, [51]).

1.5.6. Алгоритмическая сложность как количество информации

Как мы помним, алгоритмическая теория информации была призвана преодолеть определенные затруднения классической теории информации при ее использовании для сравнения моделей. Исходя из только что рассмотренных примеров, можно надеяться, что это осуществимо, по крайней мере, частично. В связи с этим рассмотрим связь между алгоритмической сложностью и энтропией.

Пусть X — некоторая случайная величина и $x_1, \ldots, x_n$ — результаты ее независимых испытаний, образующих стро-

ку $\beta = x_1 \ldots x_n$. Можно закодировать эту последовательность, используя код Хаффмана. Полученная закодированная строка будет иметь длину (вернее, математическое ожидание длины), не превосходящую $n[H(X) + 1]$.

Можно представить себе машину Тьюринга T, которая будет по закодированной с помощью Хаффмана строке $\alpha = y_1 \ldots y_n$ (здесь y_i — кодовые слова, которые могут состоять из разного числа символов) воспроизводить исходную строку $\beta : T(\alpha) = \beta$ — и останавливаться. Пусть $d[T]$ — описание машины T, необходимое некоторой универсальной машине U, чтобы повторить ее действия. Поскольку $T(\alpha') = \beta \Leftrightarrow \alpha' = \alpha$ (так построены коды Хаффмана), то длина строки α и условная алгоритмическая сложность $K(\beta \mid d[T]) = \min_{\alpha'} \left(l(\alpha') \mid U(d[T], \alpha') = \beta \right)$ строки β совпадают. Поясним, что это означает. Чтобы воспроизвести исходную строку β по ее сжатому описанию α, необходимо знать таблицу перекодировки (а также способ кодирования). Описание $d[T]$ содержит эту информацию, т. е. содержит описание (модель) случайного процесса, порождающего цепочку слов. В нашем случае эта модель считалась известной (стационарный источник без памяти с известным распределением вероятностей).

С другой стороны, математическое ожидание длины строки α можно оценить, как $nH(X)$. Таким образом, имеем оценку $K(x_1 \ldots x_n \mid d[T]) \approx nH(X)$, где машина T производит декодирование по заложенной в эту машину таблице перекодировки входной строки, закодированной алгоритмом Хаффмана.

Вероятность получить алгоритмическую сложность $K(\beta|d[T])$, меньшую, чем $nH(X)$, экспоненциально убывает с ростом разницы между ними. Например, присутствует возможность того, что все слова в строке $x_1, \ldots, x_n$ имеют наиболее высокую вероятность; алгоритмическая сложность такой строки будет существенно меньше, чем количество информации, оцененное по энтропии, но доля таких строк очень мала. Таким образом, алгоритмическая сложность согласуется с количеством информации, введенным в теории Шеннона, но она четко показывает нам, что оценка количества информации $nH(X)$ имеет смысл, только когда модель источника заранее известна.

Но что если модель нам не дана априори? Пусть μ — некоторая модель (произвольная программа для машины U). В слу-

чае, когда модель априори неизвестна, корректной оценкой количества информации в строке β следует считать безусловную алгоритмическую сложность $K(\beta) = \min_{\alpha',\mu} \left[l(\alpha') + l(\mu) \mid U(\mu, \alpha') = \beta \right]$. Здесь мы просто разбили строку, фигурирующую в определении алгоритмической сложности, на две, одну из которых интерпретируем как программу (модель), а другую — как данные. Очевидно, при $\mu = d[T]$, где T — программа, декодирующая код Хаффмана для конкретной таблицы перекодировки, и $\alpha' = \alpha$ имеем $U(d[T], \alpha) = \beta$, но равенство $K(\beta) = l(\alpha) + l(d[T])$ будет выполняться только тогда, когда не удалось найти более хорошую модель. Если исходные значения $x_1, \ldots, x_n$ действительно статистически независимы, то для большинства реализаций этого случайного процесса для его истинной модели T будет достигаться минимальная длина описания. Таким образом, возвращаемся к проблеме выбора модели.

1.6. АЛГОРИТМИЧЕСКАЯ СЛОЖНОСТЬ И СРАВНЕНИЕ ГИПОТЕЗ

1.6.1. Предсказание на основе алгоритмической вероятности

Впервые идея использования сжатия данных в целях индуктивного вывода и предсказания была высказана Р. Соломоновым [7]. Предложенный им метод он назвал *алгоритмической вероятностью* (АЛВ). По сути, подход АЛВ может быть рассмотрен как байесовский метод экстраполяции конечных бинарных строк, в котором необходимые априорные распределения вероятностей получаются в предположении, что строки были сгенерированы как выход УМТ [33].

Будем говорить, что строка α — это *описание строки* β по отношению к машине U, если $\beta = U(\alpha)$. Одна строка может иметь множество различных описаний. Пусть α_i — это i-е описание строки β, длина (количество символом) которой равна $l(\alpha_i)$. Строке α_i можно поставить в соответствие вероятность того, что она появится в качестве случайного входа к машине U. Поскольку в теории информации Шеннона количество собственной информации в некотором сообщении x равно $I(x) = = - \log_2 P(x)$, то здесь берется вероятность, равная $P(\alpha_i) = 2^{-l(\alpha_i)}$.

Нетрудно заметить, что такая вероятность получается ненормированной в том смысле, что сумма вероятностей по всем возможным строкам равна бесконечности: имеем две строки длины 1 (их вероятности 0,5), 4 строки длины 2 (их вероятности 0,25) и т. д. Эта проблема снимается, если в качестве строк рассматривать только некоторое множество префиксных кодов, что заметил в 1974 г. Л. А. Левин [89]. Сумма вероятностей для всех строк в таких множествах оказывается равной единице.

Теперь, чтобы получить полную вероятность того, что строка β будет произведена машиной U при случайном содержимом входной ленты, следует подсчитать сумму вероятностей всех ее описаний [33]:

$$P(\beta) = \sum_{i=1}^{\infty} 2^{-l(\alpha_i)}. \qquad (1.49)$$

Это распределение задает *универсальное распределение* априорных вероятностей на множестве бинарных строк. Наибольшее из слагаемых $2^{-l(\alpha_i)}$ соответствует некоторому описанию α_i, обладающему наименьшей длиной $l(\alpha_i)$, которая, по сути, является алгоритмической сложностью строки β. Таким образом, слагаемое $2^{-K(\beta)}$ является лишь одним из многих в данной сумме. Если при применении алгоритмической сложности производится выбор одного лучшего описания, то алгоритмическая вероятность возникает в результате взвешенного суммирования по всем описаниям, при этом выбор какого-либо конкретного описания не производится. Поэтому можно сказать, что алгоритмическая вероятность соотносится с алгоритмической сложностью так же, как методы предсказания и методы выбора лучшей гипотезы на основе правила Байеса (см. п. 1.2.2).

Идея АЛВ была предложена для экстраполяции бинарных строк [7]. Если на вход машины U подается некоторая строка β_0, строку β можно считать продолжением этой исходной строки с помощью описания α_i, если $\beta = U(\beta_0\alpha_i)$; вероятность этой экстраполяции равна $2^{-l(\alpha_i)}$. Если аналогичным образом взять сумму по всем описаниям, то получим распределение вероятностей по всем экстраполяциям строки β_0:

$$P(\beta \mid \beta_0) = \sum_{i=1}^{\infty} 2^{-l(\alpha_i)}, (\forall i)\big(U(\beta_0\alpha_i) = \beta\big). \qquad (1.50)$$

Следует сделать два замечания.

Во-первых, как алгоритмическая сложность, так и АЛВ, зависят от того, какая именно универсальная машина используется. Это хорошо иллюстрирует следующий пример. Пусть в качестве входа дана последовательность чисел: 1, 2, 2, 4, 8, 32, 256, ..., т. е. каждое последующее число получается перемножением двух предыдущих. Если рассматриваемая машина умеет умножать числа, то программа, печатающая эти числа на выходе, будет короче, чем, если бы машина умела только складывать числа. Это означает, что данные меры также зависят от априорной информации, заложенной в машину.

Во-вторых, как и алгоритмическая сложность, АЛВ является невычислимой [7], а вероятность, определенная таким образом, — недостижимым на практике идеалом. В связи с этим вводится «практическая» алгоритмическая вероятность, которая является функцией трех аргументов [17]:

1) собственно экстраполируемых данных;

2) априорной информации, которая уникально характеризуется нашим выбором универсальной опорной машины;

3) имеющимися вычислительными ресурсами — временем и памятью.

Были предприняты определенные попытки разработать систему индуктивного вывода, реализующую понятие практической АЛВ (теоретические основы могут быть найдены в работах [90–92]; более практическую направленность имеют статьи [93–95]).

Как замечает сам автор подхода [17], в исследованиях по искусственному интеллекту «слепой поиск» обычно непрактичен, так как размер пространства поиска увеличивается экспоненциально с увеличением размерности задачи (так называемый *«комбинаторный взрыв»*). При решении практических задач применяются различные эвристические приемы, позволяющие существенно ограничить пространство поиска на основе знаний о предметной области.

Существует несколько подходов, близких к АЛВ. Несмотря на то, что эти подходы получили гораздо большее практическое распространение, они разделяют те же сложности, что и алгоритмическая вероятность. Поэтому прежде чем переходить к более подробному рассмотрению указанных сложностей, скажем несколько слов о самих методах.

1.6.2. Алгоритмическая сложность в индуктивном выводе

Как уже упоминалось, теоретико-информационный подход к индуктивному выводу объединяет ряд методов, в основу которых положен один из принципов: минимальной длины сообщения (МДС), минимальной длины описания (МДО) и идеальная МДО.

Идею минимальной длины сообщения (независимо от алгоритмической вероятности Р. Соломонова) предложили К. Уаллас и Д. Болтон [8] в 1968 г. Передача сообщения в этом случае осуществлялась кодом, состоящим из двух частей, заменяющих отрицательные логарифмы в правиле Байеса длинами кодов Шеннона—Фано.

В 1978 г. Риссанен независимо от Уалласа, но вдохновленный идеями Р. Соломонова и А. Н. Колмогорова о предельном сжатии данных, сформулировал принцип минимальной длины описания [11] (см. также [83, 96]), используя, по существу, формальный эквивалент отрицательного логарифма вероятностей. Основной заслугой Риссанена здесь является то, что он продемонстрировал, как подходы такого рода можно использовать на практике. Он также избежал проблемы невычислимости алгоритмической сложности, ограничив пространство гипотез рекурсивными функциями определенного вида.

М. Ли и П. Витани в 1989 г. попытались обобщить ряд существовавших к тому времени подходов [12] (см. также [19, 87]), опираясь на эффективно вычислимые функции и обратившись к тестам Мартина-Лёфа [84] на случайность.

Нельзя не упомянуть о работах Хаттера (например, [97–99]) и Шмидхубера [100–102]. Хаттер предлагает при формировании распределения априорных вероятностей учитывать число шагов, необходимых программе для генерации искомой строки. Процесс перебора начинается с выполнения наиболее коротких программ, но предлагается не ждать результата каждой программы, а запускать их параллельно. Та программа, которая первой даст на выходе правильную строку, считается лучшей. Короткие программы имеют преимущество, так как запускаются раньше. Хаттер также показывает, что существует программа, которая одновременно является самой быстрой и самой короткой программой, генерирующей данную строку (точнее, эта программа длиннее самой короткой и медленнее самой быстрой програм-

мы в фиксированное число раз). Работы Шмидхубера также учитывают скорость выполнения программ при задании их априорных вероятностей и посвящены самооптимизирующемуся поиску. К сожалению, нам не известны примеры практического применения подходов Хаттера и Шмидхубера к тем задачам, о которых пойдет речь ниже, хотя эти два подхода нам представляются весьма перспективными для универсальных систем индуктивного вывода. Далее мы ограничимся рассмотрением критериев выбора гипотез, основанных только на длине описания.

Здесь будет изложена общая идея теоретико-информационного подхода без конкретизации отдельных методов (МДО, МДС, АЛВ и др.). Хотя, вероятно, авторы этих методов с этим бы не согласились. Вопрос о том, какой из этих методов лучше на практике и корректнее с теоретической точки зрения, остается дискуссионным. Так, в журнале «Computer Journal» (Volume 42, Issue 4, 1999) можно найти описания некоторых из упомянутых подходов [15; 103–105]. Также ведется обсуждение вопроса, в чем именно заключается сходство и различие некоторых из этих методов [10, 106–108].

Но поскольку мы здесь не придерживаемся абсолютно строгого изложения (интересующийся читатель может обратиться к книге М. Ли и П. Витани «An Introduction to Kolmogorov Complexity and Its Applications» [81]), и упомянутые методы в их исходной интерпретации использоваться не будут, тонкие различия между ними будут несущественны. Основной же идеей (обозначенной здесь как принцип МДО) можно считать использование корректного определения количества информации через алгоритмическую сложность (или ее вычислимую аппроксимацию) для подстановки в прологарифмированное правило Байеса:

$$K(\mu \mid \beta) = K(\beta \mid \mu) + K(\mu) - K(\beta), \qquad (1.51)$$

где β — бинарная строка, соответствующая данным наблюдений; μ— бинарная строка, соответствующая модели данных; слагаемое $K(\beta \mid \mu)$ соответствует длине наиболее короткой строки α, такой, что, если она дана на вход программе μ, то последняя напечатает на выходе строку β. Значит, данных α и программы μ полностью достаточно, чтобы воспроизвести строку β. А их суммарная сложность $K(\beta \mid \mu) + K(\mu)$ служит критерием оптимальности модели μ.

Хотя соотношение (1.51) действительно верно с точностью до константы (см., например, [87]), оно может, однако,

вызывать определенные сомнения, если μ интерпретируется как программа — тогда запись $K(\mu \mid \beta)$ бессмысленна. В связи с этим вернемся к определению алгоритмической сложности $K(\beta) = \min\limits_{\alpha',\mu'} \left[l(\alpha') + l(\mu') \mid U(\mu', \alpha') = \beta \right]$. Пусть минимум достигается на строках α и μ: $K(\beta) = l(\alpha) + l(\mu)$. Очевидно, $l(\alpha) = K(\beta \mid \mu)$. Таким образом, $l(\mu) + K(\beta \mid \mu)$ — это описание строки β, обладающее наименьшей длиной. Если есть несколько программ, которые по строке α воспроизводят строку β, то μ — наиболее короткая из них. Итак,

$$\mu_{MDL} = \arg\min_{\mu} \left[l(\mu) + K(\beta \mid \mu) \right]. \qquad (1.52)$$

Хотя уравнение (1.51), (1.52) является простым аналогом либо правила Байеса (1.4), либо уравнения (1.36), есть существенные отличия. Во-первых, теперь мы имеем возможность подсчета количества информации, содержащейся в модели (длину описания модели). Алгоритмической сложности модели соответствует некоторая априорная вероятность. В связи с этим принцип МДО часто трактуется как правило Байеса, в котором использовано универсальное распределение для гипотез. Но поскольку алгоритмическая сложность зависит от опорной машины (что будет обсуждаться позднее), такая интерпретация принципа МДО не дает о нем ясного представления. Во-вторых, различия заключаются в рассматриваемом пространстве гипотез. В данной интерпретации правила Байеса μ — произвольная бинарная строка, программа для универсальной машины Тьюринга, способная сгенерировать на выходе любую другую строку. В этом смысле пространство гипотез является универсальным.

Видно также, что АЛВ предназначена для получения распределений вероятностей по возможным предсказаниям, в то время как такие подходы, как МДС и МДО, направлены на выбор конкретной модели. Это приводит к определенным разночтениям, на которых мы сейчас кратко и остановимся.

1.6.3. Индукция и предсказание

Как уже отмечалось, программу (модель) и данные (невязки) вовсе не обязательно разделять. Кроме того, при использовании лишь одной лучшей модели для дальнейше-

го анализа его результат неизбежно окажется хуже. Использование лучшей модели, выбранной на основе алгоритмической сложности, равносильно использованию лишь одного слагаемого в формуле (1.50) при решении задачи предсказания. Если руководствоваться этими соображениями, то мы также не должны выбирать наиболее вероятное предсказание, а должны строить распределение вероятностей по всем предсказаниям, чтобы на основе этих вероятностей проводить дальнейший анализ. Например, выбирая какое-то действие, следует руководствоваться не только более вероятным исходом, но учитывать и другие возможные исходы, так как весьма маловероятный исход может иметь столь сильный эффект, что при принятии решений его нельзя игнорировать. В связи с этим Р. Соломонов характеризует подходы, основанные на алгоритмической сложности, как аппроксимации к алгоритмической вероятности [17], причем точность аппроксимации зависит от того, сколько слагаемых в уравнении (1.50) будет взято.

Но тогда возникает вопрос о необходимости привлекать построение моделей как таковое. В абстрактной идеальной системе, которую заведомо невозможно построить, выбор модели, вероятно, не потребовался бы. Однако и Р. Соломонов, являясь сторонником осуществления предсказания без выбора конкретной модели, соглашается, что построение модели имеет полезные функции (например, в науке это позволяет ученым обмениваться информацией), хотя и замечает, что выбор модели в науке (например, физического закона или теории) возможен, только если одна из моделей существенно лучше остальных [33]. Например, в социологии такой выбор пока невозможен.

Но выбор конкретной гипотезы чреват парадоксами теории подтверждения или теории принятия, даже если эта гипотеза гораздо более вероятна, чем остальные [1, гл. 13, 14]. Примером может служить парадокс лотереи: вероятность выиграть крайне мала, поэтому гипотеза, гласящая, что выигрыш возможен, просто не должна рассматриваться в процессе принятия решения [1, с. 264–265].

Сторонники индуктивного подхода (т. е. выбора модели) также не остались в стороне от обсуждения этого вопроса. Например, Уаллас признает [105], что предсказание на основе АЛВ могло бы быть лучше, но замечает, что на практике оно неосуществимо, поскольку требуется усреднение по бесконечно большому числу гипотез. Построение моделей

имеет также и неоспоримые прагматические преимущества, позволяя очень быстро проводить анализ, не проделывая каждый раз огромный объем одной и той же работы по выводу распределений вероятностей, а также обмениваться наиболее важными компактно представленными данными. Компромисс же видится в том, чтобы использовать дополнительные, менее вероятные, гипотезы только в случае необходимости, когда результат, полученный по одной гипотезе, недостаточно хорош. Это замечание по своей сути совпадает с высказыванием Р. Соломонова, особенно с учетом предложенной им практической АЛВ.

Таким образом, выбор модели оказывается все-таки необходимым. Но вместо одной лучшей модели часто нужно использовать несколько достаточно хороших моделей (придерживаться одновременно нескольких взаимоисключающих точек зрения, возможно, с разными степенями доверия, а в данном случае — с разными априорными вероятностями, не противоречит индуктивной логике). К сожалению, насчет выбора оптимального количества привлекаемых моделей пока ничего не говорится. В действительности, этот выбор является проблемой вывода следующего уровня, т. е. на вопрос о необходимом числе гипотез невозможно ответить без привлечения информации о том, зачем эти гипотезы будут использоваться и какие ограничения на ресурсы имеются. Поскольку данная проблема также является проблемой вывода, то можно надеяться, что и здесь окажется применимым принцип МДО.

К сожалению, эта проблема оказывается не единственной. Существует также ряд вопросов, возникающих и при подсчете длин описания одиночных гипотез. Эти вопросы являются общими как для предсказания, так и для выбора модели. В этом смысле они оказываются первоочередными по сравнению с только что затронутой проблемой.

1.6.4. Полнота и комбинаторный взрыв

Рассмотренные методы, обращающиеся к понятию алгоритма, обладают замечательным свойством — полнотой. Например, как указывалось выше, алгоритмическая сложность теоретически позволяет решить проблему аппроксимации набора точек функциями, вид которых априори не задан. При этом не требуется перебора всех возможных

функций: достаточно перебирать программы для УМТ, длина которых не превосходит длины строки данных. Если ограничиться рассмотрением рекурсивных функций частного вида, то также можно избежать и проблемы «зависания».

Принципиальную возможность работы с априорно неограниченными пространствами гипотез нельзя недооценить, ведь задачи такого рода постоянно встают перед человеком, что особенно хорошо видно в науке. Классическим примером может служить эволюция способов описания движения планет по небесной сфере. Эта задача совершенно аналогична аппроксимации набора точек произвольными функциями. Можно проследить, как эволюционировали представления о характере движения планет.

• При очень грубых и нерегулярных наблюдениях, вероятно, полагалось, что планеты (практически неотличимые невооруженным глазом от звезд, лишь по отсутствию мерцания) движутся по окружности вокруг Земли.

• При появлении способа определения координат на небесной сфере было замечено, что для некоторых объектов движение не является круговым, и Птолемей ввел систему эпициклов для описания их движения. Каждый такой объект теперь вращался вокруг некоторого центра, который, в свою очередь, вращался вокруг Земли. Все отклонения от этой модели относились на счет неточности наблюдений (входили в случайную составляющую).

• При развитии техники наблюдения были обнаружены регулярные (не случайные) отклонения от этой модели, что привело к введению дополнительных эпициклов. По сути, это было разложением в ряд гармонических функций. Если взять достаточное число членов, то можно описать любое периодичное движение сколь угодно точно.

• Однако при дальнейшем развитии техники (и точности наблюдений) число эпициклов продолжило расти, что послужило одним из стимулов развития представления, согласно которому все небесные тела (и Земля в том числе) вращаются вокруг Солнца по эллиптическим орбитам. Это привело сначала к созданию гелиоцентрической системы мира Коперника, а потом к выводу уравнений Кеплера.

Дальнейшее развитие представлений о движении планет Солнечной системы тесно связано с разработкой теории тяготения (сначала Ньютона, а затем Эйнштейна), поэтому эти модели опираются далеко не только на положение планет на небесной сфере и выходят еще дальше за начальное про-

странство гипотез плоских круговых движений. Но уже при переходе от системы эпициклов к гелиоцентрической системе виден выход за начальное пространство гипотез, что принципиально невозможно в системах индуктивного вывода, использующих предварительно заданные классы моделей. А именно этим недостатком обладают практически все существующие системы. Но недостаток этот принципиально разрешим в системах, основанных на алгоритмической сложности (или АЛВ), в частности, в таких системах разрешается парадокс «зелубых» изумрудов [33, 51].

Аналогично многие системы машинного обучения ограничены в том, какие концепты и закономерности они могут выучить и обнаружить, даже если пространство поиска в них бесконечно, что, видимо, связано с «неполным» набором используемых ими концептов (отсутствие «универсальности» в смысле УМТ [33]).

Часто утверждается, что выбор между геоцентрической системой мира Птолемея и гелиоцентрической системой Коперника — это не более чем вопрос удобства, а движение планет с их помощью описывается одинаково хорошо. В действительности же это не так: они одинаково хорошо описывают *имеющиеся* данные, но точность предсказания у первой существенно хуже, чем у второй (т. е. уже упоминавшаяся проблема переобучения). Это говорит о том, что в системе индуктивного вывода общего назначения пространство моделей не должно быть ограничено. Можно сказать, что с точки зрения алгоритмического подхода $y = \sum_{n=0}^{\infty} \frac{1}{n!} x^n$ и $y = \mathbf{e}^x$ — это совершенно разные гипотезы.

Но по поводу полноты существуют разные мнения. Например, Р. Бакстер [2, с. 9] критикует как статью Р. Соломонова [7], так и книгу Ли и Витани [81] за использование невычислимых функций, что является следствием полноты. М. Минский также утверждал, что «озабоченность полнотой, столь значимая для математической логики, оказалась необычайно вредной для тех, кто работает с моделями разума» [5, с. 176]. И эта критика небезосновательна.

Рассматриваемая проблема поиска минимальной программы является NP-полной, и время ее решения экспоненциально зависит от сложности задачи (число программ длины L пропорционально 2^L). Это достаточно очевидно, поскольку невозможно заранее предсказать, выдаст ли данная

программа на выходе желаемую строку, а тем более без перебора в общем случае определить по данной строке, какие программы могут ее породить. Лучший известный алгоритм предложил Левин [92], и время поиска решения оказывается не более $CT2^L$. Здесь предполагается, что существует некоторый алгоритм на некоторой машине, и он порождает искомую строку; L — длина наикратчайшего описания этого алгоритма на опорной универсальной машине Тьюринга; T — время работы алгоритма на исходной машине; C — некоторая константа, обозначающая, насколько опорная машина медленнее машины, реализующей искомый алгоритм непосредственно. Хотя максимальное время оказывается экспоненциально зависимым не от длины строки данных, а от длины наикратчайшей программы, это не является сильным утешением.

Рассмотрим следующий пример. Пусть искомый алгоритм печатает на выходе числа $x_n = 39{,}014n^2 + 12{,}827n + 56{,}403$. Коэффициенты случайны, поэтому они будут присутствовать в наикратчайшей программе без сжатия. Это означает, что потребуется перебрать не менее (а на самом деле существенно более, если учесть еще саму форму зависимости) 10^{15} программ, чтобы отыскать эту зависимость, скажем, по ста входным числам. В реальных задачах (например, при построении описания изображений) эта величина будет столь большой, что никакой прирост производительности компьютеров не сможет помочь справиться с этой проблемой. При этом человек, основываясь на знаниях о предметной области, для каждого конкретного случая способен найти алгоритм, строящий модель приемлемого качества. Например, очень часто предположение о нормальном распределении ошибок оказывается допустимым. Это позволяет применять метод наименьших квадратов. Если класс моделей задается параметрическим образом в виде порождающей модели, линейной по параметрам, то может быть использован линейный метод наименьших квадратов, который вообще не требует никакого перебора, — и это совсем не редкий случай.

В рассматриваемых подходах акцент делается на использование всех имеющихся данных о предметной области с целью получения наилучшего по точности решения, но редко рассматривается вопрос об использовании этих данных для оптимизации процедуры поиска лучшего решения. Вероятно, причина в том, что последняя задача существен-

но сложнее (попытка решения этой задачи предпринимается в уже упоминавшихся работах Хаттера и Шмидхубера, что и делает эти работы весьма интересными).

В то же время критики данных подходов не предлагают решения этой проблемы, оставляя за человеком сужение пространств гипотез и оптимизацию процедуры поиска. Это дает человеку неплохой инструмент для решения частных проблем, но не продвигает разрешение загадки индуктивного вывода и не способствует созданию «самостоятельных» систем машинного обучения. Потеря алгоритмической полноты приводит к тому, что машина не может сделать то, что не заложено (в широком смысле) в нее человеком.

Таким образом, одно из основных направлений развития методов, которые базируются на принципе МДО, заключается в автоматическом установлении ограничений на пространство гипотез и оптимизации процедур поиска лучшей модели на основе информации о предметной области. В данной книге ставится цель не установления способов решения обозначенных проблем, а лишь сбора материала о том, как это делает человек при решении сложных задач, чтобы на основании этого материала можно было делать дальнейшие заключения.

1.6.5. Проблема субъективности и инкрементное машинное обучение

Незатронутым остался еще один очень важный вопрос: зависимость алгоритмической сложности и алгоритмической вероятности от выбора опорной универсальной машины. И действительно, для разных УМТ разные алгоритмы могут оказаться наиболее короткими при одной и той же строке данных. Это означает, что разные УМТ задают разные априорные вероятности. Но предоставляет ли тогда алгоритмическая сложность в действительности решение проблемы априорных вероятностей, для чего она и привлекалась?

Обычно приводится следующий аргумент в пользу положительного ответа на данный вопрос [19, 33]. Пусть есть две опорные машины U_1 и U_2. Поскольку каждая из них может эмулировать работу другой, алгоритмическая сложность строки для второй машины не может быть больше, чем ее алгоритмическая сложность для первой машины плюс длина описания первой машины:

$$K_{U_2}(\beta) \le K_{U_1}(\beta) + l(d[U_1]). \qquad (1.53)$$

Аналогичное неравенство записывается и для логарифмов алгоритмической вероятности. Это означает, что при достаточно длинной входной последовательности модель, выбранная при использовании двух разных УМТ, будет одинаковой.

Слабость аргумента, опирающегося на большую длину входной строки, была установлена еще при обсуждении правила Байеса, так что и здесь этот аргумент не является достаточно убедительным. Тем более длина $l(d[U_1])$ может быть очень большой. Например, если одной машиной является интерпретатор языка Си, а другой — Паскаля, то эта добавка будет соответствовать длине программы-транслятора с одного языка на другой. В результате эта максимальная добавка может существенно превышать саму алгоритмическую сложность анализируемой строки.

Проблемы бы также не возникало, если бы для разных моделей алгоритмическая сложность отличалась на одну и ту же величину. Но это как раз неверно, что хорошо видно на следующем примере. Пусть обе машины соответствуют одному и тому же языку программирования, но с разными библиотечными функциями. К примеру, для одной из них не реализована функция возведения в степень. Тогда для одного и того же алгоритма придется на разных машинах использовать, к примеру, записи «pow(x, y)» и «exp(y*log (x))». Ситуация еще более усугубится, если вычисление экспоненты и логарифма потребуется проводить с помощью элементарных арифметических операций. Сходные проблемы возникнут, если для задания универсальной машины использовать формализм нейронных сетей.

Хотя проблема априорных вероятностей остается и здесь, однако она имеет несколько другой характер: пространства гипотез остаются одними и теми же, и априорные вероятности могут отличаться лишь на конечный ненулевой множитель. С другой стороны, выбор опорной машины позволяет закладывать любые априорные данные и делать это гораздо удобнее, чем путем прямой модификации априорных вероятностей, как это происходит при использовании правила Байеса. Естественно закладывать эти данные не напрямую, а на основе данных наблюдений, что характерно для инкрементного машинного обучения.

В результате этого обучения предполагается, что УМТ модифицируется таким образом, чтобы задаваемые ее архи-

тектурой априорные вероятности были согласованы с внешним миром. Простейшим примером обучения может служить добавление библиотечных функций. Исходное задание машины Тьюринга повлияет лишь на скорость обучения, и здесь эта добавка будет уже не слишком существенной, так как предполагается, что объем информации о внешнем мире будет очень большим. Хотя субъективность априорных вероятностей оказывается здесь скорее преимуществом, чем недостатком, остается важный вопрос, какой формализм все же удобнее использовать? К сожалению, детальных исследований по этому вопросу не проводилось.

Еще один принципиальный момент заключается в следующем. Если рассматривать человека в качестве универсальной машины (мы не будем касаться здесь вопроса корректности такого рассмотрения), архитектура которой задает априорные вероятности, то можно заметить, что у взрослого человека эти вероятности базируются не только на информации, полученной в течение жизни, — большой объем априорной информации закладывается в ходе эволюции. Если система машинного обучения, которой предстоит действовать в реальном мире, будет развиваться «с нуля» (например, с состояния классической машины Тьюринга), то, чтобы достигнуть уровня человека, ей потребуется проанализировать объем информации, примерно соответствующий объему всей сенсорной информации, полученной в ходе эволюции всеми «предками» данного человека, начиная с весьма простых организмов.

Иными словами, если в качестве входа такой системы машинного обучения использовать камеру, то ей потребуются многие миллиарды кадров, чтобы научиться решать проблему компьютерного зрения. Не говоря уже о проблеме вычислительных ресурсов (описанной выше), которая еще более усугубляется в данном случае, возникает проблема формирования обучающей выборки. Можно, конечно, предположить, что возможно создание системы машинного обучения, которая не предназначена для навигации в физическом мире, а обучается на основе «рафинированной» (например, текстовой) информации, но есть основания полагать, что большой объем априорной информации заложен не только в подсистемы сенсорного анализа.

Таким образом, универсальные системы машинного обучения, основанные на принципе МДО, ни в коем случае нельзя считать решением проблемы искусственного интел-

лекта (ИИ), даже если в этих системах будет решена описанная выше проблема комбинаторного взрыва. Предоставляемую ими форму потребуется заполнить содержанием — колоссальным объемом информации, источником которой и могут служить различные направления в ИИ. Но эти направления уже сейчас могут выиграть от применения принципа МДО, что будет продемонстрировано далее на конкретных примерах.

Попытки использования АЛВ для построения универсальных систем машинного обучения уже предпринимались [93–95; 109]. Такие системы начинали работу с некоторого набора простых концептов («макросов»), который постепенно пополнялся в ходе решения задач всё возрастающей сложности из обучающей выборки. Однако, как замечает автор подхода [17], желаемый результат не был достигнут: система прекрасно работала на простых примерах, но было непонятно, как ее применить для решения сложных задач. Причина же связывалась с плохим качеством обучающей выборки. Иными словами, был недооценен объем априорной информации, необходимый системе для решения реальных задач, а сконструировать обучающую выборку вручную для таких задач достаточно сложно.

Еще один важный момент, замеченный при работе над такими системами, заключается в том [17], что система никогда не использовала части решений предыдущих проблем для оптимизации поиска решений новых проблем. Другими словами, она не использовала априорную информацию о предметной области для оптимизации процедуры поиска. А именно эта проблема была сформулирована выше, в п. 1.6.4, как одна из основных проблем, которую нужно решить, чтобы получить практически применимые системы индуктивного вывода общего назначения. Пока еще трудно сказать, насколько помогут при решении этой проблемы подходы Хаттера и Шмидхубера.

Инкрементное обучение возможно также и при применении правила Байеса. В этом случае непосредственно сами априорные вероятности подвергаются модификации на основе обучающей выборки. Такой подход был назван «современным Байесианизмом» (см., например, [6]). Этот подход гораздо ближе к принципу МДО: акцент в нем делается на использование всей имеющейся информации для задания адекватных априорных вероятностей, что, в частности, позволяет решать на практике проблему переобучения.

Тем не менее, основная проблема байесовских методов — необходимость задавать пространство гипотез — остается и здесь. Такие системы не способны обнаружить не заложенные в них понятия. Более того, модель предметной области, заданная в виде априорных вероятностей всех гипотез, не является оптимальной с точки зрения принципа МДО, поэтому подобный подход можно считать неполным применением этого принципа, и он рассматриваться нами не будет. Хотя следует заметить, что сторонники этого подхода не видят у принципа МДО фундаментального преимущества [6, с. 14], так что и здесь возможны различные точки зрения.

Другим источником субъективности априорных вероятностей является способ представления данных (в случае АЛВ это вопрос, как отображать исходные данные в бинарные строки) [33]. Эта проблема аналогична проблеме выбора опорной машины. И то и другое можно охарактеризовать как выбор языка представления.

1.7. ЗАКЛЮЧЕНИЕ

Задача индуктивного вывода оказывается достаточно простой для ее автоматического решения, только если выполняется ряд принципиальных ограничений: должно быть задано фиксированное пространство гипотез; каждой гипотезе должны быть назначены априорные вероятности, неточность которых может быть компенсирована большим объемом исходных данных; должен быть задан алгоритм поиска, который может быть различным для разных пространств гипотез. При таких ограничениях возможно применение критериев, основанных на теореме Байеса.

Очевидно, что основная часть работы ложится на человека. При этом теоретической сложностью является проблема задания априорных вероятностей. Особенно отчетливо она проявляется в «больших» пространствах гипотез, в которых каждому набору данных можно поставить в соответствие большое количество гипотез, одинаково хорошо подтверждающихся этими данными. Непонимание того, как правильно следует присваивать априорные вероятности, очень хорошо демонстрируется парадоксом «зелубых» изумрудов.

Из эвристических соображений следует, что гипотеза тем вероятнее, чем она проще. Критерий простоты также об-

наруживается в науке как индикатор красоты научной теории и ее предсказательной силы. Это является сильным эмпирическим свидетельством в пользу задания априорных вероятностей гипотез на основе того, насколько они сложны или просты. Но определение сложности «на глаз» оказывается весьма ненадежным и все еще требует заметного вмешательства человека. Необходима формализация понятия простоты.

Для этого разумно привлечь теорию информации и трактовать сложность через количество информации. Но в теории Шеннона количество информации как раз и определяется через вероятность. Чтобы корректно определить количество информации в сообщении, оказывается необходимым априорно знать модель его источника. Если же модель не известна, то количество информации в любом сообщении можно считать равным одному биту, приняв модель ad hoc, что аналогично парадоксу Гудмана. Чтобы учесть сложность модели, необходимо знать ее вероятность. Получается замкнутый круг.

Если задать представление гипотез как цепочки символов, то количество информации оказывается все же возможным оценить через ее длину, что позволяет перенести проблему с задания априорных вероятностей для всех гипотез на проблему определения языка представления. Для практики это намного удобнее, однако основной проблемы не решает.

Таким образом, оказывается необходимым ввести понятие количества информации, не опираясь на понятие вероятности. Такую возможность предоставляет алгоритмическая теория информации А. Н. Колмогорова. В ней вводится понятие алгоритмической сложности, которая определяется для строки символов как длина наикратчайшей программы для универсальной машины Тьюринга, способной породить исходную строку.

Применение этой теории в целях индуктивного вывода или прогнозирования оказалось весьма плодотворным. Основное, что привносит данный подход, — алгоритмическая полнота: на пространство гипотез не накладывается никаких принципиальных ограничений, а это означает, что для его задания больше не требуется привлекать человека.

Казалось бы, множество бинарных строк конечной длины счетно. Так почему же, например, множество полиномов произвольной степени с произвольными коэффициентами

при большей мощности не обладает алгоритмической полнотой, а множество бинарных строк обладает, хотя его мощность меньше? Причина, видимо, в тех априорных вероятностях, которые индуцируются в пространстве гипотез универсальной машиной. Это хорошо иллюстрируется конечной сложностью числа «пи» или экспоненциальной зависимости (или возможностью решения проблемы эпициклов Птолемея). Все это не закладывается априори: был выбран лишь один-единственный (очень простой!) формализм, приемлемый для многих областей, а значит, обладающий огромной общностью. Иными словами, алгоритмическая сложность является адекватным определением понятия сложности, что является аналогом тезиса Чёрча—Тьюринга.

Однако априорные вероятности, определенные через алгоритмическую сложность, оказываются зависимыми от выбора универсальной машины. Не возникает ли здесь противоречия? По-видимому, нет. Если некоторая бинарная строка имеет конечную сложность на одной УМТ, то она будет иметь конечную сложность и на любой другой УМТ, хотя вероятности могут перераспределяться очень сильно. С другой стороны, возникающая при выборе опорной машины субъективность является неизбежной и позволяет, во-первых, выбирать опорные машины в соответствии с некоторым языком, показавшим свою полезность в прошлом, для эффективного решения конкретной задачи, и, во-вторых, создавать системы инкрементного машинного обучения, возможности в обучении которых ограничены только доступными ресурсами.

Отметим также следующее. В выбранном языке представления содержится априорная информация о решаемой задаче. Понимание этого позволяет предостеречься от очень распространенной ошибки, а именно: от сравнения алгоритмов анализа данных, опирающихся на разное количество априорной информации. Эта ошибка особенно часто встречается при сравнении эвристических методов (с ними мы неоднократно столкнемся в следующих главах), в которых априорная информация закладывается в неявном виде.

Перечислим некоторые преимущества использования алгоритмической сложности в индуктивном выводе:

- алгоритмическая полнота;
- возможность конструировать разумные априорные вероятности по языку описания либо использовать универсальное распределение, если подходящий язык не известен;

• возможность инкрементного обучения без ограничения на типы концептов, которые система может выучить;

• статистическая корректность (согласованность с теоремой Байеса).

Но, несмотря на эти преимущества, до окончательного решения проблемы индуктивного вывода еще далеко, возможно, гораздо дальше, чем пройденный путь. Первой принципиальной трудностью является то, что время работы существующих алгоритмов поиска минимальной программы для машины Тьюринга экспоненциально зависит от сложности задачи, т. е. от длины минимальной программы. Причем универсальных способов сокращения этого времени не существует, что приводит к комбинаторному взрыву. Если в «игрушечных мирах» подход демонстрирует хорошую работу, то при применении к реальным задачам оказывается необходимо привлекать человека для принятия решения о том, какие упрощения можно вводить и как можно при этом оптимизировать поиск на основе знания предметной области. Таким образом, возникает первый вопрос: как использовать априорную информацию не только для модификации априорных вероятностей, но и для оптимизации процедуры поиска?

Ответ, на наш взгляд, следует искать в иерархических представлениях: на верхнем уровне находится универсальная машина, обеспечивающая алгоритмическую полноту. Моделями для нее являются языки представления, каждый из которых уже, возможно, не является алгоритмически полным. Каждое из представлений может также последовательно детализироваться до более локальных представлений. Система представлений соответствует системе предметных областей, которые начинаются от наиболее общих (например, в астрономии или химии) и сужаются вплоть до конкретной задачи (например, до описания движения планет солнечной системы). Исходно модель строится в рамках некоторого локального представления, которое допускает оптимизацию поиска, и только в случае неудачи, когда входные данные оказываются несжимаемыми в данном представлении, происходит переход на следующий уровень и производится поиск нового представления (например, переход от геоцентрической системы координат в гелиоцентрическую).

Естественно, это лишь общая схема, и на этом пути возникнет множество сложностей. Однако эвристическим доводом

в его пользу, как и при первоначальном применении понятия простоты, является то, что именно такие механизмы решения сложных задач используются как естественными нейронными сетями (например, широко известный эффект адаптивного резонанса, о котором еще будет сказано при обсуждении процессов обработки сенсорной информации), так и человеком в его научной и повседневной деятельности.

Следующий важный вопрос заключается в том, как в систему индуктивного вывода должна закладываться информация о предметной области, не важно, для задания хороших априорных вероятностей или для оптимизации процедуры поиска. Такая информация может либо подаваться системе напрямую в явном виде, либо может производиться инкрементное обучение на примерах. В обоих случаях требуется решать задачу разработки некоторого языка, в рамках которого было бы удобно описывать знания о предметной области. Представление знаний — это одна из центральных проблем искусственного интеллекта, но здесь она должна рассматриваться не в общем виде, а согласованно с теоретико-информационным подходом к индуктивному выводу. Проблема инкрементного обучения также не выглядит вполне решенной в рамках алгоритмического подхода, но решение этой проблемы существенно зависит от выбранной системы представлений.

И наконец, существует ряд проблем, которые обучением «с нуля», видимо, не решить, даже имея универсальную систему машинного обучения. Причиной такого положения является колоссальный объем обучающей выборки. Наиболее очевидными примерами таких проблем являются вопросы анализа сенсорной информации. Вполне возможно, что круг этих проблем гораздо шире, чем кажется первоначально. Привлечение теоретико-информационных методов может помочь ускорить исследования в этих областях, но вряд ли системы машинного обучения смогут решить эти проблемы самостоятельно.

Таким образом, можно обозначить следующие наиболее существенные нерешенные вопросы, связанные с индуктивным выводом на основе принципа МДО:

- единообразное представление знаний о предметной области (язык, описывающий другие языки);
- инкрементное обучение — оптимальная модификация данного представления, задающего априорные вероятности гипотез, на основе полученного опыта;

• общие методы эффективного использования знаний о предметной области для избежания проблемы комбинаторного взрыва;

• применение принципа МДО в проблемах, в которых формирование обучающей выборки не представляется возможным.

Поскольку эти проблемы все время решаются человеком, ниже будут приведены примеры таких решений, чтобы на их основе можно было бы попытаться найти пути решения сформулированных выше вопросов.

Оказывается, что на практике чаще прибегают лишь к самой общей идее использования длины описания в качестве оценки качества модели, не используя формализм машин Тьюринга. При этом длина описания может не определяться точно, а оцениваться «на глазок». Для такой оценки вовлекаются, в частности, и стандартные энтропийные критерии, если оценка энтропии может быть сделана достаточно надежно.

Итак, исследователи, использующие алгоритмическую сложность в прикладных целях, обычно приносят формальную строгость в жертву удобству практического применения. Иными словами, они производят оптимизацию процедуры поиска, руководствуясь эвристическими соображениями, которые, к сожалению, далеко не всегда формулируются в явном виде. Для нас полная математическая корректность методов также будет менее значимой, чем возможность их использования, однако на вводимых при этом допущениях мы постараемся акцентировать внимание. Для более строгого и подробного теоретического рассмотрения вопросов, связанных с алгоритмической сложностью, можно порекомендовать книгу Ли и Витани [81].

Глава 2

НИЗКОУРОВНЕВЫЕ ЗАДАЧИ МАШИННОГО ОБУЧЕНИЯ

2.1. РАСПОЗНАВАНИЕ ОБРАЗОВ В КОНТЕКСТЕ МАШИННОГО ОБУЧЕНИЯ

2.1.1. Вводные замечания по проблеме машинного обучения

Машинное обучение превратилось в одну из центральных парадигм искусственного интеллекта в 1980-х годах. Некоторое время до этого господствующей тенденцией был поиск неких универсальных механизмов «чистого» мышления, воплощение которых позволило бы создать идеальный интеллект, способный решать любые задачи. Хотя в данном направлении и были достигнуты определенные успехи, они оказались гораздо скромнее, чем исходные ожидания. Неудачные попытки в создании готового идеального интеллекта привели к смещению акцента в сторону построения систем искусственного интеллекта, способных к развитию, совершенствованию, т. е. к обучению. Конечно, здесь имеется в виду стереотипное отношение к проблеме ИИ, поскольку и раньше были исследователи, занимавшиеся проблемами адаптивности и обучения (особенно в рамках бионического направления) и обращавшие внимание на социальную природу человеческого интеллекта.

Дать формальное определение обучению крайне затруднительно, но важно то, что обучение — это способность использовать предыдущий опыт для лучшего решения последующих задач. В простейшем случае обучение соответствует простому накоплению данных в процессе функционирования. Например, система машинного обучения, встретившись с задачей, которая уже была ею решена, может воспользоваться старым решением. Однако обучение этим не ограничивается: опыт решения одних задач можно использовать для решения других задач, для чего нужно обобщить свой опыт, т. е. осуществить индуктивный вывод. И действительно, связь между машинным обучением и индуктивным выводом очень тесна. Однако их нельзя отождествлять.

Машинное обучение — очень обширное направление исследований, перекрывающееся со многими другими областями и включающее рассмотрение различных проблем на основе разных подходов, для которых даже нет устоявшейся классификации. В качестве примера можно привести деление на символьные, коннекционистские (сетевые или на основе связей) и эмерджентные методы обучения [74, с. 371]. Зачастую методы из одной группы в точности копируют методы другой группы, но сущность подхода выражают в совершенно других терминах, т. е. эта классификация отражает форму методов обучения, но не их алгоритмическое содержание. Все это говорит о том, что область машинного обучения на данный момент является еще не вполне зрелой дисциплиной.

Если рассматривать машинное обучение в контексте индуктивного вывода, то подходы к обучению будут характеризоваться структурами входных данных и пространства моделей. В данной книге не ставится цель выполнить систематический обзор методов машинного обучения. Вместо этого мы попытаемся показать возможность применения принципа минимальной длины описания в тех задачах обучения, в которых исходные данные имеют количественный характер. Основная часть этих задач была поставлена в теории распознавания образов.

Некоторые методы обучения на основе символьной информации будут кратко рассмотрены в гл. 4. Хотя останутся нерассмотренными вопросы применения принципа МДО в других подходах к проблеме обучения, это вовсе не означает, что такая возможность отсутствует. Например, возрастающей популярностью начинает пользоваться употребление теоретико-информационных критериев при выборе конфигурации нейронных сетей разных типов [110–115] или сетей доверия [116–118] в задачах коннекционистского обучения.

Количественные данные, обучению по которым посвящена данная глава, обычно связаны с результатами некоторых измерений (например, выполняемых органами чувств) в отличие от дискретных или символьных данных, которые носят лингвистический или логический характер. Поскольку интерпретация сенсорных данных предшествует сознательному мышлению, такой тип обучения можно условно назвать низкоуровневым. С ним, однако, связан ряд важнейших проблем, к которым относится, например, формирование понятий.

Методы решения этих проблем в теории распознавания рассматриваются весьма абстрактно, поскольку в них не используется специфическая информация о сенсорной модальности или физическом устройстве, являющемся источником данных. Иными словами, в этих методах не привлекается априорная информация о предметной области. В то же время в них делается ряд дополнительных предположений о входном и выходном представлении данных. Эти предположения позволяют строить эффективные, с вычислительной точки зрения, методы, поэтому их можно рассматривать как некоторый промежуточный вариант между универсальными методами, не применимыми на практике, и очень частными методами, использующими анализ предметной области, проведенный человеком-исследователем. В связи с этим интересно проследить, за счет каких упрощений устраняется проблема комбинаторного взрыва при относительном сохранении общности методов.

В частности, интерес представляет существование разных типов обучения: обучение с учителем, обучение с подкреплением и обучение без учителя. При обучении с учителем системе предоставляются обучающие задачи, после решения каждой из которых ей сообщается правильный ответ, позволяющий корректировать алгоритм решения. При обучении с подкреплением вместо правильного ответа системе лишь сообщается, является ли ее решение правильным (или некоторый показатель качества решения). Существенное отличие от предыдущего случая состоит в том, что подкрепление может осуществляться средой. В случае обучения без учителя система лишена какой-либо дополнительной информации.

В действительности, при достаточно абстрактном рассмотрении все эти способы обучения идентичны, поскольку любая дополнительная обучающая информация может трактоваться просто как часть исходных данных, для которых в совокупности необходимо построить некоторую модель. Причем такой подход является общим, так как в нем делается меньше априорных предположений. Например, в нем допускается, что учитель может сообщить неверную информацию. Тем не менее разделение способов обучения оказывается очень продуктивным. Хотя, как будет показано на задачах распознавания, различные типы обучения укладываются в рамки построения моделей, дополнительная информация от учителя используется для эффек-

тивного сужения пространства моделей, вместо того чтобы просто пополнять входную информацию, на основе которой производится индуктивный вывод.

2.1.2. Основные понятия распознавания образов

Формирование распознавания образов как научной дисциплины началось в 50-х годах XX века (почти одновременно с зарождением науки об искусственном интеллекте). Многие первоначальные работы в этой области были посвящены построению автоматов, предназначенных для чтения печатных и рукописных знаков, с чем и было связано введение в употребление термина «образ». Существенное влияние на развитие этой области также оказал перцептронный подход, предложенный Ф. Розенблаттом в конце 1950-х годов. Привлекаемый для решения задач распознавания образов математический аппарат постепенно расширялся, включая теорию статистических решений, математическую логику, теорию информации, теорию формальных грамматик и т. д. Вместе с тем рос и интерес к этой области.

Задачи распознавания как прикладные задачи возникают во многих отраслях науки и техники. Однако наиболее важной разработка теории и практики распознавания видится для ИИ-проблематики. Некоторое время назад существовало мнение (во многом справедливое и сейчас), согласно которому «создание искусственного интеллекта — это, по-видимому, прежде всего построение распознающих систем, приближающихся по своим возможностям к возможностям человека в решении задач распознавания» [119, с. 7]. Описание же методов распознавания образов составляло существенную часть обзорных книг по искусственному интеллекту (см., например, [52]).

После того как в качестве отдельного вопроса ИИ выделилась область машинного обучения, включившая в свое рассмотрение под определенным ракурсом значительную часть проблематики распознавания образов, индуктивного вывода и других смежных областей, вместо распознавания образов в обзорные книги по ИИ, как правило, стал включаться раздел, посвященный машинному обучению (см., например, [5, 74]). Тем не менее наиболее развитыми здесь являются методы распознавания образов, к рассмотрению которых мы и обратимся.

Когда человек узнает своего знакомого в толпе людей, определяет, является ли данное число простым или составным, или устанавливает класс звезды по ее спектру, он решает, казалось бы, совершенно разные задачи. Однако можно заметить и некоторые общие черты. В каждой из задач присутствует некоторый объект (человек, число, звезда), представленный значениями своих признаков, а также некоторое число классов, к одному из которых необходимо отнести данный объект. Таким образом, распознавание — это отнесение некоторого неизвестного объекта по его описанию к одному из классов. Возможна и другая задача: разделить заданное множество объектов на классы (задача таксономии). В теории распознавания образов ставится следующий основной вопрос: могут ли столь разные задачи иметь одно и то же математическое описание и сходные алгоритмы решения?

Сложность этого вопроса заключается в том, что исходные описания очень сильно отличаются в различных приложениях, так же как и способы объединения объектов в классы. Тем не менее задачи, ставящиеся в области распознавания образов, являются частным случаем индуктивного вывода [120, с. 16], объединяющего не только их, но и другие задачи построения моделей. Именно в контексте индуктивного вывода и будут здесь рассмотрены некоторые вопросы распознавания образов. Чтобы более строго поставить задачу распознавания, введем следующие обозначения.

Пусть $x \in X$ — *описание объекта* (или *образ*), а X — *пространство описаний* (множество всех возможных образов).

Пока мы считаем, что множество X может быть абсолютно произвольным. В различных задачах это пространство может иметь разную мощность, разную выразительную силу и разную структуру. Практически всегда описание объекта состоит из значений его признаков, число которых в зависимости от задачи может либо быть фиксированным, либо изменяться от объекта к объекту. Сами признаки могут быть и количественными, и логическими, и символьными. Уточнение свойств пространства описаний необходимо для получения практически применимых методов, и разные уточнения приводят к разным подходам к задаче распознавания образов.

Пусть $A = \{a_1, a_2, \ldots, a_d\}$ — некоторое множество, состоящее из d элементов, $1 < d < +\infty$, где a_i — i-й *класс образов*,

а A — *множество классов* (называемое также алфавитом классов).

Для одних и тех же объектов перечень классов может быть разным в зависимости от конечной цели. Например, если в качестве объекта распознавания выступает лицо человека, то можно привести такие множества классов: {мужчина, женщина}, {ребенок, взрослый, пожилой человек}, {фас, профиль}, множество знакомых людей, выражение лица и т. д. Можно было бы предложить и такое множество классов, как, например, {грустное лицо, профиль}, т. е. попросить кого-то выбрать между тем, является ли выражение данного лица грустным или оно изображено в профиль (заметим, что в распознавании образов требуется выбрать ровно один вариант). Однако такое множество классов кажется «нелогичным» (очевидно, это связано с тем, что такие классы полностью перекрываются в пространстве X).

Решающим правилом назовем отображение $\varphi: X \to A$, которое ставит в соответствие элементу пространства описаний класс из заданного множества.

Решающее правило может также задаваться неявно через целевую функцию $\rho : X \times A \to R$, определяющую степень соответствия (например, в форме вероятности) между описанием объекта и каждым классом. Решающее правило можно определить через целевую функцию как

$$\varphi(x) = \arg\max_{a \in A} \rho(x, a). \qquad (2.1)$$

Во многих практических задачах вводится матрица потерь L_{ij}, определяющая стоимость ошибочного отнесения объекта класса i к классу j (в п. 1.2.2 приводился пример, показывающий, что такая необходимость действительно может возникнуть), а задача формулируется как минимизация ожидаемых потерь в ходе классификации. Однако учет потерь при классификации нужен лишь при принятии решения, к какому классу отнести данный объект, но не на процедуру вывода вероятностей принадлежности объекта к тому или иному классу. Конечно, в подходах, не опирающихся на теорию вероятностей, матрица потерь непосредственно влияет на решающее правило, но суть подходов не меняется и в том случае, если эта матрица не используется. В связи с этим при дальнейшем изложении матрица потерь будет опускаться.

2.1.3. Дополнительные предположения о пространстве описаний и множестве классов

Для того чтобы можно было использовать существующие методы распознавания, основные понятия необходимо уточнить. В основном речь идет о дальнейшей конкретизации структуры пространства описаний, т. е. выборе представления, в рамках которого производится исходное описание объектов. В зависимости от дополнительных предположений о пространстве описаний X практически любой математический метод распознавания образов можно отнести либо к *дискриминантным*, либо к *структурным* [121, с. 9].

В первой группе методов пространство описаний трактуется как пространство R^N (N-мерное векторное пространство) или его подпространство. Каждый объект описывается *вектором признаков* фиксированной размерности N, а само пространство X обычно называется *пространством признаков*. В этом пространстве каждому классу соответствует некоторая область. Эти области разделяются поверхностями (размерности $N - 1$), описывающимися *дискриминантными функциями*, параметры которых являются алгебраическими функциями от параметров распределений объектов внутри соответствующих классов. Разделение классов с помощью такого типа функций и является отличительной чертой дискриминантного подхода. Впервые подобный подход (как и сам термин «дискриминантный») был использован Р. Фишером в 1936 г. в целях классификации видов ириса по размерам цветка.

В том случае, когда объект описывается вектором признаков, компоненты этого вектора могут принимать не непрерывные, а дискретные значения. В предельном случае этих значений может быть только два, тогда говорят о логических признаках. Методы распознавания образов, характеризующиеся такими признаками, обычно основываются на дискретном анализе и исчислении предикатов и могут выделяться в отдельную группу логических методов. Определенное внимание построению логических систем распознавания уделено в работе [119].

Структурные методы наиболее широко определяются как методы, в которых предпринимается попытка описать объекты в терминах их частей и отношений между этими частями [122]. Таким образом, в структурных методах распознавания гораздо более важными являются не значения

отдельных признаков, а взаимосвязи между ними. Для того чтобы описать структуру объекта, исходный образ представляется в виде некоторой совокупности более простых подобразов, которые, в свою очередь, могут быть составлены из еще более простых подобразов, образуя древовидную (иерархическую) структуру. Наиболее простые подобразы называются *непроизводными элементами*. Число непроизводных элементов, входящих в некоторое структурное описание, не является фиксированным и может сильно варьироваться в зависимости от объекта. Для построения систем распознавания образов, работающих с такими представлениями, было предложено использовать математический аппарат формальных грамматик (см., например, [123–125]). Уточненный таким образом структурный подход получил название лингвистического, или синтаксического, подхода к распознаванию образов. В нем непроизводные элементы образуют алфавит (или словарь) языка описания образов, а грамматика задает правила композиции этих элементов, определяя, какие предложения в данном языке являются правильно построенными, а какие — нет. Как правило, каждый класс описывается своей грамматикой, а задача классификации сводится к определению, является ли структурное описание некоторого объекта допустимым предложением в рамках той или иной грамматики.

Типичной областью применения синтаксических методов являлось распознавание изображений (рис. 2.1). И хотя в данной области этот подход уже не столь популярен, как

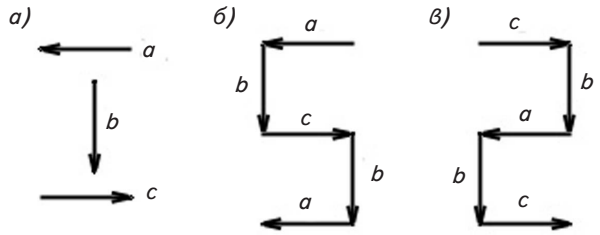

Рис. 2.1. Непроизводные элементы (*а*) и две фигуры, описанные на языке, аналогичном PDL (Picture Description Language, см. [126] или [121], с. 262–269): $a + b + c + b + a$ (*б*) и $c + b + a + b + c$ (*в*). Здесь a, b, c — непроизводные элементы, каждый из которых имеет голову (обозначена стрелкой) и хвост; выражение $a + b$ означает, что голова элемента a примыкает к хвосту элемента b

ранее, он сыграл существенную роль в развитии структурных методов как таковых.

Системы распознавания, в которых используются различные типы признаков или признаки, имеющие разную физическую природу, называются *сложными* [127]. Мы же будем здесь рассматривать дискриминантные методы, поскольку они предоставляют наиболее обширный материал для анализа; о синтаксических методах распознавания см. в гл. 4 настоящей книги; подробнее — см., например, в работе [121].

Необходимость деления на дискриминантные и структурные методы вытекает из различия в виде исходных пространств описаний. Возможно также уточнение и пространства классов. Например, множество классов может получаться произведением некоторого набора множеств, что соответствует одновременному применению нескольких классификационных схем. Если вектор, описывающий результаты классификации, полученные с помощью нескольких простых систем, используется в качестве вектора признаков для дальнейшего анализа, то подобные сложные системы называются *многоуровневыми системами распознавания*. Например, лицо по его изображению мы можем классифицировать как бородатое или безбородое, с короткими или длинными волосами и т. д., а затем, в зависимости от того, к каким классам было отнесено лицо, определить его как женское или мужское.

Возможны также и другие варианты уточнения пространства классов (например, введение иерархии классов, как это сделано в биологии для классификации видов), но мы для сокращения изложения будем рассматривать наиболее простой случай, когда множество классов задается перечислением своих элементов.

Существуют также способы деления подходов к распознаванию не по постановке задачи (не по структуре вовлекаемых пространств), а по применяемой методологии или математическому аппарату, например, деление на детерминистские и вероятностные методы и на математические и эвристические методы. Такие способы деления отражает то, каким образом соответствующая система распознавания была построена, но не то, как именно она функционирует. Действительно, многие методы, разработанные как детерминистские, были впоследствии обоснованы с позиций теории статистических решений (подобные примеры можно най-

ти в книге [128], преимущественно посвященной статистическим методам распознавания), а большинство эвристических методов получило строгое математическое описание.

Несмотря на то что многие методы распознавания являются строго математическими, сама постановка задачи (а точнее, предположения о пространстве описаний) таковой являться не может, а значит, является эвристической. Используются также дополнительные предположения, чтобы сделать методы применимыми на практике. Различные «необоснованные» предположения нас и будут больше всего интересовать. В наиболее удобном виде они сформулированы в детерминистских (и эвристических) методах, а вероятностные методы легче связать с принципом МДО, поэтому будут рассматриваться как те, так и другие.

Существуют методы, в которых не делается дополнительных предположений о пространстве описаний. Один из таких теоретических подходов — это алгебраический подход к распознаванию. Он является попыткой обобщить эвристические алгоритмы распознавания, которые выступают в качестве объектов исследования. В этом подходе семейство алгоритмов распознавания рассматривается как некоторая алгебра с введенными на ней операциями, что дает модель распознавания. Недостаток эвристических моделей заключается в том, что при их использовании не гарантируется построение алгоритма, корректно классифицирующего все образы обучающей выборки. В алгебраическом подходе этот недостаток устраняется путем расширения эвристических моделей с помощью корректирующих операций [129, с. 4]. Базисные относительно введенных операций алгоритмы используются для индуктивного построения всех прочих (в рамках данной модели распознавания) алгоритмов, среди которых ищется алгоритм, корректно решающий конкретную задачу распознавания [130].

Несмотря на то что результаты алгебраического подхода представляют определенный интерес, здесь он рассматриваться не будет по нескольким причинам. Во-первых, нас интересуют именно эвристики, содержащиеся в дискриминантных методах распознавания, к которым алгебраический подход не относится. Во-вторых, в рамках алгебраического подхода не решается ряд наиболее интересных проблем, связанных с обучением без учителя. И, в-третьих, в данном подходе, как правило, корректным алгоритмом считается произвольный алгоритм, правильно классифици-

рующий образы обучающей выборки. Выбор произвольного корректного алгоритма в качестве решения, очевидно, является недопустимым, так как будет приводить к переобучению. Выбор же критерия качества алгоритма выходит за рамки алгебраической теории распознавания.

Отметим, что первый попавшийся алгоритм будет в некотором смысле одним из самых простых, так как при поиске корректного алгоритма происходит формирование все более сложных алгоритмов из-за последовательного применения операций к базисным алгоритмам. Очевидно, что принцип МДО может быть применен и в алгебраическом подходе, который, в свою очередь, может дать подсказку, как формировать ограниченные семейства алгоритмов и осуществлять в них поиск. К сожалению, эти вопросы не исследовались, и они выходят за рамки данной книги. Читателю, заинтересовавшемуся алгебраическим подходом, рекомендуем обратиться к литературе (см., например, [129, 131] и приведенные там ссылки).

2.1.4. Постановка задачи распознавания в зависимости от количества априорной информации

В распознавании образов возникает несколько основных задач, которые хотя и могут формулироваться по-разному, но являются общими для всех упомянутых выше подходов. Это задачи классификации, распознавания и группирования (таксономии или автоматической классификации). Еще одной является задача предварительной обработки образов и выбора признаков. Она обычно рассматривается отдельно. Различие между этими задачами заключается в доступной априорной информации.

Задача *классификации* (распознавания без обучения) заключается в определении по описанию объекта того класса, к которому он принадлежит. При этом решающие правила считаются известными. В рамках дискриминантного подхода это означает, что известны разделяющие поверхности, так что для любого объекта, представленного точкой в пространстве признаков, можно определить, в какой области он расположен. В рамках синтаксического подхода эта задача соответствует ситуации, в которой известны грамматики для соответствующих классов и требуется

провести грамматический разбор, т. е. определить, является ли структурное описание объекта предложением, синтаксически правильным по отношению к какой-либо из этих грамматик.

Задача *распознавания* (обучения с учителем) заключается в построении решающих правил, которые считались известными в задаче классификации. В качестве исходной информации здесь выступает обучающая выборка. Каждый элемент выборки представляет собой описание объекта и соответствующий ему класс. В дискриминантном подходе задача распознавания сводится к построению поверхностей в пространстве признаков, разделяющих заданные в обучающей выборке множества точек. В синтаксическом методе эта задача превращается в задачу обучения грамматикам, т. е. в восстановление грамматик по заданным наборам правильно и неправильно построенных предложений. Решение задачи распознавания должно быть таковым, чтобы обеспечить наиболее высокое качество дальнейшей классификации неизвестных объектов.

В задаче *группирования* количество имеющейся информации еще меньше — в ней не определено пространство классов A, которое и требуется сформировать, опираясь на заданный набор образов, не разбитых на классы в отличие от задачи распознавания с учителем. Одной из первых работ, посвященных данной проблеме, была работа Тайрона [132].

Группирование относится к обучению без учителя (поэтому также называется распознаванием без учителя), поскольку системе, решающей эту проблему, не сообщаются «правильные ответы» для образов обучающей выборки, а разделение на классы должно быть осуществлено в соответствии со свойствами самих объектов. Формирование классов соответствует разбиению исходного множества образов на подмножества согласно некоторому критерию качества. Изучение таких критериев позволяет вскрыть смысл разделения объектов на классы. И действительно, критерий качества группирования должен отвечать на вопросы: почему нельзя объединить все объекты в один класс или, напротив, ввести для каждого объекта собственный класс? Чем хуже такие разбиения некоторого разбиения с промежуточным числом классов?

Для ответа на эти вопросы необходимо определять понятие близости или сходства образов, поскольку требуется,

чтобы подмножества, на которые производится разбиение, включали в себя объекты в некотором смысле более похожие на объекты того же подмножества, чем на объекты, отнесенные к другим подмножествам. В дискриминантном подходе близость образов трактуется как расстояние между соответствующими точками в пространстве R^N, а группирование — как выделение кластеров — компактно расположенных наборов точек. В связи с этим в рамках дискриминантного подхода задача группирования часто называется задачей *кластеризации* (или кластер-анализа).

Хотя проблема распознавания без учителя сложнее, чем распознавания с учителем, не следует думать, что решение первой проблемы могло бы полностью избавить от необходимости решать вторую проблему. С практической точки зрения это две совершенно разные задачи. Представим, например, что необходимо построить систему распознавания целей. Даже если предположить, что в случае распознавания без учителя будут правильно выделены различные типы целей, в дальнейшем по результатам классификации неизвестных целей будет нельзя осуществлять принятие решения, так как к построенным в режиме обучения без учителя классам образов не будет приписана никакая семантическая информация. В действительности, при таком подходе крайне сомнительно, что выделенные классы образов будут соответствовать именно различным типам цели, а не другим возможным причинам различия их внешнего вида. К примеру, если условие инвариантности системы распознавания по отношению к расстоянию до цели не вводить априори (и не задавать это условие неявно, приписывая, как это имеет место при обучении с учителем, цели одного типа к одному классу вне зависимости от расстояния), то классы вполне могут быть сформированы по признаку расстояния. Таким образом, в задачах, в которых классы известны априори (и именно к одному из этих классов необходимо относить новый объект в процессе классификации), использование методов обучения без учителя крайне затруднительно.

Есть и задачи, в которых, напротив, не может быть применено распознавание с учителем. Это связано либо с тем, что классы образов неизвестны самому разработчику системы распознавания, либо не представляется возможным сформировать обучающую выборку. Например, когда ученые исследуют новый тип физических объектов, то может

оказаться, что эти объекты группируются в классы на основе каких-либо своих параметров (один такой пример с разделением звезд на спектральные классы мы затронем в п. 2.4). При автоматическом решении этой проблемы классы, конечно, заранее неизвестны. Невозможность сформировать обучающую выборку может возникнуть, например, в некоторых задачах анализа изображений. Например, аэрокосмические изображения поверхности Земли могут содержать только области заранее известных типов местности (поле, лес и т. д.), однако при этом может быть неизвестно, с какой высоты, под каким углом или в какое время года производилась съемка. Из-за этих и других факторов совершенно меняется то, как выглядят соответствующие типы местности на снимке. Таким образом, для каждого снимка необходимо формировать собственную обучающую выборку. При обучении с учителем был бы необходим человек-оператор, который для фрагментов изображения указывал, к какому типу местности они принадлежат. Однако, если не ставится задача распознавания, а необходимо лишь разделить изображения на области, каждая из которых соответствовала бы лишь одному типу местности, то эту задачу можно решать на основе обучения без учителя, т. е. осуществить группирование похожих фрагментов данного изображения.

Существует несколько вариаций постановки задачи группирования, встречающихся на практике. В одной из них число классов, которые необходимо сформировать, считается известным, и эта дополнительная информация заметно снижает сложность вычислений и позволяет получить более надежное решение (разумеется, если заданное число классов соответствует действительности). Такая постановка, как правило, возникает в том случае, когда анализируемые объекты хорошо известны человеку (и классифицированы им), но формирование обучающей выборки с заданным для каждого образа классом по тем или иным причинам не представляется возможным.

В качестве примера можно привести следующую задачу автоматического анализа аэрокосмических изображений. Пусть имеются фотографии характерного ландшафта, на котором присутствуют области леса, поля и водной поверхности, которые и нужно автоматически разделить. Можно ожидать, что других типов местностей не будет и что каждый из этих типов присутствует на фотографии. На каждом изоб-

ражении эти типы местности могут выглядеть по-разному из-за сезонно-суточных изменений, изменений высоты съемки и т. д. Поэтому создать обучающую выборку, состоящую из характерных фрагментов изображений каждого типа местности с указанием соответствующего типа, которая бы подходила для произвольного изображения, проблематично. Формировать подобную обучающую выборку вручную для каждого изображения неэффективно, зато можно автоматически создавать обучающую выборку, состоящую из небольших фрагментов изображений (а именно такие и нужны в данной задаче) без указания соответствующих им типов местности. Это и означает, что имеется задача распознавания без учителя с заданным количеством классов.

Другие две разновидности проблемы группирования возникают в зависимости от того, даются ли системе распознавания все образы из обучающей выборки одновременно или последовательно. В последнем случае говорят об *инкрементном обучении*. Сейчас данный термин применим к более широкому кругу методов, однако именно с задачей распознавания без учителя в случае последовательной выборки можно связать выделение машинного обучения в отдельную область исследований (см., например, [52, с. 66]). Отличительной особенностью инкрементного подхода является то, что система распознавания может «учиться» в процессе своего функционирования: при предъявлении нового, неизвестного ей образа подобная система может не только пытаться его классифицировать на основе уже накопленной (т. е. априорной по отношению к новому образу) информации, но также и включать этот образ в обучающую выборку при классификации последующих образов.

И наконец, существует еще одна задача, в которой считается неизвестным пространство образов, задача *выбора признаков*. В ней требуется из полученных исходных данных выделить характерные свойства объектов, на основе которых сформировать пространство описаний таким образом, чтобы в этом пространстве прочие задачи распознавания решались бы легче. Для этого на основе исходных данных следует отделить признаки классов образов (или *межклассовые признаки*) от *внутриклассовых признаков*. Первые представляют собой такие характеристики, которые одинаковы для всех объектов каждого класса, но различны для объектов разных классов, в то время как вторые описывают различия объектов внутри классов. Внут-

риклассовые признаки не несут полезной информации с точки зрения распознавания, напротив, их присутствие может его осложнить. В связи с этим выбор информативных признаков, как правило, сопровождается уменьшением объема исходных данных. В случае дискриминантного подхода это означает уменьшение размерности векторов признаков, описывающих объекты. В рамках синтаксического подхода выбор признаков соответствует выбору непроизводных элементов, являющихся основой для построения грамматик. Как правило, выбор непроизводных элементов осуществляется несинтаксическими методами.

В качестве примера возьмем проблему распознавания объекта по его изображению. Исходными данными здесь является массив значений интенсивностей для каждого пикселя. Очевидно, этот массив можно представить в виде многомерного вектора признаков, но распознавание в подобном пространстве признаков неосуществимо. Даже небольшая смена ракурса приведет к тому, что значения интенсивностей большинства пикселей изменятся, т. е. образы одного класса в данном пространстве занимают области, имеющие очень сложные формы. Понятно, что каждый из этих признаков (значений интенсивностей отдельных пикселей) мало информативен: даже выкинув значительную часть пикселей, человек сможет распознать объект практически так же хорошо, как и по исходному изображению. С другой стороны, чтобы отличить, например, мяч от линейки, нам достаточно только одного такого признака, как форма. Это означает, что в рамках данной задачи существуют некоторые признаки, в пространстве которых классы образов будут хорошо отделимы с помощью сравнительно простых процедур распознавания. К сожалению, такие признаки очень сложным образом зависят от исходных описаний, с чем и связана трудность проблемы выбора признаков.

Выбор признаков может быть необходим как при обучении с учителем, так и при обучении без учителя. В первом случае задача несколько изменяется и в некотором смысле становится легче: поскольку для векторов обучающей выборки известны соответствующие им классы, то качество каждого признака может быть напрямую оценено с точки зрения их применимости для разделения имеющихся классов. Если же пространство классов априори не задано, то приходится определять критерий качества признаков на основе гораздо более общих соображений.

Отметим также следующее. Если считать номер класса, к которому принадлежит некоторый объект, признаком, то выбор такого признака соответствует также решению и задаче распознавания. Это показывает, что выбор признаков и распознавание не являются принципиально различными задачами. Обе эти задачи можно интерпретировать как проблему выбора представления, в рамках которого производится описание объекта. Разница же между ними заключается в типах привлекаемых представлений, характеризующихся степенью их локальности. Если выбор признаков можно охарактеризовать как построение распределенного представления, то группирование — как построение ядерного представления [18]. Обнаружив, что обе задачи являются предельными случаями некоторой общей проблемы, можно попытаться осуществить синтез методов решения каждой из задач. Понятие локальности представления и некоторые методы работы с представлениями, объединяющими как свойства локальных, так и распределенных представлений, будут описаны в пп. 2.4 и 2.5.

2.1.5. Задачи распознавания в терминах индуктивного вывода

Выше мы привели описания нескольких задач распознавания образов. Однако эти описания не являются математически строгими постановками задач. Чтобы посмотреть, чего именно не хватает для достижения желаемой строгости, попробуем переформулировать эти задачи в терминах индуктивного вывода. Напомним, что любая задача индуктивного вывода характеризуется исходными данными D, пространством гипотез H и критерием рациональности $r(h|D)$, служащим для сравнения качества гипотез.

В задаче классификации установить, чему соответствуют эти элементы, достаточно легко. Описание единичного объекта $x \in X$ выступает в качестве исходных данных. Пространство гипотез совпадает с множеством классов A, а критерий рациональности — это целевая функция $\rho(x, a)$. Поскольку все эти величины считаются заданными извне, то классификация осуществляется путем вычисления величин $\rho(x, a)$ для разных классов и последующего выбора класса с максимальным значением $\rho(x, a)$, что и соответствует выбору лучшей гипотезы в индуктивном выводе.

Однако уже в задаче обучения с учителем не все понятия индуктивного вывода определяются столь однозначно. Имеющиеся исходные данные — это обучающая выборка, состоящая из описаний некоторого числа объектов и информации о том, к каким классам они принадлежат: $D = \left((x_1, a_1), (x_2, a_2), \ldots, (x_M, a_M)\right)$, где M — размер выборки. Поскольку целью является нахождение решающих правил, то пространство гипотез состоит из всех возможных отображений из пространства описаний во множество классов: $\mathrm{H} = \left\{\varphi \mid \varphi : \mathrm{X} \to \mathrm{A}\right\}$ либо $\mathrm{H} = \left\{\rho \mid \rho : \mathrm{X} \times \mathrm{A} \to R\right\}$. Таким образом, исходные данные и пространство гипотез в распознавании с учителем заданы, но возникает вопрос о критерии рациональности. Как будет показано в п. 2.3, для получения применимых на практике методов пространство решающих правил, из которого осуществляется выбор, сильно сужается, а критерий для определения лучшего правила в явном виде может и не вводиться, поскольку в таких суженных пространствах удается получить беспереборный алгоритм, строящий оптимальное решающее правило.

Похожей является ситуация и для задачи группирования. Исходными данными здесь служит набор образов $D = \left(x_1, x_2, \ldots, x_M\right)$, пространство гипотез состоит из различных разбиений множества $\left\{x_i\right\}_{i=1}^{M}$ на подмножества. Однако критерий качества разбиений в общей постановке задачи группирования, как правило, не вводится, и в большинстве алгоритмов группирования полный перебор всех возможных разбиений (а их число экспоненциально зависит от размера обучающей выборки) не проводится.

В задаче выбора признаков исходные данные могут быть либо такими, как в задаче распознавания с учителем, либо такими, как в задаче группирования, но для определения пространства гипотез требуется дополнительное уточнение постановки задачи, а именно: требуется определить, каким может быть новое представление данных, т. е. новые признаки, описывающие объекты. Вместе с тем определяется и критерий, по которому оценивается качество признаков. Существуют две противоположные точки зрения на вопрос о выборе признаков [120, с. 265]. Согласно одной из них, качество признаков необходимо определять через эффективность классификации, проведенной при использовании этих признаков. В соответствии с другой точкой зрения, качество признаков можно определить исходя из свойств имеющихся объектов, без об-

ращения к другим задачам распознавания. Первый подход используется в задачах обучения с учителем, в которых пространство классов известно априори и есть возможность определить эффективность классификации по обучающей выборке. При обучении без учителя классы не известны, поэтому естественным выглядит привлечение второго подхода. Однако возникает вопрос: действительно ли оптимальные признаки могут быть выбраны без привлечения информации о том, каким образом они в дальнейшем будут использоваться? Иными словами, существует ли универсальный критерий оптимальности представления данных?

Таким образом, с точки зрения индуктивного вывода в постановках различных задач распознавания часто не хватает критерия сравнения гипотез. Он вводится после установления дополнительных ограничений, но это и позволяет получать вычислительно эффективные решения. Далее мы попытаемся определить, какие дополнительные предположения обычно вводятся человеком, которые, с одной стороны, достаточно мягкие, чтобы соблюдаться в широком классе задач, а с другой — достаточно сильные, чтобы с их помощью избегать комбинаторного взрыва.

Вместе с тем мы попробуем проследить, как можно использовать универсальный информационный критерий для обобщения существующих методов распознавания.

2.2. КЛАССИФИКАЦИЯ ОБРАЗОВ

2.2.1. Решающие функции

Мы приступаем к рассмотрению дискриминантного подхода к распознаванию образов. Как и в других подходах, здесь вводится множество классов $A = \{a_1, a_2 \ldots a_d\}$, где d — число классов. Одной из особенностей этого подхода является то, что пространство описаний X уточняется как пространство признаков, являющееся N-мерным векторным пространством $X = R^N$. Такие пространства образов наиболее типичны для задач, в которых описание объекта формируется в результате измерения его физических характеристик. Это свойственно, в частности, проблеме машинного восприятия.

Начинаем рассмотрение с задачи классификации (распознавания без обучения). Для данного вектора $\vec{x} \in X$, $\vec{x} = (x_1, \ldots, x_N)^T$ требуется принять решение о его принад-

лежности некоторому классу из заданного множества A. В рамках дискриминантного подхода принятие такого решения основывается, как правило, на одной из таких концепций, как решающие (или дискриминантные) функции, критерий минимума расстояния и оценка апостериорных вероятностей. Решающие функции играют важную роль в дискриминантном подходе; они часто могут быть построены и в рамках тех методов, где их использование в явном виде не предполагается, поэтому они будут рассматриваться в первую очередь.

Решающей функцией $\varkappa(\vec{x})$ для двух классов $a_1, a_2 \in \mathrm{A}$ называется такая функция $\varkappa : \mathrm{X} \to R$, что $\varkappa(\vec{x}) > 0$, если образ $\vec{x}$ принадлежит классу a_1, и $\varkappa(\vec{x}) < 0$, если образ $\vec{x}$ принадлежит классу a_2.

Уравнение $\varkappa(\vec{x}) = 0$ задает поверхность, разделяющую два класса. Поскольку при принятии решения об отнесении образа к тому или иному классу абсолютные значения $\varkappa(\vec{x})$ внутри классов роли не играют, всю необходимую информацию о том, как следует разделять классы, несет именно эта поверхность, описывающая границы классов в пространстве признаков (рис. 2.2).

Разделяющая поверхность позволяет легко принимать решения при классификации образов. Однако класс может быть задан не через его границы с другими классами, а как отдельная область в пространстве признаков. Если области, соответствующие двум классам, не пересекаются, то эти классы называются *разделимыми* в данном пространстве признаков. Иными словами, разделимость классов означает, что для них существует дискриминантная функция. Напротив, если области пересекаются, то такой функции не существует и классы называются *неразделимыми*. Напри-

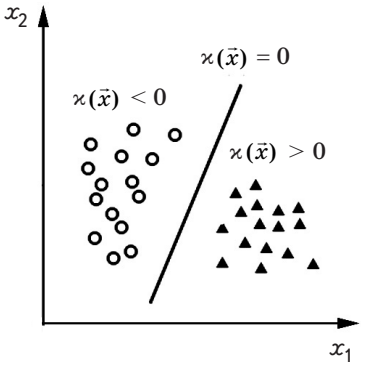

Рис. 2.2. Пример разделения образов на два класса дискриминантной функцией $\varkappa(\vec{x})$ в двумерном пространстве признаков

мер, классы, представленные на рис. 2.2, неразделимы, если пространство признаков ограничить лишь признаком x_2. В то же время они разделимы в исходном пространстве признаков.

Обычно рассматриваются не произвольные решающие функции, а лишь функции, относящиеся к некоторому параметрическому семейству, элементы которого $\varkappa(\vec{x}, \vec{w})$ определяются вектором параметров $\vec{w} = (w_1, \ldots, w_n)$, где n — число параметров. Выбор конкретных значений w_i соответствует выбору решающей функции. Определение оптимальных параметров по обучающей выборке осуществляется при решении задачи распознавания образов, которая будет рассмотрена далее. Сейчас же мы предполагаем, что они известны.

Наиболее простыми являются *линейные решающие функции*, задающиеся как

$$\varkappa(\vec{x}, \vec{w}) = w_1 x_1 + w_2 x_2 + \ldots + w_N x_N + w_{N+1}. \qquad (2.2)$$

При выборе конкретного семейства решающих функций может возникнуть ситуация, в которой данные классы не разделяются ни одной из функций семейства, несмотря на то, что сами классы разделимы. Например, если такая ситуация возникает в случае линейных решающих функций, то говорят, что эти классы не являются *линейно разделимыми* (рис. 2.3).

Итак, отличительной особенностью дискриминантных методов служит использование понятия разделяющих границ. Можно, конечно, привести абстрактный пример классов, для которых их введение бессмысленно. Рассмотрим, например, такие два класса, как рациональные и иррациональные числа. Оба множества рациональных и иррациональных чисел

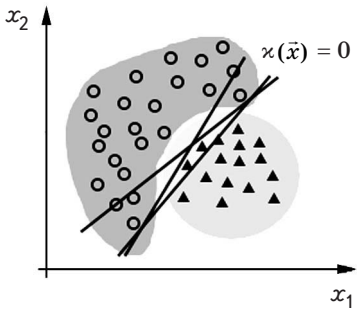

Рис. 2.3. Пример классов, не являющихся линейно разделимыми: любая попытка провести прямую, по каждую сторону от которой были бы только объекты одного класса, оказывается неудачной; при этом несложно провести нелинейную границу, которая бы разделяла классы

всюду плотны в множестве действительных чисел. Дискриминантной функцией для этих двух классов будет функция $\varkappa(x) = \varkappa(x) - 0{,}5$, где $\varkappa(x) = \lim\limits_{m\to\infty}\lim\limits_{n\to\infty}\left(\cos m!\,\pi x\right)^{2n}$ — функция Дирихле, равная единице в рациональных точках и нулю в иррациональных точках. Подобная решающая функция не обращается в ноль, т. е. у этих классов отсутствуют разделяющие границы (что, однако, самой решающей функции существовать не мешает). Похожий, но более реалистичный пример мы рассмотрим в п. 2.3.7.

На практике такие классы встречаются редко: границы классов могут варьироваться от линейных до существенно нелинейных, но сами границы все же существуют! Существование разделяющих поверхностей означает, что дискриминантная функция пересекает ноль при смене знака, что говорит об отсутствии разрывов в таких точках. Поскольку во внутренних точках классов требуется лишь сохранение знака решающей функции, то и в этих точках ее можно считать непрерывной. По сути, наличие разделяющих поверхностей соответствует непрерывности решающих функций, что и является одной из наиболее фундаментальных эвристик дискриминантного подхода. Следует заметить, что среди всех возможных функций доля непрерывных функций бесконечно мала, т. е. предположение о непрерывности, столь естественное для человека, вносит в систему распознавания огромный объем априорной информации.

Эту же эвристику можно сформулировать и в топологических понятиях. Для этого достаточно обратиться к интерпретации класса как области в пространстве признаков. Областью называют связное открытое множество (иногда, правда, условие связности опускают). Условие же открытости означает, что любая точка множества является внутренней, т. е. входит в него вместе с некоторой своей окрестностью. Очевидно, оба этих условия (связности и открытости) нарушаются при разделении чисел на рациональные и иррациональные.

Но вернемся к задаче классификации образов. На основе дискриминантной функции несложно построить решающее правило для двух классов:

$$\varphi(\vec{x}) = \begin{cases} a_1, & \varkappa(\vec{x}) > 0; \\ a_2, & \varkappa(\vec{x}) < 0. \end{cases} \qquad (2.3)$$

Применение этого решающего правила и дает решение задачи классификации.

В случае нескольких классов возможны различные определения решающей функции. Один из способов заключается в том, чтобы непосредственно воспользоваться определением для случая двух классов и ввести d^2 решающих функций $\varkappa_{ij}(\vec{x})$ (где d — это число классов), каждая из которых разделяет два разных класса a_i, $a_j \in$ A. Для таких дискриминантных функций $\varkappa_{ij}(\vec{x}) > 0$, если образ $\vec{x}$ не может принадлежать классу a_j; $\varkappa_{ij}(\vec{x}) < 0$, если образ не может принадлежать классу a_i. Решающее правило будет

$$\varphi(\vec{x}) = a_i \Leftrightarrow (\forall j)\, \varkappa_{ij}(\vec{x}) > 0. \qquad (2.4)$$

Поскольку $\varkappa_{ij}(\vec{x}) = -\varkappa_{ji}(\vec{x})$, а $\varkappa_{ii}(\vec{x})$ лишено смысла, то всего требуется построить $d(d-1)/2$ решающих функций.

Другой способ заключается в отделении данного класса одновременно от всех остальных. Для этого необходимо d дискриминантных функций $\varkappa_i(\vec{x})$, а решающее правило будет

$$\varphi(\vec{x}) = a_i \Leftrightarrow \varkappa_i(\vec{x}) > 0. \qquad (2.5)$$

Естественно, должно выполняться условие $\varkappa_i(\vec{x}) > 0 \Rightarrow$ $\Rightarrow (\forall j : j \neq i)\, \varkappa_j(\vec{x}) < 0$.

Второй вариант кажется предпочтительнее, так как требует меньшего числа решающих функций. Однако построение таких решающих функций сложнее, особенно если они выбираются из простых семейств (рис. 2.4).

В общем случае решающую функцию можно определить как функцию, разделяющую два подмножества множества классов. Очевидно, что это определение обобщает оба под-

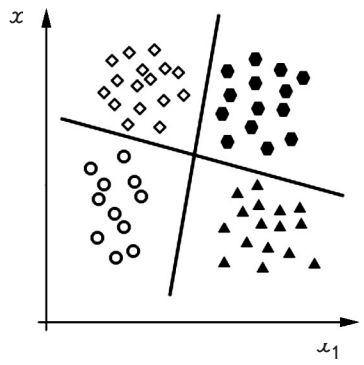

Рис. 2.4. Пример четырех классов, не каждый из которых может быть отделен линейными решающими функциями. Попарно все классы являются линейно разделимыми, поэтому достаточно шести решающих функций. Однако благодаря тому, что одни и те же решающие функции могут быть использованы для разделения различных пар классов, оказывается достаточно лишь двух решающих функций, что даже лучше четырех решающих функций, которые бы потребовались в случае отделения каждого класса от всех остальных

хода: в первом случае производится разделение таких множеств, как $\{a_i\}$ и $\{a_j\}$, а во втором — $\{a_i\}$ и $A\backslash\{a_i\}$. Разделение подмножеств классов может позволить еще уменьшить количество решающих функций (см. рис. 2.4), однако оно не может стать меньше $[\log_2 d]$.

Задача минимизации количества решающих функций, достаточных для классификации образов, особенно важна в том случае, если число классов d велико. Если представить себе, с каким числом классов объектов (или понятий) имеет дело человек, то становится ясно, что решение этой проблемы в том или ином виде потребуется при разработке универсальной системы машинного обучения. Мы, однако, этот вопрос здесь рассматривать не будем, а перейдем к изучению близкого, но несколько отличающегося по своим предпосылкам подхода к классификации, основанного на функциях расстояния.

2.2.2. Критерии, основанные на функциях расстояния

В описанном выше подходе классификация образа осуществляется на основе решающей функции, принадлежащей некоторому параметрическому семейству. Это вносит определенные ограничения на возможность разделения классов системой распознавания. Один из подходов, не обращающихся напрямую к дискриминантным функциям, использует следующие соображения. При разделении объектов на классы в каждый класс должны попадать объекты, которые чем-то схожи между собой. Степень различия между образами удобно трактовать как расстояние между ними в пространстве описаний, что соответствует введению метрики на множестве X. Использование функции расстояния в целях классификации принято считать одним из простейших и наиболее эвристических подходов [120, с. 89], и многие подобные методы действительно соответствуют этой характеристике. Однако эвристичность здесь заключается не в привлечении понятия расстояния как такового (это не накладывает никаких дополнительных ограничений на метод), а в использовании конкретной метрики. Кроме того, функции расстояния являются одним из основных средств решения проблемы кластеризации.

Поскольку сейчас мы рассматриваем задачу классификации, то мера сходства входит неотъемлемой частью в ре-

шающее правило, которое считается известным. Для простоты будем считать, что метрика пространства образов евклидова. К вопросу о выборе критерия сходства мы вернемся в п. 2.4.7 при обсуждении методов кластеризации. Сейчас же просто заметим, что евклидово расстояние между двумя образами далеко не всегда является адекватной мерой их сходства, даже когда пространство признаков является пространством R^N. В качестве примера рассмотрим изображение некоторого предмета, описанное как многомерный вектор. Другое изображение этого же предмета, но снятого немного с другого ракурса, будет описываться вектором, далеко отстоящим от вектора, соответствующего первому изображению, однако сами изображения будут казаться для человека очень похожими. Тем не менее для многих задач евклидова метрика оказывается вполне приемлемой (см., например, рис. 2.2: образы одного класса расположены друг к другу ближе, чем к образам другого класса).

В самом простом случае каждый класс может быть представлен единственным эталонным образом. Тогда новый образ, подлежащий классификации, относится к тому классу, эталонный образ которого расположен наиболее близко. Иными словами, если неизвестный объект больше всего похож на характерного представителя некоторого класса, то он относится к этому классу.

Пусть $\vec{y}_i \in \mathrm{X}$, $i = 1, \ldots, d$ — эталонные образы, заданные для каждого класса. В случае единственности эталона эти образы также называют центрами классов. Для пространства признаков с евклидовой метрикой расстояния от нового образа $\vec{x} \in \mathrm{X}$ до центров классов будет равно:

$$s_i = \|\vec{x} - \vec{y}_i\| = \sqrt{(\vec{x} - \vec{y}_i)^T (\vec{x} - \vec{y}_i)}. \qquad (2.6)$$

Рассмотрим случай двух классов. Поскольку выбирается класс, расстояние до центра которого меньше, то несложно построить решающую функцию

$$\varkappa(\vec{x}) = s_1^2 - s_2^2 = (\vec{x} - \vec{y}_1)^T (\vec{x} - \vec{y}_1) - (\vec{x} - \vec{y}_2)^T (\vec{x} - \vec{y}_2) =$$
$$= 2(\vec{y}_2 - \vec{y}_1)^T \vec{x} + (\|\vec{y}_2\|^2 - \|\vec{y}_1\|^2), \qquad (2.7)$$

из которой видно, что решающая функция линейна по $\vec{x}$ (см. также рис. 2.5). Таким образом, классификация по минимуму (евклидова) расстояния является частным случаем применения линейных решающих функций.

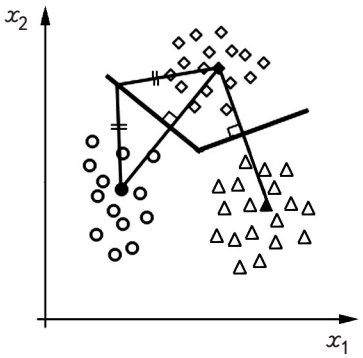

Рис. 2.5. Пример разделения классов (○, △, ◇) по минимуму расстояния. Каждый класс характеризуется единственным эталоном (●, ▲, ◆). Множество точек, равноудаленных от двух эталонов, представляет собой прямую линию

Но, несмотря на это, рассмотрение классификации по минимуму расстояния интересно тем, что в ней более отчетливо видна строящаяся модель (а мы выяснили, что классификация может интерпретироваться как построение модели). Этой моделью является эталон выбранного класса. Пусть $\vec{x} = \vec{y}_i + \Delta\vec{x}_i$. Поскольку эталоны в задаче классификации известны априори, то полным описанием объекта будет пара $(a_i, \Delta\vec{x}_i)$, по которой можно восстановить исходный образ. Далее выбирается такой номер класса, чтобы значение $s_i = \|\Delta\vec{x}_i\|$ было минимальным, что означает выбор наиболее точной модели. Понятие точности определяется выбранной метрикой.

Но возникает вопрос: моделью чего является решающая функция вида (2.3)? Заметим, что во всех рассмотренных выше примерах классы занимали сравнительно небольшие области, а решающие функции относили к каждому из классов бесконечные области. Также в случае нескольких классов их приходилось разделять попарно, т. е. решающие функции описывают (являются моделью) именно различия между классами, а не сами классы. Это может оказаться неудобным, когда системе распознавания предъявляется образ, не относящийся ни к одному из ранее выученных классов (рис. 2.6).

Существуют различные расширения подхода, использующего эталонные образы. Одно из них заключается в использовании нескольких эталонов для каждого класса. Поскольку каждый эталон можно рассматривать как центр некоторого подкласса, то такие подклассы будут разделяться линейными границами, а значит, разделяющие границы для самих классов будут кусочно-линейными (это также видно из рис. 2.5, если два нижних класса рассматривать как два

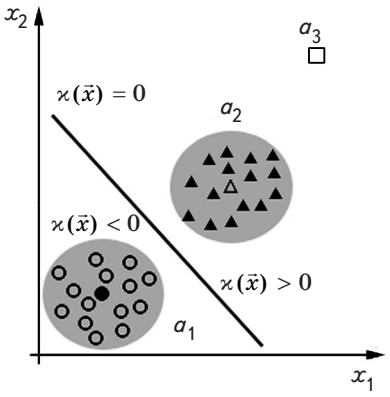

Рис. 2.6. Пример, поясняющий отличие дискриминантного метода распознавания, описывающего различия между классами с помощью решающей функции $\varkappa(\vec{x})$, от метода, описывающего сами классы через их центры и размеры. Здесь a_1, a_2 — классы, образы которых составляли обучающую выборку; a_3 — новый класс, образы которого будут относиться функцией $\varkappa(\vec{x})$ к классу a_2. Решающая функция при этом не содержит информации, позволяющей обнаружить подобные ситуации, в то же время явное описание классов дает возможность определить, насколько предъявленный объект характерен для того или иного класса

подкласса одного класса с двумя эталонами). Кусочно-линейные разделяющие поверхности также не выходят за парадигму дискриминантного подхода, но задают достаточно интересное семейство решающих функций: любую непрерывную границу можно аппроксимировать сколь угодно точно посредством кусочно-линейных функций, если не ограничивать число эталонов. Однако при таком неограниченном увеличении их числа сам смысл понятия «эталон» несколько теряется.

Проблема выбора положения (и количества) эталонов, а также метрики пространства признаков решается в рамках задач распознавания и кластеризации, поэтому мы к ним вернемся в п. 2.3.

2.2.3. Статистический подход

Однозначное отнесение образа к одному из классов при детерминистском подходе затрудняет прямое оценивание качества классификации. При статистическом подходе, напротив, полагается, что объект может принадлежать любому из классов, но с некоторой вероятностью. Сами же образы рассматриваются как отсчеты некоторого случайного вектора $\vec{X}$, имеющего значения из множества X и характеризующегося плотностью вероятности $p(\vec{x})$. Статистический подход также позволяет определять вероятность ошибочной классификации, с помощью которой и оценивается качество классификации.

Со статистической точки зрения, оптимальному качеству классификации соответствует байесовский классификатор

[120, с. 130]. Запишем правило Байеса (1.4) для данной задачи:

$$P(a_i \mid \vec{x}) = \frac{P(a_i)p(\vec{x} \mid a_i)}{p(\vec{x})}, \quad a_i \in \text{A}. \tag{2.8}$$

Напомним, что через $P(a_i \mid \vec{x})$ обозначается апостериорная вероятность класса a_i, т. е. вероятность того, что наблюдаемый образ $\vec{x}$ принадлежит классу a_i. Вероятность $P(a_i)$ — априорная вероятность получения образа, принадлежащего классу a_i, а $p(\vec{x} \mid a_i)$ — плотность распределения вероятностей образов класса a_i или правдоподобие того, что образ $\vec{x}$ принадлежит данному классу.

Рассмотрим простой пример байесовской классификации в одномерном пространстве признаков (рис. 2.7). Пусть для каждого из двух классов a_1 и a_2 есть возможность вычислить их апостериорные вероятности для каждого образа $P(a_i \mid x)$. И пусть все образы, для которых значение признака меньше некоторого порога $x < x_0$, относятся (классификатором) к классу a_1, а образы с $x > x_0$ — к классу a_2. Тогда величина P_1 (площадь под соответствующим графиком) характеризует вероятность ошибочного отнесения объекта второго класса к первому классу (или вероятность *ложной тревоги*, если класс a_1 отвечает понятию «цель»). Вероятность P_2 — это вероятность ошибочного отнесения объекта первого класса ко второму классу (вероятность *пропуска цели*).

Естественно, что эти два события (пропуск цели и ложная тревога) могут быть неравнозначными. Поэтому построение байесовского классификатора в общем случае означает минимизацию среднего риска, для чего привлекается матрица потерь. Мы договорились ее опускать, чтобы упростить запись, так как на смысл методов она не влияет (но влияет на резуль-

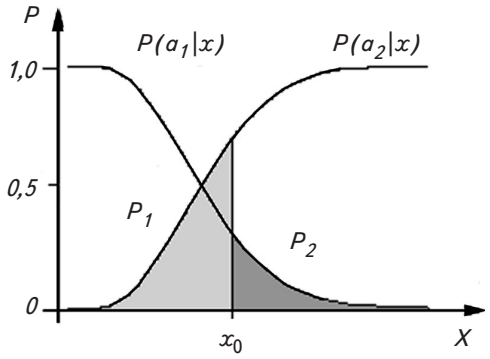

Рис. 2.7. Пример байесовской классификации для одного признака

тат, поэтому на практике, конечно, ее нужно использовать). В связи с этим вместо поиска модели, минимизирующей математическое ожидание потерь, мы будем искать наиболее вероятную модель. В задаче классификации это соответствует поиску такого класса $a_i \in A$, для которого максимальна апостериорная вероятность $P(a_i \mid \vec{x})$, а разделяющие поверхности задаются уравнениями $P(a_i \mid \vec{x}) = P(a_j \mid \vec{x})$.

Если в случае двух классов на вероятности накладывается ограничение $p(\vec{x} \mid a_1) + p(\vec{x} \mid a_2) = 1$, то это означает, что строится лишь одна стохастическая модель, которая описывает различия между классами a_1 и a_2 (иными словами, априорно предполагается, что одна из моделей верна, чем и объясняется сложность выявления объектов новых классов). В противном случае каждый класс a_i описывается отдельной стохастической моделью, задающей вероятности $p(\vec{x} \mid a_i)$.

Поскольку величина $p(\vec{x})$ не влияет на выбор класса, ее обычно не рассматривают. При классификации считается, что априорные вероятности $P(a_i)$ известны, а значения правдоподобий $p(\vec{x} \mid a_i)$ могут быть вычислены. Для этого они должны быть представлены в некотором удобном для вычисления виде. Классическим подходом, использующимся для построения байесовских классификаторов, является представление плотностей распределений условных вероятностей в виде нормального закона (см., например, [52, с. 85–90; 120, с. 136–141]):

$$p(\vec{x} \mid a_i) = \frac{1}{(2\pi)^{N/2} \left|C_i\right|^{1/2}} \exp\left[-\frac{1}{2} (\vec{x} - \vec{y}_i)^T C_i^{-1} (\vec{x} - \vec{y}_i) \right], \quad (2.9)$$

где $\vec{y}_i = E_i[\vec{x}]$ — среднее; $C_i = E_i\left[(\vec{x} - \vec{y}_i)(\vec{x} - \vec{y}_i)^T \right]$ — ковариационная матрица распределения $p(\vec{x} \mid a_i)$, которые в задаче классификации считаются известными; $\left|C_i\right|$ — определитель соответствующей матрицы.

Рассмотрим случай двух классов. Равенство апостериорных вероятностей даст уравнение разделяющей поверхности $p(a_1) p(\vec{x} \mid a_1) = p(a_2) p(\vec{x} \mid a_2)$. После логарифмирования получим

$$\begin{aligned} &2 \ln p(a_1) - \ln \left|C_1\right| - (\vec{x} - \vec{y}_1)^T C_1^{-1} (\vec{x} - \vec{y}_1) = \\ &= 2 \ln p(a_2) - \ln \left|C_2\right| - (\vec{x} - \vec{y}_2)^T C_2^{-1} (\vec{x} - \vec{y}_2). \end{aligned} \quad (2.10)$$

Таким образом, в случае нормального распределения классы разделяются поверхностью второго порядка (то, что обратное неверно, будет показано далее, в п. 2.3.4). Если плотности распределений действительно распределены по нормальному закону, то никакие поверхности другого вида не будут в среднем давать лучшего качества классификации. На самом деле, этот результат верен и для более общего случая: если плотности распределения $p(\vec{x} \mid a_1)$ и $p(\vec{x} \mid a_2)$ являются симметричными и монотонно убывающими от центра симметрии, то байесовская граница, разделяющая классы a_1 и a_2, является поверхностью не более чем второго порядка [120, с. 150]. Это, в свою очередь, говорит о том, что предположения о виде плотности распределения вероятностей являются гораздо более сильными, чем о виде разделяющих поверхностей.

Рассмотрим частный случай, когда $C_1 = C_2 = \sigma^2 E$, где E — единичная матрица. Нетрудно убедиться, что разделяющая поверхность при этом является гиперплоскостью, а априорные вероятности $P(a_{1,2})$ определяют смещение этой гиперплоскости относительно того положения, которое было бы выбрано согласно критерию минимума расстояния. Если трактовать математические ожидания $\vec{y}_i$ как эталонные образцы, то в случае произвольных ковариационных матриц максимизация апостериорной вероятности будет соответствовать минимизации расстояния Махаланобиса:

$$s(\vec{x}, \vec{y}_i) = (\vec{x} - \vec{y}_i)^T C_i^{-1} (\vec{x} - \vec{y}_i). \qquad (2.11)$$

Таким образом, между методами, применяющими функции расстояния, и методами, использующими правило Байеса, существует тесная взаимосвязь.

2.2.4. Информационный критерий

Задача классификации является сравнительно простой, по крайней мере, в рамках дискриминантного подхода. Основные сложности возникают в задачах распознавания и кластеризации. Если же они преодолены, то классификация сводится либо к вычислению решающих функций, либо к определению расстояний, либо к вычислению вероятностей по известным функциям. В этом смысле привлечение принципа минимальной длины описания вряд ли сможет улучшить имеющиеся резуль-

таты. Однако использование информационного критерия дает возможность под несколько другим углом взглянуть на данную проблему, что позволяет лучше понять суть процесса классификации. А поскольку информационный подход будет также использован в задачах распознавания и кластеризации, то целесообразно кратко остановиться на нем и здесь, как это было сделано для трех вышеописанных подходов.

Будем считать, что целью классификации является преобразование без потери информации исходного описания, заданного в виде вектора признаков $\vec{x}$, некоторого объекта в новое описание, являющееся как можно более коротким. При этом в качестве дополнительной априорной информации выступают модели μ_i, описывающие каждый класс a_i. Сам же класс (или его номер), соответствующий некоторому общему понятию, — это лишь ссылка на модель, ее «имя». Новое описание объекта должно включать номер класса i, к которому он был отнесен, и данные ε_i, описывающие его отличия от других объектов того же класса (внутриклассовые признаки). Отсутствие потерь информации означает, что $U(\mu_i, \varepsilon_i) = \vec{x}$. Тогда лучшим, согласно принципу МДО, будет класс

$$a = \arg\min_{a_i \in A} \left[l(\mu_i) + K(\vec{x} \mid \mu_i) \right]. \qquad (2.12)$$

Значения ε_i определяются при нахождении $K(\vec{x} \mid \mu_i)$.

Таким образом, классификация представляется в качестве частного случая индуктивного вывода и может быть решена с применением принципа МДО. Однако это обычно не требуется, поскольку априорное знание моделей классов μ_i и конечность их числа позволяют применять правило Байеса. Действительно, если μ_i соответствует стохастической модели, задающей плотность вероятностей $P(\vec{x} \mid a_i)$, то $K(\vec{x} \mid \mu_i) \approx -\log_2 P(\vec{x} \mid a_i)$. Поскольку выбор производится не из универсального пространства моделей, а из набора $\mu_i, \ldots, \mu_d$, то вместо опорной машины U может использоваться гораздо более ограниченное устройство, для которого $l(\mu_i) \approx -\log_2 P(a_i)$ (например, μ_i — это просто код Хаффмана, обозначающий номер соответствующей модели с учетом ее априорной вероятности). Напротив, если априорные вероятности $P(a_i)$ все же неизвестны, то можно воспользоваться универсальным распределением $2^{-l(\mu_i)}$.

С принципом МДО несложно связать и методы, основанные на минимизации расстояния. Действительно, в случае

единственного эталона новым описанием объекта является пара $(a_i, \Delta\vec{x}_i)$, а моделью класса — процедура, осуществляющая сложение невязки $\Delta\vec{x}_i$ с эталонным образом $\vec{y}_i$. Предположим, что исходные векторы заданы с некоторой фиксированной точностью. Тогда вещественные числа можно кодировать следующим образом: сначала записывать количество значащих бит, а затем сами эти биты. Тогда длина описания вектора невязок $\Delta\vec{x}_i$ будет пропорциональна логарифму от его модуля, т. е. минимизация расстояния $\|\vec{x} - \vec{y}_i\|$ в таком представлении будет равносильна минимизации длины описания. Выбор же другой метрики соответствует выбору другого представления вещественных чисел.

Модели, описывающие разные классы, могут быть разных типов, в частности, для разных классов вполне можно использовать различные функции расстояний. В каком-то смысле это делается, например, в случае расстояния Махаланобиса, которое параметризуется ковариационной матрицей, различающейся для разных классов.

Таким образом, несложно свести многие из существующих методов классификации к принципу МДО, что порой позволяет явно выделить заложенные в них ограничения. Однако попытка построить систему классификации на основе общего уравнения (2.12), в которое не заложено ограничений на тип модели μ_i, привела бы к неудаче, даже несмотря на то, что выбор производился из конечного числа моделей μ_i, ..., μ_d.

Проблема эта связана со сложностью вычисления значений $K(\vec{x} \mid \mu_i)$, а точнее, с нахождением ε_i по $\vec{x}$. Во всех практических методах вычисление внутриклассовых признаков не представляет сложности. В байесовских методах оно делается неявно при вычислении $P(\vec{x} \mid a_i)$, а в методах, использующих понятие расстояния, переход к невязкам осуществляется просто переносом одного слагаемого в другую часть равенства: $\Delta\vec{x}_i = \vec{x} - \vec{y}_i$. В то же время для нахождения таких невязок, руководствуясь формулой (2.12), придется перебирать все возможные векторы $\Delta\vec{x}_i$ в поисках такого, который бы удовлетворил условию $\vec{x} = \Delta\vec{x}_i + \vec{y}_i$.

Это связано с тем, что не существует быстрого (с полиномиальным временем) алгоритма обращения произвольного отображения $\mu : \{0,1\}^* \to \{0,1\}^*$, означающего в данном контексте модель класса.

Сделаем небольшое отступление. Из-за наличия проблемы обращения модели во многих областях отдельно изуча-

ются *генеративные* (или порождающие) модели и *дескриптивные* (или описательные) модели. Генеративные модели по описанию порождают сам объект, а дескриптивные модели, напротив, отображают исходный объект в его описание. Из-за этого для построения описания объекта на основе генеративной модели в общем случае приходится прибегать к полному перебору, а при конструировании дескриптивной модели, если она априорно неизвестна, нет возможности проверить ее адекватность. Одновременное наличие обеих моделей возможно только для сравнительно простых задач. Именно поэтому такие отрасли, как, например, компьютерное зрение (анализ изображений) и компьютерная графика (синтез изображений) развиваются практически независимо. Следует заметить, что и в человеческом мозге эта проблема полностью не решена, о чем говорит, например, то, что формирование и понимание речевых сообщений происходят во многом независимо (см., например, [133]).

Тем не менее в рамках задачи классификации эта проблема человеком все время решается за счет сужения класса рассматриваемых моделей. К сожалению, как такое сужение выполнять автоматически, еще неизвестно. Здесь же мы пытаемся пока лишь проследить, как оно осуществляется человеком.

2.3. РАСПОЗНАВАНИЕ С УЧИТЕЛЕМ

2.3.1. Линейные решающие функции и опорные векторы

Задача распознавания образов заключается в построении решающих функций, т. е. пространством моделей является некоторое множество функций (как правило, бесконечное), из которого требуется выбрать лучшую. В задаче классификации пространство моделей было конечным, поэтому естественно, что задача распознавания является существенно сложнее, а реальные системы распознавания обычно используют некоторые ограниченные множества функций (а это означает, что они способны выучить не любое понятие, представленное некоторым классом образов). Классическим методом распознавания в рамках дискриминантного подхода служит рассмотрение линейных решающих функций. Помимо того, что поиск линейных решающих функций проще как

с теоретической, так и с практической точки зрения, классификаторы, построенные на их основе, являются также и наиболее эффективными по отношению к требуемым вычислительным ресурсам [127].

Мы рассмотрим построение линейных решающих функций для случая двух классов a_1 и a_2. В задаче распознавания имеются исходные данные: $D = \left((\vec{x}_1, A_1), (\vec{x}_2, A_2), \ldots, (\vec{x}_M, A_M)\right)$, где $\vec{x}_i \in R^N$; $A_i \in \{a_1, a_2\}$ — обучающая выборка из M элементов. Необходимо построить линейную решающую функцию вида

$$\varkappa(\vec{x}, \vec{w}) = w_1 x_1 + w_2 x_2 + \ldots + w_N x_N + w_{N+1} = \vec{w}\vec{x}', \quad (2.13)$$

где $\vec{x}' = (x_1, x_2, \ldots, x_N, 1)^T$ — пополненный вектор признаков; $\vec{w} = (w_1, w_2, \ldots, w_{N+1})$ — вектор весов, который требуется определить по обучающей выборке исходя из условий $\vec{w}\vec{x}_i' > 0$, если $A_i = a_1$, и $\vec{w}\vec{x}_i' < 0$, если $A_i = a_2$.

Чтобы представить эти условия единообразно, обычно пользуются следующим приемом. Пусть $z_i = 1$, если $A_i = a_1$, и $z_i = -1$, если $A_i = a_2$. Тогда ограничения на вектор параметров будут иметь вид:

$$(\forall i)\, z_i \vec{w}\vec{x}_i' > 0. \quad (2.14)$$

Если хотя бы одно решение существует, то их бесконечно много. Это вытекает из тех соображений, что неравенства в выражении (2.14) являются строгими, т. е. ограничения на каждую компоненту вектора $\vec{w}$ при фиксированных других компонентах задаются в виде пересечения интервалов (открытых множеств), которые либо содержат бесконечно много точек, либо являются пустыми. А поскольку решений может быть бесконечно много, то необходим некоторый дополнительный критерий, однозначно определяющий, какая из разделяющих гиперплоскостей лучше.

В зависимости от критерия оптимальности и метода поиска параметров, максимизирующих этот критерий, можно построить различные процедуры нахождения линейных решающих функций. Цель книги, однако, не состоит в систематическом изложении теории распознавания образов, поэтому здесь будет рассмотрен только один такой метод — *метод опорных векторов* (SVM, Support Vector Machine). Еще один часто встречающийся метод основан на обучении многослойного перцептрона (или нейронной сети прямого распространения) [120, с. 178–189; 134, с. 79–92]. Су-

ществуют и другие методы решения этой задачи (см. например, [120]).

Основы метода опорных векторов были заложены в начале 70-х годов прошлого века: постановка и решение задачи для линейного случая были представлены, например, в работе [135]. Популярность же метода, которую он приобрел в 1990-х годах, после выхода работ [136, 137], была вызвана открытием возможности его расширения на случай нелинейных разделяющих границ (этот способ обобщения метода несколько отличен от тех, которые применялись для других линейных методов). Долгое время этот метод применялся только для обучения с учителем, однако в настоящее время появилась возможность его использования и в задачах кластеризации [138].

Смысл критерия оптимальности, привлекающегося в методе опорных векторов, заключается в следующем. Пусть есть некоторая разделяющая гиперплоскость. Ее можно перемещать параллельным переносом (меняя параметр w_{N+1}) в некоторых пределах так, что она все еще будет разделять классы. Совокупность таких параллельных гиперплоскостей образует полосу определенной ширины. В зависимости от ориентации, определяемой параметрами w_1, w_2, …, w_N, ширина полосы будет различной (рис. 2.8). Ширина полосы и является критерием качества.

Сама полоса касается одного или более векторов каждого из классов; эти векторы называются *опорными*. Среди всех гиперплоскостей, принадлежащих наиболее широкой полосе, выбирается та, расстояние до которой от опорных векторов одинаково (т. е. расположенная посередине полосы). Заметим, что положение лучшей гиперплоскости определяется лишь опорными векторами; расположение

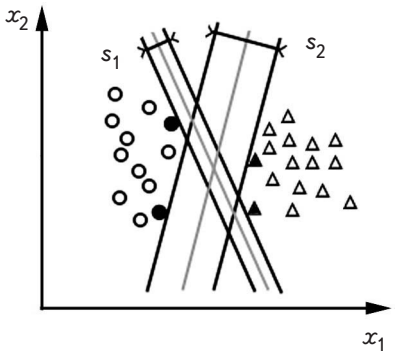

Рис. 2.8. Образы, принадлежащие двум классам (○, △) и две полосы, разделяющие эти классы. Из-за разной ориентации полосы касаются разных опорных векторов (●, ▲) и имеют разную толщину s_1 и s_2

других векторов обучающей выборки никак не учитывается. Эти условия позволяют определить лучшую гиперплоскость единственным образом (за исключением некоторых вырожденных случаев расположения векторов обучающей выборки).

Чтобы вычислить ширину полосы, используют следующий прием. Если неравенства (2.14) справедливы, то всегда можно найти такую положительную константу, после умножения на которую будут верны следующие неравенства:

$$(\forall i)\, z_i \vec{w} \vec{x}_i' \geq 1, \qquad (2.15)$$

причем хотя бы для одного вектора $\vec{x}_i'$ равенство будет точным. Гиперплоскость находится в центре соответствующей ей полосы, если расстояния от опорных векторов до нее равны. Расстояние от точки до плоскости вычисляется по формуле

$$s = \frac{\left|\vec{w}\vec{x}'\right|}{\sqrt{w_1^2 + \ldots + w_N^2}}. \qquad (2.16)$$

Наименьшее расстояние достигается на опорных векторах, для которых $\left|\vec{w}\vec{x}'\right| = 1$ (знак выражения $\vec{w}\vec{x}'$ будет различным для опорных векторов разных классов). Поскольку искомая плоскость отделена таким расстоянием от обоих классов, то ширина соответствующей полосы составит

$$s_0 = 2\left(w_1^2 + \ldots + w_N^2\right)^{-0,5}. \qquad (2.17)$$

Вместо максимизации расстояния s_0 удобнее минимизировать обратную величину при системе ограничений (2.15). Эта задача решается методом неопределенных множителей Лагранжа. Для этого составляется лагранжиан

$$L(\vec{w}, \vec{\lambda}) = \frac{1}{2}\left(w_1^2 + \ldots + w_N^2\right) - \sum_{i=1}^{M} \lambda_i \left(z_i \vec{w} \vec{x}_i' - 1\right), \qquad (2.18)$$

который минимизируется при следующих ограничениях: $\lambda_i \geq 0, i = 1, \ldots, M$ и $\lambda_i \left(z_i \vec{w} \vec{x}_i' - 1\right) = 0$ (λ_i — множители Лагранжа).

Условный экстремум лагранжиана определяется путем его дифференцирования по $\vec{w}$ и $\vec{\lambda}$ и приравнивания соответствующих производных нулю. Полное решение этой задачи достаточно громоздкое и его можно найти в соответ-

ствующей литературе (см., например, [139, 140]). Здесь мы приведем лишь результат:

$$w_k = \sum_{i=1}^{M} \lambda_i z_i x_{i,k}, \quad k = 1, \ldots, N. \tag{2.19}$$

Значение w_{N+1} получается из соотношения $z_s \vec{w} \vec{x}_s' = 1$, записанного для произвольного опорного вектора. Следует также заметить, что множители λ_i отличны от нуля только для опорных векторов. Сами же значения λ_i определяются путем минимизации квадратичной функции при некоторых ограничениях.

Как саму функцию, так и алгоритм ее минимизации, мы приводить не будем (см., например [139, 141]), но заметим, что значения коэффициентов этой функции зависят только от произведений $\vec{x}_i \vec{x}_j^T$, а не от самих векторов. Более того, подставив параметры (2.15) в уравнение решающей функции (2.13), нетрудно убедиться, что и для вычисления решающей функции достаточно знать только произведения $\vec{x}_i \vec{x}_j^T$, а не сами векторы. Более подробно метод опорных векторов, в частности его обобщение на случай нескольких классов, описан в литературе [142].

2.3.2. Обобщенные решающие функции и ядра

Не любые два набора точек в RN разделяются гиперплоскостью. Это является платой за простоту и вычислительную эффективность. Нелинейные методы сложны и трудоемки. К счастью, существует стандартный прием, позволяющий расширять процедуры построения линейных решающих функций на нелинейные. Это введение обобщенных решающих функций вида [120, с. 62]

$$\kappa(\vec{x}, \vec{w}) = w_1 f_1(\vec{x}) + w_2 f_2(\vec{x}) + \ldots + w_n f_n(\vec{x}), \quad f_i : R^N \to R. \tag{2.20}$$

В частности, несложно получить линейные решающие функции, используя $n = N + 1$ и $f_i(\vec{x}) = x_i'$.

Функции $f_i(\vec{x})$ предполагаются известными заранее, т. е. имеется возможность однозначно получить их значения для любого вектора $\vec{x}$. Тогда решающие функции вида (2.20) оказываются линейными по неизвестным параметрам, и несложно убедиться, что любой метод, предназначенный для нахождения параметров линейных решающих функций,

также будет работать и для обобщенных решающих функций.

Один из стандартных способов задания обобщенных решающих функций — представление их в виде многочленов (при этом обычно используют ортонормированные системы функций, например многочлены Лежандра или Эрмита). По сути, это означает, что для любой непрерывной (в некотором замкнутом интервале) решающей функции существует последовательность обобщенных решающих функций, равномерно сходящихся (на этом интервале) к ней. Это справедливо согласно теореме Вейерштрасса о приближении функций. Итак, если не ограничивать число n слагаемых в разложении (2.20), то можно описать любую (непрерывную) решающую функцию. Также нетрудно убедиться, что если в состав различных классов не входят идентичные векторы образов, то всегда можно найти разделяющие границы [120, с. 62].

Интересно, однако, отметить, что использование функций вида (2.20) равносильно использованию нового пространства описаний R^n с признаками $\chi_i = f_i(\vec{x})$. Во-первых, это говорит нам, что выбор подходящих признаков может сделать классы линейно разделимыми, а задачу распознавания достаточно простой. Во-вторых, привлечение обобщенных решающих функций эту задачу не решает, так как функции $f_i(\vec{x})$ задаются априорно, а не строятся автоматически.

Сопоставим эту интерпретацию с возможностью разложения решающих функций в бесконечный ряд. Получаем, что в рамках данного подхода генерируется потенциально бесконечное число признаков, что вызывает определенные сомнения. Действительно, человек стремится использовать минимально необходимое число признаков, с помощью которых он описывает те или иные классы объектов. Казалось бы, аппроксимация решающей функции многочленами математически корректна. В чем же тут проблема? Почему возникает впечатление, что такой подход может дать плохой результат?

Рассмотрим простой пример (рис. 2.9): реальные классы, по которым формируется обучающая выборка, являются неразделимыми. Тем не менее, взяв достаточно большое число членов в обобщенной решающей функции, образы из обучающей выборки можно разделить. Причем, чем больше размер обучающей выборки, тем сложнее будут решающая функция и соответствующая ей разделяющая поверх-

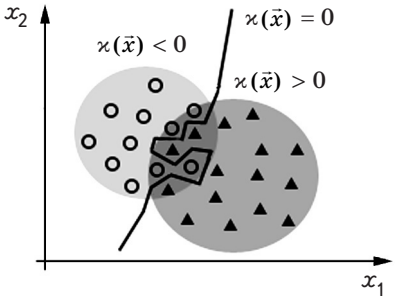

Рис. 2.9. Пример двух неразделимых классов. Использование чрезмерно сложной решающей функции дает неадекватное описание конфликтной области. Хотя данный результат приведен для кусочно-линейной решающей функции, аналогичная картина будет при задании решающей функции в виде полиномов произвольной степени

ность. Но сами классы, представленные на рис. 2.9, очень простые, а значит, и решающая функция, являющаяся моделью, описывающей различия между ними, должна быть простой.

В этом примере проявляется классический эффект переобучения, который мы обсуждали в гл. 1 книги на примере задачи аппроксимации полиномами. Если не принимать во внимание сложность модели (решающей функции), то возникает опасность получить метод, не способный к обобщению. Существуют различные приемы борьбы с переобучением. Во-первых, если интерпретировать функции $f_i(\vec{x})$ как новые признаки, то можно обратиться к методам уменьшения размерности (их мы рассмотрим в п. 2.5). Во-вторых, можно снять ограничение на то, что все векторы обучающей выборки должны разделяться, т. е. разрешить системе ошибаться на обучающих примерах. Как правило, при этом эмпирически вводится система штрафов как за подобные ошибки, так и за сложность решающей функции (например, за число параметров в ней или за число опорных векторов [137]).

Однако, как мы знаем из примера выявления экспоненциальной зависимости (см. п. 1.2.4), простое число параметров в модели не всегда является подходящим критерием ее сложности. Таким образом, несмотря на то что множество обобщенных решающих функций может совпадать с гильбертовым пространством, этот подход имеет определенные ограничения. Они могут быть несущественны для частных практических приложений, если система штрафов подобрана удачно, но попытка построить на основе подобных методов универсальную систему машинного обучения приведет к тому, что эта система будет не способна выучить многие понятия.

Поскольку в качестве примера построения решающих функций был приведен метод опорных векторов, то нельзя

не сказать про другую возможность расширения линейных методов. Если в методе опорных векторов поменять все векторы $\vec{x}_i$ на $\vec{\chi}_i = (\chi_{i,1}, \chi_{i,2}, \ldots, \chi_{i,n})$, где $\chi_{i,k} = f_k(\vec{x}_i)$, то решающая функция будет задаваться через скалярные произведения $\vec{\chi}_i \vec{\chi}_j^T$.

Пусть $F(\vec{x}) = (f_1(\vec{x}), \ldots, f_n(\vec{x}))$, тогда $\vec{\chi}_i \vec{\chi}_j^T = \langle F(\vec{x}_i), F(\vec{x}_j) \rangle$, где через $\langle\ \rangle$ обозначено скалярное произведение. Далее вводится функция $\mathrm{K}(\vec{x}_i, \vec{x}_j) = \langle F(\vec{x}_i), F(\vec{x}_j) \rangle$, называемая *ядром*. Таким образом, вместо выбора функций $f_i(\vec{x})$ достаточно задавать ядра $\mathrm{K}(\vec{x}_i, \vec{x}_j)$.

Использование ядер позволяет вместо признаков $F : R^N \to R^n$ с числовыми значениями использовать признаки $F : R^N \to L_2(R^N)$, значениями которых являются функции (пространство L_2 используется для того, чтобы можно было вычислить скалярное произведение двух функций $\langle F(\vec{x}_i), F(\vec{x}_j) \rangle$). Это позволяет легко добиваться линейной разделимости любых обучающих совокупностей при достаточно простых ядрах. Например, таким свойством обладает популярное гауссово ядро $\mathrm{K}(\vec{x}_i, \vec{x}_j) = e^{-|\vec{x}_i - \vec{x}_j|^2}$, которому соответствует функция-признак

$$[F(\vec{x})](\vec{y}) = c \exp\left[-0,5\,|\vec{y} - 2\vec{x}|^2\right]. \qquad (2.21)$$

Линейная разделимость любой (конечной) совокупности попарно различных векторов $\vec{x}_1, \ldots, \vec{x}_M$ является следствием линейной независимости функций $\exp\left[-0,5\,|\vec{y} - 2\vec{x}_i|^2\right], i = 1, \ldots, M$. Это можно интерпретировать так, что под каждый вектор обучающей выборки отводится собственная размерность в новом пространстве признаков (в результате число опорных векторов может сильно возрасти).

Сама линейная разделимость в случае гауссовых ядер не очень интересна (так как имеет тривиальную интерпретацию в рамках методов, основанных на функциях расстояния), но представляет интерес возможность одновременно использовать различные ядра.

К сожалению, хотя использование ядер имеет ряд преимуществ перед явным заданием признаков $f_i(\vec{x})$, оно не избавляет как от необходимости задавать эти ядра априори, так и от проблемы переобучения, которая приобретает особую остроту, так как использование ядер не позволяет использовать стандартные методы уменьшения размерности.

Именно поэтому здесь применяются системы штрафов за ошибки классификации на обучающих примерах и за число опорных векторов. Эти штрафы можно вводить строго на основе принципа МДО (см. п. 2.3.6).

Подробнее вопросы, связанные с теорией и практическим применением метода опорных векторов, рассмотрены в работах [141, 143].

2.3.3. Выбор эталонных образов

Методы классификации, основанные на функциях расстояния, требуют определения числа и положения эталонных образов, а также метрики пространства признаков. К решению именно этих задач сводится распознавание образов в рамках данного подхода, хотя вид метрики пространства признаков обычно считается заданным априори.

Рассмотрим сначала случай единственного эталона и известной функции расстояния. Пусть задана некоторая функция $s : \mathrm{X} \times \mathrm{X} \to R$, так что ее значения $s(\vec{x}, \vec{y})$ имеют смысл расстояния между $\vec{x}$ и $\vec{y}$. Если считать, что эталонный образ должен описывать класс безотносительно того, какие еще классы существуют, то в качестве него следует выбирать такой вектор $\vec{y}^*$, который бы минимизировал среднее расстояние до известных членов класса (был бы максимально на них похож):

$$\vec{y}^* = \arg\min_{\vec{y} \in \mathrm{X}} \left[\frac{1}{M} \sum_{i=1}^{M} s(\vec{x}_i, \vec{y}) \right], \qquad (2.22)$$

где M — число векторов обучающей выборки, принадлежащих только рассматриваемому классу.

Для произвольной функции расстояния пришлось бы исследовать все пространство признаков для нахождения эталонного образа. Однако предположение о дифференцируемости $s(\vec{x}, \vec{y})$ позволяет воспользоваться необходимым условием экстремума:

$$\sum_{i=1}^{M} \frac{\partial s(\vec{x}_i, \vec{y})}{\partial \vec{y}} = 0. \qquad (2.23)$$

Если полученная система уравнений не имеет простого решения, то можно использовать, например, метод градиент-

ного спуска (тоже требующий дифференцируемости целевой функции).

В простейшем случае функция $s(\vec{x},\vec{y}) = \|\vec{x}-\vec{y}\|^2$ соответствует квадрату евклидового расстояния, а положение эталонного образа вычисляется как

$$\vec{y}^* = \frac{1}{M}\sum_{i=1}^{M}\vec{x}_i. \qquad (2.24)$$

Чуть более сложный случай — это расстояние Махаланобиса $s(\vec{x},\vec{y}) = (\vec{x}-\vec{y})^T C^{-1}(\vec{x}-\vec{y})$. Для него положение эталонного образа также вычисляется как среднее векторов $\vec{x}_i$. Но эта функция расстояния обладает еще и набором параметров, которые следует оценить. Иными словами, здесь все же происходит выбор метрики пространства, хотя и из довольно простого семейства. Очевидно, на матрицу C должны накладываться определенные ограничения — в противном случае можно добиться сколь угодно малого среднего расстояния. Однако эти ограничения, как и саму матрицу, проще получить в рамках статистического подхода. Результатом будет

$$\vec{y}^* = \frac{1}{M}\sum_{i=1}^{M}\vec{x}_i;\ \ C = \frac{1}{M}\sum_{i=1}^{M}\left[(\vec{x}_i-\vec{y}^*)(\vec{x}_i-\vec{y}^*)^T\right]. \qquad (2.25)$$

Использование единственного эталона при сильных ограничениях на функцию расстояния позволяет разделять классы лишь в достаточно простых случаях. Для обобщения данного подхода обычно идут не по пути расширения семейства метрик, из которых производится выбор некоторой оптимальной метрики, а по пути применения нескольких эталонов для описания класса образов.

В случае нескольких эталонов возникает задача, очень похожая на задачу кластеризации. Действительно, каждый эталон можно рассматривать как центр подкласса. Тогда для заданного множества образов, соответствующих одному классу, нужно построить набор подклассов, каждый из которых описывается собственным эталоном. При этом неизвестно ни число подклассов, ни принадлежность векторов обучающей выборки. Эту задачу мы рассмотрим далее.

Но удивительно то, что в случае распознавания очень популярным оказался метод, число подклассов в котором равно числу входящих в родительский класс образов. Это метод *ближайшего соседа* (БС) и многочисленные его вариа-

ции. В простейшем случае он заключается в том, что каждый образ обучающей выборки трактуется в качестве эталона соответствующего ему класса и используется евклидово расстояние. При классификации новый образ будет относиться к тому классу, который содержит наиболее близкий к этому образу элемент.

Метод весьма успешно применяется на практике, и часто его единственным недостатком называют медленную скорость работы и требование больших объемов памяти при большом размере обучающей выборки [127]. Интересно задаться вопросом: почему в случае кластеризации каждому образу не выделяется собственный класс, а здесь для подклассов именно это и делается? Этот подход в своей начальной трактовке противоречит самому существу распознавания как построению общих понятий. Действительно, модель класса в этом случае состоит из простого перечня входящих в него элементов, т. е. совершенно не выделяются внутренние закономерности объектов, по которым произошло исходное их деление на классы. Почему же метод столь успешен?

Метод БС можно было бы рассматривать в рамках задачи классификации (обычно так и говорится: классификационное правило ближайшего соседа). Но поскольку принятие решения о принадлежности некоторого объекта классу основывается на обучающей выборке, то, очевидно, это метод распознавания (если судить по тем исходным данным, на которые он опирается). Просто стадия распознавания здесь вырождена (модель класса является моделью ad hoc). Однако обобщающие свойства все же имеются, благодаря предположению о евклидовости (мы не смотрим на точные совпадения, но ищем ближайшие образы), если оно адекватно. Можно сказать, что обобщение здесь проведено *до* построения моделей классов.

Сложно представить себе реальную ситуацию (мы не говорим о таком примере, как разделение рациональных и иррациональных чисел), в которой малые изменения векторов признаков соответствовали бы большим изменениям самих объектов. Точек, в которых происходит переход от одного класса к другому, как правило, мало (подпространство размерности на единицу меньше, чем само пространство признаков). Там, где это условие не выполняется, сам дискриминантный подход плохо применим (например, для изображений или текстов на каком-либо языке, представленных в виде векторов признаков).

Существуют различные расширения метода ближайшего соседа. Основное расширение связано с использованием k-БС правила, заключающегося в том, что для данного вектора определяется k ближайших векторов и выбирается тот класс, к которому принадлежит большее их число. Рассматриваются также методы, в которых значение k зависит от точки в пространстве признаков [128, с. 180].

Для всех этих методов характерна проблема трудоемкости вычислений. Существуют различные способы для ее уменьшения. Например, правило БС может применяться совместно с линейными решающими функциями [127]. При этом оно используется только в конфликтных областях, в которых линейная решающая функция не работает. Наиболее распространенным является привлечение методов кластеризации для объединения векторов обучающей выборки в группы и замены их на вектор средних (см., например, [128, с. 217]). Целью при этом является, правда, не построение более адекватной модели класса (хотя реально это и происходит), а уменьшение размерности данных.

Кроме малой скорости работы и эвристичности методам, основанным на правиле БС, приписывают и другие недостатки. Один из них заключается [128, с. 181] в том, что при их использовании не вводят для оценивания локальную меру расстояния. Этот и ряд других недостатков устраняются в статистическом подходе к распознаванию. Может показаться, что БС-правило имеет отдаленное отношение к статистическим методам, однако его можно интерпретировать как локальное непараметрическое оценивание плотности вероятности [128, с. 170]. Более того, правило БС и было введено из статистических соображений [144]. Этот подход привлекателен еще и тем, что в явном виде указывает ограничения, заложенные в метод (например, требование непрерывности плотности вероятности и ряд других — см. [144] или [128, с. 174]), а также может быть корректно обобщен.

2.3.4. Параметрические методы оценивания плотности вероятности

В рамках статистического подхода задача распознавания образов сводится к оцениванию плотности распределения вероятностей по конечному набору испытаний [120, с. 152;

128, с. 170]. Проблема оценивания плотностей вероятности возникает не только в распознавании образов, она относится [145] к одной из основных проблем статистического обучения. Этой проблеме посвящена обширная литература, разработано множество методов для ее решения. Мы лишь упомянем некоторые наиболее общие из этих методов, тесно связанные с задачей распознавания.

Методы оценивания плотности вероятности можно разделить на параметрические и непараметрические. Это деление несколько условно, поскольку часто непараметрическое оценивание тем или иным способом сводится к параметрическому.

Общий метод оценивания параметров основывается на правиле Байеса (или его упрощении — методе максимального правдоподобия), но примененном на этот раз к самим плотностям вероятности. В общем случае пришлось бы рассматривать плотность вероятности как случайную функцию и искать наиболее вероятную ее реализацию. В задаче распознавания, однако, очень часто пользуются предположением о том, что векторы обучающей выборки, принадлежащие одному классу, *статистически независимы и одинаково распределены* (iid, independently and identically distributed). Тогда им соответствует единственная плотность распределения вероятностей $p(\vec{x} \mid \vec{w})$ известного вида, но с неизвестными параметрами $\vec{w}$, которые требуется оценить. Естественно также считать, что плотности вероятности, описывающие разные классы, независимы, и оценивать их параметры отдельно.

Пусть $\vec{x}_1, \ldots, \vec{x}_M$ — векторы обучающей выборки, принадлежащие одному классу. Тогда, согласно теореме Байеса,

$$p(\vec{w} \mid \vec{x}_1, \ldots, \vec{x}_M) = \frac{p(\vec{w})p(\vec{x}_1, \ldots, \vec{x}_M \mid \vec{w})}{p(\vec{x}_1, \ldots, \vec{x}_M)} \qquad (2.26)$$

является апостериорной вероятностью для вектора $\vec{w}$. Статистическая независимость $\vec{x}_i$ влечет

$$p(\vec{x}_1, \ldots, \vec{x}_M \mid \vec{w}) = \prod_{i=1}^{M} p(\vec{x}_i \mid \vec{w}), \qquad (2.27)$$

а оптимальное значение вектора параметров будет

$$\vec{w}^* = \arg\max_{\vec{w}} \left[p(\vec{w}) \prod_{i=1}^{M} p(\vec{x}_i \mid \vec{w}) \right]. \qquad (2.28)$$

Следует отметить, что работать удобнее не с самой вероятностью, а с ее логарифмом (являющимся оценкой количества информации):

$$\vec{w}^* = \arg\min_{\vec{w}} \left[-\ln p(\vec{w}) - \sum_{i=1}^{M} \ln p(\vec{x}_i \mid \vec{w}) \right]. \qquad (2.29)$$

В качестве примера рассмотрим случай нормального распределения

$$p(\vec{x} \mid C, \vec{y}) = \frac{1}{(2\pi)^{N/2} |C|^{1/2}} \exp\left[-\frac{1}{2} (\vec{x} - \vec{y})^T C^{-1} (\vec{x} - \vec{y}) \right] \qquad (2.30)$$

и максимизацию правдоподобия (вместо апостериорной вероятности)

$$p(\vec{x}_1, \ldots, \vec{x}_M \mid C, \vec{y}) = \prod_{i=1}^{M} p(\vec{x}_i \mid C, \vec{y}) =$$

$$= \frac{1}{(2\pi)^{MN/2} |C|^{M/2}} \prod_{i=1}^{M} \exp\left[-\frac{1}{2} (\vec{x}_i - \vec{y})^T C^{-1} (\vec{x}_i - \vec{y}) \right]. \qquad (2.31)$$

Количество информации, которое нужно минимизировать, в этом случае будет:

$$L = -\ln p(\vec{x}_1, \ldots, \vec{x}_M \mid C, \vec{y}) = \frac{MN}{2} \ln(2\pi) + \frac{M}{2} \ln |C| +$$

$$+ \frac{1}{2} \sum_{i=1}^{M} \left[(\vec{x}_i - \vec{y})^T C^{-1} (\vec{x}_i - \vec{y}) \right]. \qquad (2.32)$$

Приравняв частные производные количества информации, взятые по параметрам плотности вероятности, можно получить систему линейных уравнений. Не так сложно убедиться, что вероятность максимальна (а значение L минимально) при значениях параметров, задаваемых формулами (2.25), которые приводились для расстояния Махаланобиса. Это также интуитивно очевидно исходя из того, что $\vec{y}$ — математическое ожидание, а C — ковариационная матрица нормального распределения.

Как отмечалось в п. 2.2.3, при использовании нормальных плотностей границы между классами описываются поверхностями второго порядка. К сожалению, не для любых клас-

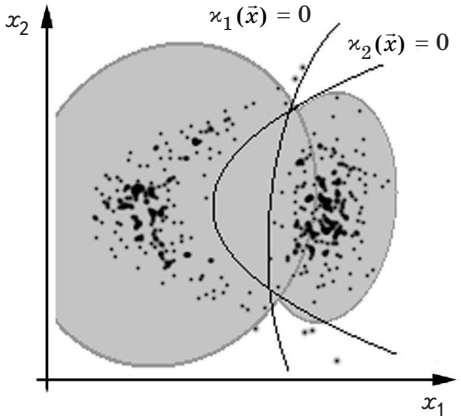

Рис. 2.10. Пример решения задачи распознавания для двух классов, разделимых кривой второго порядка $\varkappa_2(\vec{x}) = 0$. Использование метода максимального правдоподобия и описание распределения образов каждого из классов с помощью нормального закона приводит к построению разделяющей границы, которая также описывается кривой второго порядка $\varkappa_1(\vec{x}) = 0$. Однако это решение не является оптимальным

сов, разделимых такими сравнительно простыми поверхностями, использование моделей нормальных распределений приводит к построению правильных решающих функций (рис. 2.10), что показывает ограниченность данных моделей.

Отметим, что в рамках статистического подхода интеграл по пространству признаков от вероятности (2.30) должен быть равен единице. Это накладывает определенные ограничения на матрицу C, которые было бы сложно получить и обосновать в рамках подхода, основанного на функциях расстояния. Это хорошо иллюстрирует, чем именно привлекателен статистический подход.

Здесь мы продемонстрировали лишь общую схему вывода формул для параметров распределений. Аналогичным образом можно получить уравнения и в случае байесовского подхода, при котором привлекается плотность распределения априорных вероятностей $p(\vec{w})$ (см., например, [128, с. 73–82]). В работе [128, п. 2.11] также можно найти результаты для некоторых других видов плотностей.

К сожалению, в явном виде получить уравнения, по которым можно было бы вычислять параметры плотностей вероятности, удается для весьма немногих типов плотностей. Для приближенного (например, с помощью градиентного спуска) нахождения параметров используется так называемый метод *стохастической аппроксимации*, в котором вместо максимизации значения самой плотности вероятности $p(\vec{w} \mid \vec{x}_1, \ldots, \vec{x}_M)$ максимизируется регрессионная функция вида

$$\rho(\vec{w}, \vec{w}^*) = \int_X \xi(\vec{x}, \vec{w})\, p(\vec{x} \mid \vec{w}^*) d\vec{x}, \qquad (2.33)$$

где $\vec{w}^*$ — фиксированное, но неизвестное истинное значение вектора параметров, $\xi(\vec{x}, \vec{w})$ — некоторая функция, которая должна удовлетворять определенным условиям регулярности [128, с. 89]. Хотя вектор $\vec{w}^*$ неизвестен, значение $p(\vec{x} \mid \vec{w}^*)$ можно оценить в точках обучающей выборки. В простейшем случае эти значения принимаются равными единице в точках обучающей выборки, соответствующих данному классу, и равными нулю в остальных точках выборки. Тогда интеграл в выражении (2.33) оценивается через сумму

$$\rho'(\vec{w}) = \frac{1}{M} \sum_{i=1}^{M} \xi(\vec{x}_i, \vec{w}), \qquad (2.34)$$

максимум которой и будет достигаться в точке $\vec{w} = \vec{w}^*$, т. е. вместо $p(\vec{w} \mid \vec{x}_1, \ldots, \vec{x}_M)$ определяется максимальное значение $\rho'(\vec{w})$.

В качестве функции $\xi(\vec{x}, \vec{w})$ может, например, использоваться $\xi(\vec{x}, \vec{w}) = \ln p(\vec{x} \mid \vec{w})$. Тогда оценка функции регрессии $\rho'(\vec{w})$ будет совпадать с логарифмом правдоподобия $\ln p(\vec{w} \mid \vec{x}_1, \ldots, \vec{x}_M)$ [см. формулу (2.29)], т. е. стохастическая аппроксимация будет являться частным случаем байесовского оценивания. Некоторое расширение байесовских методов здесь видится лишь в том, что вместо просмотра всего пространства параметров (когда решение не удается получить в явном виде, как это было в случае с гауссовым распределением) применяется один из приближенных методов нахождения экстремума регрессионной функции $\rho'(\vec{w})$. При этом вид функции $\xi(\vec{x}, \vec{w})$ может быть подобран таким образом, чтобы эту процедуру было проще осуществлять.

Приведем еще один возможный вариант функции $\xi(\vec{x}, \vec{w})$, лучше поясняющий название метода стохастической аппроксимации. Пусть $\xi(\vec{x}, \vec{w}) = \left\| p(\vec{x}, \vec{w}) - p(\vec{x}, \vec{w}^*) \right\|^2$, а $\rho'(\vec{w})$ требуется минимизировать. Пусть $p_i = p(\vec{x}_i, \vec{w}^*)$ — оценки истинной плотности вероятности в точках обучающей выборки (как и ранее, эти значения равны единице для образов, принадлежащих данному классу, и нулю для образов, принадлежащих другим классам). Тогда задача сводится к нахождению функции $p(\vec{x}, \vec{w})$, которая проходит ближе всего (в среднеквадратичном смысле) к точкам $(\vec{x}_i, p_i)$, что является классической задачей аппроксимации.

В дальнейшем метод стохастической аппроксимации мы будем трактовать просто как способ приближенного решения задачи (2.29). Подробное изложение метода стохастической аппроксимации применительно к проблеме распознавания можно найти в работах [120, пп. 6.2–6.3; 128, пп. 2.12–2.26].

2.3.5. Непараметрические методы оценивания плотности вероятности

Зачастую возникает такая ситуация, что никаких предположений о виде плотности вероятности сделать нельзя. Тогда используют непараметрические методы оценивания. Однако и в этих случаях необходимо делать некоторые априорные допущения, такие, как, например, непрерывность или симметрия плотности распределения вероятностей [128, с. 172]. Один из широко распространенных подходов к непараметрическому оцениванию заключается в представлении неизвестной плотности в виде линейной комбинации плотностей известного (параметрического) вида — *смесей*. Мы рассмотрим *конечные смеси*, которые представляются в виде

$$p(\vec{x}) = \sum_{i=1}^{m} p(\vec{x} \mid \vec{w}_i) P(\vec{w}_i), \qquad (2.35)$$

где m — число компонентов смеси.

Чтобы подчеркнуть, что величины $P(\vec{w}_i)$ являются численными коэффициентами, мы будем использовать обозначение $P_i = P(\vec{w}_i)$. Поскольку они имеют смысл вероятностей, то для них должны выполняться ограничения $0 \le P_i \le 1$ и $P_1 + \ldots + P_m = 1$.

Таким образом, смесь представляет собой взвешенную сумму некоторого количества различных распределений. Обычно (но вовсе не обязательно) распределения принадлежат одному и тому же параметрическому семейству и различаются лишь значениями параметров.

Поскольку нас интересует оценивание плотностей распределения вероятностей, то как векторы параметров $\vec{w}_1, \ldots, \vec{w}_m$, так и коэффициенты $P_1, \ldots, P_m$, являются неизвестными. В связи с этим необходимо записать

$$p(\vec{x} \mid \vec{w}_1, \ldots, \vec{w}_m, P_1, \ldots, P_m) = \sum_{i=1}^{m} P_i\, p(\vec{x} \mid \vec{w}_i). \qquad (2.36)$$

В общем случае может быть неизвестно и число компонентов смеси m.

В распознавании образов смеси актуальны еще и по следующей причине. Каждый класс q_i имеет свою модель, выражающуюся через плотность вероятности $p(\vec{x} \mid a_i)$, а также вероятность $P(a_i)$ того, что произвольно взятый вектор будет принадлежать этому классу. Тогда верно равенство

$$p(\vec{x}) = \sum_{i=1}^{d} p(\vec{x} \mid a_i) P(a_i). \qquad (2.37)$$

Нетрудно сопоставить эту формулу с формулой (2.35). Таким образом, наличие нескольких классов, каждый из которых имеет собственную плотность вероятности, естественным образом порождает смесь. В отличие от оценивания параметров распределения для каждого класса в отдельности привлечение модели смеси для работы со всеми классами одновременно позволяет получать также значения P_i, которые в данном случае являются не чем иным, как априорными вероятностями классов, используемых в байесовском классификаторе.

Смеси полезны и как средство непараметрического оценивания плотностей вероятности отдельных классов. При этом работа со смесью может вестись абсолютно так же, как и с обычной параметрической плотностью. В частности, здесь оказывается полезным метод стохастической аппроксимации.

Один из способов использования смеси для оценивания плотностей заключается в разложении последних по базисным функциям. Если в уравнении (2.36) в качестве набора функций $\{p(\vec{x} \mid \vec{w}_i)\}_{i=1}^{m}$ использовать полную систему функций (задаваемую, как правило, априори), то с помощью смеси можно будет аппроксимировать произвольную (непрерывную) плотность вероятности. Это является замечательным свойством, когда априорные сведения о виде плотности вероятности отсутствуют. Выбор набора базисных функций, казалось бы, не накладывает никаких ограничений на то, какие плотности могут быть восстановлены, коль скоро либо этот набор является полным, либо пространство, натянутое на него, гарантированно содержит искомую плотность.

Однако хорошо прослеживается аналогия между этим подходом и методом обобщенных решающих функций. Как

и в том подходе, здесь также возникает проблема переобучения, связанная с отсутствием надлежащего критерия выбора числа компонентов в смеси. Одним из решений является привлечение теоретико-информационного критерия, который мы кратко рассмотрим далее.

Одной из наиболее популярных смесей является смесь нормальных плотностей, получающаяся подстановкой нормального распределения (2.30) в смесь (2.36) вместо плотностей $p(\vec{x} \mid \vec{w}_i)$ с различными ковариационными матрицами и векторами средних. Эта смесь имеет тесную связь с методами, базирующимися на функциях расстояния. Действительно, как мы убедились ранее, максимизация плотности вероятности в случае нормального распределения соответствует минимизации расстояния Махаланобиса. Значит, каждый компонент смеси задает положение некоторого эталонного образа с локально оцененной метрикой. Однако проблема выбора числа эталонных образов (или компонентов смеси) остается и в чистом статистическом подходе.

В частном случае для аппроксимации плотности распределения элементов одного класса можно жестко задать параметры смеси следующим образом. Число m компонентов смеси равно числу эталонных образов M. Ковариационные матрицы всех компонентов являются единичными матрицами. Вектор средних $\vec{y}_i$ i-го компонента смеси равен i-му образу обучающей выборки $\vec{y}_i = \vec{x}_i$, а коэффициенты смеси $P_i = 1/M$. Иными словами, в каждую точку обучающей выборки «помещается» нормальное распределение с единичной ковариационной матрицей. Если таким образом описать плотность вероятности каждого класса, то получим правило 1-БС. Но в отличие от этого правила в статистическом подходе можно отказаться от использования единичных ковариационных матриц, тогда появится возможность производить локальное оценивание метрики пространства признаков.

Из локального оценивания плотности вероятности можно также получить и правило k-БС. Основная идея здесь заключается в том, что для некоторой точки $\vec{x}$, в которой производится оценивание плотности, находится некоторая область (например, шар $B_r(\vec{x})$ радиуса r с центром в данной точке), содержащая помимо точки $\vec{x}$ также k элементов обучающей выборки. Отсюда можно определить локальную плотность распределения точек, а произведя соответствующую нормировку, получить и оценку плотности ве-

роятности. Строгое развитие этой простой идеи (впервые использованной в работе [144]) приводит как к получению формальных ограничений на вид плотностей вероятности, которые могут быть оценены таким способом (эти ограничения можно перенести на метод k-БС, в рамках которого их получить затруднительно), так и к построению более эффективных методов. В качестве примера можно назвать метод окон Парзена и другие методы сглаживания (развитые в работах [146–149]), использующие вместо выделения области с четкими границами сглаживание с некоторым окном или весовой функцией (также называемой ядром).

Таким образом, здесь мы дали лишь общие идеи некоторых методов оценивания плотностей вероятности и не приводили конкретных алгоритмов решения этой проблемы. Один такой алгоритм для смесей нормальных плотностей будет кратко описан в п. 2.4.4 (более подробно см., например, в работе [128]). Использование окон Парзена подробно обсуждается в работе [63]. Мы также опустили описание некоторых других подходов к распознаванию образов, например, на основе потенциальных функций. Их изложение можно найти в книгах, полностью посвященных проблемам распознавания (см., например, [120]).

2.3.6. Информационные критерии в распознавании

В качестве двух базовых подходов в распознавании образов выступают линейные дискриминантные функции и правило k-БС. Однако решающие правила, построенные на основе линейных или квадратичных дискриминантных функций, оказываются слишком простыми, чтобы адекватно описать различия между произвольными классами, а решающие правила, построенные на основе критерия k-БС, напротив, оказываются чрезмерно сложными и имеют недостаточно высокую обобщающую способность [150]. В связи с этим возникает необходимость вводить модели промежуточной сложности. В классических методах распознавания проблема выбора между решающими функциями разной степени сложности решается исходя из эвристических соображений. Принцип минимальной длины описания позволяет поставить решение этой проблемы на четкую теоретическую основу. Использование принципа МДО для этих целей начало становиться популярным сравнитель-

но недавно, поэтому работ на эту тему еще не так много. Рассмотрим кратко возможность уточнения на основе принципа МДО метода обобщенных решающих функций, метода опорных векторов и подхода на основе смеси нормальных распределений, хотя, конечно, этими методами возможность применения принципа МДО не ограничена.

Напомним, что в методе обобщенных решающих функций неизвестная решающая функция представляется в виде линейной комбинации базисных функций. Такое разложение имеет тенденцию содержать как можно больше членов, чтобы минимизировать ошибку на обучающей выборке, в результате чего качество классификации образов, не вошедших в обучающую выборку, может оказаться неудовлетворительным. Примером обращения к принципу МДО для избежания подобного эффекта переобучения может служить работа [150].

В этой работе базисные функции $f_i(\vec{x})$ [см. уравнение (2.20)], которые интерпретируются как новые признаки, строятся как произведения нормированных на промежутке [−1, 1] полиномов Лежандра, аргументами которых являются разные компоненты вектора $\vec{x}$. Это делается для того, чтобы на основе ортонормированной системы функций со скалярным аргументом получить систему ортонормированных функций с векторным аргументом. Для построения нелинейной дискриминантной функции сначала формируется большое пространство признаков, соответствующих всем произведениям полиномов Лежандра ограниченной степени (ограничение выбирается так, чтобы размерность нового пространства признаков была минимально больше размера обучающей выборки).

В этом пространстве рассматриваются различные подпространства, критерий выбора между которыми основан на принципе МДО ([150]):

$$MDL(\mathrm{T}) = \frac{M}{2}\log_2 \frac{\varepsilon^2(\mathrm{T})}{M} + \frac{n}{2}\log_2 M, \qquad (2.38)$$

где n — размерность рассматриваемого подпространства признаков T, а

$$\varepsilon^2(\mathrm{T}) = \sum_{i=1}^{M}\left[z_i - \sum_{j=1}^{n} w_j f_j(\vec{x}_i) \right]^2 \qquad (2.39)$$

z_i имеет тот же смысл, что и в п. 2.3.1. Первая часть суммы (2.38), очевидно, отвечает длине описания отклонений соответствующей решающей функции от идеальных значений z_i на векторах обучающей выборки в предположении о нормальном распределении ошибок. Заметим, однако, что такое вычисление количества информации не вполне корректно (см. п. 1.3.3) и может вызвать определенные проблемы, когда значение $\varepsilon^2(\mathrm{T})$ становится меньше единицы и начинает приближаться к нулю. В связи с этим формула (2.38) явно нуждается в некотором уточнении, однако обычно просто предполагается, что размерность подпространства T меньше, чем размер обучающей выборки M.

Второе слагаемое штрафует сложность пространства T, указывая длину его описания: n — это число признаков, которые требуется описать, а $\frac{1}{2}\log_2 M$ — количество бит информации, необходимых для описания одного признака. Проще понять, почему это слагаемое имеет именно такой вид, из следующих рассуждений. Размерность общего пространства признаков не превосходит M, поэтому при выборе подпространства T происходит выбор n чисел, соответствующих номерам признаков, каждое из которых не превосходит M. Такой выбор можно сделать C_M^n способами. При небольших значениях M или n эту величину можно вычислить непосредственно. Однако когда оба значения M и n заметно отличаются от единицы, вычисление числа сочетаний становится затруднительным и необходимо использовать асимптотические оценки. При $M >> 1$ и $1 << n << M/2$ используется оценка $\log_2 C_M^n \approx \frac{n}{2}\log_2 M$. Заметим, что при $n = 1$ данная оценка дает ошибку в два раза: $\log_2 C_M^1 = \log_2 M$.

В этой работе также применен эвристический алгоритм последовательного выбора признаков, позволяющий избежать перебора всех возможных подпространств, и сообщается об успешном применении разработанного метода.

Мы реализовали часть описанного метода (без процедуры выбора подпространств признаков) и провели ряд экспериментов, один из которых представлен на рис. 2.11. Результаты этих экспериментов подтверждают сообщающуюся в статье [150] информацию, что на основе информационного критерия может быть выбрана решающая функция адекватной сложности.

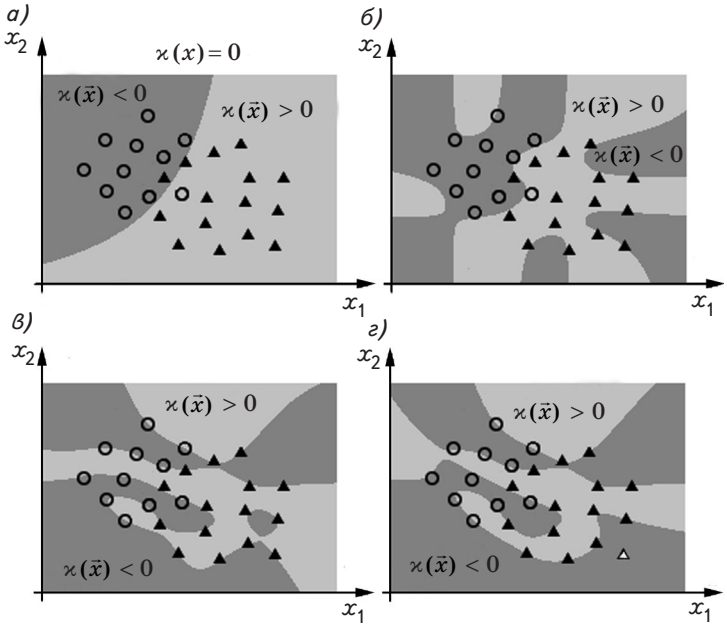

Рис. 2.11. Разделение двух классов (○ и ▲) посредством обобщенных решающих функций с разным числом признаков: *а* — четыре признака ($1, x_1, x_2, x_1x_2$), приводящие к хорошему обобщению при неверной классификации двух образов; *б* — 25 признаков, приводящих к неверной, но с меньшей ошибкой, классификации двух образов; *в* — 36 признаков, приводящих к верной классификации всех образов; *г* — 36 признаков, но один образ (△) исключен из выборки. Видно, что в этом случае образ «△» классифицируется неверно с помощью построенного по другим 25 векторам решающего правила, что свидетельствует об очень низкой обобщающей способности. Также видно, что исключение одного вектора сильно влияет на результирующие области, что говорит о плохой робастности метода при большом числе компонентов разложения. При корректном вычислении количества информации решение *а* оказывается более предпочтительным, чем *б* и *в*

Отметим, что использование полиномов Лежандра для построения новых признаков непринципиально, т. е. принцип МДО может применяться для того, чтобы избежать переобучения для других признаков. Однако формула для вычисления длины описания здесь выводится человеком на основе дополнительных предположений. Это позволяет получить более строгие критерии, чем чисто эвристические штрафы, но также требует привлечения эвристически вводимых упрощений.

В работе [151] представлен несколько более строгий подход к вычислению длин описания дискриминантных моделей в приложении к методу опорных векторов (со ссылками на более ранние работы на эту же тему), чем упомянутая выше работа [150], посвященная обобщенным решающим функциям. Чтобы не перегружать изложение техническими деталями, рассмотрим в несколько упрощенном виде основные идеи, заложенные в этом подходе.

Чтобы корректно вычислять длины описаний, необходимо не упускать из виду все компоненты описания, чтобы быть уверенным, что не произошло потери информации. Для этого представим, что существуют отправитель (источник) и получатель (приемник) сообщений. Отправитель посылает сжатое описание имеющегося набора данных. Если получатель сможет правильно восстановить любые исходные данные по полученному описанию, значит, потери информации не произошло и длина описания вычисляется корректно. При таком стиле рассуждений становится необходимо в явном виде сформулировать, какой априорной информацией располагает приемник.

В задаче распознавания считается [151], что получатель сообщения знает образы обучающей выборки $\vec{x}_1, \ldots, \vec{x}_M$, но не знает их принадлежности к классам $A_1, \ldots, A_M$. Отправитель и получатель также должны иметь возможность заранее «договориться» об алгоритме кодирования, который должен быть определен до получения обучающей выборки и должен быть одинаковым для любой выборки. В простейшем случае отправитель посылает сообщение, составленное из самих меток классов $A_1, \ldots, A_M$. Это модель ad hoc. Но даже для нее приемник сообщения должен знать, что получает именно список номеров классов, а не, скажем, программу для машины Тьюринга, которая их генерирует. Более содержательные модели должны устанавливать зависимость A_i от $\vec{x}_i$. Иными словами, отправитель передает получателю описание правила классификации, на основе которого получатель восстанавливает по известным ему значениям $\vec{x}_i$ значения A_i. Поскольку не всякое классификационное правило позволяет получить истинные значения A_i для всех $\vec{x}_i$, то посылаемое сообщение должно состоять из двух частей: описания классификационного правила и перечня векторов обучающей выборки, для которых классификация по этому правилу выполняется неверно. На основе этой информации получатель сможет правильно вос-

становить значения A_i, т. е. потери информации не происходит.

Рассмотрим оценку длин описания этих двух частей сообщения для метода опорных векторов. Будем считать, что есть только два класса, а разделяющая гиперплоскость проходит через начало координат. Первая часть сообщения должна содержать информацию об ориентации гиперплоскости, построенной методом опорных векторов.

Вопрос заключается в том, с какой точностью следует описывать ориентацию гиперплоскости. Чем шире полоса, тем с большей ошибкой может быть задана ориентация соответствующей гиперплоскости без опасности неверной классификации, а значит, тем меньше бит информации нужно потратить на передачу вектора ориентации. В частности, это обосновывает выбор ориентации гиперплоскости, дающей наиболее широкую полосу (на чем и базируется метод опорных векторов) с точки зрения МДО-принципа. Связь допустимой ошибки ориентации с шириной полосы хорошо видна на примере двух полос, представленных на рис. 2.8.

Если все образы обучающей выборки заключить в гиперсферу радиуса R, то величина s/R (s — ширина некоторой полосы) будет описывать допустимую ошибку каждого угла ориентации. Код для описания ориентации можно построить разными способами. Например, можно равномерно размещать точки на гиперсфере единичного радиуса и выбирать такую наименьшую плотность размещения точек, чтобы хотя бы одна из них оказалась вблизи (на меньшем расстоянии, чем ошибка, определяемая шириной полосы) истинной ориентации разделяющей гиперплоскости, а затем пересылать номер точки, задающей приблизительную ориентацию. Поскольку при разной плотности расположения точек их размещение на гиперсфере будет различаться, то нужно также передавать информацию о плотности точек (или об их количестве). Если рассматривать только такие расположения точек, которые будут по плотности больше в 2^k раз по сравнению с некоторым начальным распределением, то для передачи информации о распределении достаточно передавать его номер k. Все номера (натуральные числа) можно передавать, например, с помощью кодов с саморазграничением (см. п. 1.5.4). Можно показать, что число точек, расположенных на сфере, для надлежащего кодирования ориентации разделяющей плоскости должно

быть $n = c(R/s)^{N-1}$, где c — определенная константа ([151]). Для кодирования номера точки на сфере, задающей приблизительную ориентацию, и для кодирования варианта размещения с помощью кодов с саморазграничением нужно примерно $\log_2 n$ и $\log_2 \log_2 n$ бит соответственно. Приблизительно можно считать, что требуется около

$$L_1 = (N-1)\log_2 (R/s) \tag{2.40}$$

бит информации для описания ориентации. Это и составляет длину описания первой части сообщения.

Рассмотрим описание второй части сообщения. Нам нужно переслать номера всех образов обучающей выборки, которые классифицируются неправильно. Пусть таких образов n_{err}, каждый из них — это число от 1 до M. В сообщение сначала необходимо включить код для числа n_{err}, а затем коды для n_{err} номеров образов, что можно сделать, снова использовав коды с саморазграничением. Тогда суммарную длину этой части сообщения можно приближенно оценить как

$$\log_2 n_{err} + n_{err} \log_2 M. \tag{2.41}$$

Сложив длины описаний обеих частей, получим критерий для выбора ориентации разделяющей плоскости и ширины полосы. Варьируя ширину полосы, мы получаем менее точную ориентацию плоскости, в результате чего некоторые образы могут быть классифицированы неправильно. Минимизируя общую длину описания, мы также получаем и точность решения. Полученную оценку длины описания можно использовать на практике, хотя в приведенных рассуждениях присутствует ряд неточностей. Более подробный вывод и более точные формулы см. в работе [151].

Мы рассмотрели случай фиксированного пространства признаков. Однако обращение к принципу МДО было аргументировано именно возможностью с его помощью выбирать среди различных пространств признаков. Вместо пространства признаков в методе опорных векторов выступают ядро и опорные векторы. Поскольку при классификации используются только опорные векторы, то размерность пространства признаков (явно в методе не вводящееся) совпадает с числом этих векторов, поэтому в формулу (2.40) вместо N нужно подставить число опорных векторов n_{sv}. Чем некоторое ядро лучше описывает распределение векторов обучающей выборки, тем меньше значение n_{sv} и тем короче суммарная длина описания, что дает корректный

критерий для выбора между ядрами. В простейшем случае можно задать перечень используемых ядер и пронумеровать их. Тогда отправитель сообщения должен указать лишь код номера ядра в дополнение к прочей информации. В более сложном случае, когда ядро принадлежит некоторому параметрическому семейству (например, если ядро гауссово), помимо типа ядра также требуется передавать и значения его параметров.

Мы не будем показывать, как это делается для метода опорных векторов, а рассмотрим еще один пример применения принципа МДО, на этот раз к выбору количества компонентов в смеси нормальных распределений (имеющих очевидную связь с гауссовыми ядрами).

Рассмотрим смесь (2.36) в случае, когда ее компоненты распределены нормально (2.30). Пусть m — число компонентов смеси. Для каждого m можно определить параметры смеси $\mu_m = \{P_i, \vec{y}_i, C_i\}_{i=1}^m$ с помощью некоторого итеративного алгоритма (например, ожидание максимизации). Проблема, возникающая при использовании классических методов, заключается в том, чтобы среди этих решений выбрать лучшее. Для этого определим длину описания исходного набора данных $(\vec{x}_1, \ldots, \vec{x}_M)$ в рамках модели смеси. Если нам известны параметры смеси, то оценкой длины описания этого набора данных будет являться следующая величина:

$$L = -\sum_{i=1}^{M} \log_2 p(\vec{x}_i \mid \mu_m) = -\sum_{i=1}^{M} \log_2 \left(\sum_{j=1}^{m} P_j p(\vec{x}_i \mid \vec{y}_j, C_j) \right). \quad (2.42)$$

Однако набор параметров смеси μ_m неизвестен априори, поэтому его длину описания также необходимо учитывать. Смесь описывается $n_p = (m-1) + m\left(N + N(N+1)/2\right)$ параметрами: $m - 1$ параметров — это коэффициенты перед компонентами смеси P_i (поскольку их сумма равна 1, то есть возможность один коэффициент не описывать); $N + N(N + 1)/2$ параметров описывают вектор средних $\vec{y}_i$ и (симметричную) ковариационную матрицу C_i для каждого из m компонентов смеси (N — размерность пространства признаков). Как и в формуле (2.38), число бит информации на каждый признак (а параметры смеси, по сути, образуют пространство признаков) берется равным $\frac{1}{2}\log_2 M$. Тогда получаем критерий для выбора числа компонентов смеси (см., например, [152]):

$$MDL(m,\mu) = L + \frac{n_p}{2}\log_2 M. \qquad (2.43)$$

Таким образом, проблема выбора числа компонентов смеси решена. Аналогично можно получить решение в случае, когда смесь описывает плотность вероятности сразу нескольких классов [152, 153]. Тогда необходимо учитывать принадлежность образов обучающей выборки различным классам и описывать неверно классифицированные образы (как это можно сделать, пояснялось на предыдущем примере). Вообще проблема выбора числа компонентов смеси — это проблема обучения без учителя, и мы рассмотрим ее в п. 2.4.5 под несколько другим углом.

2.3.7. Принцип МДО и априорные ограничения методов распознавания

Многие существующие методы могут быть обобщены и улучшены в рамках информационного подхода, так что здесь имеется широкая сфера для приложения усилий ученых и инженеров. Однако во всех рассмотренных выше примерах принцип МДО применялся в качестве вспомогательного метода, надстраивающегося над разработанными в рамках других теорий подходами.

В гл. 1 было показано, что задачу индуктивного вывода можно рассматривать в рамках принципа МДО без привлечения дополнительных средств, но при этом решение оказывается не применимым на практике. В связи с этим естественно предположить, что привлечение различных методов связано не с корректностью предлагаемого ими решения, а с его вычислительной эффективностью. Интересно посмотреть, какие дополнительные элементы, содержащиеся в этих методах, позволяют им быть эффективными и могут ли они быть использованы без потери общности принципа МДО.

При постановке чистой задачи индуктивного вывода имеется набор данных $D = \left((\vec{x}_1, A_1), (\vec{x}_2, A_2), \ldots, (\vec{x}_M, A_M)\right)$, для которого нужно построить единую модель, выявляющую закономерности, присутствующие в этих данных. В качестве этих закономерностей будут выступать как зависимости A_i от $\vec{x}_i$, так и связи последующей пары $(\vec{x}_i, A_i)$ со всеми предыдущими (а также зависимости между отдельными битами в представлениях чисел). Эта единая модель будет находиться как

$$\mu_{MDL} = \arg\min_{\mu} \left[l(\mu) \mid U(\mu) = D \right]. \qquad (2.44)$$

Заметим, что под $\vec{x}_i$ и A_i будем понимать некоторые их представления в виде строк символов. Очевидно, выбор этого представления субъективен.

В задаче распознавания предполагается, что пары $(\vec{x}_i, A_i)$ между собой никак не связаны, поэтому следует искать модель другого вида. Можно предложить два варианта.

Вариант 1. Для непосредственного восстановления решающего правила следует искать

$$\varphi_{MDL} = \arg\min_{\varphi} \left[l(\varphi) \mid (\forall i) U(\varphi, \vec{x}_i) = A_i \right]. \qquad (2.45)$$

Именно этот вариант (хотя и для специально сконструированных решающих правил) использовался в примере, посвященном методу опорных векторов (см. п. 2.3.6). Векторы $\vec{x}_i$ считались априорно известными и не описывались. В дополнение решающее правило разбивалось на две части, чтобы при выборе из фиксированного множества решающих правил для любого из них обеспечить выполнение условия $(\forall i) U(\varphi, \vec{x}_i) = A_i$.

Вариант 2. В явном виде следует строить модели для каждого класса образов. Производится построение d отдельных моделей (d — число классов):

$$\psi_{j,MDL} = \arg\min_{\psi} \left[l(\psi) + \sum_{i=1}^{M_j} K(\beta_i^{(j)} \mid \psi) \,\middle|\, (\forall i) U(\psi, \beta_i^{(j)}) = \vec{x}_i^{(j)} \right], \qquad (2.46)$$

где $\psi_{j,\,MDL}$ — модель класса a_j; M_j — число образов обучающей выборки, принадлежащих этому классу; $\vec{x}_i^{(j)}$ — i-й образ, принадлежащий j-му классу; $\beta_i^{(j)}$ можно интерпретировать как описание внутриклассовых признаков вектора $\vec{x}_i^{(j)}$.

Формула (2.46) аналогична формуле (1.52), введенной в п. 1.6.2. Именно построение подобных моделей (хотя и в неявном виде) было использовано при выводе уравнений (2.38) и (2.43). Действительно, если считать, что имеются источник и приемник сообщений, то в сообщении передается следующая информация: $\left\{ \psi_{j,MDL}, \beta_1^{(j)}, \ldots, \beta_{M_j}^{(j)} \right\}_{j=1}^{d}$, на осно-

ве которой происходит восстановление исходного набора данных — векторов $\vec{x}_i^{(j)}$. Так, для уравнения (2.43) сумма $\sum_{i=1}^{M_j} K(\beta_i^{(j)} \mid \psi)$ соответствует минус логарифму правдоподобия, а величина $l(\psi)$ — длине описания параметров смеси.

При решении задачи классификации для нового вектора $\vec{x}$ при первом варианте решения проблемы достаточно вычислить значение $U(\varphi_{MDL}, \vec{x})$ [или в другой записи $\varphi_{MDL}(\vec{x})$]. Для второго варианта оказывается необходимым для каждого класса a_j определять строку $\beta^{(j)}$ минимальной длины, такую, что $\psi_{j,MDL}(\beta^{(j)}) = \vec{x}$, и выбирать тот класс, для которого оказывается минимальной длина описания $l(\psi_{j,MDL}) + K(\beta^{(j)} \mid \psi_{j,MDL})$. Поиск $\beta^{(j)}$ в общем случае может вызвать определенные проблемы. Отображение $\varphi_{MDL}(\vec{x})$ можно считать дескриптивной моделью, в то время как отображения $\psi_{j,\,MDL}$ — генеративными моделями для соответствующих классов. В задаче распознавания дескриптивные модели выглядят предпочтительнее (и, как мы увидим в следующей части, есть все основания думать, что именно такие модели используются в биологических системах при обработке сенсорной информации). Однако такие модели удобно строить лишь при распознавании с учителем, при котором можно определить, адекватно ли модель описывает имеющиеся данные. К сожалению, при распознавании без учителя их построение затруднительно.

При сравнении формулы (2.44) с формулами (2.45) и (2.46) хорошо видно, насколько упрощается задача распознавания благодаря предположению о том, что векторы статистически независимы и одинаково распределены. Вместо строки, состоящей из M элементов, рассматриваются M одноэлементных строк (или вместо упорядоченной совокупности элементов рассматривается неупорядоченная). Это предположение должно оказаться очень полезным: даже если оно в действительности не выполняется, его использование может позволить во многих случаях получить неплохое приближение к истинному решению существенно быстрее, чем получить точное решение.

Другим приемом является само разделение задачи распознавания и классификации. Действительно, при полностью корректном решении задачи классификации на основе данных из обучающей выборки пришлось бы для каждого нового образа решать задачу распознавания d раз, поскольку решающее правило имело бы вид

$$A_{M+1} = \arg\min_{A}\left[MDL\left((\vec{x}_1, A_1), (\vec{x}_2, A_2), \ldots, (\vec{x}_M, A_M), (\vec{x}, A)\right)\right], \quad (2.47)$$

где *MDL* (...) — минимальная длина описания некоторого набора данных.

Необходимость решения задач распознавания можно пояснить следующим образом. Классификационное правило строится на основе обучающей выборки. Как только мы классифицируем очередной образ, мы должны включать его в обучающую выборку, а значит, должны исправить решающее правило, что, вообще говоря, может повлиять на классификацию всех предыдущих образов. Поскольку новый образ можно отнести к любому классу с какой-то степенью достоверности, то следует рассмотреть каждую из гипотез. Однако интуитивно кажется, что такой пересмотр всех своих старых знаний на основе новой информации вряд ли должен быть часто необходимым. Поэтому получение новой порции информации должно приводить лишь к небольшой коррекции имеющихся представлений. Если такую коррекцию не осуществлять вообще, то будет иметь место полное разделение задач классификации и распознавания. В противном случае приходим к идее инкрементного обучения.

Следовательно, разделение задач классификации и распознавания является очень мощной эвристикой, позволяющей получать загрубленное решение, но за счет значительной экономии вычислительных ресурсов. Инкрементное обучение существенно уточняет это решение, позволяя системе доучиваться в процессе работы за счет непринципиального усложнения вычислений.

Эти две эвристики (предположение о статистической независимости и разделение задач), хотя и являются очень мощными, не позволяют получить приемлемое по времени работы решение на основе подхода МДО и универсального пространства моделей. В методах распознавания, очевидно, применяются некоторые дополнительные предположения, приводящие к эффективным алгоритмам.

Одним из центральных упрощений в дискриминантном подходе к распознаванию является предположение о непрерывности решающей функции. Рассмотрим следующий пример.

Пусть есть два класса, заданных перечнем образов: {(18; 7), (6; 3), (11; 17), (2; 14)} и {(13,334; 16,825), (21,12; 6,1),

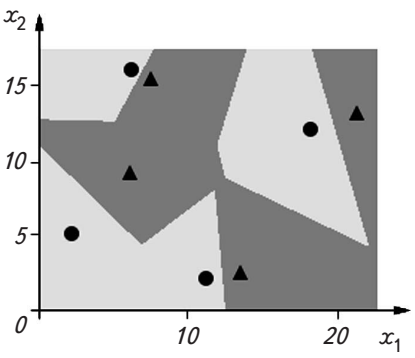

Рис. 2.12. Неудачная попытка разделения двух классов — целых (●) и дробных (▲) чисел при использовании правила 1-БС. Для разделения классов с помощью дискриминантных функций сложность последних должна соответствовать сложности самих исходных данных. Это может служить надежным индикатором того, что выбранное семейство решающих функций не позволяет описать понятие, заложенное в представленные классы (в данном случае понятие целых и дробных чисел), и от одного из априорных предположений следует отказаться

(5,85; 10,13), (7,31, 3,7)}. Каждый образ описывается двумя признаками, а в качестве пространства признаков выступает R^2, что вполне согласуется с дискриминантным подходом. Требуется построить решающее правило и определить класс, к которому относится образ (14; 20). На основе, например, правила ближайшего соседа результат будет выглядеть так, как показано на рис. 2.12. Но человек, не особо задумываясь и не имея дополнительных априорных сведений, построит совершенно другое решающее правило: векторы с целочисленными компонентами относятся к первому классу, а с дробными — ко второму. В рамках дискриминантного подхода такая решающая функция построена быть не может ни одним из существующих подходов, так как нарушается предположение о непрерывности. Конечно, такая задача легко решается в рамках синтаксического подхода, но по исходному пространству описаний это определить нельзя.

Поскольку очень широкий класс методов опирается на понятие непрерывности (или понятие количественной природы исходных данных), целесообразно закладывать это понятие в систему машинного обучения априори. Однако предположение о непрерывности должно выступать лишь в качестве альтернативы, так как оно ограничивает множество понятий, которые могут быть выявлены. Выбор между альтернативами может осуществляться следующим образом: если при имеющихся априорных предположениях не удается получить описания данных более короткого, чем при принятии модели ad hoc, то пространство моделей следует либо изменить (использовать отрицание некоторого ап-

риорного предположения), либо расширить (отказаться от одного из таких предположений).

Не следует абсолютизировать и возможности человека: для разделения классов из большинства ранее приведенных примеров (см. рис. 2.2–2.11) человеку потребовалось бы изобразить точки обучающей выборки на плоскости и воспользоваться возможностями своей зрительной системы. Используя численные данные, задачу распознавания ему решить было бы гораздо сложнее. В последнем примере (см. рис. 2.12), напротив, графическое представление информации мало о чем говорит, и задача решается при использовании символьных представлений. Таким образом, универсальность кроется в использовании различных классов представлений, а не в единой процедуре решения проблем.

Заметим также, что все описанные здесь методы распознавания имеют дополнительные ограничения помимо условия непрерывности решающей функции. Зачастую они связаны с некоторыми предположениями о линейности или нормальности. Вернемся к примеру с обобщенными решающими функциями. Если новые признаки являются полиномами от исходных признаков, то, как уже отмечалось, разделяющая граница, содержащая экспоненциальную зависимость, восстановлена быть не может. При этом полиномиальные решающие функции будут иметь тенденцию содержать как можно больше слагаемых, т. е. будут столь же сложны, как и модель ad hoc.

Увеличение сложности модели при поступлении новых данных может служить критерием недостаточности выбранного пространства моделей и необходимости обратиться к новому пространству. И только при исчерпании всех частных возможностей можно переходить к универсальному пространству моделей для поиска недостающего признака. Обнаружив этот признак один раз, в дальнейшем его можно использовать в качестве альтернативы при построении пространства моделей.

2.3.8. Пример практического приложения: распознавание целей

В качестве практической задачи, для которой выигрыш от привлечения теоретико-информационного подхода может оказаться значимым, можно привести задачу распознава-

ния малоразмерных целей. Малый объем исходной информации, искаженной шумом, при существенной неопределенности параметров цели (типа, формы, пространственной ориентации, дальности), характеризующих эту задачу, является настоящим вызовом методам распознавания. Другой крайний случай представляет собой задача интерпретации изображений с высокой детальностью, но с большой неопределенностью по содержанию (по присутствующим на изображении объектам, их взаимному расположению и т. д.).

Задача распознавания целей сопряжена с задачами их обнаружения, отделения от фона, вычисления признаков и ряда других. Здесь мы рассматриваем лишь задачу распознавания, хотя и при решении других задач теоретико-информационный подход может оказаться полезным (см., например, [154]).

Один из актуальных вопросов, связанных с распознаванием малоразмерных целей, заключается в следующем: какова предельная дальность, на которой возможно робастное (с определенным уровнем достоверности) распознавание? Мы полагаем, что теоретический предел дальности можно установить на основе информационного критерия. В этот критерий должны войти количество информации, содержащейся в изображении объекта (это количество информации зависит от площади, занимаемой объектом на изображении, шумов, устраняющих часть информации с изображения), а также неопределенность (также выражающаяся в битах), имеющаяся для параметров объекта. Если количество информации в изображении превышает неопределенность в параметрах объекта, то эта неопределенность может быть полностью устранена. Если же количества информации на изображении недостаточно, то распознавание, в принципе, невозможно со 100%-ной вероятностью.

Предельное удаление, на котором количество информации об объекте приближается к степени неопределенности его параметров, представляет наиболее важный случай. В зависимости от того, извлекается ли вся полезная информация из изображения или нет, предельная дальность распознавания цели может меняться. И при построении решающей функции очень важно корректно учитывать точность, с которой обучающие примеры классифицируются решающей функцией, и ее сложность, в противном случае возникает опасность переобучения (как правило, малоразмер-

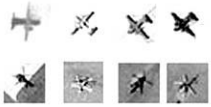

Рис. 2.13. Примеры приближенных изображений целей, предназначенных для распознавания

ные цели наблюдаются в сложной фоноцелевой обстановке). В этих условиях эффективность методов распознавания становится принципиальной.

Для примера используем аэрокосмические изображения целей, подобных представленным на рис. 2.13. Наша задача состоит не в решении проблемы распознавания малоразмерных целей (эта проблема заслуживает отдельного рассмотрения), а в сравнении методов распознавания на реальных данных, поэтому мы используем изображения относительно крупных целей, а малый объем исходных данных будет достигаться за счет использования простых признаков (проблема выбора признаков будет рассмотрена в п. 2.5): отношения площади объекта к квадрату его периметра и степени вытянутости объекта (отношение осей эллипса, наиболее соответствующего объекту).

На рис. 2.14, 2.15 представлены результаты применения решающих правил, построенных с помощью метода нормальных смесей при разном числе компонентов и с помощью метода обобщенных решающих функций при разном числе параметров соответственно. В табл. 2.1, 2.2 приведены значения длин описания [см. (2.38) и (2.43)], а также вероятности распознавания, полученные по изображениям, не вошедшим в обучающую выборку. Еще раз отметим, что критерии качества, в которых не учитывается сложность мо-

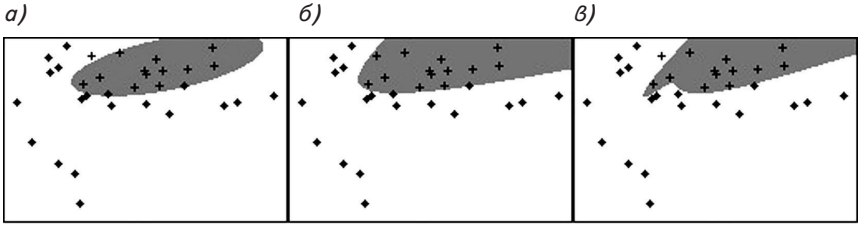

Рис. 2.14. Разделение пространства признаков на области решающими функциями, построенными методом нормальных смесей при разном числе компонентов: *а* — описание каждого из классов однокомпонентной смесью; *б* — описание более компактного класса смесью с одним компонентом, а более протяженного класса — смесью с двумя компонентами; *в* — описание обоих классов двухкомпонентными смесями (образы, принадлежащие классам, отмечены разными символами — «♦» и «+»)

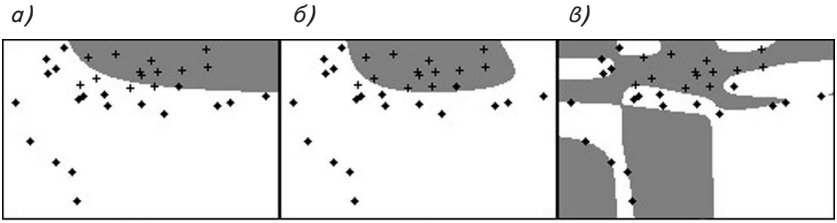

Рис. 2.15. Разделение пространства признаков на области решающими функциями, построенными методом обобщенных решающих функций при разном числе параметров: *а* — четыре параметра; *б* — девять параметров; *в* — 25 параметров. При увеличении числа параметров наблюдается тенденция к построению таких решающих правил, что образы оказываются вблизи границ областей, описывающих классы образов

дели, будут максимальны при максимальном числе параметров, что вовсе не соответствует вероятности правильного распознавания образов, не вошедших в обучающую выборку.

Выбор между решающими функциями различной сложности в классических методах распознавания затруднителен. Часто эти методы требуют определенной настройки, за счет чего теряется адаптивность методов. К примеру, нередко используемые при распознавании целей нейронные сети с обратным распространением ошибки требуют задания числа скрытых слоев. Если же настройка оказывается неточной, то может быть выбрана любая строка приведенной таблицы, что существенно сказывается на вероятности правильного распознавания. Можно сказать, что для данного примера привлечение принципа МДО позволяет повысить

Таблица 2.1

Результаты тестирования методов конечных смесей (m_1, m_2 — количества компонентов смесей, описывающих первый и второй классы образов; *MDL* — минимальная (для данного количества параметров) длина описания; p — вероятность правильного распознавания)

m_1	m_2	*MDL*, бит	p, %
1	1	387	88
1	2	380	95
2	2	386	89

Таблица 2.2

Результаты тестирования методов обобщенных решающих функций (m — число параметров обобщенной решающей функции)

m	*MDL*, бит	p, %
4	23	78
9	19	91
25	25	63

вероятность правильного распознавания с 88 до 95 %, что является весьма существенным.

Как видно из таблицы, информационные критерии качества решающей функции достаточно хорошо соответствуют вероятности распознавания на всем пространстве образов. Руководствуясь этими критериями, можно выбрать решающую функцию адекватной сложности, что, в конечном счете, приводит к увеличению дискриминантных свойств методов распознавания. Таким образом, теоретико-информационный подход к распознаванию действительно оказывается полезным при решении конкретных практических задач.

Следует отметить существенное различие длины описания, полученной методом конечных смесей и методом обобщенных решающих функций. Это вызвано тем, что в первом случае описывались сами классы образов, а во втором — лишь различия между ними. Если обратиться к аналогии с отправлением сообщения, то в методе конечных смесей в сообщении передавались закодированные значения векторов признаков $\vec{x}_i$ наряду со значениями классов A_i, а в методе обобщенных решающих функций передавались только A_i (полагалось, что значения $\vec{x}_i$ известны получателю априори). Оба подхода допустимы, однако сравнение значений длин описания для них напрямую невозможно, коль скоро осуществляется описание различных данных.

2.4. ГРУППИРОВАНИЕ ОБРАЗОВ В ПРОСТРАНСТВЕ ПРИЗНАКОВ

2.4.1. Проблема обучения без учителя

Задача распознавания образов заключается в построении решающих правил по обучающей выборке, состоящей из объектов, для которых известно, к каким классам они принадлежат. Однако информация о принадлежности тоже должна быть каким-то образом получена. В ряде задач эта информация может быть задана человеком (или другим «учителем»), но во многих, пожалуй, наиболее важных случаях, она отсутствует, и классы необходимо формировать автоматически. Задача группирования и заключается в том, чтобы разбить имеющееся множество образов на подмножества, каждое из которых определяет некоторый класс.

Группирование является частным случаем обучения без учителя. При обучении без учителя отсутствует обратная связь от среды, которая бы говорила, является ли выход системы машинного обучения корректным или нет, и система сама должна искать закономерности или структуру во входных данных [18, с. 38]. Найденные закономерности выступают в качестве результата работы подобной системы. Поскольку эти закономерности описывают входные данные, то общей целью обучения без учителя можно считать построение нового представления данных, которое отражало бы их структуру. Иными словами, основной целью здесь является построение модели источника данных, что в точности соответствует проблеме индуктивного вывода.

Обычно вид строящихся в процессе обучения представлений сильно ограничен самой постановкой более частной задачи обучения. Например, при группировании предполагается, что любой объект может быть отнесен к одному классу из конечного множества. Это частично определяет структуру строящегося представления: в новом описании объекта зарезервирована ячейка для хранения номера класса. При выборе признаков, являющемся другой задачей обучения без учителя, в основном рассматриваются только представления, состоящие из некоторого набора чисел (как правило, количество этих чисел одинаково для любого описываемого объекта). Более того, конкретные методы решения таких задач, как группирование или выбор признаков, еще сильнее сужают вид представления. При этом внимание акцентируется именно на той части представления, которая предопределена постановкой задачи. Так, в задаче группирования рассматриваемой частью представления является именно номер класса, к которому относится тот или иной образ, а внутриклассовые признаки обычно в явном виде не строятся.

Таким образом, практически в любой подход к обучению без учителя явно или неявно закладываются дополнительные априорные предположения о природе входных данных. Чтобы убедиться в этом, достаточно представить, что будет, если попытаться даже не решить, а просто поставить задачу группирования, например, для такой совокупности векторов: $\left\{(x_i, e^{x_i})\right\}_{i=1}^{M}$, где x_i — некоторые числа. Сама постановка задачи построения нового представления как задачи группирования здесь бессмысленна (рис. 2.16). Можно было

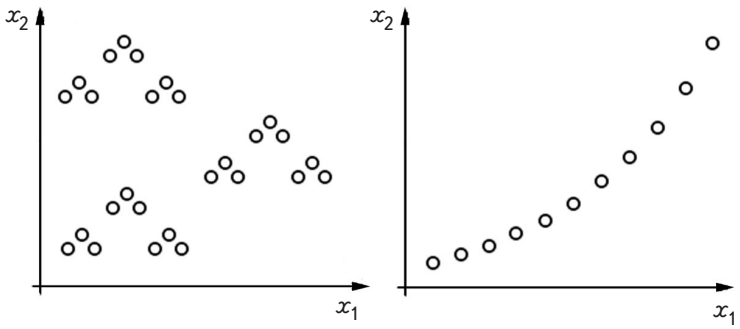

Рис. 2.16. Примеры наборов образов, для которых решение задачи группирования может оказаться проблематичным, хотя их размещение в пространстве признаков обладает отчетливой структурой, описываемой не в терминах классовой принадлежности

бы возразить, что это частный случай группирования, поскольку просто имеется один класс, но вряд ли многие существующие методы группирования смогут этот класс корректно выделить, хотя, если ставить задачу вписывания кривой, она будет успешно решена.

Поскольку все эти задачи (группирование, выбор признаков, вписывание кривой и некоторые другие) могут рассматриваться в качестве уточнений проблемы обучения без учителя (или индуктивного вывода) и поскольку методы решения одних задач плохо применимы для решения других задач, но являются эффективными с вычислительной точки зрения, интересно посмотреть, какие общие ограничения на строящиеся описания заложены в каждой из них. Подход, описанный в гл. 1 книги, универсален и может быть применен для решения любой из этих задач, но на практике он бесполезен. Возможно, анализ вводимых ограничений даст подсказку, как избежать проблемы комбинаторного взрыва.

С другой стороны, в рамках подхода МДО становится понятно, что различные задачи обучения без учителя могут быть унифицированы, и оказывается, что существующие методы обучения без учителя представляют собой лишь ограниченное подпространство пространства всех возможных алгоритмов обучения, возникающих в обобщенном подходе [18, с. 14].

Одна из основных характеристик, по которым различаются методы обучения без учителя, заключается в степени локальности получаемых с помощью них представлений [18,

с. 39]. Смысл этой характеристики наиболее очевиден на примере нейронных сетей. Пусть у некоторой нейронной сети есть входной и выходной слои, активность нейронов которых соответствует входным и выходным данным соответственно. Если при любых входных данных активируется лишь один нейрон выходного слоя, то такая сеть полностью воплощает локальное представление. Если же при произвольных входных данных активными являются многие нейроны выходного слоя, то построенное представление данных является распределенным. Понятие локальности мы попытаемся сформулировать и в рамках некоторых других подходов. Пока заметим, что локальные представления являются решением задачи группирования (один нейрон выходного слоя соответствует одному классу), а распределенные представления являются результатом решения задачи выделения признаков (активность нейрона определяет значение некоторого признака для данного образа).

2.4.2. Задача группирования

Рассмотрим задачу группирования в рамках дискриминантного подхода. Как и раньше, пространством признаков является $X = R^N$. В качестве исходных данных в этой задаче выступает (неупорядоченное) множество векторов $\{\vec{x}_1, \vec{x}_2, \ldots, \vec{x}_M\}$. Требуется разделить это множество на непересекающиеся подмножества (задающие классы эквивалентности) или определить значения $1 \le A_i \le d$ для каждого образа, где $d \le M$ — число подмножеств. В зависимости от постановки задачи число классов d либо может быть задано априорно, либо его также необходимо определить в процессе обучения.

Если в качестве гипотезы выдвигается некоторый случайный вариант группирования, то для него может быть решена задача распознавания образов. Причем если в данном разбиении нет идентичных образов, отнесенных к различным классам, то можно построить решающее правило, с помощью которого можно было бы правильно классифицировать все образы обучающей выборки. Таким образом, классы могут быть сформированы произвольным образом и использованы в дальнейшем для распознавания и классификации. Новые объекты будут относиться к этим произвольно выбранным классам. Задача группирования ста-

новится бессмысленной, поскольку допускает любое решение (отсутствие «правильного» решения типично для обучения без учителя), если не использовать дополнительные ограничения, позволяющие выбирать лучший способ разделения объектов на классы. Обычно рассматривается два варианта введения таких ограничений.

В первом случае явно вводится некоторая целевая функция, которая может быть вычислена для каждой гипотезы группирования, что позволяет выбрать лучшую из них. Эта целевая функция может быть сконструирована исходя из знания предметной области, тогда она будет содержать, частично или полностью, информацию о том, какие классы требуется получить в процессе группирования. Больший интерес представляет случай, при котором такая информация не закладывается и целевая функция является достаточно универсальной. Очевидно, что такой целевой функцией может служить длина описания исходных данных в рамках той или иной гипотезы, сложенная с длиной описания (сложностью) самой гипотезы. Часто другие используемые целевые функции можно представить как частный случай критерия МДО.

Однако вычисление целевой функции для каждой гипотезы группирования имеет следующий недостаток. Даже если число классов считается известным, то существует M^d вариантов группирования (если считать, что класс может быть пустым, а варианты группирования, переходящие друг в друга простой переиндексацией классов, являются различными). Неудивительно, что перебор всех возможных вариантов группирования и выбор лучшего из них по какому-либо критерию называется «решением возрастающей сложности» [128, с. 241]. В связи с этим необходимо использовать более простые, но приближенные решения. Проблема заключается в том, чтобы избавиться от экспоненциально возрастающей сложности, получив решение приемлемого качества. Один из популярных методов решения этой проблемы мы рассмотрим в п. 2.4.4 при обсуждении моделей смесей.

При другом подходе к проблеме выбора между различными гипотезами группирования глобальная целевая функция не вводится. Вместо этого полагается, что объекты, попадающие в один класс, должны быть как можно более похожи друг на друга. Поскольку в дискриминантном подходе степень сходства интерпретируется как расстояние

между образами в пространстве признаков, то группирование в этом случае называется кластеризацией, хотя этот термин может применяться и для прочих методов распознавания без учителя в рамках дискриминантного подхода.

Функция расстояния определяется лишь для пар векторов, тогда как аргументом целевой функции выступает их произвольное множество. Это вносит существенные ограничения на возможности методов кластеризации, но и делает их чрезвычайно эффективными с вычислительной точки зрения. Именно с них мы и начнем рассмотрение.

2.4.3. Кластеризация на основе функций расстояния

Интуитивно кажется, что при группировании в один класс должны попадать объекты, «похожие» друг на друга. Степенью сходства двух объектов удобно считать расстояние между соответствующими образами в пространстве признаков. Тогда задача группирования сводится к получению классов с минимальным средним расстоянием между образами. Подобные классы образов принято называть кластерами. При этом, как правило, используется евклидово расстояние, или расстояние Махаланобиса, но могут применяться и другие (в том числе и неметрические) критерии сходства (например, [120, с. 102–104]):

$$s(\vec{x},\vec{y}) = \frac{\vec{x}^T\vec{y}}{\|\vec{x}\|\cdot\|\vec{y}\|}, \quad \text{или} \quad s(\vec{x},\vec{y}) = \frac{\vec{x}^T\vec{y}}{\vec{x}^T\vec{x} + \vec{y}^T\vec{y} - \vec{x}^T\vec{y}}.$$

Выбор меры сходства основывается на знании природы входных данных. Например, если объектами являются прямые линии, а векторы признаков — это векторы, описывающие их направление, то использовать в качестве меры сходства угол между ними будет предпочтительнее, чем евклидово расстояние. Если же рассматриваются отрезки прямых линий, то необходимо также учитывать и различие их длин, что приведет к специфической для данных объектов мере сходства. Выбор меры сходства обычно задается человеком, а не осуществляется автоматически. Использование конкретной меры мало влияет на суть алгоритмов кластеризации, поэтому их можно описывать в предположении, что расстояние является евклидовым. Евклидово расстояние применяется в тех случаях, когда отсутствует какая-либо априорная информация о природе объектов.

Рассмотрим кратко несколько алгоритмов кластеризации. Один из них основан на вычислении k внутригрупповых средних, или кратко, *алгоритм k средних* [155]. Этот алгоритм требует задания числа кластеров, исторически обозначаемых через k. Мы, однако, как и выше, будем использовать переменную d для обозначения числа классов и $A = \{a_1, \ldots, a_d\}$ — для обозначения множества классов. Алгоритм состоит из следующих шагов:

1) каждому из d кластеров произвольным образом назначаются их центры (или эталонные образы) $\vec{y}_i$; в качестве этих центров обычно выступают первые d образов обучающей выборки $\vec{y}_i = \vec{x}_i$, $i = 1, \ldots, d$;

2) каждый образ выборки относится к тому классу, расстояние до центра которого минимально: $A_i = \arg\min_{a_j \in A} \left(s(\vec{x}_i, \vec{y}_j) \right)$, $i = 1, \ldots, M$;

3) центры кластеров пересчитываются исходя из того, какие образы к каждому из них были отнесены: $\vec{y}_j = \frac{1}{M_j} \sum_{(\forall i) A_i = a_j} \vec{x}_i$, где M_j — число образов, попавших в класс a_j;

4) шаги 2) и 3) повторяются, пока не будет достигнута сходимость, т. е. пока классы не перестанут изменяться.

Существуют различные модификации алгоритма k средних, например, в работе [156] предложен алгоритм, в котором евклидово расстояние было взвешенным по каждой координате. Аналогично можно использовать и расстояние Махаланобиса, вычисляемое для классов на каждой итерации.

Хотя алгоритм k средних основывается на функции расстояния, нетрудно заметить, что он минимизирует глобальную меру — суммарное среднеквадратичное отклонение образов от центров своих классов.

Одним из ограничений алгоритма k средних является необходимость задания числа классов. Чтобы решить эту проблему, можно выполнить алгоритм для различного числа классов и выбрать среди решений некоторое лучшее. При выборе по минимальной сумме расстояний от образов до центров классов предпочтение будет отдаваться решениям с большим числом классов, поэтому требуется применять более сложные критерии.

Но существуют алгоритмы, не требующие задания числа классов. Простейший такой алгоритм требует задания порога s^* на размер кластера и состоит из следующих шагов:

1) сформировать один кластер ($d = 1$) из первого образа обучающей выборки и положить $\vec{y}_1 = \vec{x}_1$;

2) выбрать следующий, еще не рассмотренный, вектор $\vec{x}_i$ и определить минимальное расстояние $s' = \min\limits_{j=1,\ \ldots,\ d} s(\vec{x}_i, \vec{y}_j)$; если это расстояние меньше порога $s' < s^*$, то отнести образ к соответствующему классу, в противном случае — увеличить число классов d на единицу и сформировать новый класс с центром $\vec{y}_d = \vec{x}_i$;

3) повторить шаг 2) последовательно для всех образов обучающей выборки.

Этот алгоритм требует лишь однократного отнесения каждого образа к некоторому классу и является достаточно эффективным для вычисления, но при этом его результат, помимо всего прочего, в большой степени зависит от порядка предъявления образов.

Хотя для работы данного алгоритма не требуется знание числа классов, ему необходим размер кластеров s^*, причем этот размер является одинаковым для всех классов. Эти два параметра (число классов и размер кластеров) несут примерно одинаковое количество (хотя и немного разной) априорной информации. Поэтому нельзя сказать, что этот подход предпочтительнее предыдущего. Если он применяется на практике и порог s^* неизвестен, то алгоритм желательно выполнять для различных значений порога и использовать некоторый критерий для выбора лучшего решения.

Приведенный алгоритм представляет некоторый интерес как простейший пример инкрементного обучения. Действительно, его работу можно интерпретировать следующим образом: если новый встреченный объект не похож ни на один уже известный класс объектов, то это объект нового, ранее не встречавшегося класса. Если же объект может быть надежно классифицирован, то он пополняет информацию о выбранном классе: на его основе уточняется модель класса (в данном случае смещается вектор средних).

Существуют алгоритмы, использующие другие стратегии построения кластеров. Например, в алгоритме *максиминного* (максимально-минимального) *расстояния* сначала выделяются наиболее удаленные (наиболее различные) образы, служащие прототипами классов. Таким образом, помимо максимизации сходства объектов, относящихся к одним и тем же классам, максимизируется и различие объектов, принадлежащих к разным классам.

Алгоритм ISODATA (Iterative Self-Organizing Data Analysis Techniques), развитый в работах [157–159], основывается на алгоритме k средних, но включает набор оказавшихся полезными на практике эвристик и параметры по их настройке. Одним из задаваемых априори параметров является желаемое число кластеров K, которое выступает в качестве рекомендации: в результате работы алгоритма может быть построено как меньшее, так и большее число кластеров. Подробно алгоритм описан в работах [120, с. 112–120; 134, с. 40–45]). Приведем основные эвристики.

1. Ликвидируются кластеры, в состав которых входит меньшее, чем заданное, число элементов.

2. Для каждого текущего кластера определяется направление максимальной вытянутости. Если размер некоторого кластера в этом направлении больше заданного порога, то кластер расщепляется на два. Расщепление происходит только в том случае, если кластеров мало (меньше $K/2$) или размер данного кластера больше, чем средний размер кластеров.

3. Попарно сливаются кластеры, расстояния между центрами которых меньше заданного порога. Слияние происходит, если число кластеров больше $2K$.

Хотя эти эвристики поиска опираются на функцию расстояния, их можно применять для произвольных алгоритмов группирования. Но для этого потребуется введение более обоснованных критериев, по которым можно было бы осуществлять удаление и создание кластеров, а также их слияние и расщепление.

Перечисленные методы кластеризации преимущественно являются итерационными. Это позволяет не перебирать все возможные варианты группирования, но делает эти методы чувствительными к начальной гипотезе о положении центров кластеров либо к порядку предъявления векторов обучающей выборки. Использующиеся в алгоритме ISODATA эвристики помогают не только подбирать более подходящее число классов, но и находить более приемлемое решение (рис. 2.17), несколько ослабляя (но не убирая полностью) зависимость от начальной гипотезы.

Другой проблемой алгоритмов кластеризации является то, что задание априори меры расстояния сильно ограничивает их возможности. Например, при использовании евклидова расстояния строящиеся классы всегда будут линейно разделимыми, а понятия, соответствующие этим классам,

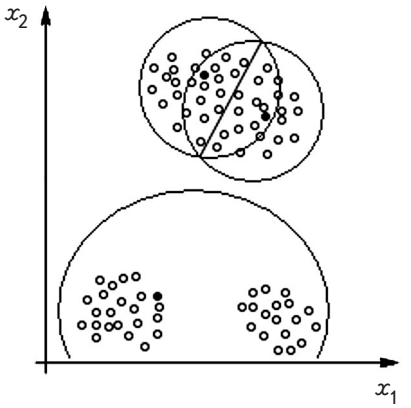

Рис. 2.17. Пример негативного влияния на результат кластеризации неудачного выбора начальных центров (●) в методе k средних. Слияние двух верхних кластеров и расщепление нижнего кластера, осуществляющиеся алгоритмом ISODATA при удачно настроенных параметрах, позволяют получить верное решение

т. е. те понятия, которые может обнаружить система машинного обучения, достаточно простыми.

Одним из фундаментальных ограничений методов, основанных на мерах сходства объектов, является то, что их решения локальны в том смысле, что рассматривают лишь отношения между парами образов, в то время как образы могут находиться в более сложных отношениях. Это ограничение относится ко всем подобным методам, независимо от конкретных алгоритмов группирования. Для пояснения рассмотрим пример, представленный на рис. 2.18. Векторы образуют две окружности. Методами кластеризации (если, конечно, не использовать специально подобранные метрики) нужного результата не получить. Даже если отказать совокупностям образов, формирующих окружности, называться классами, этот случай будет оставаться примером задачи группирования. В то же время задачи группирования подобного типа встречаются весьма часто, например, в компьютерном зрении (об этом будет говориться далее). Они могут

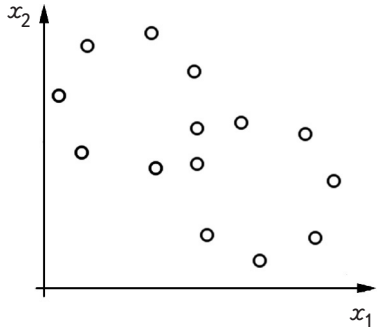

Рис. 2.18. Пример набора образов, для корректного группирования которого требуется построение глобальной модели для всего гипотетического класса: в результате группирования образы должны сформировать две окружности. Использование только функций расстояния, вычисляющихся для пары образов, не дает желаемого результата

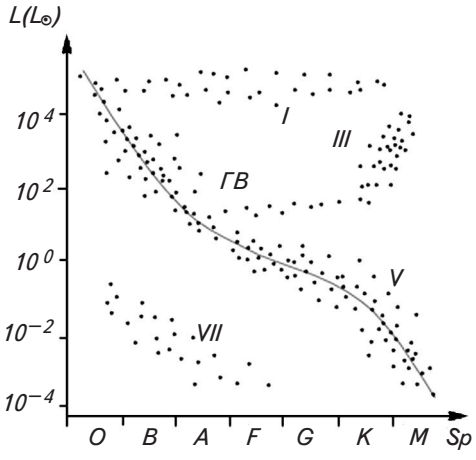

Рис. 2.19. Диаграмма Герцшпрунга—Рассела (Г—Р), устанавливающая зависимость между светимостью звезд и их спектральным классом. Диаграмма Г—Р — это плоскость, на которую точками нанесены звезды в зависимости от их светимости (или абсолютной звездной величины) и спектрального класса (или температуры). Оказывается, что в данных координатах звезды располагаются не хаотично, а образуют группы (классы светимости). Эти группы имеют глубокий физический смысл: они соответствуют различным стадиям жизни звезд:
I — сверхгиганты; *III* — красные гиганты; *ГВ* — горизонтальная ветвь; *V* — главная последовательность; *VII* — белые карлики; $L_\odot$ — светимость звезды в светимостях Солнца; *Sp* — спектральный класс

встречаться и в классическом распознавании образов, если какие-то объекты характеризуются не конкретными значениями своих признаков, а функциональной связью между ними. К сожалению, в таких случаях проблема либо перекладывается на выбор признаков, либо разрабатываются узкоспециализированные методы группирования.

В качестве реального примера, в котором некоторый класс характеризуется функциональной зависимостью между параметрами, можно привести диаграмму Герцшпрунга—Рассела (рис. 2.19). Звезды класса светимости V (главная последовательность, на которой, в частности, сейчас находится Солнце) описываются нелинейной зависимостью между светимостью и спектральным классом. При этом формы разных классов светимости могут сильно различаться между собой. Решение проблемы группирования для подобных случаев может оказаться полезным для автоматического выявления эмпирических закономерностей в физических данных, что необходимо для автоматизации научных исследований. С другой стороны, этот пример показывает, что алгоритмы кластеризации, которые строят лишь выпуклые классы, сильно сужают возможности систем машинного обучения по выявлению адекватных понятий, а значит, эти ограничения необходимо преодолевать.

Во всех перечисленных алгоритмах кластеризации используются некоторые параметры, либо напрямую, либо косвенно задающие число кластеров. Даже в алгоритмах наподобие ISODATA эта информация частично содержится в настроечных параметрах эвристик поиска. Если параметры априорно неизвестны, то алгоритмы приходится исполнять несколько раз для различных значений параметров. Но при этом возникает необходимость выбора между различными решениями, т. е. возникает необходимость введения некоторого глобального критерия качества. К методам, в которых в явном виде производится максимизация такого критерия, мы сейчас и обратимся.

2.4.4. Использование смесей в задаче группирования

Одним из универсальных критериев качества некоторой гипотезы (а каждый вариант решения задачи группирования представляет собой гипотезу) является ее апостериорная вероятность. Пусть $h \in \mathrm{H}$ — гипотеза, соответствующая некоторому способу разделения векторов обучающей выборки на группы; H — множество всех возможных вариантов группирования. Тогда, согласно правилу Байеса (1.4), верно

$$P(h \mid D) = \frac{P(h)P(D \mid h)}{P(D)}, \qquad (2.48)$$

где $D = \left(\vec{x}_1, \vec{x}_2, \ldots, \vec{x}_M\right)$ — данные наблюдений.

Поскольку векторы выборки считаются независимыми, то лучшая гипотеза соответствует максимизации

$$h_{\max} = \arg\max_{h \in \mathrm{H}} \left[P(h) \prod_{i=1}^{M} p(\vec{x}_i \mid h) \right]$$

или

$$h_{\max} = \arg\min_{h \in \mathrm{H}} \left[-\log_2 P(h) + L(D \mid h) \right],$$

где $L(D \mid h) = -\sum_{i=1}^{M} \log_2 p(\vec{x}_i \mid h)$ — минус логарифм правдоподобия. Возникает вопрос, как для данной гипотезы группирования h вычислить плотность вероятности $p(\vec{x} \mid h)$. Проблему оценивания сложности гипотезы $-\log_2 P(h)$ мы пока

отложим и рассмотрим решение на основе правила максимального правдоподобия (МП):

$$h_{ML} = \arg\min_{h \in \mathrm{H}} \left[-\sum_{i=1}^{M} \log_2 p(\vec{x}_i \mid h) \right]. \qquad (2.49)$$

Поскольку в рамках гипотезы h каждый вектор отнесен к некоторому классу, то можно описать каждый класс собственной плотностью вероятности $p_i(\vec{x})$, $i = 1, \ldots, d$, где d — общее число классов (в рамках гипотезы h). Пусть вероятность того, что произвольно взятый вектор принадлежит i-му классу, равна P_i. Тогда

$$p(\vec{x}_i \mid h) = \sum_{j=1}^{d} P_j p_j(\vec{x}_i). \qquad (2.50)$$

Каждую плотность $p_i(\vec{x})$ в отдельности можно оценить методами, о которых мы говорили в пп. 2.3.4 и 2.3.5, после чего оценить вероятности P_i. Подставив найденные плотности вероятностей (2.50) в уравнение (2.49), получим решение проблемы группирования. Как уже было сказано в п. 2.4.2, этот подход является подходом возрастающей сложности, так как размер пространства гипотез H растет очень быстро, что неприемлемо на практике.

Стандартное решение этой проблемы заключается в следующем. Пусть все плотности $p_i(\vec{x})$ принадлежат некоторому параметрическому семейству плотностей распределения вероятностей. Тогда каждая плотность $p_i(\vec{x})$ полностью определяется вектором параметров $\vec{w}_i$, так что $p_i(\vec{x}) = p(\vec{x} \mid \vec{w}_i)$. Уравнение (2.50) примет вид

$$p(\vec{x} \mid h) = \sum_{i=1}^{d} P_i p(\vec{x} \mid \vec{w}_i). \qquad (2.51)$$

Правая часть этого уравнения представляет собой смесь [см. также уравнение (2.36)]. Пусть $\vec{w} = (\vec{w}_1, \ldots, \vec{w}_d, P_1, \ldots, P_d)$ — вектор параметров, описывающий эту смесь. Этот вектор можно оценить на основе предполагаемой гипотезы группирования h. Однако его можно оценивать и непосредственно. Тогда приходим к формулировке проблемы группирования как оценивания параметров смеси:

$$p(\vec{x} \mid \vec{w}) = \sum_{i=1}^{d} P_i p(\vec{x} \mid \vec{w}_i). \qquad (2.52)$$

В этом уравнении гипотеза h заменена вектором параметров $\vec{w}$. Заметим, что связь между ними неоднозначна.

Пространство гипотез H было дискретно. Теперь же пространство поиска стало непрерывным: вектор параметров $\vec{w}$ может плавно изменяться. Однако для решений с различным числом кластеров размерность этого вектора должна быть различной, т. е. число компонентов смеси не определено. Проблему выбора числа классов мы рассмотрим в следующем параграфе, а пока будем считать, что число компонентов смеси d известно.

Чаще всего считается, что модели с одинаковым числом параметров обладают одинаковой сложностью. Хотя это предположение не совсем верно, оно может служить неплохим приближением и позволяет применять метод МП. Наиболее популярными и давно применяемыми (см., например, [160, 161]) являются использование смесей нормальных плотностей и максимизация правдоподобия с помощью алгоритма *ожидания-максимизации* (ОМ; expectation maximization, EM).

Алгоритм ОМ широко применяется для одновременного нахождения параметров модели и недостающих данных. *Задачу с недостающими данными* можно сформулировать следующим образом. Пусть $p(D, D' \mid \vec{w})$ — некоторая стохастическая модель, причем D — известные данные; D' — недостающие данные; $\vec{w}$ — параметры модели, которые необходимо определить. Если бы данные D' были также известны, то задача свелась к поиску таких значений параметров $\vec{w}$, которые бы максимизировали правдоподобие $p(D, D' \mid \vec{w})$ или, в общем случае, апостериорную вероятность $p(\vec{w} \mid D, D')$. Поскольку эти данные неизвестны, то их также необходимо определить. Алгоритм ОМ решает эту задачу, используя некоторое исходное предположение о значении вектора параметров и итерационно выполняя два шага [162]:

1) шаг ожидания — руководствуясь стохастической моделью при текущем значении параметров, оценить недостающие данные;

2) шаг максимизации — используя исходные данные и текущую оценку недостающих данных, получить новую оценку параметров модели, максимизирующих правдоподобие всей совокупности данных.

Поясним работу алгоритма ОМ на задаче группирования.

Рассмотрим смесь нормальных плотностей для простого одномерного случая

$$p(x \mid \vec{w}) = \sum_{i=1}^{d} P_i p(x \mid y_i, \sigma_i) = \sum_{i=1}^{d} \frac{P_i}{\sqrt{2\pi}\sigma_i} \exp\left[-\frac{(x - y_i)^2}{2\sigma_i^2} \right]. \quad (2.53)$$

Пусть

$$P_{ij} = \frac{P_j p(x_i \mid y_j, \sigma_j)}{\sum_{l=1}^{d} P_l p(x_i \mid y_l, \sigma_l)}, \; i = 1, \ldots, M \quad (2.54)$$

является вероятностью того, что i-й образ обучающей выборки принадлежит j-му классу. Эти вероятности являются недостающими данными в задаче группирования, и их можно вычислить, зная параметры смеси. Если же вероятности P_{ij} известны, то несложно найти параметры смеси, максимизирующие правдоподобие:

$$P_j = \frac{1}{M} \sum_{i=1}^{M} P_{ij}; \quad (2.55)$$

$$y_j = \frac{1}{MP_j} \sum_{i=1}^{M} P_{ij} x_i; \quad (2.56)$$

$$\sigma_j^2 = \frac{1}{MP_j} \sum_{i=1}^{M} P_{ij} \left(x_i - y_j \right)^2. \quad (2.57)$$

Шаг ожидания алгоритма ОМ состоит в вычислении вероятностей (2.54), а шаг максимизации — в вычислении параметров смеси (2.55)–(2.57). Итеративное выполнение этих шагов позволяет добиться максимизации правдоподобия. Обычно алгоритм останавливается, когда правдоподобие (или его логарифм) начинает мало изменяться на шаге максимизации или выполнено большое число итераций. Не представляет трудности обобщить алгоритм на многомерный случай; одномерный случай приведен лишь для большей наглядности.

Алгоритм ОМ требует задания начальной гипотезы. Если первым выполняется шаг ожидания, то требуется задать какие-либо значения параметров смеси. Если же сначала

выполняется шаг максимизации, то необходимы значения вероятностей P_{ij}. Эти значения задаются случайным образом. В зависимости от начальной гипотезы алгоритм может сойтись к различным решениям, поэтому иногда он выполняется несколько раз и из полученных решений выбирается лучшее.

Нетрудно заметить, что действие алгоритма ОМ для случая смеси нормальных плотностей очень похоже на работу метода k средних. Действительно, выбираются некоторые начальные положения и размеры кластеров, а затем они итерационно подправляются исходя из того, какой вектор попал в какой кластер. Здесь, однако, векторы не жестко относятся к единственному классу, а принадлежат всем классам с некоторой вероятностью. Также оценивается размер кластера (а в многомерном случае ковариационная матрица, т. е. расстояние Махаланобиса). Этот алгоритм является более строгим со статистической точки зрения, но его суть та же, что и у метода k средних.

Однако привлечение нормальных плотностей в качестве компонентов смеси не может быть обосновано в рамках наглядной геометрической интерпретации, как обосновывается использование евклидова расстояния в методах, основанных на функциях расстояний. Здесь модели смеси представляют очень широкие возможности в плане выбора привлекаемого семейства плотностей. Зададимся вопросом: а что будет, если вместо смеси нормальных плотностей использовать разложение по произвольным базисным функциям? Например, можно использовать систему многочленов, ортонормированных в некоторой области пространства признаков, содержащей обучающую выборку.

Гауссова функция быстро спадает при удалении точки от среднего значения. В связи с этим произвольный вектор, как правило, имеет высокое значение вероятности P_{ij} лишь для одного компонента смеси. В случае с многочленами это будет уже неверно, и каждый вектор будет принадлежать с высокой вероятностью сразу нескольким «классам». Более того, эти «классы» не будут соответствовать некоторым локализованным областям в пространстве признаков, т. е. интуитивное понятие класса в данном случае нарушается. С точки зрения представлений данных эти два случая соответствуют локальным и распределенным представлениям, и между ними есть целый спектр представлений с разной степенью локальности. Полностью распределенные

представления будут рассмотрены в разделе, посвященном задаче выбора признаков, а сейчас вернемся к проблеме группирования.

Использование в качестве модели каждого класса одной моды, представляющей собой нормальную плотность вероятности (или другой плотности простого параметрического семейства), вносит сильное ограничение. В частности, при использовании нормальных плотностей строящиеся классы всегда разделимы кривыми второго порядка. Как и в случае распознавания образов, можно попытаться каждый класс описывать смесью. Таким образом, каждый компонент $p_j(\vec{x})$ смеси (2.50) сам задается в виде смеси

$$p_j(\vec{x}) = \sum_{l=1}^{m_j} \tilde{P}_{j,l}\, p(\vec{x} \mid \vec{w}_{j,l}), \qquad (2.58)$$

где $\tilde{P}_{j,l}$ — коэффициент при l-м компоненте j-й смеси, а $\vec{w}_{j,l}$ — параметры этого компонента. Но такая «двойная» смесь легко превращается в простую смесь

$$p(\vec{x} \mid \vec{w}) = \sum_{j=1}^{d} \sum_{l=1}^{m_j} P_j \tilde{P}_{j,l}\, p(\vec{x} \mid \vec{w}_{j,l}), \qquad (2.59)$$

состоящую из $m_1 + \ldots + m_l$ компонентов. Отсюда несложно заметить, что при фиксированном наборе отдельных мод $p(\vec{x} \mid \vec{w}_{j,l})$ способ их отнесения к тем или иным классам не будет сказываться на правдоподобии данных. При этом общее число параметров будет одинаковым для различных решений с одинаковым числом классов и одинаковым числом мод. Таким образом, из критерия максимального правдоподобия совершенно непонятно, чем выделение отдельного класса под каждую моду распределения (2.59) хуже построения небольшого числа классов сложной формы, в совокупности включающих те же моды.

При решении задачи распознавания привлечение смесей позволяло описывать классы неэллиптической формы; здесь же это не приводит к успеху. Причину этого можно понять из анализа рис. 2.20, из которого видно, что сами моды необходимо группировать, т. е. объединять в классы на основе общности их свойств. Признаками, описывающими моды, являются векторы их параметров. Таким образом, для корректного решения проблемы группирования и получения в результате классов со сложной формой необ-

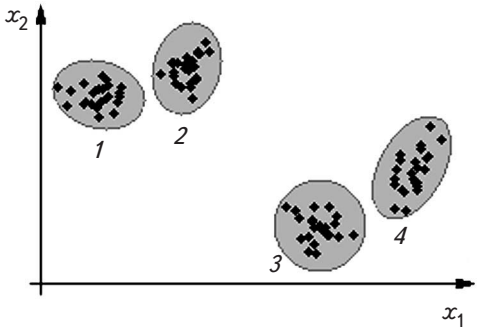

Рис. 2.20. Набор точек, распределение которых описывается смесью, состоящей из четырех компонентов. При попытке построить два кластера, каждый из которых описывался бы смесью, состоящей из двух компонентов, решение, объединяющее компоненты *1* и *2* и компоненты *3* и *4*, и решение, объединяющее компоненты *1* и *3* и компоненты *2* и *4*, будут обладать одинаковым минус логарифмом правдоподобия, равным 2940 бит. К такому же результату приведет использование единой смеси, состоящей из четырех компонентов

ходимо учитывать взаимосвязи между параметрами, описывающими компоненты смеси. Этот вопрос редко рассматривается, хотя имеет непосредственную связь с проблемой выбора числа кластеров при группировании (или, в общем случае, с проблемой оценивания сложности параметрической модели). К рассмотрению проблемы выбора числа кластеров мы сейчас и перейдем.

2.4.5. Критерии выбора числа кластеров

Проблема выбора числа кластеров в задаче группирования очень важна, поскольку предположение о том, что число кластеров известно, является очень сильным ограничением на класс решаемых задач. В то же время во многих методах эта проблема решается чисто эвристически и зачастую не вполне корректно, так что методы оказываются склонными либо формировать чрезмерно большое, либо слишком малое число кластеров по сравнению с оптимальным их количеством. Мы уже касались этой проблемы и способа ее решения для задачи распознавания на основе МДО-принципа (см. п. 2.3.6); теперь же рассмотрим ее несколько подробнее применительно к задаче группирования.

Приведем несколько критериев, которые применяются в методах, базирующихся на стохастическом подходе к проблеме группирования. Все эти методы используют следующие два шага [2, с. 158]:

1) сначала получается набор решений задачи группирования с фиксированным числом кластеров d; при каждом числе d находится (например, с помощью алгоритма ОМ) решение, которое максимизирует правдоподобие;

2) полученные решения сравниваются на основе некоторого критерия и из них выбирается лучшее.

Заметим, что такой подход имеет определенные ограничения, которые могут частично сниматься в некоторых эвристических подходах. Например, в алгоритме ISODATA число классов динамически изменялось во время поиска параметров самих классов, что позволяло в ряде случаев выходить из локальных минимумов (см. рис. 2.17).

Первый шаг был рассмотрен выше. Теперь будем считать, что параметры смеси оценены для некоторого набора значений числа классов. Поскольку для каждой оценки вектор параметров является фиксированным и известным, в записи будем его опускать. Тогда пусть $p_i^{(d)}(\vec{x})$ — i-й компонент смеси для d классов, а $P_i^{(d)}$ — коэффициент, указывающий вес, с которым i-й компонент входит в смесь (или безусловная вероятность того, что произвольный объект будет принадлежать i-му классу); $n_p^{(d)}$ — число параметров смеси для решения с d классами.

Как и ранее, обозначим

$$P_{ij}^{(d)} = \frac{P_j^{(d)} p_j^{(d)}(\vec{x}_i)}{\sum_{l=1}^{d} P_l^{(d)} p_l^{(d)}(\vec{x}_i)} \qquad (2.60)$$

вероятность того, что i-й образ обучающей выборки принадлежит j-му классу в рамках решения для d классов.

Минус логарифм правдоподобия данных в рамках данной модели смеси будет

$$L^{(d)} = -\sum_{i=1}^{M} \log_2 \left[\sum_{l=1}^{d} P_l^{(d)} p_l^{(d)}(\vec{x}_i) \right]. \qquad (2.61)$$

Общепринятыми являются следующие критерии.

1. *Коэффициент распределения* (Partition Coefficient, PC, [163]):

$$PC(d) = \frac{1}{M} \sum_{i=1}^{M} \sum_{j=1}^{d} \left(P_{ij}^{(d)} \right)^2. \qquad (2.62)$$

Он представляет собой один из эмпирически введенных критериев.

2. *Информационный критерий*, предложенный Акайке (An Information Criterion, AIC, [164]):

$$AIC(d) = L^{(d)} + n_p^{(d)}. \qquad (2.63)$$

Это один из первых критериев, в котором правдоподобие данных штрафуется в зависимости от сложности модели (числа параметров в ней). Однако этот критерий имеет тенденцию к переобучению.

3. *Информационный критерий Байеса* (Bayes Information Criterion, BIC, [165]):

$$BIC(d) = L^{(d)} + \frac{n_p^{(d)}}{2} \log_2 M. \qquad (2.64)$$

В п. 2.3.6 было показано, что этот критерий может быть получен в качестве частного случая в рамках МДО-подхода. Более того, одновременно с работой [165] и независимо от нее он был выведен в работе Риссанена [11], в которой вводится термин «минимальная длина описания». В связи с этим критерий этот часто отождествляется с принципом МДО вообще и противопоставляется принципу минимальной длины сообщения — МДС (см., например, [2]), в рамках которого получается несколько другое решение при привлечении других упрощений (см., например, [9]).

4. *Критерии, основанные на принципе МДО.*

В простейшем случае эти критерии совпадают с информационным критерием Байеса. Под простейшим случаем подразумевается предположение независимости параметров смеси. При привлечении более сложных схем кодирования (т. е. более богатых языков представления) такие зависимости могут учитываться, что приводит к более сложным выражениям для длин описания моделей. Это дает более адекватную оценку сложности модели, чем простой учет числа параметров, но не позволяет формировать классы неэллиптической формы.

Этим критерии выбора числа компонентов не исчерпываются (см., например, [2, с. 158–162, 166, 167]). Видно, что привлекаемые критерии могут достаточно сильно различаться, поэтому некоторые из них должны быть не вполне корректными. Как с теоретической, так и с экспериментальной точки зрения, наилучшими являются критерии, которые согласуются с принципом МДО.

Тем не менее информационные критерии, применяемые на практике, являются во многих отношениях приближенными. Например, логарифм правдоподобия используется в качестве оценки длины описания данных в рамках некоторой модели не совсем корректно: нетрудно предложить модель, в которой смесь состоит из M компонентов, при этом каждая плотность вероятности является дельта-функцией $p_i^{(d)}(\vec{x}) = \delta(\vec{x} - \vec{x}_i)$, $P_i^{(d)} = 1/M$, $i = 1, \ldots, M$. В этом случае минус логарифм правдоподобия равен минус бесконечности, т. е. эта модель ad hoc оказывается «идеальным» решением. Обычно эту проблему обходят, а не решают, просто не допуская вырожденных или близких к ним решений (в случае смеси нормальных плотностей избегают ковариационных матриц с определителем, равным нулю). Очевидно, проблема состоит в том, что действительные числа считаются заданными с бесконечной точностью, в то время как на описание вещественных параметров моделей (в частности, на вектор средних) выделяется конечное число бит. Описание сложности модели также является приближенным, поскольку не определяется для каждого параметра, сколько бит необходимо выделить на его описание. Если в случае смеси дельта-функций на описание параметров (т. е. смещений дельта-функций) было бы выделено конечное число бит, то эти смещения не совпали бы с векторами обучающей выборки, а значит, $p_j^{(d)}(\vec{x}_j)$ оказались нулевыми и минус логарифм правдоподобия стал бы вместо минус бесконечности равен плюс бесконечности.

Схема более строгого подхода была описана при рассмотрении выбора параметров в методе опорных векторов на основе принципа МДО. В этой схеме в явном виде строятся представления данных, поэтому оценка количества информации является корректной. Хотя из-за того, что используемые представления не являются алгоритмически полными, полученная оценка количества информации не совпадает с алгоритмической сложностью и в этом смысле также является приближенной.

Ограниченность привлекаемых на практике представлений (например, в методах, опирающихся на смеси нормальных плотностей) проиллюстрирована на рис. 2.21 и в табл. 2.3 (см. также рис. 2.19). Видно, что при использовании моделей классов, которые не могут захватить подлежащую структуру данных, происходит выделение избыточного числа кластеров: группирование при $d = 5$ дает правильные

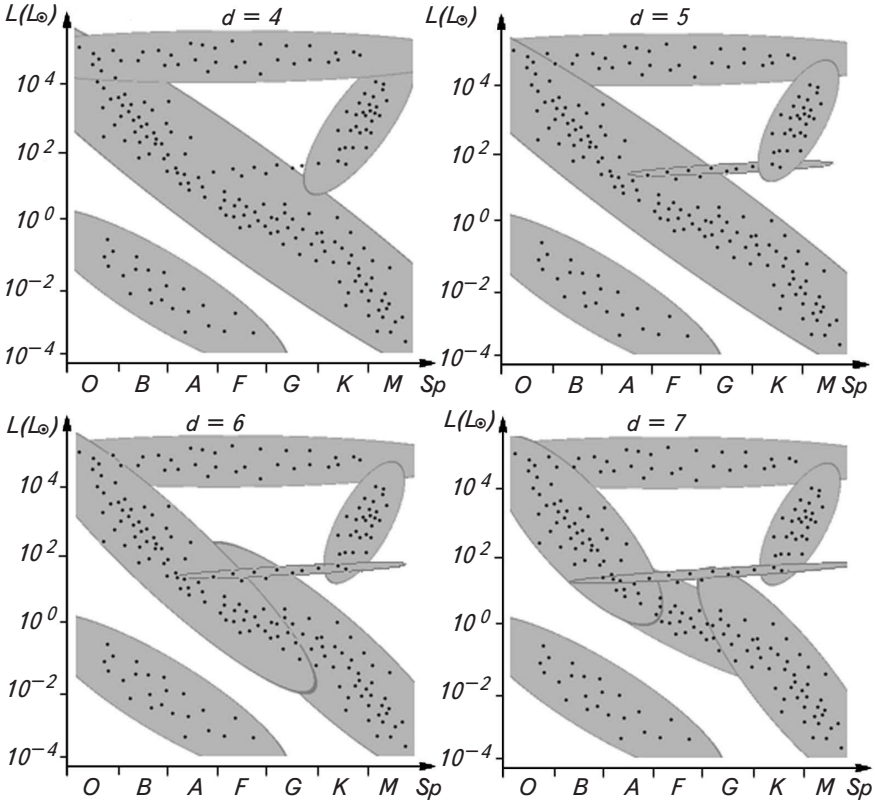

Рис. 2.21. Результаты группирования набора точек, представленных на рис. 2.19, для различного числа классов d (группирование выполнено с помощью алгоритма ожидания — максимизации, примененного к смеси нормальных плотностей)

кластеры, однако распределение звезд главной последовательности недостаточно хорошо описывается нормальным законом, в связи с чем меньшей длиной описания в рамках данного представления обладает решение при $d = 6$. При использовании нормальных плотностей для описания вероятностных свойств классов прослеживается тенденция разбивать классы сложной формы на подклассы. С другой стороны, именно ограниченность методов позволяет им быть вычислительно эффективными.

Основным подходом к получению кластеров неэллиптической формы является использование приема, подобного обобщенным решающим функциям, т. е. группирование осуществляется в новом пространстве признаков, получен-

Таблица 2.3

В таблице приведены значения минус логарифма правдоподобия, сложности модели, входящей в критерий *BIC*, и суммарное значение этого критерия [см. формулу (2.64)] для моделей, представленных на рис. 2.21. Для сравнения приведены соответствующие значения слишком простой ($d = 1$) и слишком сложной ($d = 10$) модели

Число классов d	$L^{(d)}$, бит	Сложность модели	$BIC(d)$
1	3691,8	23,1	3714,9
4	3431,6	92,4	3524,0
5	3407,4	115,5	3522,9
6	**3377,0**	**138,6**	**3515,6**
7	3355,8	161,7	3517,5
10	3334,6	231,0	3565,6

ном из исходного пространства добавлением признаков, являющихся функциями уже имеющихся. Однако число признаков, которые могут быть добавлены, является большим, поэтому среди них нужно выбрать в некотором смысле оптимальные признаки, т. е. решить задачу выбора признаков. Но прежде, чем перейти к этой задаче, сделаем несколько дополнительных замечаний о проблеме группирования.

2.4.6. Основные упрощения в постановке задачи группирования

Рассмотренные информационные критерии не относятся всецело к распознаванию образов. Они могут применяться при выборе любой параметрической стохастической модели. Однако задача группирования отличается от произвольной задачи построения модели тем, что в ней делаются некоторые дополнительные предположения об исходных данных и о пространстве моделей.

Первое предположение касается исходных данных. Так же как и в случае обучения с учителем, нам надо было бы решить полную задачу построения модели для данных $D = (\vec{x}_1, \vec{x}_2, \ldots, \vec{x}_M)$: $\mu_{MDL} = \arg\min_{\mu} \left[l(\mu) \,|\, U(\mu) = D \right]$. Однако множество векторов обучающей выборки считается неупорядоченным. Это позволяет искать модель, порождающую ис-

ходные данные D с точностью до перестановки векторов, т. е. ослабляет требования к модели и она может быть проще.

Сделаем небольшое отступление о предположении неупорядоченности. У кого-то из читателей может возникнуть вопрос, почему это предположение рассматривается как некоторое дополнительное ограничение, ведь оно абсолютно естественно. Однако, как известно, наиболее очевидные утверждения часто бывают и наиболее сложно доказуемыми. Представим себе существо, живущее в некотором виртуальном мире. Ему в качестве сенсорного входа поступает непрерывный поток данных. На основе чего оно может считать, что какие-то блоки этих данных можно свободно менять местами так, что это не скажется на результате индуктивного вывода?

Относительная неизменность объектов (и отделимость объектов друг от друга) — это свойство нашего мира, и в качестве априорной информации это заложено в нас, что позволяет оптимальнее решать задачи распознавания. Однако это предположение выполняется лишь приближенно и может нарушаться, и тогда, например, кто-то может с удивлением обнаружить, что его знакомый, которого он долго не видел, стал совершенно другим человеком. Можно представить себе мир, который сложно разделить на объекты, а каждый отдельный фрагмент мира уникален и изменяется согласно собственным внутренним закономерностям. В качестве таких «миров» можно было бы назвать облачный покров Земли или атмосферу Солнца. Человек, глядя на облако, пытается описать его привычными формами, однако стоит на пару минут отвести взгляд, как потом уже очень сложно узнать то облако.

Помимо предположения о неупорядоченности образов обучающей выборки производится поиск не единой модели для всех данных, а считается, что данные порождены несколькими независимыми источниками, модель каждого из которых не связана с моделью прочих источников. Опять же, возможность существования независимых причин — предположение не столь несомненное, как мы привыкли считать. Вполне возможно, оно и в нашем мире никогда не выполняется абсолютно строго. Но даже если это и так, то, руководствуясь им, можно получать хоть и приближенные, но гораздо более эффективные алгоритмы построения моделей.

Уместно задать вопрос: а что такое понятие вообще и нужно ли объекты делить на классы? Нужно ли разуму ис-

пользовать понятия, чтобы мыслить, или можно работать со всем набором данных своего опыта и строить для них единую модель? Эти вопросы аналогичны тому, который был поставлен в п. 1.6.3, посвященном обсуждению необходимости введения моделей как таковых. Действительно, понятие — это лишь ссылка на модель. Даже при решении задачи группирования с использованием смесей происходило описание всех объектов единой стохастической моделью $p(\vec{x} \mid \vec{w})$. Отдельные классы возникали лишь постольку, поскольку эта единая модель представлялась в виде суммы отдельных, не зависящих друг от друга компонентов. Однако это очень частный случай. Существование независимых моделей — эвристика, помогающая облегчить процесс построения общей модели и выполняющаяся лишь приближенно.

На практике, однако, применяют более сильные упрощения, используя ограниченные представления и приближенные алгоритмы оптимизации. Основной эвристикой кластеризации был переход от дискретного пространства гипотез группирования к непрерывному пространству параметров, в котором возможно применение процедур типа градиентного спуска. Единственный оставшийся дискретный параметр — это число классов. По нему осуществляется перебор, а принцип МДО позволяет корректно объединить длины описаний для непрерывных и дискретных параметров и выбрать лучшее решение (сложность классических методов как раз и заключалась в задании надлежащего критерия для смешанного пространства). Универсальность количества информации как целевой функции и делает его привлекательным.

Помимо градиентных методов при кластеризации привлекались и другие эвристики поиска. Например, в алгоритме ISODATA применялись эвристики, позволяющие в первом приближении избавиться от перебора по числу кластеров d. Использование в качестве критерия качества длины описания в стиле МДО могло бы сделать принятие решений о слиянии и расщеплении кластеров более строгим, а также помогло бы избавиться от некоторых настроечных параметров, что улучшило бы алгоритм ISODATA и сделало бы его более гибким в применении. Либо наоборот, эти эвристики можно было бы внести в алгоритмы, использующие модели смесей. К сожалению, работы в данном направлении практически отсутствуют.

Разработка строгих критериев по слиянию и расщеплению кластеров должна оказаться очень полезной для систем инкрементного обучения, например, чтобы по мере поступления новых неклассифицированных образов не приходилось заново решать проблему группирования для всех накопленных данных. К примеру, человек, не знакомый с различными направлениями в живописи, будет грубо классифицировать картины, скажем, как нарисованные карандашом или краской. Однако по мере накопления опыта эти классы будут постепенно расщепляться. И наоборот, какие-то объекты или явления, ранее считавшиеся совершенно различными, могут быть объединены в один класс с накоплением знаний и наблюдением промежуточных вариантов. Например, это относится к электромагнитному излучению различных длин волн.

Еще одна эвристика, с которой мы столкнулись при обсуждении смесей, каждый компонент которой также являлся смесью, — использование иерархических моделей. Здесь сначала оказалось необходимым группировать образы в кластеры, а затем группировать сами кластеры на основе сходства между ними. К сожалению, привлечение иерархических представлений в данной задаче мало изучено, поэтому подробно этот вопрос будет рассмотрен далее, в разделе, посвященном проблемам восприятия, для решения которых иерархические представления оказываются чрезвычайно полезны.

2.5. ВЫБОР ПРИЗНАКОВ

2.5.1. Общие замечания о проблеме выбора признаков

При решении рассмотренных проблем классификации, распознавания и группирования предполагалось, что предварительно каким-то образом задано некоторое пространство признаков X. Однако для многих реальных объектов составить исчерпывающий перечень признаков, которые бы их полностью описывали, достаточно проблематично. Тем более, что для разных объектов эти признаки могут оказаться различными: например, вряд ли имеет смысл говорить о лептонном заряде футбольного мяча или материале, из которого «изготовлен» электрон. При этом объекты, как правило, могут быть описаны большим числом признаков,

многие из которых бесполезны в конкретной задаче распознавания. Таким образом, до решения любой задачи распознавания необходимо выбрать различительные признаки объектов и определить процедуры их измерения.

При решении задач распознавания в конкретных приложениях, таких как медицинская диагностика, распознавание рукописных текстов, идентификация личности по биометрическим данным и т. д., выбор признаков осуществляется человеком-экспертом исходя из знания проблемы. При этом принимаются во внимание многие дополнительные факторы, такие как сложность измерения того или иного признака. Например, при идентификации личности дактилоскопия является далеко не самым надежным, но одним из наиболее дешевых и быстрых способов, поэтому часто ее использование оказывается предпочтительнее.

Несложно убедиться, что выбор подобных проблемно-зависимых признаков опирается не только на знания о предметной области, но и на общие представления о действительности. Для автоматического решения этой задачи необходимо либо создание узкоспециализированных алгоритмов, либо возможность получения и использования компьютером соответствующих знаний. Первое не представляет особого теоретического интереса, в то время как второе выходит далеко за рамки проблемной области распознавания образов.

В связи с этим при решении задачи автоматического выбора признаков предполагается, что система распознавания не обладает априорной информацией о предметной области и ей на вход подается описание объектов, содержащее значения фиксированных признаков. Входные значения признаков могут трактоваться как сенсорные данные, являющиеся результатом физических измерений и поступающие на вход машинной системы, которая на них реагирует. Следует заметить, что использование заданного извне пространства признаков не приводит к тому, что машинная система будет обладать какими-то принципиальными ограничениями по сравнению с человеком: последнему в качестве сенсорного входа также подаются образы, принадлежащие предзаданному и фиксированному пространству признаков. Ограничения же автоматического выбора признаков возникают из-за другого упрощения, связанного с отсутствием априорной информации о задаче.

Итак, сформулируем задачу выбора признаков в рамках дискриминантного подхода. Пусть дано исходное простран-

ство признаков $X = R^N$ (проблема формирования исходного пространства в рамках теории распознавания образов не решается). На его основе требуется построить новое пространство $X' = R^n$ размерности n и отображение $F : X \to X'$, обладающие определенными желаемыми свойствами или являющиеся в некотором смысле оптимальными. Существуют два основных подхода к определению критерия оптимальности. В одном из них качество признаков определяется исходя из качества классификации на основе выбранных признаков. При другом подходе считается, что выбор признаков может быть осуществлен безотносительно того, где и как эти признаки будут в дальнейшем использоваться [120, с. 265]. В действительности эти подходы не противоречат друг другу, поскольку в них предполагаются различные исходные данные. В первом случае считается, что дана обучающая выборка векторов, для которых известны принадлежности к классам. Во втором же случае эта принадлежность считается неизвестной. Таким образом, имеют место проблемы обучения с учителем и без учителя соответственно. В связи с этим выделяются и две основные задачи: преобразование кластеризации и собственно выбор признаков.

Преобразование кластеризации заключается в нахождении такого преобразования F, которое бы максимизировало различия между классами и минимизировало различия между образами внутри классов. При этом, как правило, стремятся также уменьшить и размерность пространства признаков.

Цели *выбора признаков* менее четкие, что неудивительно для задачи обучения без учителя. Здесь в качестве общей цели рассматривается снижение размерности пространства признаков, при котором исходные признаки, которые, возможно, плохо характеризуют классифицируемые объекты, заменяются признаками, соответствующими «истинным свойствам» объектов [52, с. 216]. Подобная нестрого сформулированная цель мало помогает решению проблемы, и критерии оптимальности, уточняющие ее, обычно вводятся совместно с дополнительными упрощениями. Однако выбор признаков может быть рассмотрен как частный случай построения представления данных или индуктивного вывода. Такой взгляд на проблему позволяет строго ввести критерий оптимальности, что мы и попытаемся показать. Несмотря на то что при этом выбор признаков перестает рассматриваться как предварительный шаг перед

другими задачами распознавания, становится заметной очень тесная связь между этими разными задачами.

Задачи, аналогичные задаче выбора признаков, возникают не только в распознавании образов, но и во многих других дисциплинах, таких как статистика, анализ временных рядов, обработка сигналов, изучение искусственных и естественных нейронных сетей и т. д. Рассмотрение этих задач как общей проблемы построения представления позволяет абстрагироваться от конкретной дисциплины. Мы, однако, будем рассматривать методы выбора признаков в контексте распознавания образов, чтобы не нарушать единства терминологии и целостности изложения, и начнем с задачи преобразования кластеризации, ставящейся при обучении с учителем.

2.5.2. Преобразование кластеризации при обучении с учителем

Рассмотрим проблему выбора признаков в случае обучения с учителем. Как и в задаче распознавания, здесь в качестве исходных данных выступает набор векторов $\{\vec{x}_1, \ldots, \vec{x}_M\}$, $\vec{x}_i \in \mathrm{X}$, для которых известна принадлежность классам: i-й вектор принадлежит $A_i \in \mathrm{A}$ классу. Как и ранее, обозначим через $x_1^{(k)}, \ldots, x_{M_k}^{(k)}$ образы, принадлежащие k-му классу, $M_1 + \ldots + M_d = M$. Прежде чем переходить к формальным методам выбора признаков, попробуем сначала понять, в чем заключается смысл самой задачи.

Как уже отмечалось выше, в интуитивном понимании образы, принадлежащие одному классу, должны быть похожи между собой или должны иметь близкие значения некоторых признаков. Однако могут присутствовать и признаки, которые сильно различаются для объектов данного класса. Например, разный цвет двух автомобилей не мешает им быть одной модели или, тем более, являться автомобилями. Таким образом, различные признаки могут в разной степени характеризовать тот или иной класс образов. Наилучшими будут инвариантные признаки, т. е. признаки, одинаковые для всех образов класса. Из этих соображений задачу выбора признаков можно трактовать как задачу выбора инвариантных признаков. Однако для признаков с вещественными значениями характерна погрешность их из-

мерения, поэтому задачу следует ставить как присвоение весов q_1, ..., q_N, с которыми должны учитываться признаки в процессе классификации.

Мерой сходства двух образов является расстояние между ними. В случае евклидова расстояния с учетом весов получаем

$$s_{\vec{q}}^2(\vec{x},\vec{y}) = \sum_{i=1}^{N} q_i \left(x_i - y_i\right)^2. \qquad (2.65)$$

Следует найти такие веса, которые бы максимизировали среднее расстояние между образами различных классов и минимизировали среднее расстояние между образами, принадлежащими одному и тому же классу:

$$L(\vec{q}) = \frac{1}{d^2}\sum_{k=1}^{d}\sum_{l=1}^{d} S_{\vec{q}}^2\left(\left\{\vec{x}_i^{(k)}\right\}_{i=1}^{M_k}, \left\{\vec{x}_j^{(l)}\right\}_{j=1}^{M_l}\right) - \frac{1}{d}\sum_{k=1}^{d} S_{\vec{q}}^2\left(\left\{\vec{x}_i^{(k)}\right\}_{i=1}^{M_k}\right), \qquad (2.66)$$

где $S_{\vec{q}}(\cdot)$ и $S_{\vec{q}}(\cdot,\cdot)$ — средние расстояния внутри множества и между двумя множествами соответственно (с учетом весов признаков). Для их вычисления необходимо усреднить расстояния между всеми парами точек, однако в случае евклидовой метрики вместо этого можно считать расстояния до векторов средних.

Пусть $\vec{y}^{(k)}$ — эталонный образ (вектор средних) k-го класса. Тогда внутриклассовое расстояние будет равно

$$S_{\vec{q}}^2\left(\left\{\vec{x}_i^{(k)}\right\}_{i=1}^{M_k}\right) = \frac{1}{M_k}\sum_{j=1}^{M_k}\sum_{i=1}^{N} q_i\left(x_{j,i}^{(k)} - y_i^{(k)}\right)^2. \qquad (2.67)$$

Преобразуем среднее внутриклассовое расстояние:

$$\frac{1}{d}\sum_{k=1}^{M_k} S_{\vec{q}}^2\left(\left\{\vec{x}_i^{(k)}\right\}_{i=1}^{M_k}\right) = \frac{1}{d}\sum_{k=1}^{M_k}\frac{1}{M_k}\sum_{j=1}^{M_k}\sum_{i=1}^{N} q_i\left(x_{j,i}^{(k)} - y_i^{(k)}\right)^2 =$$

$$= \sum_{i=1}^{N} q_i \frac{1}{d}\sum_{k=1}^{M_k}\frac{1}{M_k}\sum_{j=1}^{M_k}\left(x_{j,i}^{(k)} - y_i^{(k)}\right)^2 = \sum_{i=1}^{N} q_i \sigma_i^2,$$

где σ_i^2 — средняя по всем классам дисперсия i-го признака.

Расстояние между классами образов в простейшем случае может быть вычислено как расстояние между их эталонными образами. Тогда

$$S_{\vec{q}}^2\left(\left\{\vec{x}_i^{(k)}\right\}_{i=1}^{M_k},\left\{\vec{x}_j^{(l)}\right\}_{j=1}^{M_l}\right)=\sum_{i=1}^{N}q_i\left(y_i^{(k)}-y_i^{(l)}\right)^2. \qquad (2.68)$$

Среднее расстояние между классами будет равно

$$\frac{1}{d^2}\sum_{k=1}^{d}\sum_{l=1}^{d}\sum_{i=1}^{N}q_i\left(y_i^{(k)}-y_i^{(l)}\right)^2=$$

$$=\sum_{i=1}^{N}q_i\frac{1}{d^2}\sum_{k=1}^{d}\sum_{l=1}^{d}\left(y_i^{(k)}-y_i^{(l)}\right)^2=\sum_{i=1}^{N}q_i\Sigma_i^2, \qquad (2.69)$$

где Σ_i^2 — среднее расстояние вдоль i-го признака между классами образов.

Тогда целевую функцию можно записать как

$$L(\vec{q})=\sum_{i=1}^{N}q_i\Sigma_i^2-\sum_{i=1}^{N}q_i\sigma_i^2. \qquad (2.70)$$

Здесь дисперсии σ_i^2 и Σ_i^2 зависят только от распределения образов внутри классов и от взаимного расположения классов, но не зависят от вектора параметров $\vec{q}$.

Очевидно, для нахождения единственного минимума целевой функции необходимо ввести некоторое ограничение на вектор весов. В качестве такого ограничения можно использовать, например, условие

$$\sum_{i=1}^{N}q_i^2=1. \qquad (2.71)$$

Тогда получаем задачу нахождения условного экстремума, для которой необходимо составить функцию Лагранжа

$$L(\vec{q},\lambda)=L(\vec{q})-\lambda\left(\sum_{i=1}^{N}q_i^2-1\right). \qquad (2.72)$$

Используя условие экстремума

$$\frac{\partial L(\vec{q},\lambda)}{\partial q_i}=\left(\Sigma_i^2-\sigma_i^2\right)-2\lambda q_i=0, \qquad (2.73)$$

несложно получить значения весов:

$$q_i = \frac{1}{2\lambda}\left(\Sigma_i^2 - \sigma_i^2\right), \qquad (2.74)$$

где $2\lambda = \left[\sum_{j=1}^{N}\left(\Sigma_j^2 - \sigma_j^2\right)^2\right]^{1/2}$.

Однако если вычислять межклассовые расстояния как расстояния между эталонными образами, то веса могут получиться отрицательными. По этой и ряду других причин подобный способ вычисления расстояний между классами является не вполне корректным: правильнее считать среднее расстояние от образов одного класса до эталонного образа другого класса. В этом случае межклассовые расстояния всегда будут превосходить внутриклассовые.

Руководствуясь несколько другой эвристически сконструированной функцией потерь и условием $\prod_{i=1}^{N} q_i = 1$, описывающим сохранение элемента объема, можно получить более удобные в использовании веса:

$$q_i = \frac{1}{\lambda}\Sigma_i^2 / \sigma_i^2, \qquad (2.75)$$

где $\lambda = \left(\prod_{i=1}^{N} \frac{\sigma_i^2}{\Sigma_i^2}\right)^{1/N}$.

Такие веса вполне могут использоваться на практике (рис. 2.22).

Использование евклидова расстояния в качестве меры сходства имеет очевидные ограничения. Более универсальным критерием, как уже неоднократно отмечалось, является количество информации. В классическом (шенноновском) подходе среднее количество информации выражается через энтропию, для вычисления которой необходимо знать соответствующие вероятности. Поскольку речь идет об обучении с учителем, то можно полагать, что плотности распределения вероятностей $p_k(\vec{x}), k = 1, \ldots, d$ для каждого класса вычислены. Тогда аналог расстояния от образа $\vec{x}$ до класса k будет равен

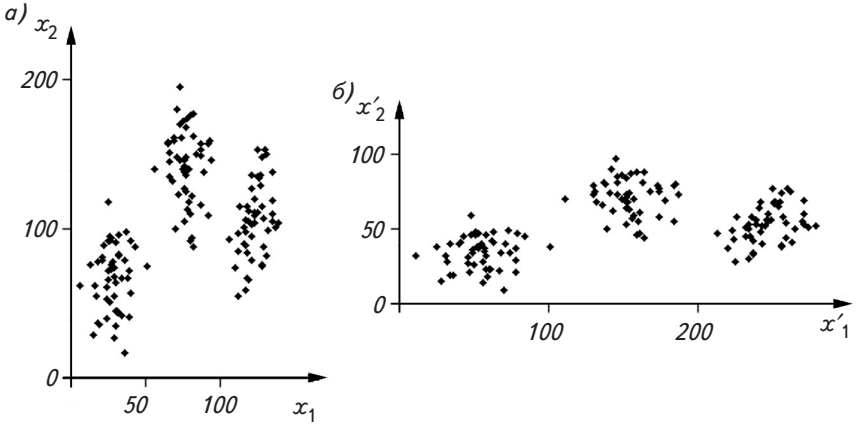

Рис. 2.22. Пример преобразования кластеризации: *а* — образы, принадлежащие трем классам, в исходном пространстве признаков (по этим данным были определены веса признаков $q_1 \approx 4$ и $q_2 \approx 0{,}25$); *б* — результат преобразования кластеризации, представленный как переход к новым признакам $x'_1 = \sqrt{q_1}x_1$ и $x'_2 = \sqrt{q_2}x_2$

$$I(\vec{x} \mid a_k) = -\log_2 \left(p_k(\vec{x}) / \nu_0(\vec{x}) \right), \qquad (2.76)$$

т. е. собственному количеству информации, содержащейся в образе, в предположении, что он принадлежит k-му классу. Здесь $\nu_0(\vec{x})$ — некоторая опорная плотность (см. п. 1.3.3). Связь количества информации $I(\vec{x} \mid a_k)$ с расстоянием становится очевидной, если в качестве плотности вероятности подставить нормальное распределение.

Теперь вместо среднего внутриклассового расстояния мы можем использовать энтропию

$$H_k = -\int_X p_k(\vec{x}) \log_2 \frac{p_k(\vec{x})}{\nu_0(\vec{x})} d\vec{x}. \qquad (2.77)$$

Величина H_k определяет среднее количество информации, содержащееся в произвольном образе k-го класса, если известна его принадлежность этому классу.

Аналог расстояния от класса l до класса k следует определить как среднее «расстояние» образов класса l до класса k:

$$H_{k,l} = -\int_X p_l(\vec{x}) \log_2 \frac{p_k(\vec{x})}{\nu_0(\vec{x})} d\vec{x}. \qquad (2.78)$$

Эта величина характеризует среднюю длину оптимального кода, кодирующего произвольный образ класса l в предположении, что он принадлежит классу k.

Как и выше, для случая евклидова расстояния, нас интересует ситуация, в которой расстояние между классами образов как можно больше, а внутриклассовые расстояния как можно меньше. Тогда в качестве целевой функции (рассматривая пару классов) следует взять

$$J_{k,l} = H_{k,l} + H_{l,k} - H_k - H_l. \qquad (2.79)$$

С помощью несложных преобразований получаем

$$J_{k,l} = \int_X \left(p_k(\vec{x}) - p_l(\vec{x})\right) \log_2 \frac{p_k(\vec{x})}{p_l(\vec{x})} d\vec{x}. \qquad (2.80)$$

Эта величина, называемая *дивергенцией*, выражает полную среднюю информацию для различения двух классов. Часто вывод уравнения дивергенции осуществляется при не вполне корректном задании энтропии (игнорируется необходимость введения опорной плотности), что, к счастью, не сказывается на результате, поскольку в процессе преобразований опорная плотность сокращается. Отсутствие опорной плотности в уравнении для дивергенции делает последнюю удобной в использовании.

Заметим, что критерий дивергенции можно свести к рассмотренным ранее критериям, основанным на функциях расстояния, если плотности вероятности положить равными нормальным распределениям частного вида. Можно также дать нестрогое пояснение, почему веса признаков (2.75) предпочтительнее весов (2.74). Действительно, в случае нормального распределения в одномерном случае энтропия пропорциональна логарифму дисперсии. В уравнении для дивергенции энтропии складываются и вычитаются, поэтому внутриклассовые и межклассовые расстояния необходимо делить и перемножать между собой.

Таким образом, концепция дивергенции позволяет строго ввести степень различия между классами. При этом также учитываются стохастические модели классов $p_k(\vec{x})$, для конкретного вида которых не представляет сложности уточнить уравнение (2.80). Если в результате оно получается не интегрируемым аналитически, то дивергенция может быть оценена путем замены интеграла по всему пространству признаков на суммирование по точкам обучающей выборки. Для определения весов признаков необходимо рассчитать значения дивергенции не по самим плотностям вероятности $p_k(\vec{x})$, а по их сечениям по каждому признаку.

Описанные выше подходы позволяют упорядочивать признаки по весам, т. е. по степени их эффективности, и выби-

рать из них некоторое количество наиболее значимых для уменьшения размерности пространства признаков. Однако при этом не происходит построения новых признаков. Можно было бы тестировать каждый новый признак с помощью одного из введенных выше критериев, но этот подход имеет очевидный недостаток. Пусть мы хотим, например, построить новый признак, являющийся линейной комбинацией уже существующих признаков, но ее коэффициенты неизвестны. Поскольку перебор всех возможных признаков, заданных таким параметрическим способом, бесполезен, возникает вопрос: как определить оптимальные значения коэффициентов линейной комбинации?

Для решения этой проблемы необходимо вернуться к постановке задачи выбора признаков как к задаче определения оптимального отображения $F : \mathrm{X} \to \mathrm{X}'$ из исходного пространства признаков в новое пространство. Отображение F выбирается принадлежащим некоторому параметрическому семейству $F(\vec{x}, \vec{w})$, где $\vec{w}$ — вектор параметров, оптимальное значение которых необходимо определить. Для этого максимизируем дивергенцию

$$\vec{w}^* = \arg\max_{\vec{w}} \left\{ \int_{\mathrm{X}} \left[p_k\left(F\left(\vec{x}, \vec{w}\right)\right) - p_l\left(F\left(\vec{x}, \vec{w}\right)\right) \right] \log_2 \frac{p_k\left(F\left(\vec{x}, \vec{w}\right)\right)}{p_l\left(F\left(\vec{x}, \vec{w}\right)\right)} d\vec{x} \right\}. \tag{2.81}$$

Даже для линейного преобразования координат $F(\vec{x}, W) = \vec{x}^T W \vec{x}$, где W — матрица преобразования, аналитическое решение найдено лишь для случаев, когда $p_k(\vec{x})$ — нормальные распределения частного вида (например, с одинаковыми ковариационными матрицами для всех классов). В противном случае необходимо применять приближенные методы. Здесь они не описываются, поскольку и те и другие методы громоздки и их значимость ограничивается практическим применением. Подробнее они рассмотрены в работах [120, с. 311–327; 168]).

2.5.3. Проблема выбора признаков при обучении без учителя

При выборе признаков, как и при нахождении преобразования кластеризации, требуется найти некоторое оптимальное преобразование $F : \mathrm{X} \to \mathrm{X}'$, где $\mathrm{X} = R^N$ и $\mathrm{X}' = R^n$

в дискриминантном подходе. Однако в качестве исходных данных здесь выступает набор векторов $\{\vec{x}_i\}_{i=1}^{N}$, $\vec{x}_i \in R^M$ без информации об их принадлежности классам. Несложно заметить, что в постановке задачи нигде в явном виде не указывается, что исходные объекты, векторы признаков которых поступают на вход системы, могут относиться к различным классам. Чтобы связать качество признаков с эффективностью классификации, нам бы пришлось для каждого возможного набора признаков решать задачу группирования и оценивать дивергенцию формируемых классов. Очевидно, такой путь приводит к методам, обладающим высокой вычислительной сложностью. Напротив, можно было бы сначала решить задачу группирования, а затем — выбора признаков с помощью преобразования кластеризации. Такой подход в ряде случаев допустим, но в целом при его применении теряется смысл процедуры выбора признаков как предварительной обработки данных, помогающей снизить размерность данных и решить последующие задачи распознавания.

В связи с этим обычно рассматривают критерии оптимальности преобразования F, напрямую не связанные с проблемами распознавания образов. Методы снижения размерности, использующие такие критерии, применяются и при решении других проблем (например, при сжатии данных, в когнитивной графике, для подавления шумов измерений и т. д.). Широкая применимость таких методов говорит в пользу корректности использования критериев, не опирающихся на качество классификации. Более того, как уже замечалось, обе проблемы — группирование и выбор признаков — могут рассматриваться как частные случаи проблемы построения представления данных, и для них могут быть использованы одинаковые критерии, определяющие качество решения.

Исторически первыми и наиболее разработанными методами выбора признаков в случае обучения без учителя являются методы, использующие статистические моменты второго порядка (см. п. 1.4.2). В качестве критерия качества в этих методах выступает точность, с которой образы описываются в новом пространстве признаков уменьшенной размерности. Потеря же точности описания трактуется с точки зрения евклидова расстояния. Применение этих методов в некоторых приложениях, в частности в распоз-

навании образов, показало их ограниченность. Это вызвало интерес, подкрепленный некоторыми нейрофизиологическими данными, к методам, ведущим поиск наиболее «интересных» признаков, привлекая статистики более высоких порядков. В итоге исследователи пришли к критерию, указывающему, что наилучшие признаки должны быть статистически независимыми, что, очевидно, подразумевает и уменьшение длины описания.

К сожалению, в большинстве работ рассматриваются лишь линейные преобразования пространства признаков, что позволяет значительно упростить задачу как с вычислительной, так и с концептуальной точки зрения. Простейшим способом введения нелинейных признаков является прием, использованный для построения обобщенных решающих функций, что накладывает очевидные ограничения, пример которых будет приведен ниже. Другой способ — применение непараметрических оценок преобразования F так, чтобы оно локально подстраивалось под особенности распределения данных. Недостатком этого подхода является то, что строящиеся признаки получаются избыточно сложными (признаки ad hoc).

Поиск нелинейных преобразований даже очень частного вида является сложной задачей, хорошее решение которой пока неизвестно. В связи с этим, чтобы отчетливее пояснить суть различных подходов к выбору признаков, не углубляясь в технические детали, рассмотрим поиск лишь линейных преобразований, и начнем мы с методов второго порядка. Некоторые нелинейные методы описаны в литературе, см., например, [169–171].

2.5.4. Анализ главных компонент и факторный анализ

К классическим методам второго порядка, предназначенным для выбора признаков, относятся *анализ главных компонент* (АГК; principal component analysis, PCA) и *факторный анализ* (ФА; factor analysis, FA). Эти методы очень похожи, если сравнивать их по результирующим формулам, поэтому они иногда отождествляются или один метод рассматривается в качестве частного случая другого метода, хотя исходные предпосылки в них различаются. Начнем описание с АГК.

Предположим, что мы хотим уменьшить размерность векторов признаков таким образом, чтобы по образам, описанным с помощью новых признаков, можно было бы как можно более точно восстановить исходные образы. Рассмотрим сначала случай $n = 1$.

Поскольку мы ограничились лишь линейными преобразованиями пространства X, то новый признак должен являться линейной комбинацией исходных признаков, т. е. должен определять некоторое направление $\vec{w}_1$ в пространстве X. Это направление называется первой *главной компонентой*. Условие минимальной потери точности означает, что проекция векторов обучающей выборки на это направление должна обладать максимальной дисперсией

$$\vec{w}_1 = \arg\max_{\|\vec{w}\|=1} \sum_{i=1}^{M} \left(\vec{w}^T(\vec{x}_i - \vec{y})\right)^2, \qquad (2.82)$$

где $\vec{y} = \frac{1}{M}\sum_{i=1}^{M} \vec{x}_i$ — вектор средних.

Значение найденного таким образом признака для i-го вектора будет равно $x'_{i,1} = \vec{w}_1^T \vec{x}_i$. Но поскольку вектор $\vec{w}_1$ соответствует некоторому направлению в исходном пространстве, то $\vec{w}_1\vec{w}_1^T\vec{x}_i$ — проекция i-го вектора на данное направление, а $\vec{x}_i - \vec{w}_1\vec{w}_1^T\vec{x}_i$ — его проекция на $N-1$-мерное пространство, перпендикулярное к этому направлению, т. е. тот остаток от вектора $\vec{x}_i$, который не описывается новым признаком. В таком $N-1$-мерном пространстве можно найти следующее направление, проекция векторов обучающей выборки на которое обладает максимальной дисперсией. После $k-1$ таких итераций остатки будут иметь вид

$$\vec{x}_i^{(k-1)} = \vec{x}_i - \sum_{j=1}^{k-1} \vec{w}_j\vec{w}_j^T\vec{x}_i, \qquad (2.83)$$

и на их основе можно будет найти очередную k-ю главную компоненту $\vec{w}_k$ абсолютно так же, как была найдена первая и все последующие компоненты. Отметим, что направления, соответствующие главным компонентам, получаются ортогональными.

Оказывается, что поиск n главных компонент совпадает с нахождением n собственных векторов ковариационной

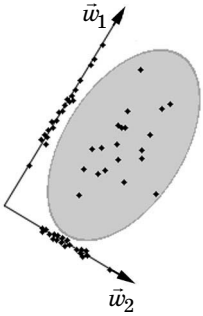

Рис. 2.23. Пример определения главных компонент. Проекция векторов выборки на направление $\vec{w}_1$ обладает максимальной дисперсией, т. е. позволяет максимально полно объяснить вариативность данных. В данном случае главные компоненты адекватно отражают структуру данных

матрицы $C = \frac{1}{M}\sum_{i=1}^{M}(\vec{x}_i - \vec{y})(\vec{x}_i - \vec{y})^T$, соответствующих n наибольшим собственным числам. Это дает возможность не искать последовательно главные компоненты, максимизируя дисперсию проекции векторов обучающей выборки, а использовать стандартные операции с матрицами для определения собственных векторов и чисел. Собственные векторы соответствуют направлению осей эллипсоида, вписанного в данные (рис. 2.23), а собственные числа — размерам осей (точнее, их квадратам).

В факторном анализе, в отличие от АГК, не минимизируются погрешности произвольного описания, а производится построение оптимальной модели объектов одного типа. Векторы обучающей выборки — это измерения характеристик указанных объектов, но эти измерения выявляют не «истинные» свойства или признаки объектов (или скрытые факторы), которые недоступны наблюдателю, а некоторые внешние проявления этих факторов. В своих проявлениях, доступных наблюдателю, факторы смешаны друг с другом и зашумлены. Если предполагать, что факторы смешиваются линейным образом, то моделью, описывающей измерение признаков некоторого объекта, будет

$$\vec{x} = W\vec{\chi} + \vec{\nu}, \qquad (2.84)$$

где $\vec{\chi}$ — вектор скрытых факторов (признаков); W — матрица, определяющая связь между скрытыми признаками и наблюдаемыми признаками; вектор $\vec{\nu}$ представляет собой шум. Разные объекты обладают различными значениями скрытых признаков, но одной и той же матрицей W, так как она описывает природу признаков.

Поскольку ФА — это метод второго порядка, то при отсутствии шума следовало бы минимизировать величину

$$\varepsilon = \sum_{i=1}^{M} \left(\vec{x}_i - W\vec{\chi}_i\right)^2, \qquad (2.85)$$

определяющую точность, с которой модель описывает процесс порождения данных. Это полностью соответствует тому, что делается в АГК. Помимо того что этот факт позволяет применять и в ФА прием, использованный в АГК, — искать скрытые факторы (или «истинные» признаки объектов) как собственные векторы ковариационной матрицы C, он также имеет крайне важное следствие, и суть этого следствия состоит в том, что поиск оптимального представления данных идентичен поиску оптимальной модели источника, порождающего эти данные!

Следует, однако, заметить, что, исходя из уравнения для погрешности (2.85), факторы определяются неоднозначно. Действительно, если взять подпространство $X' \subseteq X$, натянутое на факторы, то любая полная система векторов в этом подпространстве, называемом *факторным* или *АГК-подпространством*, будет соответствовать минимуму погрешности (2.85). Таким образом, помимо минимизации погрешности необходимо использовать дополнительный критерий для выбора конкретных факторов. Часто используется принцип, близкий по смыслу к принципу бритвы Оккама, который гласит, что факторы должны быть такими, чтобы матрица W была как можно проще, т. е. содержала как можно больше нулевых элементов. Это означает наиболее простую связь между скрытыми и наблюдаемыми признаками [172].

Еще одна деталь, отличающая ФА от АГК, заключается в том, что в АГК размерность n пространства X' является либо заданной, либо может быть вычислена, если задана погрешность, с которой в новом пространстве признаков описываются образы. Другими словами, в АГК размерность n определяется тем, где и как будет использоваться новое представление. Например, в задачах когнитивной графики $n = 2$. Факторный же анализ претендует на восстановление истинных признаков объектов, поэтому число таких признаков должно определяться вместе с ними самими, для чего приходится привлекать различные эвристические критерии [52, с. 220] либо информационные критерии для выбора априорных вероятностей в байесовском подходе [173].

И наконец, различие ФА и АГК, которое сказывается на конечных формулах, заключается в введении слагаемого $\vec{\nu}$,

описывающего шум. При наличии шума, который обычно предполагается гауссовым, необходимо искать собственные векторы и собственные числа не ковариационной матрицы C, а матрицы $C - C[\tilde{\nu}]$, где $C[\tilde{\nu}]$ — ковариационная матрица шума. Если эта матрица известна, то содержательно задача не меняется, в противном случае необходимы более сложные методы анализа.

Факторный анализ (как, впрочем, и анализ главных компонент) можно рассматривать как процесс построения стохастической модели объектов, представленных векторами признаков, как если бы эти объекты принадлежали одному классу. Этот процесс опирается на метод максимального правдоподобия и принцип максимума энтропии, а стохастическая модель выбирается из семейства нормальных плотностей. Продемонстрируем данное утверждение. Пусть векторы $\{\vec{x}_i\}_{i=1}^{M}$ распределены нормально с плотностью вероятности $p(\vec{x} \mid C, \vec{y})$. Параметры $C, \vec{y}$ могут быть оценены по векторам обучающей выборки с помощью метода максимального правдоподобия. Поскольку факторы связаны с этими векторами линейной зависимостью, то они также распределены нормально с плотностью вероятности $p(W\vec{\chi} \mid C, \vec{y})$. Руководствуясь принципом максимума энтропии для выбора наиболее информативных признаков, оптимальную матрицу W находят из уравнения

$$H = -\int_{\mathrm{X}'} p(W\vec{\chi} \mid C, \vec{y}) \log_2 p(W\vec{\chi} \mid C, \vec{y})\, d\vec{\chi} \qquad (2.86)$$

при ограничении $\|W\| = 1$.

Как мы убедились в п. 1.4.3, максимизация энтропии в случае гауссова распределения равносильна максимизации дисперсии (в одномерном случае). В многомерном же случае будет максимизироваться сумма собственных чисел ковариационной матрицы распределения вероятностей, а в данной ситуации — распределения $p(W\vec{\chi} \mid C, \vec{y})$, являющегося проекцией распределения $p(\vec{x} \mid C, \vec{y})$ в некоторое подпространство $\mathrm{X}' \subseteq \mathrm{X}$. Таким образом, энтропия будет максимальна, когда подпространство X′ — это АГК-подпространство, натянутое на первые n собственных векторов ковариационной матрицы C. В случае других подпространств энтропия будет меньше.

Критерий максимума энтропии оказывается так же широко применимым в случае выбора признаков, как и концепция дивергенции в случае преобразования кластериза-

ции. Если рассматривать произвольные плотности вероятности, а не только нормальные распределения, то, основываясь на этом критерии, можно построить более универсальные методы. Но, прежде чем переходить к методам, не ограничивающимся моментами второго порядка, отметим следующее.

Как было указано выше, в АГК и ФА предполагается, что все образы обучающей выборки относятся к объектам одного класса. Это является очевидным ограничением, особенно в задачах распознавания образов (рис. 2.24). Можно, однако, вспомнить методы группирования, основанные на смеси нормальных плотностей. Там вместе с разделением объектов на классы происходила и оценка параметров распределений для каждого класса. Иными словами, каждый класс характеризовался собственной ковариационной матрицей, а значит, и своими скрытыми признаками, отличными от скрытых признаков других классов. Хотя в данном методе группирования в явном виде выбор признаков не осуществлялся, его нетрудно расширить, чтобы помимо поиска классов образов производился и выбор признаков. Очевидно также, что методы группирования, в которых предполагается, что все классы описываются одинаковыми признаками, и методы выбора признаков, в которых предполагается, что все образы принадлежат одному классу, являются ограниченными. В связи с этим следует рассматривать методы, решающие обе проблемы одновременно, т. е. методы, которые строят представления, объединяющие свойства локальных и распределенных представлений. Мы кратко остановимся на них в п. 2.5.7.

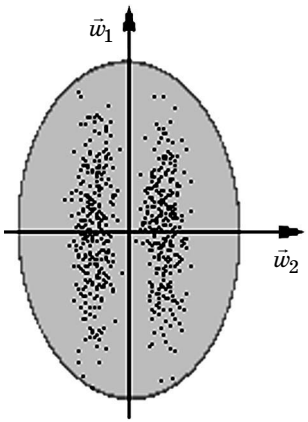

Рис. 2.24. Классический пример ситуации, в которой применение анализа главных компонент повлечет потерю важной информации. Векторы разделяются на два класса. Выбор же первой главной компоненты, обладающей максимальной дисперсией, приведет к полной неразделимости классов

Еще один недостаток методов второго порядка можно проиллюстрировать на основе следующих рассуждений. Пусть компоненты векторов, представленных в исходных данных, выражают некоторые физические величины, например, массу и расстояние. Тогда изменение единиц измерения (скажем, грамм на килограмм) может привести к выделению других факторов. Это значит, что различные компоненты (признаки), составляющие вектор, должны априори полагаться в некотором смысле равноправными.

Несмотря на свою ограниченность, методы второго порядка обладают определенной привлекательностью. Они опираются лишь на информацию из ковариационной матрицы и вектора средних, вычислительно просты и используют лишь классические операции с матрицами, не требуя разработки процедур поиска в пространстве параметров преобразования. В связи с этим для образов, содержащих очень большое число признаков, АГК и ФА могут стать наиболее подходящими методами предварительного выбора признаков. И конечно, они будут наиболее оптимальными, если векторы действительно распределены нормально. Более подробно эти вопросы рассмотрены в соответствующей литературе [174, 175].

2.5.5. Уменьшение избыточности данных и поиск интересных направлений в пространстве признаков

В общем случае плотность вероятности не описывается своими вторыми моментами и возникает необходимость использовать более сложные методы. Исторически такие методы разрабатывались в двух направлениях: уменьшение избыточности данных и поиск «интересных» направлений в пространстве признаков [172]. Методы, разработанные в рамках обоих подходов, оказались тесно связанными и впоследствии привели к подходу, называемому анализом независимых компонент. Поскольку этот подход объединяет предыдущие два подхода, более ранние подходы мы опишем лишь кратко.

Происхождение идеи выбора интересных направлений может стать понятным из анализа рис. 2.24. Из него следует, что проекции векторов на направление $\vec{w}_1$ распределены нормально, в то время как распределение их проек-

ций на направление $\vec{w}_2$ сильно отличается от нормального закона. И хотя дисперсия вдоль второй главной компоненты меньше, чем вдоль первой, это направление кажется более значимым, или интересным. Если пытаться найти такие направления, не производя в явном виде кластеризацию, то это приводит к концепции «интересных» направлений в пространстве признаков как направлений, вдоль которых распределение данных наиболее сильно отличается от гауссова распределения (гауссово распределение считается наименее интересным [176]).

Как указывалось в п. 1.4.2, нормальное распределение обладает максимальной энтропией при фиксированной ковариационной матрице. Для любого другого распределения энтропия строго меньше. В связи с этим одним из наиболее популярных критериев «интересности» направления является энтропия вдоль него при фиксированной дисперсии. Пусть $\vec{X}$ — случайный вектор, реализациями которого являются векторы обучающей выборки. Тогда наиболее интересным направлением будет

$$\vec{w}_1 = \underset{\vec{w}:\sigma^2\left(\vec{w}^T\vec{X}\right)=1}{\arg\min} H(\vec{w}^T\vec{X}), \qquad (2.87)$$

где σ^2 — дисперсия; H — энтропия случайной величины.

Как уже отмечалось, вычисление энтропии непрерывной случайной величины является сложной задачей, поэтому часто используют упрощенные критерии, определяющие степень отличия распределения от нормального закона (см., например, [172] и приведенные там же ссылки).

Во втором подходе, в рамках которого исследуются методы, не ограничивающиеся анализом ковариационной матрицы, выбираются компоненты, позволяющие уменьшить избыточность данных. Этот подход был развит во многом на основе нейрофизиологических данных, полученных в работах Барлоу, Филда и некоторых других авторов. Напомним, что, согласно этим работам, в естественных нейронных сетях (выполняющих первичную обработку сенсорной информации) производится уменьшение избыточности в том смысле, что нейроны одного слоя нейронной сети настраивают свои связи таким образом, чтобы как можно реже активироваться совместно. Иными словами, нейроны выделяют максимально независимые признаки, в связи с чем их активность является некоррелированной.

Эти идеи, почерпнутые из естественных нейронных сетей, были перенесены на искусственные нейронные сети, из-за чего данный критерий развивался преимущественно в рамках нейросетевого подхода. Однако избыточность в смысле совместной активации нейронов может быть перенесена и на избыточность в вероятностном или теоретико-информационном смысле. К рассмотрению метода, получающегося в результате такого перенесения, мы сейчас и перейдем.

2.5.6. Анализ независимых компонент

На языке теории вероятностей избыточность признаков можно интерпретировать как наличие статистической зависимости между ними. Стремление уменьшить эту избыточность соответствует поиску независимых признаков. Именно эта цель преследуется в методе, известном как *анализ независимых компонент* (АНК; independent component analysis, ICA). Как указано в обзоре [172], посвященном АНК, впервые этот метод был сформулирован в диссертационной работе на французском языке [177]. Однако, поскольку идеи анализа независимых компонент тесно перекликаются с подходами, основанными на уменьшении избыточности, методы, похожие по описанию на АНК, можно найти и в более ранних работах.

Существуют несколько формальных определений АНК, различающихся примерно так же, как АГК отличается от ФА. Другими словами, эти различия вызваны тем, производится поиск компонент или факторов и присутствует ли в модели шум. Еще одним отличием между определениями является привлекаемый критерий, указывающий степень статистической независимости. Хотя все такие критерии имеют экстремум при полной статистической независимости случайных величин (признаков), разные критерии могут привести к различным результатам, поскольку при рассмотрении ограниченного множества преобразований $F : \mathrm{X} \to \mathrm{X}'$ получить полностью независимые признаки, как правило, нельзя (рис. 2.25).

Наиболее популярный критерий статистической независимости опирается на понятия взаимной информации и совместной энтропии. Пусть $\vec{X}' = (X'_1, \ldots, X'_n)$ — случайный вектор, полученный из случайного вектора $\vec{X} = (X_1, \ldots, X_N)$ в результате применения регулярного преобразования F.

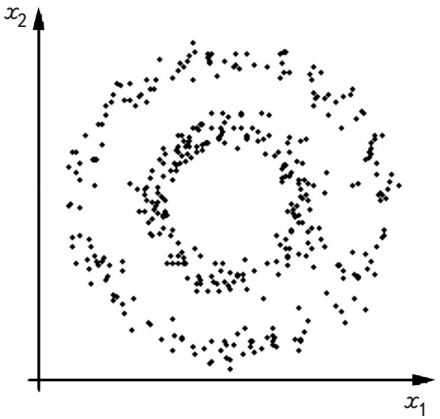

Рис. 2.25. Набор образов, для которых независимыми признаками являются расстояние от центра окружностей и соответствующий угол направления. Никакие линейные комбинации признаков x_1 и x_2 не могут привести к уменьшению избыточности данных. Более того, независимые признаки нельзя найти и нелинейными методами, аналогичными обобщенным решающим функциям

Как отмечалось в п. 1.3.2, статистическая независимость случайных величин $X'_1, \ldots, X'_n$ равносильна условию

$$H(\vec{X}') = H(X'_1) + \ldots + H(X'_n). \qquad (2.88)$$

Отсюда можно получить критерий, указывающий степень статистической независимости:

$$\mathrm{I}(X'_1, \ldots, X'_n) = H(X'_1) + \ldots + H(X'_n) - H(\vec{X}'), \qquad (2.89)$$

который является не чем иным, как средней взаимной информацией случайных величин $X'_1, \ldots, X'_n$. Средняя взаимная информация — это численная оценка информационной избыточности признаков, поэтому данный критерий можно было бы получить сразу же, если вместо статистической избыточности была рассмотрена информационная.

Помимо тесной связи с методами уменьшения избыточности АНК является и обобщением методов второго порядка. Действительно, как указывалось в п. 1.4.4, отсутствие корреляции между признаками является частным случаем их статистической независимости. Факторный анализ можно рассматривать именно как поиск признаков, между которыми отсутствует корреляция. Значит, ФА получается из АНК при предположении о нормальном распределении случайных величин.

Наиболее часто рассматривается линейный АНК, в котором $\vec{X}' = W\vec{X}$. Если энтропия вычисляется корректно с учетом опорной плотности, которая преобразуется вместе со случайной величиной, то $H(\vec{X}') = H(\vec{X})$ при любом невырожденном преобразовании F (см. п. 1.3.3). Если же опорная плотность не учитывается, то $H(\vec{X}')$ и $H(\vec{X})$ различаются

на величину, пропорциональную $\log_2 |W|$ (в линейном случае). В связи с этим при поиске линейного преобразования удобно накладывать ограничение $|W| = 1$.

При использовании АНК двумя основными проблемами являются вычисление энтропии непрерывной случайной величины и поиск оптимальных параметров преобразования. Вычисление энтропии подразумевает оценивание плотности вероятности случайной величины, хотя в явном виде оно может и не осуществляться. Проблема оценивания плотности вероятности была центральной для задач распознавания, и мы ее достаточно подробно рассматривали. Интересно отметить, что если для ее решения используется смесь нормальных плотностей, то это означает, что для каждой гипотезы преобразования $\vec{X}' = W\vec{X}$ будет решаться задача группирования, что заставляет задуматься о целесообразности независимого решения задач выбора признаков и группирования. Однако в анализе независимых компонент обычно используют более простые методы вычисления энтропии (см., например, [172]), избегая явного построения стохастической модели данных.

Поиск матрицы преобразования является нелинейной оптимизационной задачей и чаще всего решается с помощью различных итеративных методов, например, стохастического градиентного спуска или некоторых специфических для АНК алгоритмов [178–181]. Существуют также методы поиска нелинейных преобразований пространства признаков [182, 183]. Анализ независимых компонент является перспективным и бурно развивающимся направлением исследований, но, к сожалению, мы вынуждены опустить многие практические и теоретические вопросы АНК, в частности, вопросы поиска оптимального преобразования. Эти и многие другие вопросы освещены в литературе (см., например, [172, 184], а также специальный номер журнала «Journal of Machine Learning Research», vol. 4, в частности, статьи [169, 185, 186]).

Заметим, что в анализе независимых компонент признаки получаются неупорядоченными и не указывается, какие из них следует исключить, а какие — оставить. Это не является недостатком данного метода, а лишь означает, что в нем решается именно проблема поиска оптимального представления, а не уменьшения размерности. И действительно, независимость найденных компонент подразумевает, что они порождены независимыми причинами, что и по-

зволяет разделить эти причины. Исключение же каких-то признаков означает потерю информации. Однако корректное оценивание важности информации возможно лишь в рамках той задачи, в которой она будет использоваться. Например, исключать ли в процессе анализа рукописного текста из рассмотрения информацию о почерке человека или, наоборот, игнорировать содержание текста, нельзя решить, не обращаясь к тому, с какой целью это делается. Возможно, производится идентификация личности по почерку, а возможно — решается задача распознавания самого текста. Если задача ставится именно как понижение размерности, то могут быть использованы и такие общие критерии, как информативность признаков.

Как видно из рис. 2.26, на основе независимых компонент могут быть построены признаки, гораздо лучше отражающие структуру данных, чем с помощью методов второго порядка. Однако сами признаки могут иметь совершенно различный смысл. Это означает, что полностью распределенные представления являются не вполне адекватными, если в пространстве признаков присутствуют кластеры. Если в случае малого (по сравнению с числом исходных признаков) числа кластеров выделяемые признаки оказы-

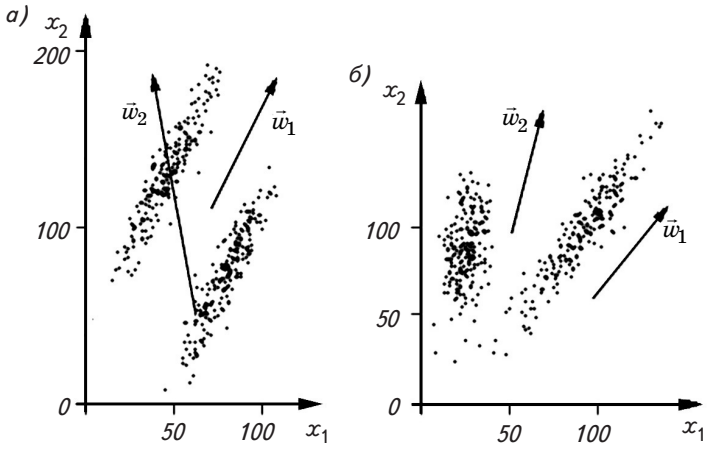

Рис. 2.26. Примеры независимых компонент, использование которых означает переход в косоугольную систему координат, где они будут ортогональны: *а* — одна независимая компонента соответствует признаку, общему для двух классов, другая — является межклассовым признаком; *б* — независимые компоненты выделяют два признака, каждый из которых характеризует лишь один класс

ваются вполне пригодными для последующего группирования, то при большом числе кластеров ситуация ухудшается. Поскольку как полностью локальные, так и полностью распределенные представления имеют свои ограничения, необходимо развивать методы, которые строят общие представления, объединяющие оба типа представлений.

2.5.7. Представления информации, объединяющие свойства распределенных и локальных представлений

Выше уже отмечалось, что простейшие представления, имеющие свойства как локальных, так и распределенных представлений, можно построить на основе смеси нормальных плотностей. Действительно, в этих методах для каждой моды плотности вероятности оцениваются ковариационные матрицы, главные компоненты которых — это новые признаки, собственные для каждого класса (т. е. внутриклассовые признаки). Однако классы сами по себе могут обладать сложной структурой, которая не захватывается методами второго порядка, такими как АГК. Напомним, что при решении задачи группирования не удается в явном виде применять модели смесей для описания классов, поскольку последние «распадаются» на отдельные подклассы. Включение анализа независимых компонент в процедуру группирования, вероятно, поможет частично решить эту проблему.

С другой стороны, модели смеси могут использоваться в процессе оценивания энтропии в стандартной процедуре АНК. Поскольку независимые компоненты в данном случае будут общими для всех классов, то при правильной постановке проблемы в результате можно получить оптимальные межклассовые признаки.

Таким образом, можно предложить следующую последовательность шагов:

1) применить АНК для определения оптимальных межклассовых признаков;

2) выполнить группирование в новом пространстве признаков;

3) определить независимые признаки для каждого найденного класса.

После решения всех трех задач будет получено представление, в котором каждый образ будет описываться номе-

ром класса, к которому он принадлежит, а также перечнем значений признаков, специфических для данного класса. Такое представление, действительно, объединяет свойства как локальных, так и распределенных представлений.

Однако для того, чтобы получить оптимальное совместное решение всех трех проблем, для каждой гипотезы решения первой задачи должна решаться вторая задача. Для каждой же гипотезы решения второй задачи должна решаться третья задача, и только затем должно выбираться лучшее решение. Такое одновременное решение всех трех задач абсолютно неэффективно. С другой стороны, группирование легче осуществлять, если использовать независимые компоненты, найденные обычным методом АНК, а не исходные признаки. Внутриклассовые же признаки могут быть получены и для классов, построенных обычными методами группирования. Это дает первое приближение к решению, которое можно использовать затем, чтобы заново по очереди решить все три задачи. Поскольку некоторая модель данных уже будет доступной после начальной итерации, то первая задача из выбора признаков превратится в проблему нахождения преобразования кластеризации, а вторая задача — из группирования в распознавание. В третьей же задаче построенные на предыдущей итерации внутриклассовые признаки могут быть использованы в качестве начального приближения к нахождению уточненных внутриклассовых признаков.

Такой подход похож по смыслу на алгоритм ожидания-максимизации в задачах с недостающими данными. Однако, чтобы построить корректный метод решения трех перечисленных задач, необходима целевая функция, объединяющая все три шага. Более подробное описание алгоритма здесь не приводится, но вопрос о целевой функции мы рассмотрим. Естественным выбором в данном случае является длина описания.

2.5.8. Информационный критерий качества представления

Прежде чем переходить к вопросу о целевой функции для объединенных представлений, обсудим связь принципа МДО и метода выбора признаков в АНК. Для этого зададимся вопросом: какой смысл (с точки зрения подхода

МДО) строить независимые признаки, если сумма энтропии отдельных компонент случайного вектора в любом случае будет не меньше, чем энтропия самого случайного вектора?

Предположим, что для случайного вектора $\vec{X}$ известна плотность распределения вероятностей. И пусть каждая компонента вектора исходно представляется с некоторой точностью и принадлежит некоторому диапазону, так что может принимать r различных значений. Оптимальный код Хаффмана для такого случайного вектора потребует таблицу перекодировки, содержащую r^N элементов. Это означает, что уже для достаточно малых значений r и N размер таблицы перекодировки будет существенно превосходить количество реализаций случайного вектора, которые требуется закодировать. В связи с этим каждая компонента вектора должна кодироваться отдельно от остальных, но такой код будет неоптимальным, избыточным.

Хотя эти рассуждения являются сильно упрощенными, они показывают, почему (с точки зрения оптимального кодирования) необходимо использовать независимые компоненты. Действительно, значение каждой из статистически не зависимых компонент можно кодировать отдельно, без учета значений других компонент. Это означает, что для получения оптимального кода требуется N таблиц перекодировки суммарного размера rN вместо одной таблицы размера r^N. Таким образом, использование независимых признаков позволяет минимизировать длину описания набора данных либо за счет уменьшения избыточности данных, либо за счет уменьшения размеров таблицы перекодировки (точнее, длины описания стохастической модели, которая может быть представлена и другими способами).

С другой стороны, выбор независимых признаков подразумевает выявление «истинных причин», или скрытых факторов, а не некоторых следствий из них, в которых разные причины переплетены сложным образом, что делает эти следствия статистически зависимыми между собой. Другими словами, минимизация длины описания позволяет проникать в истинную суть явлений и событий, насколько это возможно при имеющейся априорной и апостериорной информации. Это является еще одним неформальным доводом в пользу принципа МДО, а также пояснением к смыслу функционирования методов на его основе.

Итак, анализ независимых компонент может рассматриваться как результат применения принципа МДО к задаче

выбора признаков. В действительности методы, подобные АНК, выводились как обобщение АГК, причем независимо от работ, посвященных самому АНК (см., например, [18, с. 53–76]). Также предпринимались попытки обобщить анализ независимых компонент в рамках теоретико-информационного подхода [187, 188], хотя не всегда авторы при этом обращались именно к принципу МДО.

Рассмотрение проблемы выбора признаков в рамках теоретико-информационного подхода позволяет расширить классический АНК. Во-первых, если необходимо осуществлять выбор между признаками разной степени сложности, то принцип МДО может помочь в этом. Такая необходимость может возникнуть, например, в нелинейном АНК или в методах распознавания, использующих обобщенные решающие функции, применение которых подразумевает необходимость решения проблемы выбора признаков. Во-вторых, принцип МДО может потребоваться, если для вычисления энтропии производится оценивание плотности вероятности.

Более важным, однако, может оказаться то, что критерий длины описания позволяет ввести единую целевую функцию для задач выбора признаков и группирования и рассматривать совместное решение этих двух задач как построение единого представления данных, включающего элементы как распределенного, так и локального представлений.

Итак, пусть есть набор образов, составляющих обучающую выборку. Эти образы необходимо разделить на классы, для каждого из которых нужно выбрать собственные признаки. Воспользуемся формулой (2.64), которая, хотя и обладает рядом недостатков, вполне пригодна для оценивания длины описания произвольной параметрической модели:

$$MDL(d) = L^{(d)} + \frac{n_p^{(d)}}{2}\log_2 M, \qquad (2.90)$$

где $L^{(d)}$ — максимально достижимое правдоподобие данных при числе кластеров, равном d; $n_p^{(d)}$ — число параметров, описывающих плотность распределения вероятности; M — число векторов обучающей выборки.

С каждым классом свяжем новые признаки, определяемые отображениями $F_k^{(d)}(\vec{x})$, где k — номер некоторого класса. Пусть для каждого кластера число новых признаков равно числу исходных признаков (мы рассматриваем представления без потери информации, поэтому это пред-

положение, вероятнее всего, будет верно, однако здесь оно делается лишь для упрощения записи).

Для независимых признаков, принадлежащих к k-му классу, верно разложение

$$p_k\left(F_k^{(d)}(\vec{x})\right) = p_{k,l}\left(F_{k,l}^{(d)}(\vec{x})\right)\ldots p_{k,N}\left(F_{k,N}^{(d)}(\vec{x})\right), \qquad (2.91)$$

где $p_{k,l}$ — сечение плотности вероятности образов k-го класса по l-му признаку; $F_{k,l}^{(d)}(\vec{x})$ — l-й выбранный признак k-го класса.

В связи с этим плотность вероятности можно оценивать отдельно не только для каждого кластера, но и для каждого признака в кластере. Благодаря тому что плотность вероятности в новом пространстве признаков факторизуется, то логарифм от величины $p_k\left(F_k^{(d)}(\vec{x})\right)$ превратится в сумму. Правдоподобие данных $L^{(d)}$ вычисляется так же, как и в методах группирования (оно заменяет эмпирическую оценку энтропии, использующуюся в АНК), хотя и при использовании факторизованных плотностей вероятности, а вот общее число параметров модели изменится.

Во-первых, уменьшится число параметров, необходимое для описания плотности распределения. Действительно, если $p_{k,l}$ — нормальные распределения, то каждое из них описывается двумя параметрами, а для описания факторизованной плотности p_k необходимо лишь $2N$ параметров. Раньше при использовании зависимых признаков необходимо было описывать ковариационную матрицу размера $N \times N$, которая содержала $N(N + 1)/2$ различных параметров, а также вектор средних, состоящий из N параметров. Однако этот выигрыш в числе параметров появился лишь за счет использования преобразований $F_k^{(d)}(\vec{x})$. Если эти преобразования линейны и получены с помощью АГК, то каждое из них описывается как $N(N - 1)/2$ параметров, что в сумме с $2N$ параметрами, описывающими плотности $p_{k,l}$, дает в точности то число параметров, которое раньше было необходимо для описания плотности p_k.

Однако есть определенные отличия. Во-первых, преобразования $F_k^{(d)}(\vec{x})$ могут быть нелинейными, что увеличит число параметров (выигрыш по сравнению с линейным случаем может быть достигнут за счет увеличения правдоподобия). Во-вторых, что более интересно, можно использовать одни и те же признаки для нескольких классов. При решении задачи группирования возникла проблема получения

кластеров неэллиптической формы: использование модели смеси для кластера приводило к «распаду» последнего на субкластеры, которые необходимо было группировать на основе какого-то критерия их сходства. Очевидно, таким критерием сходства может служить возможность использования одинаковых признаков. Использование одинаковых признаков для нескольких кластеров может привести к некоторому уменьшению правдоподобия данных, так как это более простая модель по сравнению с использованием собственных признаков для каждого класса, но уменьшит число параметров. Значит, если использование общих признаков для нескольких классов приводит к уменьшению общей длины описания, то эти классы нужно рассматривать как подклассы класса сложной формы. Можно предложить и более сложные схемы, в которых объединяемые классы будут обладать не всеми, а лишь некоторыми общими признаками, а также схемы, в которых объединение происходит по сходству параметров плотностей распределения вероятностей.

Для иллюстрации основной идеи вернемся к рис. 2.21. В этом примере звезды главной последовательности при использовании смеси нормальных плотностей разбиваются на несколько классов, которые могут быть объединены в один класс, если для него будет выбрано два признака, один из которых является нелинейной комбинацией исходных двух. Такой нелинейный признак непригоден для описания звезд других спектральных классов, поэтому сгруппированы будут только звезды главной последовательности, что и требуется. Таким образом, решив сначала проблему группирования при использовании кластеров простых форм, далее для этих кластеров производится выбор индивидуальных признаков, а затем — постепенное объединение кластеров. При попытке объединить кластеры необходимо выбрать новые признаки (возможно, более сложные, чем имеющиеся) и, если использование этих общих признаков уменьшает длину описания, принять решение о том, что эти кластеры являются субкластерами большего кластера.

Итак, при совместном решении задачи кластеризации и выбора признаков может использоваться тот же критерий, основанный на МДО-принципе, который использовался в задаче группирования. Небольшие модификации критерия связаны со способом вычисления плотности вероятности и числа параметров. Сильнее меняются алгоритмы решения этой задачи, которым еще необходимо посвятить много ис-

следований. Мы же здесь лишь слегка наметили общую схему группирования и выделения внутриклассовых признаков, даже не затронув не менее важную проблему одновременного выбора межклассовых признаков.

2.5.9. Пример практического приложения: выбор текстурных признаков

Рассмотрим пример того, как можно применить принцип МДО при решении конкретной практической задачи выбора признаков. Для наглядности приведем упрощенный подход, который, однако, позволяет получить интересные результаты и показать, какие трудности могут возникнуть при применении принципа МДО к реальным данным.

Мы выбрали проблему текстурного анализа изображений. Понятие текстуры является одним из фундаментальных понятий иконики, науки об изображениях. В простейшем случае текстура возникает из-за периодического повторения значений коэффициента отражения точек некоторой физической поверхности, что вызывает периодические (но, возможно, искаженные перспективой) повторения значений интенсивностей пикселей изображения. Природные объекты, однако, не обладают подобной регулярной периодической структурой. Тем не менее пространственные вариации значений коэффициента отражения у таких объектов также содержат некие закономерности, различные для разных объектов. При этом повторяются не интенсивности отдельных пикселей, а значения некоторых параметров (текстурных признаков) в локальных распределениях интенсивностей пикселей. Это позволяет выделять на изображении области, соответствующие разным объектам, что и является целью текстурной сегментации изображения. Текстурный анализ может использоваться не только для выделения объектов на изображении, но и непосредственно для их распознавания. К примеру, дым в задаче раннего обнаружения лесных пожаров в первую очередь распознается по характерным для него значениям текстурных признаков. Другие задачи текстурного анализа — определение формы объектов по изменению параметров их текстуры, синтез текстуры (например, в целях создания камуфляжной раскраски) и т. д.

Тип закономерности, характеризующей некоторую текстуру, может быть произвольным, что делает затруднитель-

ным формальное определение понятия текстуры и заметно осложняет текстурный анализ изображений. В связи с этим было предложено множество текстурных признаков, которые могут быть постоянными (или изменяться в некоторых пределах) внутри области с одной текстурой и принимать другие значения в области с другой текстурой. Текстурные признаки могут основываться на статистических моментах изображения (например, среднее и дисперсия интенсивностей пикселей в данной области) или моделях марковских случайных полей, коэффициентах локального Фурье или вейвлет-разложения (функциях Габора) и других математических преобразованиях (см., например, [189–193]). Существует также множество эвристических признаков, таких, как, например, число переходов (на единицу площади) через ноль второй производной изображения.

Зачастую бывает невозможно сказать заранее, какую группу признаков следует использовать для различения тех или иных текстур. В связи с этим методы текстурного анализа обычно привлекают большое количество текстурных признаков.

Один из классических подходов к текстурному анализу заключается в следующем: в исследуемом изображении вокруг каждого пикселя берется его окрестность некоторого (сравнительно небольшого) размера, по которой вычисляется вектор текстурных признаков. Таким способом формируется выборка векторов, являющаяся исходными данными для одной из задач распознавания образов. Может решаться как задача обучения без учителя, так и задача обучения с учителем. Последняя имеет место, когда есть возможность заранее подготовить изображения с типичными текстурами, что позволяет в обучающей выборке для каждого вектора устанавливать, к какому классу он принадлежит. Обученная система текстурного анализа далее может работать с новыми изображениями, относя каждый пиксель по его окрестности к тому или иному классу текстуры.

Существует также другой подход к текстурному анализу, больше применимый к задаче выделения отдельных объектов на изображении, а не к их распознаванию. Этот подход использует общие методы сегментации, которые мы рассмотрим позднее (см. пп. 2.6.3 и 3.2.3).

Задача распознавания образов, возникающая в текстурном анализе, отличается сравнительно большим пространством признаков. И хотя может показаться, что признаки

лишними не бывают, ниже будет показано, что перед применением методов распознавания малоинформативные признаки необходимо исключать.

Часто можно встретить, к примеру, методы текстурного анализа, в которых плотность вероятности каждого класса задается в несокращенном пространстве текстурных признаков размерности n, где $n \sim 10^1 \div 10^2$, и описывается гауссианой, имеющей порядка n^2 (т. е. сотни или даже тысячи) параметров! На первый взгляд может показаться удивительным, что этот хорошо обоснованный с позиций теории вероятностей метод распознавания может работать в данном случае хуже, чем примитивный метод сравнения с эталоном в признаковом пространстве с евклидовой метрикой, когда значения всех признаков приведены к диапазону [0, 1]. Когда специалист по обработке изображений сталкивается с такой ситуацией, у него, естественно, может возникнуть недоверие к изощренным методам распознавания образов, и предпочтение он будет отдавать ранним эвристическим методам. Именно поэтому так важно понимать условия применения тех или иных методов.

Ниже мы приведем алгоритм выбора наиболее информативных признаков. Сразу отметим, что этот алгоритм используется лишь для демонстрации необходимости сокращения пространства признаков, и вряд ли его в таком виде следует использовать на практике (хотя он позволяет получить результаты лучшие, чем результаты, полученные методом нормальных смесей без сокращения числа признаков).

На рис. 2.27 приведены образцы текстур двух классов. Вокруг каждого пикселя каждого из этих изображений была взята окрестность, по которой был подсчитан вектор текстурных признаков, принадлежащий одному из двух классов. Таким образом, была сформирована обучающая выборка. В качестве текстурных признаков было использовано 18 из признаков, предложенных Хараликом [193]. На практике, когда число классов текстур больше, может использоваться заметно большее число признаков для различения этих классов.

Построение решающей функции, служащей для дальнейшей классификации точек на новом изображении, производилось методом обобщенных решающих функций, описанным в пп. 2.3.2 и 2.3.6. При этом использовались квадратичные решающие функции, число параметров которых

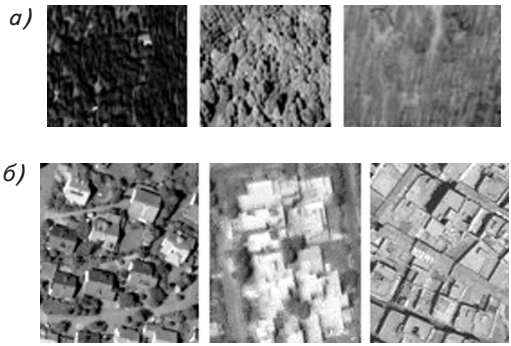

Рис. 2.27. Образцы текстур изображений:
а — леса; *б* — строений

по порядку величины соответствует числу параметров нормального распределения, заданного на том же пространстве признаков. Соответственно, пространство обобщенных признаков имело размерность $n(n+1)/2+1$. В самой процедуре распознавания не осуществлялся выбор подпространства в этом пространстве обобщенных признаков.

Для выбора признаков применялся следующий итеративный алгоритм. Сначала процедура распознавания выполнялась для каждого признака в отдельности. Выбирался признак, дающий наименьшую длину описания (2.38). Далее процедура распознавания выполнялась по очереди для всех пар признаков, в которых первый признак был зафиксированным лучшим признаком, выбранным на предыдущем шаге. Среди всех пар признаков выбиралась та, которая давала минимальную длину описания. Таким же образом производилось дальнейшее добавление признаков. Алгоритм останавливался тогда, когда добавление нового признака не приводило к уменьшению длины описания, т. е. когда первое слагаемое в правой части уравнения (2.38) начинало уменьшаться медленнее, чем возрастало второе слагаемое.

В табл. 2.4 приведены результирующие длины описания данных обучающей выборки при использовании разного числа признаков, длина описания самой решающей функции, суммарная длина описания, а также вероятности правильного распознавания, оцененные по тестовой выборке, полученной по изображениям, не вошедшим в обучающую выборку.

Первый выбранный признак обеспечивает 96,6 % правильного распознавания. Если бы второй выбранный при-

Таблица 2.4

Длины описания и вероятности распознавания обучающей выборки, полученной по образцам текстур (рис. 2.28) при разном числе текстурных признаков

Число признаков n	L (данные)	L (модель)	L (сумма)	Процент распознавания
1	58,3	7,4	65,7	96,6
2	43,2	14,9	58,1	97,6
3	28,6	22,3	50,9	98,0
4	19,2	37,2	56,4	97,7
10	0,4	208,4	208,8	95,2

знак использовался в одиночку, то он обеспечил бы распознавание с 86,7%-ной вероятностью, а третий — с 76,4%-ной вероятностью. Три этих признака позволяют добиться 98 % распознавания. Дальнейшее увеличение числа признаков ведет не только к увеличению общей длины описания, но и к уменьшению вероятности распознавания на примерах, не вошедших в обучающую выборку. На всякий случай отметим, что связь между длиной описания, полученной для обучающей выборки, и вероятностью распознавания, полученной на тестовой выборке, не является строгой и часто может нарушаться, особенно если обучающая (или тестовая) выборка не является репрезентативной.

Таким образом, из приведенных в табл. 2.4 данных видно, что использование лишь части признаков может оказаться лучше использования всех признаков и выбор лучших признаков может осуществляться на основе информационного критерия. Оптимальное число признаков зависит от взаимного расположения классов в признаковом пространстве.

Более наглядным может оказаться результат обученного классификатора на новых изображениях. На рис. 2.28, *а* представлено изображение, для которого осуществлялся текстурный анализ классификатором, обученным по образцам текстуры 2.27, при использовании разного числа признаков. Как видно из рисунка, привлечение трех признаков является предпочтительным, а в случае четырех признаков уже начинает проявляться эффект переобучения.

Конечно, в данном примере мы использовали решающие функции весьма ограниченного вида: $\sum_{i,j=1}^{n} w_{ij} x_i x_j$, в которых

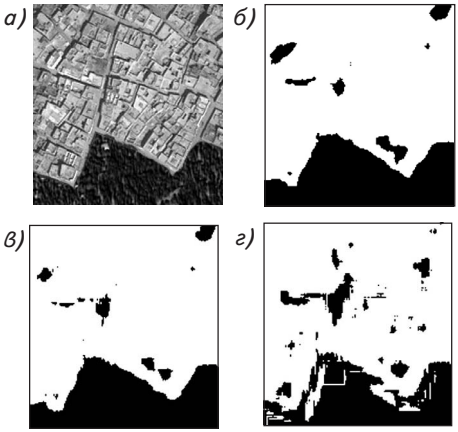

Рис. 2.28. Исходное изображение (*а*) и результаты его текстурного анализа при использовании двух (*б*), трех (*в*), четырех (*г*) текстурных признаков (белые области классифицированы как дома, черные — как лес; обучение классификатора производилось по небольшому числу образцов текстуры, представленных на рис. 2.27)

добавление одного признака приводит к значительному увеличению числа параметров. Вместо этого можно было бы использовать более гибкий подход, кодируя лишь часть параметров обобщенной решающей функции (обобщенных признаков), тогда добавление одного нового признака могло бы не так сильно сказываться на сложности модели, что позволило бы использовать большее число признаков без опасения столкнуться с эффектом переобучения. В этом случае контроль над сложностью решающей функции необходимо выполнять внутри метода распознавания образов.

Отметим следующий важный момент. Может сложиться впечатление, что, увеличивая число признаков, мы увеличиваем объем исходных данных, которые мы описываем, поэтому сравнивать длины описаний при использовании разного числа признаков некорректно. Это было бы абсолютно верно, если бы решалась задача обучения без учителя, в которой от источника к приемнику сообщения передаются закодированные исходные векторы из обучающей выборки. Здесь же получатель сообщения должен лишь восстановить для заранее известных ему векторов обучающей выборки их принадлежность классам по переданной ему информации о разделяющей поверхности. Если какие-то признаки не используются при конструировании разделяющей поверхности, то их можно игнорировать при передаче сообщения. И наоборот, игнорирование каких-то признаков уменьшает сложность разделяющей поверхности, но может вызывать увеличение числа исключений (векторов обучающей выборки, неверно классифицируемых посредством этой разделяющей поверхности), информацию о которых

также нужно передавать. В случае же выбора признаков при обучении без учителя признаки не могут полностью игнорироваться, но могут выделяться из общего вектора признаков и кодироваться независимым образом, что также будет способствовать общему упрощению модели. Еще раз подчеркнем, что при сравнении длин описаний нужно очень внимательно следить за тем, чтобы эти описания относились к одним и тем же исходным данным. Это условие служит своего рода нормировкой длин описаний.

Отметим еще один важный момент. Проблема текстурного анализа изображений характеризуется достаточно большими обучающими выборками, которые могут составлять миллионы векторов. При этом, как видно из рис. 2.27, каждый текстурный класс неоднороден и включает несколько подклассов, в каждом из которых большое число векторов (а промежутки в пространстве признаков между этими подклассами не заполнены). Из-за такой специфики начальных данных сложность строящейся модели существенно меньше длины описания самих данных. Небольшие изменения сложности модели могут привести к достаточно большим случайным изменениям длины описания данных. В результате, даже с учетом сложности модели, может иметь место тенденция к переобучению, которая наиболее ярко проявляется, когда пространство моделей не вполне адекватно регулярностям, содержащимся в данных (например, если границы классов описываются только кривыми второго порядка, а в действительности форма этих границ сложнее).

Мы привели пример удачного решения проблемы выбора признаков, однако возможны ситуации, когда системы машинного обучения, построенные на основе принципа МДО, будут вести себя так, будто они подвержены эффекту переобучения.

Приведем простой пример. Пусть есть некоторая выборка и две модели. Если мы эту выборку увеличим в два раза, просто продублировав ее элементы, то сложности этих двух моделей не изменятся, а соответствующие длины описания данных увеличатся в два раза. При фиксированной сложности моделей можно легко добиться того, чтобы была выбрана более сложная модель (дающая чуть меньшую длину описания данных без учета сложности самой модели), просто достаточное число раз повторив выборку. В действительности именно этот случай имеет место при текстурном анализе: в выборке лишь несколько образцов текстур, каж-

дый из которых многократно (с небольшими вариациями) повторен. Этот пример, казалось бы, приводит к мысли о том, что принцип МДО неверен. Однако это не так. Легко построить модель, которая будет эффективно сжимать данные с повторяющейся информацией. Иными словами, дело не в самом принципе МДО, а в пространстве моделей, которое не содержит компонента, способного описать такого рода регулярность данных.

Тем не менее указанная проблема на практике встречается довольно часто, и о ней нужно помнить. Самое простое решение — использовать только некоторую часть обучающей выборки, чтобы избежать большого числа повторений похожих элементов. Это решение, однако, слишком эвристично и ставит дополнительные вопросы о том, до какой степени уменьшать выборку, не скажется ли это на качестве системы распознавания и т. д. Более строгий метод заключается в том, чтобы построить такое пространство моделей, в котором подобного рода регулярности могли бы быть описаны явно. В частности, можно описывать каждый класс с помощью нескольких эталонов (или конечной смесью с несколькими компонентами) или предложить многоуровневую систему распознавания, в которой на нижнем уровне выполняется векторное квантование (группы близко расположенных векторов заменяются на некоторое усредненное значение), которое также может выполняться на основе принципа МДО [167]. Более подробный анализ проблемы больших выборок выходит за пределы данной работы.

2.6. РЕГРЕССИЯ И СЕГМЕНТАЦИЯ

2.6.1. Задача регрессии

Распознавание образов (в рамках дискриминантного подхода) может рассматриваться как поиск отображения из непрерывного множества значений в дискретное множество. Отличный тип моделей строится в другой фундаментальной задаче статистического анализа и машинного обучения. Это задача регрессии, в которой производится поиск отображения из одного непрерывного множества в другое непрерывное множество. Хотя задача регрессии является не менее важной, чем группирование, остановимся на ней достаточно кратко, поскольку частные ее варианты были не-

однократно рассмотрены в процессе изложения предыдущего материала.

Итак, сформулируем задачу регрессии. Пусть $\vec{Z}$ и $\vec{Y}$ — случайные векторы, причем $\vec{Z}$ — это вектор независимых переменных размерности N, а $\vec{Y}$ — вектор зависимых переменных размерности N_1. И пусть есть обучающая выборка — набор из M пар векторов $\left(\vec{z}_i, \vec{y}_i\right)_{i=1}^{M}$, являющихся отсчетами соответствующих случайных векторов. Необходимо построить модель, с помощью которой можно было бы по новым отсчетам вектора $\vec{Z}$ предсказывать значение вектора $\vec{Y}$. Для этого вводится матрица факторов (или переменных) регрессии $\hat{X} = \left(\vec{X}_1, \ldots, \vec{X}_n\right)$ размера $N_1 \times n$, получающихся из независимых переменных в результате применения некоторого преобразования $\hat{X} = F(\vec{Z})$ (в простейшем случае преобразование может быть тождественным и факторы регрессии будут совпадать с независимыми переменными). Поскольку $\vec{Z}$ — случайный вектор, то и матрица $\hat{X}$ является случайной и будет различной для разных реализаций вектора $\vec{Z}$.

Чаще всего рассматривается линейная регрессионная модель, которая вводится как

$$\vec{Y} = \hat{X}\,\vec{w} + \vec{R}, \tag{2.92}$$

где $\vec{w}$ — n неизвестных коэффициентов регрессии, которые необходимо определить по обучающей выборке; $\vec{R}$ — случайный вектор невязок (или ошибок регрессии).

Классический метод нахождения параметров регрессионной модели — это метод наименьших квадратов, в котором производится поиск модели, для которой достигается минимум суммы квадратов невязок. Вектор невязок для каждой реализации случайных векторов будет

$$\vec{r}_i = \vec{y}_i - \hat{X}_i\vec{w}. \tag{2.93}$$

Тогда целевая функция задается как

$$L(\vec{w}) = \sum_{i=1}^{M} \left\|\vec{r}_i\right\|^2 = \sum_{i=1}^{M} \left\|\vec{y}_i - \hat{X}_i\vec{w}\right\|^2. \tag{2.94}$$

В результате дифференцирования по $\vec{w}$ и приравнивания частных производных нулю получается система линейных уравнений, для решения которой существуют стандартные методы.

Метод наименьших квадратов обладает очевидными ограничениями. Как уже отмечалось, критерий среднеквадратичного отклонения можно получить в результате применения метода максимального правдоподобия при использовании модели гауссова шума частного вида. Метод максимального правдоподобия при необходимости позволяет расширить метод наименьших квадратов на случай других моделей шума. Однако в классическом статистическом подходе остается еще одна проблема, к описанию которой мы и перейдем.

2.6.2. Проблема выбора факторов и ее решение с помощью принципа МДО

В общем случае неизвестными в дополнение к параметрам регрессии являются также факторы, т. е. связь F между независимыми переменными и факторами не дана априори и ее требуется определить. Выбор факторов влияет на сложность регрессионной модели, и метод максимального правдоподобия, игнорирующий априорные вероятности моделей, перестает давать адекватные результаты, что и составляет корень проблемы классических статистических методов.

Как и в случае обобщенных решающих функций, служащих для формирования новых признаков, факторы обычно выбираются из множества, строящегося на основе некоторой системы функций $f_1, f_2, \ldots$, где $f_i : R^N \to R^{N_1}$. Для некоторого подмножества этого множества функций может быть сформулирована обычная задача регрессии:

$$\vec{Y} = w_1 f_{i_1}(\vec{Z}) + \ldots + w_n f_{i_n}(\vec{Z}) + \vec{R}. \qquad (2.95)$$

В этом смысле проблемы выбора факторов в регрессии и выбора признаков в распознавании являются очень близкими.

Часто в качестве системы функций $\{f_i\}$ используется некоторый базис в гильбертовом пространстве, например, система ортонормированных полиномов [194], или вейвлет-разложения [195]. В итоге может быть построено сколь угодно много факторов.

Трудность же классических статистических подходов заключается в том, что, взяв достаточно большое число факторов регрессии, можно получить равную нулю дисперсию

для $\vec{r}_i$. Причем этот результат может быть достигнут для сколь угодно большого числа регрессионных моделей, опирающихся на разные факторы. Осуществить выбор между этими моделями, не привлекая дополнительных соображений, не представляется возможным. Таким образом, использование регрессионных моделей с большим числом факторов приводит к проблеме переобучения.

Вернемся к полиномиальной аппроксимации и рассмотрим ее в рамках регрессионного анализа (в одномерном случае $N = 1$ и $N_1 = 1$) в качестве наглядного примера, иллюстрирующего проблему переобучения. Для простоты факторы, среди которых осуществляется выбор, будут мономами произвольной степени от единственной независимой переменной: $X_k = Z^k$.

Обратимся к рис. 2.29, на котором представлены набор точек и четыре полинома разных степеней, обладающих минимальной среднеквадратичной ошибкой для данной степени. Параметры полиномов определялись по десяти точкам. Одиннадцатая точка (на рисунке — ○) в обучающую выборку не входила. Несмотря на то что полиномы с бóльшим числом параметров n обладают меньшей среднеквадратичной ошибкой, они хуже предсказывают положение точек, не вошедших в обучающую выборку.

В классическом статистическом анализе проблема переобучения решается различными эвристическими методa-

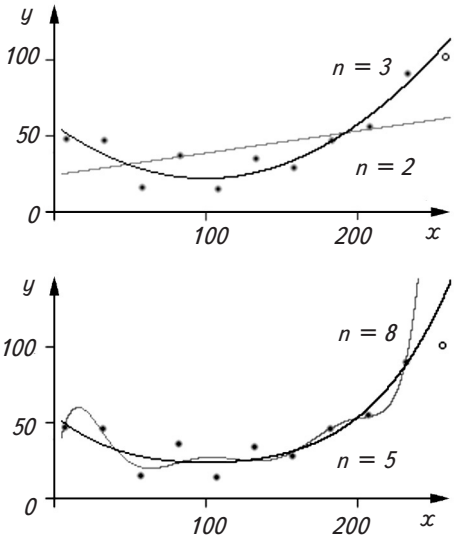

Рис. 2.29. Пример задачи регрессии с неизвестным набором факторов: аппроксимация набора точек полиномами произвольной степени (n). Минимальное СКО достигается при максимальной степени полинома, что приводит, однако, к наихудшему предсказанию значения функции для точек, не вошедших в выборку (точка, обозначенная ○)

ми, такими как методы перекрестной проверки, в которых лишь часть данных используется для оценивания параметров, а другая часть — для определения точности решения. Более строгое решение заключается в оценивании априорных вероятностей моделей на основе принципа минимальной длины описания.

Рассмотрим теоретико-информационный подход к регрессии. Пусть есть источник и приемник информации. И пусть требуется передать значения векторов $\vec{y}_i$, при этом значения вектора $\vec{z}_i$ получателю сообщения известны. Вместо того чтобы передавать сами значения $\vec{y}_i$, можно передать модель, описывающую, как по значениям $\vec{z}_i$ восстановить $\vec{y}_i$. Чем лучше модель описывает данные, тем более компактным будет передаваемое сообщение.

В случае линейной регрессионной модели необходимо передать описание преобразования F, вектор коэффициентов регрессии $\vec{w}$, а также невязки $\vec{r}_i$. Хорошей оценкой длины описания невязок является минус логарифм правдоподобия данных:

$$L_r = -\sum_{i=1}^{M} \log_2 p(\vec{r}_i). \tag{2.96}$$

Как уже неоднократно отмечалось, такое определение количества информации для непрерывной случайной величины, не учитывающее необходимость введения опорной плотности, является некорректным. Особенно отчетливо это видно в случае нормального распределения невязок с одинаковой дисперсией по всем направлениям. Тогда эту длину описания с точностью до аддитивной константы можно оценивать по формуле

$$L_r = M \log_2 \sigma(\vec{r}_i), \tag{2.97}$$

где $\sigma(\vec{r}_i)$ — среднеквадратичное отклонение.

Если дисперсия стремится к нулю, количество информации стремится к минус бесконечности, что является абсурдным результатом. Простым способом преодоления этой проблемы является использование другой оценки:

$$L_r = \frac{M}{2} \log_2 \left(\sigma^2(\vec{r}_i) + \varepsilon^2 \right), \tag{2.98}$$

где ε — погрешность задания непрерывных величин; если эта величина неизвестна, можно воспользоваться «честным» способом вычисления длины описания, в явном виде при-

меняя некоторую схему кодирования с вычислением длины получающегося сообщения.

Второй частью сообщения является описание регрессионной модели, которое состоит из описания преобразования F и вектора коэффициентов $\vec{w}$. Длина этого описания соответствует минус логарифму априорной вероятности модели, т. е. штрафует сложность модели. Здесь можно применить те же критерии, что и в случае проблемы группирования, а именно: *AIC*, *BIC* и др. В частности, можно использовать наиболее простой критерий, основанный на принципе МДО: $L_p = \frac{1}{2} n \log_2 M$ — длину описания параметрической части модели.

Вообще этот критерий сильно упрощен. Более сложные МДО-критерии приведены, например, в работах [2, 194]. Но привлекательнее является оценивание числа бит, которые следует отводить под каждый коэффициент регрессии при построении описания. Это позволяет не только более корректно определить длину описания модели, но и оценить точность каждого параметра. Здесь же все параметры считаются равноправными независимо от того, с каким фактором они перемножаются.

Покажем функционирование МДО-принципа на самом простом критерии:

$$MDL = L_r + L_p = M \log_2 \sigma[\vec{r}_i] + \frac{n}{2} \log_2 M \qquad (2.99)$$

Таблица 2.5

Длины описаний для оптимальных полиномов восьми различных степеней, аппроксимирующих набор точек, представленных на рис. 2.29

n	L_p	$\sigma(\vec{r}_i)$	L_r	MDL
1	1,66	20,83	43,81	45,47
2	3,32	18,05	41,74	45,06
3	4,98	8,36	30,64	35,62
4	6,64	8,13	30,23	36,87
5	8,30	7,96	29,93	38,23
6	9,97	7,62	29,32	39,28
7	11,63	7,55	29,16	40,79
8	13,29	6,65	27,33	40,62

и используем его для случая, изображенного на рис. 2.29. В табл. 2.5 представлены суммарные длины описания, получающиеся для полиномов различной степени. Из таблицы видно, что минимальная длина описания достигается для полинома второй степени ($n = 3$). Эта модель не только дает наилучшую точность предсказания положения точек, не вошедших в обучающую выборку, но и соответствует истинной функции, использованной для порождения данных.

Регрессия была одной из первых задач наряду с распознаванием, для которых был применен принцип МДО. И сейчас продолжаются работы в данном направлении (см., например, [2, 194, 196]), в частности, с помощью МДО решаются задачи выделения сигнала на фоне существенных шумов [197–200]. Еще одно направление исследований, связанное с регрессией, посвящено проблеме сегментации, решение которой требует совместного осуществления группирования и регрессии.

2.6.3. Задача сегментации

Задача сегментации возникает всякий раз, когда требуется некоторый массив данных разбить на однородные порции, причем понятие однородности может быть достаточно сложным, например, при разделении слитной речи на слова, а также изображения — на различные объекты и фон. Другой пример — это сегментация экспериментальных кривых. При этом, как правило, фиксируются изменяющиеся во времени характеристики некоторого объекта. Построенную кривую разбивают на отдельные участки, в которых изменение характеристик объекта подвержено какой-то одной закономерности, а различие закономерностей на разных участках кривой связано с переходом между дискретными состояниями объекта.

Таким образом, при сегментации необходимо использовать модели, являющиеся гибридными моделями регрессии и группирования: отдельные элементы исходных данных должны быть объединены в группы, для каждой из которых строится собственная регрессионная модель. Как и в случае задачи регрессии, в качестве исходных данных выступает набор пар векторов $\left(\vec{z}_i, \vec{y}_i\right)_{i=1}^{M}$. Независимые переменные $\vec{z}$ обычно являются пространственными или вре-

менными координатами, а значениями зависимых переменных — измерения некоторых (физических) характеристик в данной точке пространства или в данный момент времени. При формулировании задачи сегментации практически всегда используется предположение о пространственно-временной однородности нашего мира. Это предположение выражается в способе постановки задачи группирования в рамках сегментации, а именно: не допускается произвольное группирование элементов данных $(\vec{z}_i, \vec{y}_i)$. Вместо этого в пространстве независимых переменных должны быть выделены области, в каждой из которых применяется своя регрессионная модель (рис. 2.30).

В такой постановке задача сегментация может быть сформулирована следующим образом. Пусть есть набор измерений: $(\vec{z}_i, \vec{y}_i)_{i=1}^{M}$, где $\vec{z}_i \in \mathrm{Z} = R^N$ и $\vec{y}_i \in \mathrm{Y} = R^{N_1}$. Необходимо сформировать области G_1, ..., G_d, $G_k \subset \mathrm{Z}$ и для каждого набора пар $\{(\vec{z}_i, \vec{y}_i) : \vec{z}_i \in G_k\}$ построить собственную регрессионную модель: $g_k(\vec{z}, \vec{w}_k) : G_k \to \mathrm{Y}$, например, $\vec{Y} = w_{k,1} f_{k,1}(\vec{Z}) + \ldots + w_{k,n_k} f_{k,n_k}(\vec{Z}) + \vec{R}$.

Прямой метод решения задачи сегментации заключался бы в переборе всех возможных способов группирования и построении для каждого из них регрессионной модели. Однако этот метод ресурсоемок и непрактичен, хотя подобный исчерпывающий поиск и будет давать оптимальный (с точки зрения выбранного критерия) результат. Вместо этого можно применять различные эвристики поиска, аналогич-

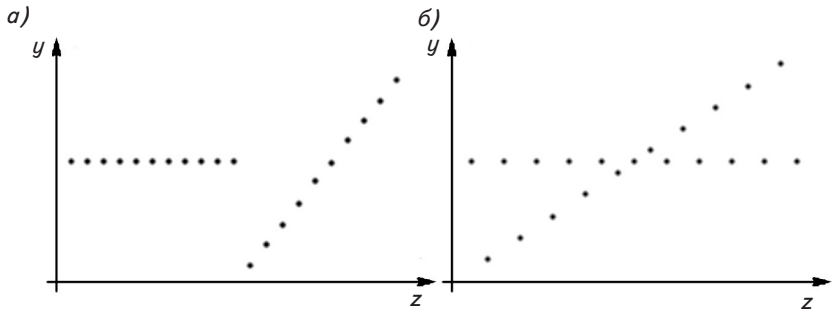

Рис. 2.30. Пример, иллюстрирующий ограничения, накладываемые сегментацией на области: *а* — расположение точек описывается двумя регрессионными моделями, введенными на областях (отрезках на оси *z*); *б* — группирование точек на основе выделения областей невозможно, хотя для этого случая также достаточно лишь двух регрессионных моделей

ные тем, которые использовались в алгоритме ISODATA. В дополнение к приемам слияния и расщепления кластеров (а в данном случае сегментов) можно воспользоваться тем, что достаточно часто точки $\vec{z}_i$ образуют прямоугольную сетку. Это позволяет определять те точки, которые располагаются на границах областей G_i, и улучшать качество сегментации, пытаясь перенести такие точки из одной области в другую и корректируя при этом параметры регрессионных моделей. Для каждой конкретной задачи могут вводиться и дополнительные эвристики.

Помимо организации поиска необходимо, конечно, решить вопрос о критерии качества сегментации. Невязки модели сегментации определяются как

$$\vec{r}_i = \vec{y}_i - g_k(\vec{z}_i, \vec{w}_k), \qquad (2.100)$$

где $k : \vec{z}_i \in G_k$.

Оценив плотность распределения вероятностей невязок, можно использовать метод максимального правдоподобия. Однако, если число сегментов априорно неизвестно, этот метод будет иметь тенденцию выделять под каждый элемент данных отдельный сегмент, для которого строится собственная регрессионная модель, точно описывающая единственную попавшую в сегмент пару $(\vec{z}_i, \vec{y}_i)$. Если же допускаются и регрессионные модели разной степени сложности, то другой крайностью будет объединение всех точек в один сегмент с регрессионной моделью, содержащей большое число факторов. Помимо различных искусственных приемов, позволяющих избежать этих двух вырожденных случаев, решением проблемы выбора оптимальной модели сегментации могут быть теоретико-информационные критерии.

2.6.4. Информационный критерий качества сегментации

Для определения критерия качества модели сегментации представим, что есть отправитель и получатель сообщения. Отправителю известны значения как $\vec{z}_i$, так и $\vec{y}_i$, в то время как получателю известны значения лишь независимых переменных. Какую информацию необходимо включить отправителю сообщения, чтобы получатель смог однозначно восстановить значения $\vec{y}_i$? Для этого получателю необходимо знать параметры регрессионных моделей $\vec{w}_k$ (и соответ-

ствующие факторы) для каждого сегмента, информацию о самих областях G_k, чтобы для каждой точки $\vec{z}_i$ была возможность узнать, какую именно регрессионную модель использовать, а также значения невязок $\vec{r}_i$.

Правдоподобие связано лишь с длиной описания невязок. Методы, не учитывающие другие составляющие, разумеется, будут иметь склонность к переобучению. Можно, например, опустить описание областей, тогда общая длина описания сведется к сумме длин описаний регрессионных моделей каждого сегмента. Способ вычисления этих длин был описан выше, в п. 2.6.2 [см., например, уравнение (2.99)]. Такое упрощение часто бывает вполне допустимо, поскольку почти не мешает выбору подходящего числа сегментов. Действительно, поскольку каждому сегменту соответствует регрессионная модель, которая должна быть описана, то чем больше выбрано сегментов, тем больше будет и суммарная длина описания модели сегментации. Небольшой недоучет сложности модели (которая состоит из большого числа сегментов), возникающий при отбрасывании длины описания областей, позволяет получить более простые алгоритмы. Еще более сильное упрощение заключается в том, чтобы использовать единую стохастическую модель для описания невязок, задаваемых уравнением (2.100), а сложность модели вычислять через суммарное число параметров во всех регрессионных моделях. Тогда можно непосредственно использовать уравнение (2.99).

Тем не менее в ряде задач именно описание G_k областей является наиболее существенным. Это характерно, например, для задач сегментации изображений. Здесь процесс описания областей по смыслу соответствует определению формы объектов, которая содержит важнейшую информацию, необходимую в задачах анализа изображений. В связи с этим в общей постановке задачи сегментации вопрос описания областей игнорировать нельзя.

Области могут быть представлены своими границами. Обычно предполагается, что эти границы гладкие, поэтому они могут быть аппроксимированы с помощью некоторого параметрического семейства гладких функций. Это задача регрессии. Если же считать, что поверхности областей гладки почти всюду, то придем к постановке задачи описания областей через их границы в качестве задачи сегментации, но в пространстве размерности на единицу меньше, чем исходное пространство. Эти рассуждения можно продолжить

и дальше, в итоге получится цепочка взаимосвязанных общим критерием длины описания задач сегментации в пространствах уменьшающейся размерности.

Рассмотрим два «соседних» уровня:

$$L = \sum_k L\left((\vec{z}_i, \vec{y}_i) : \vec{z}_i \in G_k\right) + \sum_k L(\delta G_k), \qquad (2.101)$$

где $L\left((\vec{z}_i, \vec{y}_i) : \vec{z}_i \in G_k\right)$ — длина описания регрессионной модели для точек, попавших в область G_k [например, с помощью уравнения (2.99)], а $L(\delta G_k)$ — длина описания границы соответствующей области.

Если значения $L(\delta G_k)$ не учитываются, то границы областей будут получаться изрезанными: любой локальный выброс, произошедший из-за шума, может быть отнесен к неверной области лишь на основе значений зависимых переменных в данной точке, невзирая на ее пространственное положение. В то же время чем проще граница области, тем меньше ее длина описания $L(\delta G_k)$. Значит, учет этих величин должен приводить к получению более гладких границ областей и более адекватного результата сегментации. Если этот учет корректен, то также не будут получаться и слишком сглаженные границы, поскольку это приведет к понижению правдоподобия данных в рамках регрессионных моделей.

Таким образом, задачи, которые можно было бы решать отдельно (собственно сегментацию исходных данных и описание границ областей), оказываются связанными общим критерием качества. Первая сумма в уравнении (2.101) соответствует качеству регрессионных моделей, а вторая сумма может быть проинтерпретирована как критерий качества группирования. Как и при совместном решении проблем выбора признаков и кластеризации, в данном случае принцип минимальной длины описания позволил связать две отдельные, на первый взгляд, задачи. Большая значимость данного результата будет показана при обсуждении вопросов интерпретации изображений в гл. 3, в которой мы неоднократно столкнемся с необходимостью привлечения моделей сегментации. В связи с тем что к вопросу сегментации мы еще вернемся на конкретных примерах, здесь он был описан достаточно кратко. Более детальное описание теоретико-информационных методов сегментации можно найти, например, в работах [2, 201, 202].

2.7. ЗАКЛЮЧЕНИЕ

Задача сегментации, как и все предыдущие задачи, может быть выражена через поиск такой наикратчайшей программы для универсальной машины Тьюринга, которая бы на выходе порождала данную обучающую выборку. Задача сегментации отличается от задач группирования, выбора признаков и регрессии лишь дополнительными предположениями о структуре пространства моделей. Исходные данные могут рассматриваться во всех этих задачах как одинаковые. Действительно, если в задаче регрессии или сегментации объединить зависимые и независимые переменные, то полученный вектор будет иметь тот же смысл, что и вектор признаков в двух предыдущих задачах. Таким образом, если в универсальном (алгоритмическом) пространстве моделей вводить некоторые общие ограничивающие предположения, то это приводит к нескольким задачам построения моделей. Причем эти задачи не только отличаются формулировками, но и требуют разработки различных методов решения, коль скоро они должны быть применимы на практике. Часто эти методы оказываются достаточно изощренными, поэтому сомнительно, что можно автоматически формировать процедуру построения модели при произвольных ограничениях на пространство моделей на ранних этапах функционирования системы машинного обучения.

В данной главе мы рассматривали лишь дискриминантные методы, которые оперируют с данными, имеющими примерно одно и то же исходное представление. Как отмечалось в п. 2.1.3, в распознавании образов существуют различные подходы, отличающиеся структурой пространства признаков, с которым они оперируют. Таким образом, большое разнообразие методов построения моделей увеличивается также в связи с дополнительными предположениями о структуре исходного представления данных.

Многие практические методы оказываются слишком частными, работающими лишь при сильных ограничениях на пространство моделей и привлекающими дополнительную априорную информацию. Системы машинного обучения, основанные на этих методах, оказываются ограниченными в тех понятиях, распознаванию которых они могут научиться. С другой стороны, подходы к индуктивному выводу, основанные на алгоритмической сложности, являют-

ся слишком общими и слишком «теоретическими» для использования на практике. Основная трудность заключается в том, чтобы свести эти две крайности. Для достижения этой цели можно как постепенно обобщать и объединять частные методы, так и обогащать различными эвристиками универсальный метод. Устойчивая тенденция к постепенному обобщению частных методов действительно наблюдается.

Выделим и основные приемы и допущения, которые мы обнаружили при обсуждении дискриминантных методов (многие из этих допущений не связаны непосредственно с дискриминантным подходом).

• Неупорядоченность или независимость отдельных порций данных (во всех задачах отсчеты одной и той же случайной величины считались независимыми и одинаково распределенными) — независимость в пространстве данных.

• Разделение модели на независимые составляющие (в задачах распознавания строилась отдельная модель для каждого класса, а в задаче сегментации — для каждого сегмента, при выборе признаков производился поиск независимых компонент) — независимость в пространстве моделей.

• Векторы, расположенные на бесконечно близком расстоянии, почти всегда эквивалентны (в задачах распознавания класс занимает некоторую область в пространстве признаков) — непрерывность в пространстве данных.

• Малое изменение обучающей выборки почти всегда влечет малое изменение модели (в инкрементном обучении при появлении нового образа модель класса не строится заново, а лишь корректируется) — непрерывность в пространстве моделей.

• Существование обобщенных моделей (при классификации осуществляется индуктивный вывод на очень ограниченном пространстве моделей, сформированных на этапе распознавания; если при классификации происходит построение модели индивидуального объекта, то при распознавании — обобщенной модели для целой их совокупности).

Все эти предположения могут нарушаться, и строящиеся на их основе модели не будут оптимальными. Однако, как уже отмечалось, отказ от оптимальности является обязательным для получения эффективных методов.

Ограниченность ресурсов (памяти и времени) даже может вводиться как формальное условие задачи индуктив-

ного вывода [33]. Обычно ограничения на ресурсы вводятся как некоторые жесткие пороги, что не всегда адекватно отражает суть проблемы ресурсов. В связи с этим было бы интересно модифицировать принцип МДО таким образом, чтобы в нем учитывалась не только длина строящейся программы, но и скорость ее работы (или число элементарных операций, выполняющихся в процессе генерации исходного набора данных). К примеру, с точки зрения такого гипотетического критерия сам поиск наикратчайшей программы является неоптимальным алгоритмом построения моделей и должен быть заменен приближенным, но более быстрым алгоритмом (способ точного определения наикратчайшей программы можно сравнить с моделью ad hoc). Теоретические исследования в этом направлении ведутся (например, уже упоминавшиеся работы Хаттера [97–99] и Шмидхубера [100–102]), но на данный момент малочисленны. Возможно, именно здесь скрывается возможность еще более приблизиться к построению самооптимизирующихся алгоритмов поиска моделей с малой длиной описания.

К сожалению, даже многие из общих допущений, перечисленных выше, затруднительно использовать в алгоритмическом подходе к построению моделей. В то же время практические методы привлекают еще более сильные ограничения. Вполне возможно, что при создании универсальной системы машинного обучения многие подобные частные методы (а также прочие методы, разрабатываемые в рамках синтаксического и логического подходов) необходимо будет закладывать априорно, однако как при этом не допустить внесения принципиальных ограничений на возможности системы к обучению, остается пока еще неясным.

Нецелесообразность развития систем машинного обучения «с нуля» еще более отчетливо видна, если в качестве исходных данных выступают сенсорные данные, полученные из реального мира с помощью сенсоров таких модальностей, как, например, зрение или слух. Изучение проблемы интерпретации сенсорной информации также может помочь получить некоторые подсказки, касающиеся того, как в биологических системах происходит эффективное построение моделей на основе больших потоков данных. В то же время и области компьютерного зрения и распознавания речи могут выиграть от использования принципа МДО.

Глава 3
МАШИННОЕ ВОСПРИЯТИЕ

3.1. ПРЕДСТАВЛЕНИЕ ИЗОБРАЖЕНИЙ В СИСТЕМАХ КОМПЬЮТЕРНОГО ЗРЕНИЯ

3.1.1. Машинное восприятие в контексте искусственного интеллекта

Долгое время стереотипным бытовым представлением о воплощении искусственного интеллекта были антропоморфные роботы. Естественно, такие роботы должны были быть наделены органами чувств, сходными с теми, которые есть у человека. Очувствление роботов, которое бы позволило придать им адаптационные способности, нашло широкое практическое применение и в автоматизации производства, и в задачах, требующих проведения работ в экстремальных условиях [203]. Но, чтобы наделить робота способностью видеть или слышать, недостаточно лишь снабдить его видеокамерой и микрофоном: поступающая от них информация будет бесполезна без алгоритмов ее интерпретации. Подобные алгоритмы с соответствующими сенсорными датчиками составляют систему восприятия. Целью системы восприятия, биологической или машинной, является построение модели реального мира и использование этой модели для взаимодействия с миром [204].

Модель, построенная системой восприятия и представленная в символьной форме, может выступать в качестве входа в программы искусственного интеллекта. В связи с этим среди специалистов по ИИ достаточно широко распространено мнение, согласно которому восприятие может быть представлено отдельным вспомогательным модулем, разработка которого мало связана с созданием собственно интеллекта [205]. Более того, алгоритмы интерпретации сенсорной информации разных модальностей изучаются в рамках различных научных направлений и опираются на весьма специфические результаты исследований. Области их практического применения также мало связаны между собой и выходят далеко за рамки робототехники. Казалось бы, все это подтверждает целесообразность изучения вопросов восприятия разных типов как замкнутых проблем, не

зависимых друг от друга и от вопросов искусственного интеллекта.

Хотя такое разделение задач и было необходимо в начале исследований, в настоящее время все больше возрастает уверенность в том, что полное разделение восприятия и мышления — это слишком грубое приближение. Эти процессы могут заметно выиграть от активного взаимодействия друг с другом [205]. Влияние информации, полученной от сенсора одного типа, на процесс интерпретации информации, полученной от сенсора другого типа, также может использоваться для достижения взаимного улучшения результатов интерпретации сенсорной информации разных модальностей [206]. Данная точка зрения подтверждается многими нейрофизиологическими и психофизическими данными [207, с. 104–109]. Это говорит о целесообразности использования контекстного и высокоуровневого знания в процессе восприятия (рис. 3.1).

В то же время и сам процесс мышления тесно связан с восприятием. Человек оперирует понятиями, основанными на сенсорной информации (причем это могут быть не только внешние сенсоры типа зрения, слуха, обоняния, осязания, но также и «внутренние сенсоры», интероцепторы, выполняющие различную диагностику организма, а также сигналы от вестибулярного аппарата). Помимо этого человек обладает воображением — «сложной подсистемой, осуществляющей визуальные рассуждения посредством символьных манипуляций с пространственными представлениями» [205]. Широко известно, что человек использует воображение не только, скажем, для планировки комнаты, но и для решения многих других (в том числе и научных) проблем. Яркой иллюстрацией тому может служить такое направление, как когнитивная графика, в котором понятия (и взаимосвязи между ними) некоторой предметной области представляются в удобной для человека форме, а именно: в форме зрительных образов, что позволяет ему видеть

Рис. 3.1. Иллюстрация эффекта перцептивной готовности как влияния высокоуровневого знания на процесс интерпретации. Если бы на изображении были представлены только столбец букв или только строка цифр, то интерпретация символа «0» произошла бы неосознанно, причем в двух случаях интерпретация была бы разной (другие примеры проявления этого эффекта можно найти по адресу http://www.psy.msu.ru/illusion/set.html на сайте факультета психологии МГУ)

(в прямом смысле) законы данной предметной области, которые бы он не смог увидеть (в переносном смысле), используя некоторое традиционное символьное представление.

Зададимся, например, вопросом: смог бы человек играть в шахматы, если бы вместо доски ему сообщался набор чисел, описывающих абстрактную игровую ситуацию? А смог бы он *научиться* играть в шахматы, если бы правила описывались как допустимые манипуляции над символьными строками? Очевидно, в процессе игры человек активно использует воображение, основой которого является зрительное восприятие. Компьютерные программы, играющие в шахматы, этого не делают. В то же время они и не учились играть в шахматы по описанию правил, вместо этого они были сконструированы на основе знаний экспертов. В адрес таких программ часто также слышны упреки, что они побеждают за счет «грубой силы», а не интеллекта.

В другой игре — «го» — число вариантов существенно больше, чем в шахматах, и подход на основе «грубой силы» там не помогает. Компьютерным программам, играющим в «го», пока еще далеко до тех успехов, которые продемонстрировали шахматные программы. Справедливости ради необходимо сказать, что это, возможно, связано с тем, что создание шахматных программ, способных победить чемпиона мира среди людей, было в свое время вызовом, брошенным специалистам по искусственному интеллекту, так что на решение этой проблемы были потрачены существенно более значительные усилия, чем на программы, играющие в «го».

Наиболее успешные шахматные программы создавались без оглядки на вопросы машинного восприятия, и «думают» они в процессе игры совсем не так, как человек, которого они, однако, способны превзойти. Шахматные программы, при разработке которых ставилась цель моделирования мышления человека (что включало, в частности, распознавание образов), а не наиболее эффективной игры, оказывались слабее [208]. Возможно, и при создании искусственного интеллекта как такового можно полностью избежать необходимости разработки системы машинного восприятия (даже в качестве отдельного модуля), если рассматривать ИИ, который, скажем, получал бы в качестве входной информации только текст. Но будет ли способен такой интеллект понимать те концепты, которыми оперирует? Можно ли корректно использовать термины, для которых имеют-

ся лишь формальные их определения через другие такие же термины? Человек избегает пользоваться такими терминами. Чтобы понять смысл какого-либо понятия, он, как правило, стремится найти аналогии с *наглядными* понятиями, максимально приближенными к сенсорному опыту. Для этого, например, потенциальные функции представляют в виде рельефа местности, покрытой холмами и расщелинами, а элементарные частицы — в виде маленьких шариков, несмотря на то что их оптического изображения быть просто не может. Возможно, из-за этого человеком так тяжело воспринимаются концепции или теории, противоречащие сенсорному опыту, такие как общая теория относительности или теория большого взрыва. Для достижения их понимания приводятся аналогии, например, расширение Вселенной представляется как надувание шарика, на поверхности которого нарисованы галактики.

Однако остается неясным, является ли использование чувственного опыта в качестве опоры при формировании понятий (в том числе и абстрактных) принципиально необходимым для интеллекта или же, наоборот, это является существенной слабостью человеческого разума, которую нужно преодолеть при создании ИИ. Чтобы не заниматься спекуляциями, мы не будем пытаться ответить на подобные вопросы. Тем не менее следует отметить, что путь наделения машины восприятием — это один из возможных подходов к ИИ, в рамках которого могут быть решены некоторые проблемы (в дополнение к упоминавшейся выше проблеме смысла), свойственные другим подходам.

Например, при создании экспертных систем центральной проблемой является построение базы знаний, которое, по сути, осуществляется вручную: инженер по знаниям опрашивает экспертов в конкретной предметной области с целью выявления базовых концепций и отношений между ними, на основе чего разрабатывает структуру организации знаний [209, с. 218]. И лишь более поздние этапы допускают частичную автоматизацию. Например, формулирование правил может осуществляться по описанию предполагаемого процесса решения экспертами некоторых типичных задач. Экспертные системы предназначаются для работы в достаточно узких предметных областях. Человек же оперирует огромным объемом знаний. Если передача опыта специалистов экспертным системам является весьма трудоемкой задачей, то можно представить, сколь сложно было бы в рам-

ках этого подхода заложить в компьютер знания, сопоставимые по объему со всеми знаниями одного человека или, тем более, человечества. Но даже если бы это и удалось сделать, то такая система не смогла бы самостоятельно научиться чему-то принципиально новому без участия человека. Автоматизация процесса приобретения знаний становится все более актуальной [209, с. 450]. Наделение машинной системы восприятием позволяет, по крайней мере, решить проблему источника информации, на основе которой должно осуществляться обучение.

Шире эта проблема может быть сформулирована как необходимость помещения развивающегося интеллекта в достаточно сложную среду. Физический мир и человеческий социум являются естественным выбором такой среды. Более того, некоторые исследователи настаивают на том, что это единственный выбор. Например, Брукс считает [74, с. 256], что интеллект, являясь продуктом взаимодействия некоторой системы со своим окружением, не может возникнуть в «невоплощенных» системах, таких как системы доказательства теорем или классические экспертные системы.

Не поддерживая столь категоричную точку зрения, мы тем не менее считаем, что заниматься проблемами машинного восприятия в контексте проблематики искусственного интеллекта весьма полезно. Кроме того, в данной области имеется множество практически важных приложений, таких как адаптивные роботы, системы голосового управления, беспилотные летательные аппараты и многие другие. Сами задачи, возникающие здесь, являются очень сложными как по объему данных, требующих быстрой интерпретации, так и по сложности используемых моделей, поэтому методы искусственного интеллекта находят здесь свое применение. И в то же время биологические системы восприятия изучены весьма детально, поэтому многому можно поучиться у природы в плане решения сложных задач.

Однако главная причина, по которой мы рассматриваем проблемы машинного восприятия, заключается в том, что процесс интерпретации сенсорной информации можно трактовать как индуктивный вывод на основе сенсорных данных, а значит, здесь может быть использован принцип минимальной длины описания. Поскольку алгоритмы восприятия для любой сенсорной модальности осуществляют индуктивный вывод, то между ними должно быть много

общего, но в то же время для эффективности они должны быть специализированными. В чем именно это выражается, попытаемся выяснить на конкретных примерах.

3.1.2. Интерпретация изображений как центральная проблема компьютерного зрения

Область компьютерного зрения включает большой комплекс задач по автоматическому анализу изображений, таких как распознавание целей, лиц или рукописного текста, извлечение изображений из баз данных по их текстовому описанию или наброску от руки, обзор производственных помещений, автоматический контроль качества выпускаемой продукции, различные биомедицинские приложения и многое другое. Для решения большинства этих задач разрабатываются специализированные системы анализа изображений, причем получающиеся решения часто оказываются применимы лишь в том окружении, для которого они конструировались. К примеру, система обнаружения пешеходов на проезжей части, разработанная без учета возможности того, что может идти дождь или снег, при ухудшении погоды, вероятно, будет давать некорректный результат, а система анализа аэрокосмических изображений Земли, разработанная для работы с равнинной местностью, скорее всего, не сможет быть применена для горной местности.

Зрительная система человека гораздо более гибкая. Она может решать многие задачи, с которыми сталкивается впервые, причем делает это невероятно быстро, без заметных усилий со стороны человека, и часто лучше, чем узкоспециализированные системы компьютерного зрения [210, с. 11]. Как оказалось, создание системы компьютерного зрения с подобными возможностями — чрезвычайно трудная задача, причем проблема состоит не столько в вычислительных ресурсах, сколько в разработке надлежащих алгоритмов.

В систему компьютерного зрения изображение поступает в виде массива данных. При незначительной смене ракурса или освещения могут одновременно измениться *все* элементы этого массива (рис. 3.2, *а*, *б*). Пусть, например, необходимо обнаружить различия, присутствующие на паре изображений. Если эти изображения идентичны, за исключением нескольких деталей, то задача не представляет сложности: достаточно лишь «вычесть» одно изображение

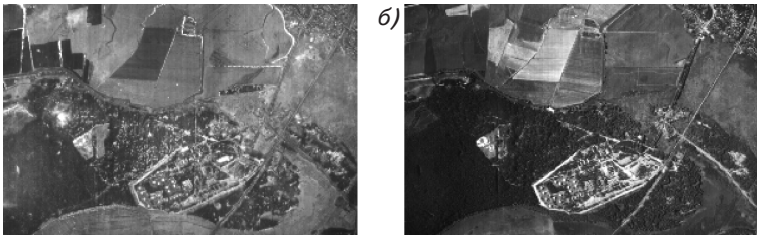

Рис. 3.2. Пример двух разносезонных аэрокосмических изображений одного и того же участка местности. Человек способен отождествить различные точки на этих изображениях, несмотря на то что интенсивности отдельных пикселей, а также соотношения между ними различны

из другого. Но что если изображенная сцена осталась прежней, а лишь сместился источник освещения? В дополнение к тому, что изменится освещенность разных участков сцены, также сменят свое положение тени и блики, однако все эти изменения будут согласованы между собой. Человек их воспримет правильно: как изменение положения источника освещения при неизменном содержании сцены. Но как сделать, чтобы и для компьютера два таких изображения не «выглядели» полностью различными?

Возьмем другую задачу: распознавание объекта по его изображению. Как, например, суметь автоматически различить крысу и мышь (примерно одинакового цвета и размера), которые сняты с одного ракурса, и при этом узнать мышь в разных ракурсах?

Очевидно, что проблема кроется в исходном представлении изображений, не отражающем в явном виде содержание той части физического мира, которая присутствует на изображении. Проблема построения более подходящего представления, которое бы позволяло восстанавливать информацию о физическом мире, включая геометрические и физические свойства видимых поверхностей, — это общая проблема для большинства задач компьютерного зрения. Доминирующая в области компьютерного зрения реконструкционная парадигма рассматривает решение этой проблемы как первичную цель, достижение которой необходимо для создания системы машинного зрения общего назначения.

Заметим, что такая реконструкционная парадигма хорошо согласуется с целями, которые должна преследовать система машинного восприятия (см. п. 3.1.1). Тем не менее

иногда этот подход подвергается критике за непрактичность и утверждается, что сама эта первичная цель некорректна и ее необходимо изменить [211]. Действительно, в последнее время в области машинного зрения отмечается определенный кризис, связанный с тем, что из-за несовершенства теоретических подходов и применения большого числа эвристических и слабо проверенных методов до сих пор не удалось построить системы технического зрения, близкие по своим возможностям к зрительному аппарату животных и человека [204, 212, 213, с. 12]. В качестве альтернативы предлагается рассматривать лишь вопросы построения систем машинного зрения, предназначенных для решения конкретных практических задач [211]. Сторонники такого целевого подхода настаивают также на том, что и зрительная система человека представляет собой не более чем комплекс специализированных устройств по решению частных задач.

Тем не менее рядом авторов реконструкционный подход отстаивается не только как жизнеспособный, но и как принципиально необходимый в рамках науки об изображениях — иконики [205]. Ограничение же исследований узкими прикладными задачами рассматривается как игнорирование проблемы, а не ее решение. Однако для преодоления кризиса реконструкционного подхода все еще требуется разработка теоретически обоснованных методов анализа изображений, и принцип минимальной длины описания может здесь в определенной мере быть полезен. Действительно, построение описания изображения в рамках некоторого представления (или *интерпретация* изображения) — это задача индуктивного вывода, в котором длина описания является строгим критерием оптимальности модели. Само же представление задает распределение априорных вероятностей в пространстве моделей. Часто при разработке алгоритмов интерпретации изображений эти составляющие индуктивного вывода не адресуются непосредственно, а задаются косвенно через введение различных эвристик. Хотя такой подход и позволяет решать частные практические задачи, он существенно затрудняет достижение основной цели реконструкционного подхода.

Следует особо отметить, что выбор оптимального представления изображений представляет собой эмпирическую задачу, поскольку представление должно быть согласовано со свойствами физического мира. Эта задача не может быть решена на основе абстрактных теоретических рассуждений,

поэтому целесообразно обратиться к тому эмпирическому опыту, который накоплен исследователями в области компьютерного зрения. В связи с этим, прежде чем переходить к описанию работ по применению теоретико-информационных методов в целях интерпретации изображений, кратко рассмотрим существующие на данный момент способы представления изображений в системах компьютерного зрения.

Обычно при классификации методов интерпретации по типам привлекаемых представлений различают три уровня абстракции [214]: 1) нижний (или пиксельный); 2) средний (или промежуточный символьный); 3) верхний (или семантический). Каждый из этих уровней включает большое разнообразие различных представлений и может быть дальше уточнен. Такое деление также не вполне строгое: некоторые представления могут быть отнесены к разным уровням. В частности, здесь мы отделяем пиксельный уровень от низкоуровневых математических представлений.

3.1.3. Представления в виде необработанных данных: пиксельный уровень

В задачах автоматического анализа изображений в качестве исходного представления, из которого осуществляется отображение в некоторое конечное представление, обычно выступает представление изображения в виде массива «сырых» данных — набора результатов измерений, выполненных для некоторой сцены. Под сценой обычно понимается набор физических объектов или некоторый фрагмент реального мира [210, с. 11]. Чаще всего в качестве измеряемой физической величины выступает интенсивность электромагнитного излучения в заданном направлении (из некоторого спектрального диапазона, например, видимого, ИК-, радио- или рентгеновского). Однако измеряемая характеристика может быть и совершенно другой: например, цифровая карта высот, гидроакустические данные или распределение плотности некоторых элементарных частиц в зависимости от направления также могут рассматриваться в качестве изображений.

Мельчайшим элементом изображения при таком начальном представлении является *пиксель* (pixel, picture element), содержащий результат единичного измерения данной фи-

зической величины. В связи с этим такие представления также называют представлениями на уровне пикселей [215].

Пиксели, образующие изображение, с соответствующими им значениями интенсивностей организованы в массив, размерность которого определяется природой данных. Упорядочение пикселей в этом массиве соответствует пространственной организации сцены. Для многих задач массив пикселей является двумерным, но есть приложения (например, медицинские), в которых размерность массива может быть больше двух. В случае объемных изображений их элементы принято называть *вокселями* (voxel, volume pixel). Последовательность изображений может трактоваться как изображение большей размерности (на уровне пиксельных представлений время ничем не отличается от дополнительной пространственной координаты).

Основной проблемой, связанной с представлениями на пиксельном уровне, является проблема эффективности этих представлений в целях их хранения и передачи. Это особенно актуально для объемных изображений, для которых существует несколько альтернативных способов представления [214], таких, как, например, трехмерный массив вокселей, стереопара, проекция на гауссову сферу и т. д. Однако разработка таких представлений вызвана необходимостью решения чисто технических проблем, не связанных с задачами анализа изображений, поэтому здесь мы на них останавливаться не будем.

Представления на пиксельном уровне являются исходными для любых приложений интерпретации изображений с помощью цифровых вычислительных машин. Именно потому, что такое представление является общим для различных задач компьютерной обработки изображений, часто говорят, что «изображения — это массивы пикселей» [214]. Это может вызвать определенную терминологическую путаницу, например, при обсуждении особенностей зрительного восприятия животных и человека. В связи с этим здесь будет отделяться изображение (содержание) от его конкретного представления (формы).

Представления на пиксельном уровне содержат в себе всю имеющуюся информацию о наблюдаемой сцене, но в форме, неудобной для автоматического анализа [210, с. 12]. Это и вызывает необходимость привлечения других представлений изображений с целью извлечения содержащейся в них релевантной информации.

3.1.4. Низкоуровневые представления: математические модели изображений

Представления на пиксельном уровне не говорят нам о том, как следует выполнять даже такие простейшие операции над изображениями, как, например, масштабирование или вращение. В связи с этим возникает естественное желание представить изображение как элемент некоторого математического пространства, чтобы воспользоваться уже введенными на нем операциями. Часто отображение из пиксельного представления в выбранное математическое пространство является взаимно однозначным и непосредственно выражается через исходные значения интенсивностей, а результаты математических операций над изображениями снова представляются в виде массива пикселей. Из-за этого интерпретации изображений в качестве массивов пикселей и в качестве математических объектов часто смешивают, объединяя их в один класс [214, 215], а сам процесс такой переработки изображений выделяют в отдельный тип задач — обработка изображений [216, с. 140]. Хотя математические модели изображений можно условно отнести к пиксельному уровню, между ними существуют принципиальные отличия.

Интерпретация изображения в качестве элемента математического пространства позволяет распространить формальные операции, введенные на этом пространстве, и на изображения. Это дает обширный набор строгих внутренне непротиворечивых средств анализа и преобразования изображений. К примеру, если выбранное пространство является метрическим, то появляется возможность формального определения «расстояния» (т. е. степени сходства) между изображениями.

К сожалению, существующие на данный момент строгие математические модели изображений являются достаточно низкоуровневыми и имеют ограниченную область применения. Уместно задать вопрос [217], адекватна ли трактовка изображения как некоторой непрерывной функции?

Три основных класса математических моделей изображений включают: представления в виде случайных полей, функциональные представления и вейвлет-представления [217]. Вейвлет-представления можно рассматривать как особого вида функциональные представления, которые призваны отразить иерархическую природу изображений, поэтому более подробно о них см. в п. 3.1.7.

Функциональные модели. При использовании функциональных моделей изображение интерпретируется как функция из некоторого (например, гильбертова) пространства: $f : G \to V, G \subseteq R^n, V \subseteq R^m$, где G — область определения функции; V — область ее значений; n — размерность изображения, обычно $n = 2$; m — размерность вектора физических величин, измеренных для каждой точки, например, для полутоновых изображений $m = 1$, а для цветных *RGB*-изображений $m = 3$.

Функциональное представление является базовым для проведения таких операций над изображениями, как пространственное преобразование изображений, преобразования яркости, фильтрация и др. Пространственное преобразование изображения (масштабирование, вращение и т. д.) осуществляется с помощью смены системы координат в области G:

$$f_2(\vec{x}') = f(g(\vec{x})), g : G \to G_2, \qquad (3.1)$$

где g — функция, ставящая в соответствие каждой точке из области определения G исходного изображения f точку в области определения G_2 преобразованного изображения f_2 (рис. 3.3, *а*, *б*).

Преобразование яркости (например, изменение яркости или контраста для всего изображения) или преобразование

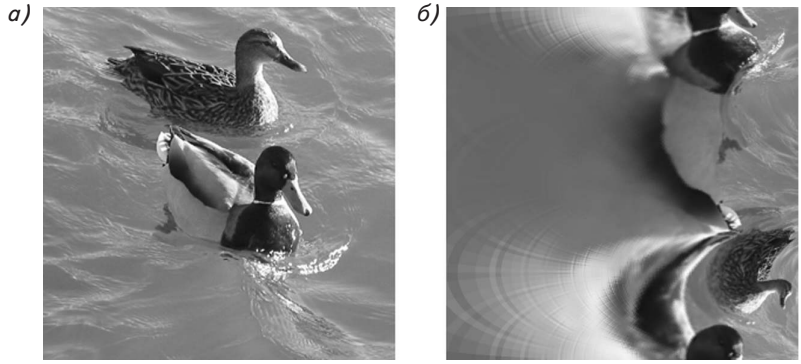

Рис. 3.3. Использование математического представления для перевода изображения (*а*) в полярно-логарифмические координаты (*б*). В отличие от обычных полярных координат, здесь по оси абсцисс откладывается не радиус, а его логарифм. Изменение масштаба и вращение исходного изображения приводят к его сдвигу в полярно-логарифмических координатах

цвета (если V — цветовое пространство) может быть представлено следующим образом:

$$f_2(\vec{x}) = h(f(\vec{x})), h : V \to V_2, \tag{3.2}$$

где h — функция, зависящая только от значения функции f в данной точке и ставящая в соответствие этому значению новое значение, возможно, из другой области V_2. Изменению яркости соответствует функция h.

Фильтрация изображений является более широким понятием, чем приведенные выше преобразования двух типов, так как в общем случае зависит от всего содержимого изображения. Если $\Phi = \{f \mid f : G \to V\}$ — функциональное пространство, к которому принадлежит данное изображение, а Φ_2 — пространство, к которому принадлежит обработанное изображение, то фильтрация изображения — это произвольное отображение из пространства Φ в Φ_2: $T : \Phi \to \Phi_2$.

Наиболее часто используется линейная фильтрация, которая может быть представлена в виде операции свертки с некоторым ядром φ:

$$f_2 = T(f) = f \otimes \varphi, \tag{3.3}$$

где $(f \otimes \varphi)(\vec{x}) = \int\limits_G f(\vec{y})\varphi(\vec{x} - \vec{y})d\vec{y}$, что связано с существованием эффективных схем вычисления операции свертки (не только на цифровых вычислительных машинах, но также и с помощью методов когерентной оптики) и с прозрачностью ее результатов. К примеру, с помощью линейной фильтрации представляются такие операции, как сглаживание изображения или его дифференцирование.

Еще одной важной возможностью, предоставляемой функциональными моделями изображений, является смена базиса в исходном функциональном пространстве Ф. Частным, но очень важным, примером является преобразование Фурье:

$$F(\vec{\omega}) = \int\limits_G f(\vec{x})\mathbf{e}^{-i(\vec{\omega}, \vec{x})}d\vec{x}. \tag{3.4}$$

Функциональные и другие математические представления позволяют формально ввести понятие инварианта. Желание работать с представлениями, инвариантными к некоторому типу преобразований, совершенно естественно

проистекает из того факта, что сцена с одним и тем же содержанием может совершенно по-разному (при попиксельном представлении) выглядеть при различных условиях наблюдения. Примером теоретического подхода к анализу сцен, привлекающего понятие инварианта, является подход на основе групп Ли на плоскости [218].

Поскольку каждое изображение исходно представляется в виде набора точек (конечного массива данных), а число функций данного класса, как правило, бесконечно, то необходимо вводить некоторые ограничения, позволяющие из множества всех подходящих функций выбрать лучшую.

Часто это достигается тем, что рассматривается ограниченный класс функций, так что по набору точек, образующих данное изображение, выбор функции однозначен. Однако в ряде случаев удобнее накладывать интегральные ограничения из некоторых модельных соображений. Эти ограничения часто представляются в виде минимизации некоторого функционала, и задача построения модели изображения превращается в задачу вариационного исчисления.

Наиболее простыми функционалами, подвергающимися минимизации, являются следующие [219]:

$$L_2(f) = \int_G |f(x)|^2 \, dx; \quad L_1(|\nabla f|) = \int_G |\nabla f(x)| \, dx;$$

$$L_2(|\nabla f|) = \int_G |\nabla f(x)|^2 \, dx. \qquad (3.5)$$

Последняя норма является наиболее широко используемой (это так называемая регуляризация Тихонова [220]).

Выбор минимизируемого функционала (также называемого мерой сложности изображения) часто осуществляется на основе исследований, посвященных определению статистических свойств естественных изображений и особенностей зрительного восприятия животных и человека [39, 221]. Этот подход также применяется при привлечении статистических моделей изображений: если изображение представляется в виде случайной функции, то выбор ее наиболее вероятной реализации осуществляется посредством минимизации некоторого функционала при соблюдении ограничений, налагаемых исходными данными.

Вероятностные модели. В общем случае вероятностные модели являются более общими, чем функциональные, так как представляют изображения некоторыми случайными

функциями, по отношению к которым регулярные функции являются весьма частным подклассом. Однако зачастую регулярная составляющая в вероятностных моделях гораздо проще, чем в функциональных, а сами модели призваны описать статистические свойства изображений, не отражая при этом их пространственной структуры [222]. Стремление включить в стохастическую модель информацию о пространственных положениях элементов приводит, как правило, к использованию различных моделей случайных полей (см., например, [223–225]).

Стохастические модели изображений берут свое начало из теории цифровой обработки сигналов. Исходно они предназначались для решения задач подавления шума, помехоустойчивого кодирования и сжатия, в частности, в целях передачи изображений по каналам связи (см., например, [226, с. 439–440, 652–657, 745–766], а также приведенные там ссылки). Сейчас же стохастические модели широко используются для описания текстуры [227, 228], реставрации изображений [217, 229], для декомпозиции изображений на области [229, 230] и т. д.

Вероятностные методы также привлекаются для описания изображений не на пиксельном уровне, а на более высоких уровнях. Например, существуют вероятностные модели для контуров на изображениях [213, с. 31–33], для структурных описаний [121, с. 142–216] и т. д. Но их, скорее, следует отнести к представлениям соответствующих уровней.

Помимо стохастических моделей, описывающих какой-то один из аспектов изображений, в статистическом анализе изображений ставится и общая проблема, которая может быть сформулирована следующим образом [231]: можно ли найти единую стохастическую модель для изображений?

Ответ на данный вопрос находится в рамках порождающих стохастических моделей изображений. В таких моделях предполагается, что есть набор скрытых переменных $\vec{\chi} = (\chi_1, \ldots, \chi_N)$ с заданной плотностью распределения априорных вероятностей $p(\vec{\chi})$, а также функция построения изображения, задающая распределение вероятностей по изображениям при данных скрытых переменных: $p(f \mid \vec{\chi})$, где f — изображение.

Моделью конкретного изображения будет являться набор значений этих скрытых переменных, апостериорные вероятности которых можно оценить по правилу Байеса:

$$p(\vec{\chi} \mid f) \sim p(f \mid \vec{\chi}) p(\vec{\chi}). \qquad (3.6)$$

Вектор $\vec{\chi}$ выступает в качестве описания изображения. Правило Байеса будет выполняться и для описаний, сформированных в рамках любого другого представления, что, на первый взгляд, делает статистический подход универсальным средством. Однако здесь возникают две принципиальные трудности. Во-первых, возникает уже знакомая проблема задания априорных вероятностей $p(\vec{\chi})$. Во-вторых, распределение $p(f \mid \vec{\chi})$ должно задаваться в явном виде либо должна существовать возможность его вычислить, что для описаний, представляющих интерес, неосуществимо. Например, если дано словесное описание изображения, то как определить вероятность того, что некоторое изображение соответствует этому описанию? Такие вероятности нельзя вывести теоретически и нельзя оценить по примерам реальных изображений. Из-за этого описания $\vec{\chi}$ в стохастических моделях изображений довольно простые.

В качестве примера приведем простейшую модель [219], в которой интенсивности отдельных пикселей распределены по одному и тому же нормальному закону, характеризующемуся двумя параметрами: средним a и дисперсией σ:

$$p\left(f \mid (a, \sigma)\right) = \prod_{\vec{x} \in G} \frac{1}{\sqrt{2\pi\sigma^2}} \mathbf{e}^{-\left(\frac{f(\vec{x})-a}{\sqrt{2}\sigma}\right)^2}. \qquad (3.7)$$

Эта модель, очевидно, очень упрощенная, так как она не учитывает пространственные зависимости, которые как раз и характеризуют изображения. Однако в силу своей простоты она является удобной стартовой точкой при построении более сложных статистических моделей [219]. Более сложные модели могут включать учет статистик высоких порядков, ограничения на пространственный спектр или некоторые общие предположения о статистических свойствах изображений, такие как инвариантность к масштабу или стационарность (см., например, [231]).

Для выбора распределения $p(f \mid \vec{\chi})$, необходимого при построении модели некоторого изображения, важным является учет информации о статистиках естественных сцен и об особенностях в строении систем зрения у животных и человека. Изучение этих вопросов составляет две отдельные,

но тесно связанные ветви исследований в области статистического моделирования изображений, и им посвящена обширная литература (см., например, [232–234] и [35–42] соответственно). Наличие четкой связи между ответами нейронов в естественных зрительных системах и статистиками естественных сцен показывает важность учета этой информации при построении систем машинного зрения.

Описанный выше подход неявно предполагает [231], что существует некоторое распределение безусловных вероятностей по изображениям:

$$p(f) = \int_{\vec{\chi}} p(f \mid \vec{\chi}) d\vec{\chi}. \tag{3.8}$$

Очевидно, значения $p(f)$ будут сильно различаться для конкретных приложений (для очень ограниченных задач возможно даже явно установить это распределение). Поэтому, если и существует некая универсальная стохастическая модель изображений, она должна опираться лишь на самые общие свойства физического мира, не обращаясь к информации о конкретных объектах (и, естественно, ее использование в частных приложениях будет менее эффективным).

Скрытые переменные должны каким-то образом описывать содержание сцены. Если это содержание достаточно произвольно, то построить стохастическую модель изображений сцены, которая могла бы быть применима на практике, оказывается проблематичным. Это относится и к математическим представлениям вообще: в них приходится использовать определенные упрощения, так что они оказываются недостаточно выразительными для описания содержания сцен.

Таким образом, низкоуровневые методы позволяют решить множество проблем обработки изображений, однако существует ряд задач анализа изображений, в которых их применение не приводит к удовлетворительному результату. В качестве примера таких задач можно привести совмещение разносезонных изображений (см., например, рис. 3.2) или изображений, полученных при использовании различных типов сенсоров, выявление изменений, некоторые задачи распознавания объектов сложной формы и т. д. В этих задачах низкоуровневые представления используются как отправная точка для построения промежуточных символьных представлений.

3.1.5. Средний уровень: структурные методы

В качестве конечной цели интерпретации изображений часто рассматривается присвоение группам пикселей меток, соответствующих некоторым объектам реального мира, таким как дом, дерево, дорога и т. д. Однако если бы существовала прямая взаимосвязь между пиксельным и семантическим содержанием изображения, то эта задача была бы давно решена, поэтому общепринятым стало предположение, что в процессе интерпретации изображений должны использоваться некоторые промежуточные представления [214].

В качестве основных причин различия внешнего вида одного и того же объекта на разных изображениях можно назвать смену ракурса (пространственное преобразование), изменение освещения (преобразование интенсивности), смену типа сенсора, собственную изменчивость объекта (например, сезонные изменения на аэрокосмических фотографиях). Следовательно, для назначения меток необходимо построить описания изображений, инвариантные перечисленным преобразованиям. К сожалению, математические модели позволяют добиться инвариантности только по отношению к весьма ограниченным классам преобразований. Изменчивость других типов, вызванная, например, сменой сенсора или собственными изменениями объекта, оказывается трудноформализуемой, так как сильно зависит от свойств объектов наблюдения.

Поскольку различные изображения соответствуют одному и тому же объекту, получению интерпретации изображения, инвариантной некоторым преобразованиям, также соответствует и уменьшение размерности данных. Одна из проблем заключается в том, чтобы построить инвариантные представления, потеряв при этом как можно меньше полезной информации.

Представления, решающие в той или иной степени эту задачу, принято выделять в отдельный класс представлений среднего уровня. Поскольку результатом их применения является неизобразительная информация, их также называют промежуточными символьными представлениями. Они обращаются к структуре изображения, т. е. к взаимосвязям между различными пикселями, поэтому такие представления часто называют структурными.

Взаимосвязи между группами пикселей подразумевают привлечение статистик высоких порядков (их сложно оце-

нить по изображениям) либо негладких функций (для них сложно определить аналитические модели), что и вызывает сложности при использовании строгих математических моделей для описания структуры изображений.

Вместо использования абстрактных представлений, опирающихся на понятие инвариантности, Д. Марр предположил, что основным предназначением зрения является реконструкция форм и местоположений объектов по изображениям [235, с. 51], т. е. восстановление их физических характеристик. Таким образом, согласно этому подходу, интересующее нас промежуточное представление должно содержать описание реальных поверхностей, присутствующих в физическом пространстве. Для получения этого представления им было предложено использовать ряд вспомогательных представлений, таких как необработанный первоначальный эскиз и 2,5-мерный эскиз, содержащих «характерные объекты», соответствующие реальным физическим особенностям наблюдаемых поверхностей [235, с. 57].

Необходимость нескольких представлений промежуточного уровня также часто иллюстрируют, опираясь на аналогию анализа текста (или речи) и изображений [214]. Так же как буквы группируются в слова, слова в предложения, а предложения в текст, пиксели следует группировать в знаки промежуточного представления, которые уже, в свою очередь, образуют конкретные объекты. Подобная аналогия, однако, не говорит, сколько и каких должно быть промежуточных уровней. Мы рассмотрим три таких уровня: контурные представления, представления в виде непроизводных структурных элементов и представления в виде составных структурных элементов.

Контурные представления. Под контуром обычно понимается местоположение локального изменения или резкого перепада яркости на изображении [236]. Контуры служат простейшими признаками изображений, используемыми в целях анализа изображений для выделения границ объектов, и являются основой для построения структурных элементов. Хотя результат выделения контуров еще сложно назвать структурным описанием изображения, его также нельзя отнести и к низкоуровневым представлениям. Процедуры построения контурных описаний изображений можно разделить на глобальные и локальные.

Для глобальных процедур характерно разбиение изображения на однородные области (сегментация), на основе ко-

торых и строятся контуры как границы этих областей [237]. Возможны также и другие процедуры получения представлений, аналогичных по смыслу контурным, например, преобразование срединной оси. Хотя такие представления весьма различны, они содержат идентичную информацию. Однако какие-либо из них могут быть менее удобными. Так, например, срединная ось обладает повышенной чувствительностью к малым возмущениям формы соответствующей области (см., например, [235, с. 310]).

Локальные процедуры основываются либо на определении цепочек максимумов на градиентном поле (или пересечений нуля второй производной), либо на непосредственной аппроксимации яркостных переходов [236]. Поскольку выделение локальных яркостных переходов может тоже рассматриваться как вариант сегментации [213, гл. 6.1], то локальные и глобальные процедуры имеют много общего. Тем не менее локальные процедуры существенно проще в реализации и более правдоподобны с нейрофизиологической точки зрения, поэтому они более популярны.

Построение градиентного поля полутонового изображения осуществляется с помощью операторов, выполняющих дискретное дифференцирование. Классическими операторами, служащими для этой цели, являются операторы Робертса [238], Превитт [239], Собела [240, с. 291] (рис. 3.4, *а*, *б* и 3.5) и др. Этим трем операторам соответствуют маски, являющиеся дискретным аналогом ядер интеграла свертки:

$$H_1 = \begin{pmatrix} -1 & 0 \\ 0 & 1 \end{pmatrix}; \; H_2 = \begin{pmatrix} 0 & -1 \\ 1 & 0 \end{pmatrix}; \tag{3.9а}$$

$$H_1 = \begin{pmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{pmatrix}; \; H_2 = \begin{pmatrix} -1 & -1 & -1 \\ 0 & 0 & 0 \\ 1 & 1 & 1 \end{pmatrix}; \tag{3.9б}$$

$$H_1 = \begin{pmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{pmatrix}; \; H_2 = \begin{pmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{pmatrix}. \tag{3.9в}$$

Могут также использоваться и маски бóльших размеров. Поскольку таким операторам присуще свойство подавления шума, увеличение размера маски приводит к большей

а) *б)*

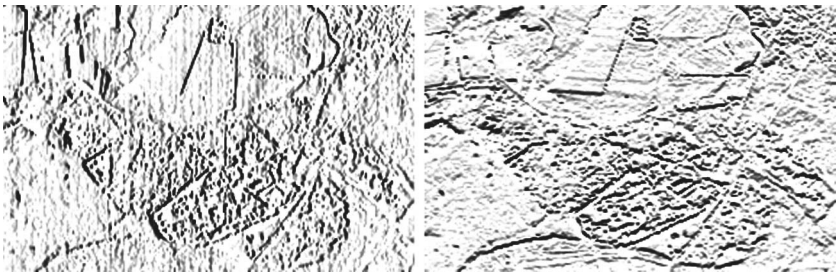

Рис. 3.4. Результат численного дифференцирования изображения, приведеного на рис. 3.2, *а*, с помощью оператора Собела. Два приведенных изображения (инвертированных для лучшего восприятия) соответствуют модулям частных производных по x и по y (*а*, *б*), полученных с помощью пары масок (3.9в). Маска накладывается в каждой точке изображения и указывает веса, с которыми необходимо суммировать интенсивности соседних пикселей для получения результирующей интенсивности

Рис. 3.5. Поле модуля градиента изображения, приведенного на рис. 3.2, *а*, также являющееся изображением. Для получения контурного описания изображения необходимо на градиентном поле отследить цепочки максимумов. Присутствие таких длинных цепочек является эмпирическим фактом

робастности процедур извлечения контуров, однако и увеличивает вероятность обнаружения ложных контуров [236]. Помимо получения более гладких контуров, более удобных для дальнейшего анализа, привлечение масок разных размеров обосновывается иерархичностью организации физического мира [235, с. 67]. Отметим, что и в зрительной системе существуют каналы, настроенные на различную пространственную частоту [241].

В компьютерном зрении сейчас популярны такие методы обнаружения контуров, как предложенные Канни [242], Деришем [243] (рис. 3.6, *а*, *б*), Линденбергом [244]. Предлагаются и другие методы (см., например, [245–247]).

Таким образом, построение градиентного поля изображения основывается на какой-либо математической модели. Затем осуществляется выделение контуров как цепочек максимумов на этом поле. Однако при этом возникает проблема обоснования такой формализации понятия контура

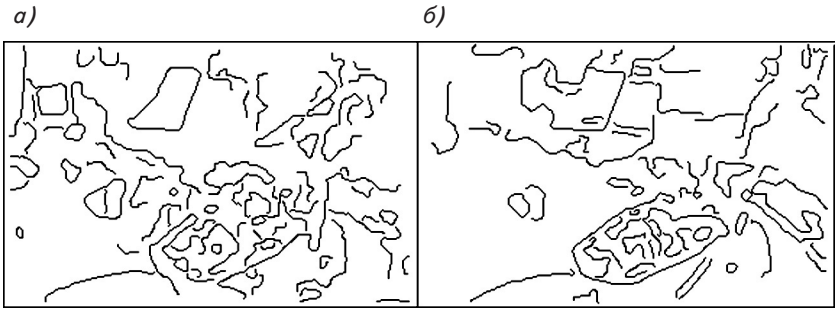

Рис. 3.6. Контуры, выделенные с помощью фильтрации Дериша на изображениях, представленных на рис. 3.2, *а*, *б* (соответственно). В то время как интенсивности отдельных пикселей на двух изображениях полностью различны, контуры имеют много общих элементов. Поскольку при выделении контуров часть информации теряется, человеку, конечно, предпочтительнее исходные изображения

(о существовании этой проблемы свидетельствует разнообразие подходов к обнаружению контуров).

Как правило, приводятся следующие аргументы в пользу привлечения контуров [213, с. 43]:

• контур является концентратором информации в изображении;

• контур полностью характеризует форму объектов на изображении;

• контуры объекта, в отличие от его остальных точек, устойчивы на изображениях, полученных в разное время, разных ракурсах, условиях погоды и при смене датчика;

• контурные точки составляют незначительную часть всех точек изображения, поэтому работа с ними позволяет резко сократить объем вычислений.

В несколько других терминах аргументация использования контуров заключается в том, что нарушения непрерывности в физическом пространстве (поверхности) порождают также и перепады яркости на изображении [235, с. 64]. А значит, последние могут быть использованы для получения информации о реальных поверхностях.

Хотя все эти аргументы убедительны, они все же относятся к тем контурам, которые бы хотелось получить в результате работы детектора контуров, а не к привлекаемым в них моделям изображений. Таким образом, процедура перехода от низкоуровневых представлений изображений к контурным представлениям требует дальнейшего теоретического обоснования.

Сами контурные представления также допускают использование строгой математической теории (например, контур может быть представлен в виде комплекснозначного сигнала [213]). В рамках таких теорий контурное описание может рассматриваться как конечный результат процедуры интерпретации изображения и использоваться для дальнейшего анализа.

Популярным является использование контуров в целях совмещения пары изображений или изображения с векторной моделью (например, картой местности или чертежом детали) на основе преобразования расстояний [248, 249], описание формы объектов или областей по их контурам, например, с помощью методов математической морфологии [210], для решения задачи стереоописса [250] и т. д. Контурные представления также служат основой для построения структурных описаний изображений.

Непроизводные структурные элементы. Если исходные изображения можно считать (как правило) двумерным сигналом, контуры на изображениях — одномерным сигналом, то следующие по уровню абстракции промежуточные символьные представления изображений являются безразмерными (или, условно, с размерностью, равной нулю). Однако подходы к получению таких представлений могут быть разными.

Один из подходов заключается в применении детекторов признаков на изображениях (см., например, [251]). Этот подход относится к широко распространенному классу методов анализа изображений на основе признаков. Выделение признаков производится путем применения некоторых локальных операторов (или шаблонов) к окрестностям каждой точки на изображении с последующим пороговым ограничением. В результате выделяется некоторое число признаковых точек (или точек интереса [252, 253]), в которых отклик был достаточно большим. Этим точкам назначаются метки, соответствующие искомому признаку. Если оператор имеет несколько свободных параметров, то значения параметров, при которых отклик в данной точке максимален, также приписываются найденному признаку в качестве его характеристик.

Простейшими выделяемыми признаками могут быть перепады яркости (края), которые в отличие от контуров представляются в виде совокупности отдельных точек (т. е. нульмерного многообразия). Другими признаками являются углы

(существуют модели для углов, находящихся на соединениях отрезков вида «Г», «Т», «Y» и «X» [254–256]), полосы, пятна [257]. Используются также детекторы признаков, предназначенные для узких предметных областей (например, для обнаружения лиц или распознавания рукописных текстов). Исследуется и противоположная задача, заключающаяся в автоматическом построении детекторов признаков на основе содержащейся в изображении информации [219].

Другой подход получения промежуточных символьных представлений использует понятие не признаков, а структурных элементов, которые строятся на основе контурных (или аналогичных) представлений. Получение структурных элементов по контурам требует решения задачи сегментации последних. Как отмечалось в п. 2.6.3, общая задача сегментации (в частности, самих изображений или их контуров) включает совместное решение двух проблем: группирования и регрессии. В зависимости от применяемой регрессионной модели в качестве структурных элементов могут выделяться отрезки прямых линий [258], дуги окружностей [259] или эллипсы [260] и т. д.

Оценивание параметров регрессионной модели обычно осуществляется на основе критерия среднеквадратичного отклонения (при использовании метода наименьших квадратов) и гораздо реже — на основе теоретико-информационных критериев. В последнем случае основной упор делается не на проблему регрессии, а на проблему группирования. Следует заметить, что игнорирование первой проблемы возможно лишь для простых моделей отдельных сегментов, например, для случая, когда в качестве структурных элементов выступают отрезки прямых линий. Задача группирования может решаться либо постепенным объединением меньших сегментов в более длинные участки контуров, либо поиском граничных точек сегментов, в качестве которых обычно выступают точки максимальной кривизны (рис. 3.7).

Причиной существования двух различных парадигм построения промежуточных символьных описаний является, вероятно, разработка двух формальных теорий распознавания образов — дискриминантной и синтаксической и попытки их применения к анализу и распознаванию изображений.

Действительно, при дискриминантном подходе исходное описание объектов задается в виде вектора признаков, а задача распознавания сводится к поиску решающих функций,

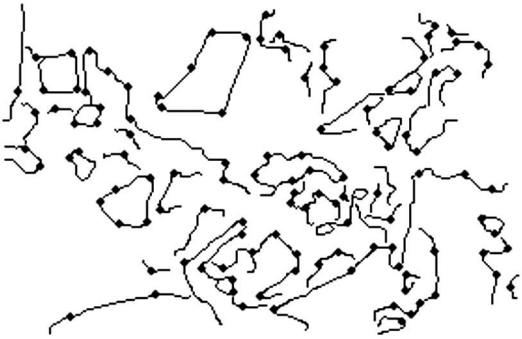

Рис. 3.7. Точки максимальной кривизны, выделенные на контурах, по отношению к контурам аналогичны самим контурам по отношению к изображениям. Соединив точки максимальной кривизны прямыми линиями, можно получить контурное описание, близкое к исходному

инвариантных по отношению к значениям внутриклассовых признаков. Интерпретация изображения как некоторого вектора признаков приводит к задаче поиска инвариантных признаков для их использования при классификации.

При синтаксическом подходе, обращающемся, как правило, к теории формальных грамматик, для описания объектов привлекаются непроизводные элементы, образующие некоторый алфавит. При этом значимыми для идентификации объекта являются не сами элементы, а те отношения, которые они образуют между собой. Через такие отношения описывается структура объекта. В связи с этим долгое время синтаксические методы отождествлялись со структурными методами вообще.

Однако оба крайних подхода к распознаванию образов оказались несостоятельными для задач интерпретации изображений: инвариантные признаки оказались слишком сложными, чтобы их можно было выделить общими методами в рамках дискриминантного подхода, а формальные грамматики — недостаточно гибкими для удобного описания многообразия встречающихся на изображениях структур. Сейчас различия между структурными и признаковыми подходами в данной области не столь отчетливы: например, структурные элементы могут называться признаками, основанными на краях [251, п. 2.3], или иметь численные параметры, такие как длина, ориентация и т. д.

Существует также и промежуточный подход, при котором выделяются геометрические элементы на основе краевых точек, не объединенных в связанные контуры [261–264]. Этот подход более популярен в промышленных приложениях и фотограмметрии, поскольку краевые точки могут быть локализованы с субпиксельной точностью, что позволяет выполнять более точные измерения параметров геометрических элементов, например, с помощью метода наименьших квадратов [261, 262].

При этом, однако, точки, принадлежащие разным элементам, не разделены. Поэтому здесь применяются специфические методы, которые одновременно решают проблему группирования и регрессии. Среди этих методов наибольшей популярностью пользуется преобразование Хо [263, 264].

Промежуточные символьные представления еще более разнообразны, чем детекторы контуров. Строгие методы работы с такими представлениями существуют (например, на основе теории формальных грамматик [121, 265] или на основе теории графов [266, 267, с. 42–94]), однако в них предполагается, что исходное представление дано априори. Обоснования же методов построения самих представлений гораздо слабее. Согласно информационной теории зрения Д. Марра, цель выделения структурных элементов заключается в том, чтобы определить положение физических нарушений непрерывностей [235, с. 105], однако самим Марром не приводится строгого доказательства того, что такое восстановление достигается каким-либо алгоритмом.

Попыткой дать строгое развитие этому подходу является, например, работа [219]. Выбор признаков рассматривается в ней с точки зрения максимизации содержащейся в них информации. При этом использование конкретных стохастических моделей изображений позволяет получить некоторые общепринятые признаки (например, края) в качестве наиболее информативных, причем информативность признаков имеет строгое количественное выражение.

Составные структурные элементы. Дальнейшее развитие структурного подхода заключается в формировании на основе непроизводных элементов составных структурных элементов. Такие элементы естественным образом возникают в рамках синтаксического подхода в качестве нетерминальных символов формальной грамматики. Терминальные же символы соответствуют непроизводным элементам и базовым пространственным отношениям между ними. Хотя

синтаксические методы уже не слишком популярны, сама идея использования таких элементов остается весьма перспективной.

Типы составных структурных элементов могут быть как проблемно-зависимыми, так и достаточно общими. Примером первых могут служить, например, такие элементы, составленные из отрезков прямых линий, как проекции прямоугольных параллелепипедов на плоскость (в более простом случае ищутся также прямоугольники и параллелограммы), построение которых на аэрокосмических изображениях необходимо для обнаружения зданий [258, 268, 269].

Существуют различные подходы к группированию непроизводных структурных элементов с целью образования более сложных структурных элементов, вид которых не зависит от предметной области. К таким элементам относятся, например, «L», «U» и другие соединения, пары параллельных прямых линий, пятна, составленные из пятен меньших размеров, и т. д. [235, с. 104; 270]. Однако все подходы к группированию в том или ином виде опираются на подобие структурных элементов. На основе подобия составные структурные элементы могут дальше группироваться, формируя более крупномасштабные элементы изображения.

Попытки строгого обоснования различных типов составных структурных элементов практически отсутствуют. Имеется лишь общая концепция Д. Марра (см. [235]) об иерархической пространственной организации реального мира, которую также должны отражать иерархические структурные представления изображений.

Составные структурные элементы хорошо различимы между собой (при условии, что они надежно выделены), поэтому такие элементы являются удобным средством для описания объектов, присутствующих на изображении, в целях распознавания последних. В результате распознавания может быть построено описание изображения в терминах присутствующих на нем объектов и пространственных отношений между ними, что соответствует верхнему уровню.

3.1.6. Верхний уровень: методы, основанные на знаниях

Как уже отмечалось, в качестве конечной цели зрительного анализа часто рассматривают назначение семантических меток областям на изображении, т. е. описание изоб-

ражений сцен на естественном языке. Тогда в качестве конечного представления изображений рассматривается некоторая система представления знаний.

Более того, многие авторы указывают на необходимость использования знаний и в самом процессе анализа изображений [271, 272]. Аргументация здесь используется примерно следующая: если мы издали видим темное пятно над столом, то мы можем догадаться, что это телефон, хотя информации в самом изображении телефона для этого недостаточно, т. е. мы используем для анализа изображения высокоуровневую информацию [272]. В связи с этим полагается, что и весь процесс обработки зрительной информации должен вестись под управлением знаний (подход сверху вниз). В качестве основной аргументации [214] в пользу этого выступает утверждение о недостоверности результатов, полученных в подходах, ведомых данными (подходах снизу вверх), и о проблеме комбинаторного взрыва количества возможных различных интерпретаций.

Простым примером неявного использования знаний может служить распознавание конкретных объектов на изображениях. И действительно, в этих задачах ведется целенаправленный поиск объектов, принадлежащих определенным (обычно очень узким) классам. Знание свойств этих объектов используется для их распознавания.

В классическом распознавании изображений знания описываются в терминах тех представлений, которые используются в самих алгоритмах распознавания. Распознавание может вестись на основе каких-либо низкоуровневых представлений, тогда и объект описывается соответствующим образом, например, в виде его инвариантных признаков [273, 274], коэффициентов Фурье или вейвлет-преобразования [275, 276] и т. д. Распознаваемые объекты могут храниться в виде своих контурных [248, 277] или структурных [121, с. 282–285; 278] описаний, если привлекаются более абстрактные представления. Таким образом, распознавание часто сводится к получению описания изображения и его сравнению (вычисление некоторой меры сходства) с описанием объекта, поиск которого производится.

Если в классическом распознавании изображений знания представляются неявно, то в последние десятилетия появилась тенденция явного представления знаний в системах автоматической интерпретации изображений. Здесь выделяют три основных типа знаний [271]: перцепционное,

семантическое и функциональное. Перцепционное знание позволяет интерпретировать изображения в терминах линий, областей и т. д., в то время как семантическое знание описывает определенные абстрактные понятия, такие как форма или конкретные объекты и отношения между ними. Функциональное знание предназначено для регулирования процесса интерпретации изображения в зависимости от предметной области и поставленной цели.

Существуют следующие подходы к представлению знаний в системах интерпретации изображений: семантические сети [279–282], объектно-ориентированные представления и фреймы [272, 283, 284], продукционные системы [271, 279], мультиагентный подход [285–287], представления, основанные на логике предикатов, и др. Часто для представления знаний различного типа в одной системе может привлекаться несколько представлений. Но все эти представления заимствованы из экспертных систем. Мы приведем лишь общие свойства таких представлений в рамках задач анализа изображений. Подробнее см. в работах [209, 272].

Как правило, системы, основанные на знаниях, априорно имеют высокоуровневое описание сцены (например, [279]). Задача заключается в привязке этого высокоуровневого описания к изображению, а вовсе не в построении самого описания. Высокоуровневое описание сцены задается не полностью: могут быть неточно известны положения объектов, какие-то объекты или их части в априорном описании могут быть пропущены из-за неполноты описания или могут быть указаны лишние, если часть объекта закрыта. Например, при распознавании объекта неточно известны его координаты на изображении и неизвестно, какой именно объект из данной совокупности присутствует. Однако выбор всегда осуществляется из малого числа альтернативных интерпретаций, каждая из которых выдвигается в качестве гипотезы и проверяется на соответствие изображению. Это хотя и позволяет избежать комбинаторного взрыва числа возможных интерпретаций, характерного для подхода снизу вверх, но делает методы, основанные на знаниях, способными работать лишь в сильно ограниченных предметных областях. Если же столь жесткие ограничения на имеющуюся априорную информацию не накладываются, то подходу сверху вниз проблема комбинаторного взрыва свойственна в еще большей степени, чем подходу снизу вверх.

Более того, при разработке систем, основанных на знаниях, также возникает потребность решать проблему построения промежуточных описаний изображений. Например, в работах [279, 280] используется одна и та же система AIDA (семантические знания представляются в ней с помощью семантических сетей, а функциональные — с помощью продукционных правил) для двух разных задач: восстановление трехмерной формы близких объектов и выбор опорных точек на аэрокосмических изображениях. Однако же в каждом из применений этой системы оказывается необходимым решать свои задачи более низкого уровня: сегментация по яркостному изображению и карте глубины в первом случае и выделение узких полос на изображении (параллельных контуров на градиентном изображении) во втором случае. В системах, подобных системе AIDA, также используется ряд концептов (например, полигон) и взаимосвязей между ними (например, понятие перпендикулярности), которые неявно закладываются в алгоритмы первоначального анализа изображений. Лишь иногда эти понятия выделяются в явном виде в качестве перцепционных знаний. А поскольку полнота системы низкоуровневых понятий не исследуется, то далеко не любые высокоуровневые знания могут быть представлены в рамках выбранного формализма (например, семантических сетей).

Таким образом, если подобные системы применять к конкретным задачам, значительная часть работы ложится на человека — выбирать низкоуровневые признаки и реализовывать эвристические процедуры их выделения. При этом универсальный набор необходимых низкоуровневых признаков не обсуждается. Следовательно, прежде чем рассматривать возможность создания универсальной системы зрения, способной строить семантическое описание произвольной сцены, необходимо решить эту же проблему для построения промежуточного символьного описания.

Также понятно, что построение этих описаний в зрительной системе животных и человека первично по сравнению с построением высокоуровневых описаний сцен. Трудно утверждать, что младенцы обладают какими-либо семантическими знаниями об анализируемых их зрительной системой сценах. Хотя у человека действительно могут использоваться высокоуровневые знания для управления восприятием (например, этим объясняется общеизвестный эффект перцептивной готовности, о котором упоминалось в п. 3.1.1), но на-

копление этих знаний осуществляется на основе более низкоуровневых представлений изображений. К сожалению, в большинстве существующих систем машинного зрения знания о предметной области закладываются вручную. Это делает процесс создания таких систем трудоемким, а сами системы — узкоспециализированными.

Более обещающим кажется принципиально иной подход, развиваемый в работах [206, 288–290]. В указанных работах исследуется проблема одновременного выделения слов в слитной речи и распознавания объектов на изображении и последующее их связывание. В дальнейшем сформированные таким образом лингвистические единицы, связанные со зрительными понятиями, используются как для более надежной сегментации слитной речи на отдельные слова, так и для управления зонами внимания зрительной системы. Именно такой подход, а не искусственное занесение знаний в систему интерпретации изображений, кажется наиболее перспективным для создания систем, основанных на знаниях. К сожалению, в этих работах используется очень простая модель зрительного восприятия, а основное внимание уделяется вопросам анализа речевой информации. Для расширения возможностей подобного подхода, очевидно, необходимо решить проблему построения несемантического представления изображений, носящего общий характер и не зависящего от предметных знаний. Более подробно проблему интеграции сенсорной информации различных модальностей мы рассмотрим в п. 3.4. Поскольку мы считаем, что приобретение знаний некоторой системой машинного восприятия должно происходить в процессе обучения и при использовании нескольких сенсорных модальностей, то, рассматривая далее применение принципа МДО при интерпретации единичных изображений, мы ограничимся промежуточными символьными представлениями. Но сначала необходимо остановиться на важных для анализа изображений иерархических представлениях.

3.1.7. Иерархические представления изображений

Принято считать, что статистические решения о принадлежности некоторого сигнала с шумами к определенному классу должны основываться на как можно большем числе наблюдаемых отсчетов сигнала и что промежуточные

дискретные решения относительно подмножеств отсчетов вредны, так как они разрушают информацию. Однако опыт решения задач анализа изображений доказывает целесообразность принятия подобных промежуточных решений [291]. Действительно, изображения могут трактоваться как зашумленный сигнал, отсчетами которого являются интенсивности отдельных пикселей (или клеток растра), а интерпретация изображений — как статистический вывод. Вывод, дающий наиболее надежные результаты, должен был бы опираться на исходные значения интенсивностей всех пикселей изображения. Почему же подобные методы анализа изображений (например, методы сравнения с эталоном в распознавании и корреляционные методы в совмещении изображений) оказались недостаточно эффективными?

Один из возможных ответов на этот вопрос можно сформулировать следующим образом. Проблема интерпретации является NP-полной, поэтому сложность нахождения точного решения этой проблемы возрастает не менее чем экспоненциально с увеличением размера изображения [237]. Это ограничение является принципиальным, оно подразумевает, что требуется искать субоптимальные методы интерпретации. Принятие промежуточных решений (или построение промежуточных представлений) и является принципиальным подходом к решению проблемы комбинаторного взрыва.

Заметим, что эта проблема очень похожа на проблему, описанную в п. 1.6.4 и связанную с NP-полнотой задачи индуктивного вывода, из-за которой попытка найти оптимальную модель по некоторым исходным данным неизбежно приводит к комбинаторному взрыву, коль скоро пространство моделей не ограничено. Эта связь неудивительна, поскольку процесс интерпретации изображений является частным случаем индуктивного вывода, причем здесь объем исходных данных очень большой, поэтому способы избежания NP-полноты в интерпретации изображений, приводящие к приемлемым решениям, могут оказаться очень полезными и для индуктивного вывода в целом.

При исследовании существующих методов интерпретации изображений можно выделить два различных способа введения иерархичности (промежуточных решений) в процесс анализа изображений: иерархичность по пространственному масштабу (методы с переменной разрешающей способностью [292–296]) и иерархичность по уров-

ням абстрактности привлекаемых представлений (многоуровневые методы [281, 297]).

Существуют два основных аргумента в пользу привлечения методов с переменной разрешающей способностью. Первый аргумент апеллирует к природе изображений: изображения не содержат предустановленного характерного масштаба в отличие, например, от речевой или тактильной информации [231]. В более широкой формулировке иерархичность пространственной организации постулируется как одно из основных общих свойств видимого мира [235, с. 59], которое должно учитываться при разработке представлений изображений. В качестве более практического аргумента приводится возможность с помощью методов с переменной разрешающей способностью заметно увеличивать производительность алгоритмов: приближенное решение сначала находится на основе изображений с уменьшенным разрешением, а затем это решение лишь уточняется при постепенном увеличении разрешения, вместо того чтобы решать полную задачу на наиболее детальном уровне (см., например, [298]).

Очевидно, вероятность получить неоптимальное решение возрастает, поскольку при нахождении начального приближения используется не вся имеющаяся информация, но при этом достигается значительный выигрыш в производительности, поэтому при одинаковом времени работы иерархических и неиерархических методов результат первых оказывается лучше.

Представления, использующие несколько масштабных уровней, существуют для различных уровней абстракции, хотя наиболее исследованными являются низкоуровневые представления, такие как пирамидально-рекурсивные структуры [210, 293, 294], вейвлет- и фрактальные представления [295, 296] и т. д. Целесообразность построения контурных и структурных представлений на нескольких масштабных уровнях (рис. 3.8, *а*–*г*) была указана сравнительно давно [235, с. 66–89], и в некоторых подходах такие представления в определенной степени используются [270], однако сами представления детально не исследовались. При построении представлений, основанных на знаниях, пространственная иерархичность может осуществляться с помощью отношений «часть—целое» [280, 281].

Другой тип иерархичности — по уровням абстракции — имеет еще более слабое обоснование. Аргументы, выдвига-

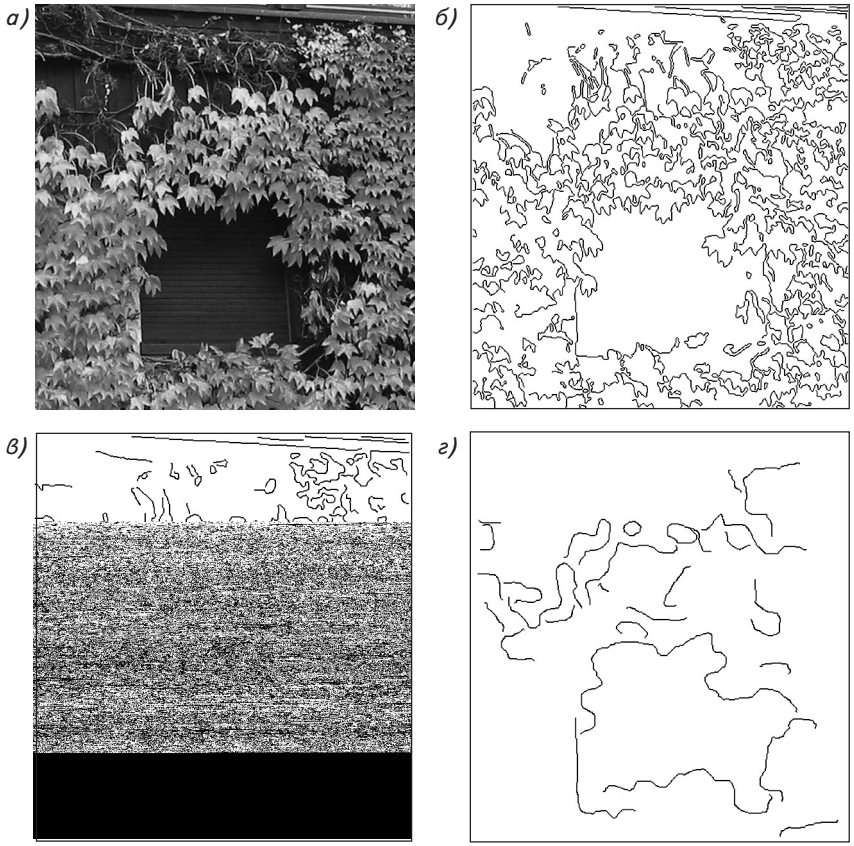

Рис. 3.8. Исходное изображение (*а*) и контуры, выделенные с помощью фильтрации Дериша на разных масштабных уровнях (*б–г*). Видно, что объекты, очерчивающиеся контурами на разных масштабах, могут быть различными

емые в защиту необходимости вводить промежуточные представления, приводились выше. Как правило, они носят чисто эмпирический характер. Многоуровневыми методами обычно называют методы, привлекающие одновременно несколько представлений. В таких методах сначала последовательно строятся представления возрастающих уровней абстракции, затем задача анализа изображений решается на наиболее высоком уровне с последующим уточнением на более низких уровнях.

Здесь так же, как и в методах с переменной разрешающей способностью, сначала находится грубое решение на основе данных с уменьшенной размерностью, а затем оно

уточняется. Есть, однако, и существенное различие: понижение размерности данных здесь осуществляется при выделении наиболее релевантной информации или инвариантных признаков, в то время как в методах переменной разрешающей способности понижение размерности данных осуществляется независимо от содержания изображения.

Представляет интерес применение многоуровневого подхода к самой проблеме интерпретации изображений. Базовые идеи здесь были заложены С. Гроссбергом [299, 300] в его теории адаптивного резонанса. Они заключаются в том, что, получив интерпретацию исходных данных на более высоком уровне, мы можем уточнить более низкоуровневую интерпретацию. К примеру, при чтении рукописного текста, написанного не слишком хорошим почерком, в некотором слове человек для одной из букв может иметь несколько гипотез ее значения. Правильная гипотеза позволит получить осмысленное слово (элемент представления следующего уровня) и будет поддержана за счет положительной обратной связи с этого уровня. Другие же гипотезы, не дающие хороших интерпретаций на следующем уровне, будут подавлены отрицательной обратной связью. Таким образом, правильные интерпретации на различных уровнях как бы входят в резонанс, что позволяет бороться с неоднозначностями, возникающими в подходе снизу вверх.

К сожалению, концепция адаптивного резонанса получила развитие лишь применительно к искусственным нейронным сетям (сети типа ART, Adaptive Resonance Theory [301–303]), хотя сама идея очень продуктивна и общеприменима к любым иерархическим методам, поскольку позволяет снизить отрицательный эффект от принятия промежуточных решений. К примеру, нечто похожее привлекалось в п. 2.5.7 в целях совместного использования локальных и распределенных представлений. В связи с этим крайне интересно было бы объединить теорию адаптивного резонанса и принцип минимальной длины описания. В п. 3.5 мы попытаемся продемонстрировать, что это действительно возможно.

Таким образом, в задачах автоматического анализа изображений привлекаются представления нескольких уровней абстракции. Многие задачи могут быть решены при использовании различных представлений с разной эффективностью. Как правило, структурные методы являются наиболее робастными, а низкоуровневые математические представле-

ния дают наиболее точные результаты. Представления различных уровней могут быть описаны с помощью формальных (внутренне непротиворечивых) систем, однако выбор самих представлений и тем более построение алгоритмов перехода от одного представления к другому является гораздо менее обоснованным и более эвристичным. В результате подобной нестрогости в процессе интерпретации изображений при переходе от более низких уровней описания к более высоким уровням происходит постепенная потеря значимой информации, вследствие чего современные системы машинного зрения оказываются несравнимо менее эффективными или менее универсальными, чем зрительный аппарат животных и человека.

В результате многолетнего опыта применения подходов снизу вверх и сверху вниз многие исследователи пришли к выводу [214], что процесс интерпретации изображений должен производиться под управлением данных, но с использованием обратных связей с верхних уровней. Об этом также свидетельствуют работы по адаптивному резонансу [300], основанные на нейрофизиологических данных. Однако, как и при прямом распространении информации, которая осуществляется алгоритмами интерпретации в процессе перехода между представлениями различных уровней, алгоритмы обратного распространения информации являются зачастую эвристическими, что сильно снижает их эффективность.

Таким образом, в рамках реконструкционной парадигмы решение проблемы зрения видится в создании многоуровневых систем интерпретации изображений с привлечением обратных связей. Однако для достижения этой цели необходима разработка подхода к построению таких обобщенных теоретических моделей изображений, которые бы единообразно включали представления различных уровней и связывали их между собой. Такой подход позволил бы далее исследовать конкретные модели изображений (выбор наилучшей модели, очевидно, является эмпирической проблемой, так как зависит от свойств физического мира).

Столь широкая проблема выходит далеко за рамки данной книги. Здесь мы лишь попытаемся показать возможность применения принципа минимальной длины описания для решения задачи построения многоуровневого представления, ограниченного составными структурными элементами. Останется незатронутой проблема восстановления карт отражательной способности видимых поверхностей и трех-

мерного описания сцены, которое может осуществляться на основе этого представления. Для некоторых типов изображений (например, аэрокосмических) эта проблема не слишком актуальна, но в общем случае она должна быть решена после построения структурного описания.

3.2. ПРИНЦИП МИНИМАЛЬНОЙ ДЛИНЫ ОПИСАНИЯ В ИНТЕРПРЕТАЦИИ ИЗОБРАЖЕНИЙ

3.2.1. Выбор представления изображений с теоретико-информационной точки зрения

Мировой опыт в области интерпретации изображений показывает, что процесс интерпретации должен быть многоуровневым, и указывает, какие именно уровни желательно использовать. Однако отсутствуют объективные критерии сравнения качества работы различных алгоритмов, строящих описания изображений. Зачастую качество описания определяется на основе субъективного мнения эксперта (т. е. эвристически) либо по эффективности применения этого описания для решения какой-нибудь последующей задачи зрения (т. е. эмпирически). Установление строгого критерия для сравнения качества описаний изображений позволило бы вести целенаправленное улучшение алгоритмов интерпретации. Для введения такого критерия необходимо свести проблему интерпретации изображений к некоторой общей математической проблеме.

Давайте взглянем на процесс интерпретации изображений как на частный случай индуктивного вывода (о корректности такого рассмотрения см., например, [2, с. 2, 3]). Напомним (см. п. 1.1), что индуктивный вывод характеризуется такими основными элементами, как данные наблюдений D, пространство гипотез H и критерий рациональности $r(h|D)$. Исходными данными здесь являются изображения, представленные на пиксельном уровне, а отдельная гипотеза h — это гипотеза, описывающая содержание изображения в некоторых терминах (например, опорные точки, структурные элементы или метки, соответствующие конкретным объектам). Пространство гипотез и критерий рациональности в явном виде определяются лишь в рамках низкоуровневых представлений — стохастических или функциональных моделей изображений.

Общий критерий выбора между такими моделями изображений можно задать с помощью правила Байеса: $r(h|D) = P(h)|D) \sim P(h)P(D|h)$. Если $P(h)$ и $P(D|h)$ могут быть вычислены, то задача интерпретации изображений решена. К сожалению, определить оба эти множителя крайне сложно. Априорные вероятности гипотез, задание которых составляет общую проблему вероятностных методов, можно оценить с помощью обучения с учителем, но это реализуемо, только если пространство гипотез крайне мало. Например, если речь идет о распознавании объектов из очень ограниченного числа классов, то человек может для обучающей выборки изображений указать правильную гипотезу о содержании каждого изображения (при этом человек как бы передает свое знание априорных вероятностей компьютерной системе). Вычисление вероятностей $P(D|h)$ опирается на модели процесса порождения изображений (о них мы упоминали в п. 3.1.4). Но лишь для достаточно простых моделей удается определить $P(D|h)$. В совокупности H, $P(h)$ и $P(D|h)$ составляют модель предметной области. На языке теории вероятности эти модели имеют крайне сложное описание; сам же байесовский метод основывается на предположении о верности модели предметной области, т. е. захватывает лишь один уровень индуктивного вывода. Остается неясным, как сравнивать разные пространства гипотез и задавать на них априорные вероятности.

Для решения этих проблем мы попытаемся воспользоваться принципом МДО. В информационном подходе выбор пространства гипотез с соответствующими им априорными вероятностями заменяется выбором представления, в рамках которого описывается содержание изображения. В рамках конкретного представления следует выбрать ту гипотезу о содержании изображения, которая позволяет описать это изображение наиболее коротко.

Выбор между различными представлениями сам по себе может осуществляться на основе информационной меры, т. е. по суммарной длине описаний некоторого ансамбля изображений. Несмотря на возможность введения строгого критерия качества представления, предложение конкретных представлений, из которых следует определить лучшее, должно осуществляться человеком. Можно было бы предложить алгоритм, перебирающий программы для машины Тьюринга и выбирающий наиболее короткую программу, порождающую на выходе данное изображение, но такой алгоритм не-

пременно натолкнулся бы на проблему комбинаторного взрыва. Отметим, что проблемой выбора оптимального представления изображений занимается большое количество исследователей, опирающихся не только на информацию, содержащуюся в конкретных изображениях, но и на обширные знания свойств внешнего мира, а также использующих в качестве подсказок данные о функционировании своей собственной системы зрения, на эволюционное формирование которой ушли миллионы лет.

В области компьютерного зрения исследователями ведется подобный, но более направленный перебор представлений. Как указывалось ранее, общий вид представлений можно считать установленным — это иерархические структурные представления, среди которых и нужно выбрать оптимальное. Теоретико-информационный подход удобен также тем, что он в равной степени может быть применен для разных уровней представления.

Использование принципа минимальной длины описания в различных задачах компьютерного зрения начало приобретать популярность в 90-х годах прошлого века и продолжает завоевывать все больше сторонников. Применение информационного критерия качества описания в рамках различных представлений изображения или серии изображений было опробовано в следующих задачах:

- сегментация (по текстуре или цвету) изображения [167, 304, 305];
- выделение признаков на изображении [219];
- построение структурных элементов изображения [306] и их группирование [307], а также описание формы границ областей [308, 309];
- распознавание объектов на изображении [310] и распознавание рукописных символов [311, 312];
- оценивание параметров пространственного преобразования между парой изображений одной и той же сцены, снятой с разных ракурсов, по набору опорных точек [313] и собственно сопоставление и совмещение пары изображений (нахождение соответствия между точками изображений) [314];
- оценивание поля движения по видеосерии [315–317];
- и др. (см., например, [318–320]).

Из приведенного перечня видно, что теоретико-информационные методы помогают решать многие задачи анализа изображений, а длина описания может служить критерием каче-

ства описания изображений на разных уровнях абстрактности, начиная с пиксельного и заканчивая семантическим.

Тем не менее можно сказать, что применение принципа МДО в данном научном направлении только начинается. Такая точка зрения связана не только с тем, что в подавляющем большинстве работ в области компьютерного зрения продолжают использоваться такие классические методы выбора модели, как метод наименьших квадратов или метод максимального правдоподобия, но и тем, что в работах, в которых принцип МДО применяется, он обычно применяется в достаточно ограниченной форме. Поясним это на конкретных примерах.

Так, в работе [311] для распознавания рукописных символов их изображения описываются вектором признаков, компоненты которого соответствуют координатам характеристических точек на изображении. Извлечение признаков (определение количества и положения характеристических точек) осуществляется эвристической процедурой, и лишь дальнейший анализ (распознавание) изображения символа, представленного в виде вектора признаков, ведется в соответствии с принципом МДО.

В другой работе [306] рассматривается вопрос аппроксимации краев, обнаруженных на изображении. В то время как аппроксимация ведется на основе принципа МДО, края считаются заданными в качестве исходных данных, а в самой работе для их обнаружения производится ограничение по порогу амплитуды градиентного поля. Иными словами, решается проблема интерпретации не исходного изображения, а некоторого его промежуточного представления, в котором часть информации уже потеряна.

Приведем еще один пример. В работе [304] рассматривается проблема сегментации изображений на основе принципа МДО, при этом в целевой функции не учитывается сложность границ областей, что приводит к построению описаний изображений, эквивалентных по уровню абстрактности контурным описаниям. Еще одним показательным моментом в этой работе является способ текстурной сегментации. Для ее реализации интенсивность в каждой точке изображения заменяется вектором текстурных признаков (таких, как, например, среднее и дисперсия интенсивностей в малом окне с центром в данной точке или количество яркостных переходов разных знаков). Сегментации подвергается такое преобразованное многокомпонентное изображение. Хотя это и

приводит к неплохим результатам с точки зрения выделения областей однородной текстуры, но не является последовательным применением принципа МДО, поскольку может привести к увеличению длины описания отсегментированного преобразованного изображения по сравнению с исходным изображением.

Все упомянутые работы, бесспорно, вносят большой вклад в развитие теоретико-информационного подхода к интерпретации изображений, но они обращаются к различным аспектам одной общей задачи. Мы же попытаемся рассмотреть задачу целиком, хотя предложенные решения будут получены в рамках сильных упрощений. Поскольку основной целью реконструкционного подхода в иконике является получение представления изображений, которое бы позволяло описывать в явном виде физические свойства объектов, составляющих сцену, свое изложение мы начнем с тех ограничений, которые могут быть наложены на эти свойства.

3.2.2. Общие предположения о свойствах изображений

Пространство моделей (или соответствующее представление), из которого происходит выбор модели, наилучшим образом описывающей данные наблюдений, является метамоделью (моделью предметной области). Выбор конкретной метамодели означает привлечение некоторой априорной информации, играющей роль ограничений, накладываемых в процессе интерпретации изображений на их содержание. Естественно, метамодели могут иметь различную длину описания или информативность, т. е. накладывать разное число ограничений. Пространство моделей с минимальным количеством априорной информации — это пространство программ для некоторой универсальной машины Тьюринга. Но помимо того, что использование подобного представления невозможно на практике из-за отсутствия эффективных методов вычисления алгоритмической сложности, универсальное распределение вероятностей, видимо, сильно отличается от распределения вероятностей на множестве изображений.

Напротив, в рамках целевого подхода в иконике, противопоставляемого реконструкционной парадигме, используются очень сильные ограничения, черпаемые из той конкретной прикладной задачи, для которой разрабатывается система машинного зрения [205]. Иными словами, в рам-

ках этого подхода разработчиками каждый раз строится новая метамодель, приспособленная под частную задачу, что делает такие методы эффективными, но узкоспециализированными.

В самом же реконструкционном подходе исследователи стремятся использовать общие ограничения, накладываемые физическим миром, что позволяет строить методы более универсальные, чем методы, разрабатываемые в рамках целевого подхода, и более эффективные, чем методы, не использующие никаких специфичных для иконики ограничений. Впервые достаточно подробный список таких физических ограничений был сформулирован Д. Марром [235, с. 57–63] в книге «Зрение. Информационный подход к изучению представления и обработки зрительных образов». В ней разрабатывается информационная теория зрения, т. е. зрение рассматривается как процесс преобразования визуальной информации из некоторого исходного представления в более информативное конечное представление. К сожалению, в книге используется неформальное понятие информации на уровне словесного описания того, что именно должно осуществляться в процессе зрения, но не привлекается формальная теория информации на алгоритмическом уровне, отвечающем на вопрос, как именно осуществляется процесс зрения.

Представляется, что теория Д. Марра может быть переформулирована в теоретико-информационных терминах и приведена в соответствие с принципом МДО, что позволит сделать ее более строгой и построить более надежные алгоритмы зрения. К сожалению, эта задача выходит за рамки данной книги. Здесь мы лишь воспользуемся физическими предположениями Д. Марра для выбора представления изображений, минимизирующего их длину описания. Процитируем эти предположения.

1. Поверхности как реальные объекты: весь видимый мир можно рассматривать как некоторую композицию гладких поверхностей, функции отражательной способности которых могут отличаться сложной пространственной структурой.

2. Иерархическая организация: пространственная организация функции отражательной способности некоторой поверхности часто порождается совместным воздействием целого ряда различных процессов, каждый из которых относится к отдельному уровню.

3. Подобие: объекты, появляющиеся на некоторой поверхности в результате некоторого процесса порождения отражательной способности, действующего на некотором определенном масштабном уровне, обычно обладают большим сходством по размерам, локальному контрасту, цвету и пространственной организации между собой, чем с другими объектами этой же поверхности.

4. Пространственная непрерывность: характерные объекты, возникающие на некоторой поверхности в результате действия какого-то одного процесса, помимо того, что они обладают «внутренним» подобием, часто образуют определенную пространственную организацию, принимающую вид кривых, прямых и, возможно, более сложных конфигураций.

5. Непрерывность нарушений непрерывности: геометрическое место разрывов по глубине или ориентации поверхности почти везде гладко.

6. Непрерывность движения: при наличии любого нарушения непрерывности движения более чем в одной точке (например, вдоль некоторой прямой) следует считать, что имеет место граница объекта.

Эти пункты могут дополняться законами распространения излучения, с помощью которых можно построить, например, модели стереозрения или процесса затенения и образования теней. Корректное разделение параметров источников освещения и функций отражательных способностей видимых поверхностей как независимых факторов должно приводить к уменьшению длины описания. Значит, верно и обратное: критерий МДО может использоваться для оптимального восстановления пространственной организации сцены. К сожалению, в этом направлении сделано немногое, поэтому преимущественно сосредоточимся на промежуточных символьных представлениях изображений, которые опираются на предположения Д. Марра.

Рассмотрим первое предположение. Заметим, что каждая видимая поверхность представлена на изображении некоторой областью. Функция отражательной способности этой поверхности под воздействием освещения порождает распределение яркости внутри соответствующей области изображения, специфическое для данной области. Иными словами, первое предположение говорит о том, что для описания изображения необходимо использовать модели сегментации. Изображение должно быть разбито на области, и рас-

пределения яркости внутри каждой из них необходимо описать с помощью отдельной модели.

Сам Д. Марр критиковал идею сегментации изображений [235, с. 275–276], широко распространенную в компьютерном зрении. Однако тогда под сегментацией понималось разделение изображения на объекты и фон. Именно неопределенность этих понятий и была причиной критики. Здесь же сегментация рассматривается с точки зрения построения модели и не привлекает понятия объекта и фона. Более того, если первое предположение Д. Марра соблюдается, то оптимальная модель сегментации выделит как раз те области на изображении, которые соответствуют видимым поверхностям. Как было показано в п. 2.6, выбор регрессионных моделей адекватной сложности равно, как и выбор оптимального числа сегментов, целесообразно осуществлять, руководствуясь принципом МДО.

Рассмотрим пятое предположение. Геометрическое место точек разрывов по глубине или ориентации поверхностей соответствует на изображении границам областей (контурам). Данное предположение говорит о том, что границы областей почти всюду гладки, а значит, для их описания также должны использоваться модели сегментации: точки нарушения гладкости разделяют сегменты на контурах, каждый из которых описывается некоторой гладкой функцией. Очевидно, и здесь может использоваться информационный критерий. Более того, обе задачи — сегментация изображения на области и сегментация контуров (или границ областей) — оказываются связанными единой целевой функцией (см. п. 2.6.4), что в алгоритмах компьютерного зрения практически всегда игнорируется.

Второе, третье и четвертое предположения Д. Марра также можно интерпретировать с позиции уменьшения длины описания. Эти предположения указывают на то, какие регрессионные модели следует использовать для описания содержимого областей. Характерные детали, присутствующие на видимых поверхностях, должны иерархически группироваться на основе подобия их внутренних признаков (размеров, ориентации и т. д.) и местоположения (если они образуют некоторые регулярные структуры). Последовательное группирование характерных деталей вполне соответствует построению составных структурных элементов.

Сходство структурных элементов, как и регулярности в их взаимном расположении, подразумевает наличие общей

информации в их описаниях. Значит, для выработки корректного критерия формирования составных структурных элементов может быть использован принцип минимальной длины описания, возможность чего также будет рассмотрена ниже.

Шестое предположение имеет отношение к динамическим сценам. Хотя здесь мы рассмотрим вопросы интерпретации лишь отдельных изображений статических сцен, описание движения, как отмечалось в п. 3.2.1, также может быть включено в общую схему, в которой принцип МДО привлекается в качестве критерия [315–317].

Итак, построение промежуточного символьного описания изображения подразумевает его сегментацию на области (или извлечение контурной информации), построение непроизводных структурных элементов на основе контуров или границ областей и формирование составных структурных элементов посредством группирования непроизводных элементов. Рассмотрим каждый из этих шагов в отдельности, а затем (см. п. 2.5) обсудим вопрос об обратных связях, позволяющих снимать неопределенности, возникающие на нижних уровнях в методах, управляемых данными.

Но прежде чем переходить к проблеме сегментации, сделаем небольшое отступление и отметим, что предположения Д. Марра, явно сформулированные им для изображений, во многом похожи на те предположения, которые неявно делались для произвольного пространства признаков в дискриминантных методах распознавания образов. Действительно, при распознавании образов обычно предполагается (см. п. 2.3.7), что каждый класс занимает некоторую область с гладкими или кусочно-гладкими границами. Подобное предположение о непрерывности, сделанное для изображений, чья природа сравнительно четко определена, более обоснованно, чем это же предположение, сделанное для произвольного абстрактного пространства признаков. Тем не менее не для всех типов изображений оно вполне справедливо (рис. 3.9, *а*, *б*).

В то же время, хотя зрительная система человека и оптимизирована под конкретную среду, она достаточно универсальна для того, чтобы позволить человеку работать и с такими изображениями, с которыми он в естественных условиях не мог бы столкнуться. Так, эксперт, взглянув на изображения, представленные на рис. 3.9, *а*, *б*, получил бы много полезной информации о представленных на них

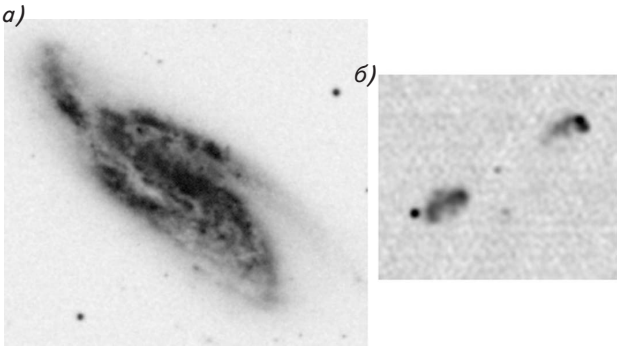

Рис. 3.9. Пример изображений, для которых предположения Д. Марра не вполне выполняются: *а* — оптическое изображение галактики (негатив); *б* — изображение внегалактического радиоисточника (на частоте 1400 МГц)

объектах. Это говорит о том, что любые предположения, закладываемые в систему зрения, не должны накладывать слишком жестких ограничений на возможность различения и отождествления произвольных изображений заранее непредусмотренных типов.

3.2.3. Сегментация изображений на однородные области

Поступающее на вход системы машинного зрения изображение, представленное на пиксельном уровне, сначала должно быть описано в терминах областей и их границ. Эту задачу сегментации можно трактовать как обычную задачу сегментации, рассмотренную в п. 2.6, но со специфичными классами регрессионных моделей. Вид регулярной составляющей регрессионной модели области, как правило, берется очень простым (часто — константой), в то время как стохастическая составляющая модели берется более сложной, чем модель гауссова шума, обычно применяемая при решении общей задачи регрессии или сегментации. Во-первых, распределение интенсивностей отдельных пикселей вовсе не гауссово. Во-вторых, между интенсивностями соседних пикселей существуют сложные статистические зависимости, которые различаются для разных видимых поверхностей.

Большинство методов сегментации изображений не восстанавливает модели, описывающие содержимое областей,

а использует переход от интенсивностей к текстурным признакам, отражающим локальные статистические свойства изображения, сравнительно постоянные для каждой поверхности. Здесь мы, однако, будем придерживаться общего подхода к сегментации.

Так же как и в общей задаче сегментации, при сегментации изображений возникает проблема выбора между моделями различной степени сложности. Эта проблема свойственна и для методов, использующих текстурные признаки, однако в них она адресуется в неявном виде.

Итак, пусть G — это область, на которой задано изображение $f(x,y): G \to R$. Задача заключается в том, чтобы разбить эту область на d областей G_1, ..., G_d, таких, что $G_1 \cup G_2 \cup ... \cup G_d = G$ и $G_k \cap G_l \neq \varnothing \Leftrightarrow k = l$, где d также неизвестно. Введем обозначение: $f_k(x,y) = f(x,y)\big|_{G_k}$ — сужение изображения f (интерпретируемого как двумерная функция) на область G_k. Каждая из функций f_k описывается собственной регрессионной моделью $g_k(x,y,\vec{w}_k): G_k \to R$, где $\vec{w}_k$ — ее вектор параметров, которые необходимо определить в процессе сегментации.

Определение семейства функций, описывающих содержимое отдельных областей, — это отдельная сложная задача, связанная с формализацией понятия текстуры и выходящая за рамки данной книги. Здесь будут использованы лишь достаточно простые регрессионные модели.

Воспользуемся теоретико-информационным критерием (2.101), введенным в п. 2.6.4 для общей задачи сегментации. Эта целевая функция состоит из двух частей: суммы длин описания содержания областей и суммы длин описания границ областей.

Рассмотрим сначала случай, при котором для описания содержания областей никакие регулярные модели не привлекаются, а интенсивности отдельных пикселей считаются независимыми результатами испытаний некоторой случайной величины, распределение вероятностей которой постоянно внутри каждой области. Тогда длина описания содержания каждой области будет равна энтропии интенсивностей ее пикселей $H(f_k)$, умноженной на число пикселей $\|G_k\|$ в области G_k. Эта величина является оценкой математического ожидания длины описания интенсивностей пикселей области G_k, закодированных с помощью кода Хаффмана (см. п. 1.3.4). Для декодирования такого кода требуется также таблица перекодировки, объем которой

можно оценить как $N_{\text{int}} \log_2 N_{\text{int}}$ бит, где N_{int} — число различных уровней интенсивности. Следовательно,

$$L\left(f_k(x,y)\right) = H(f_k)\left\|G_k\right\| + N_{\text{int}} \log_2 N_{\text{int}}. \qquad (3.10)$$

Энтропия $H(f_k)$, как правило, определяется с помощью оценивания распределения вероятностей интенсивностей пикселей в области. В случае полутоновых изображений с не слишком высоким числом градаций (например, до 256) может непосредственно строиться гистограмма (число пикселей в области с данным уровнем интенсивности), а для вычисления энтропии может использоваться формула энтропии дискретной случайной величины.

Для цветных изображений, для которых каждому пикселю соответствует несколько значений интенсивностей в разных спектральных полосах, гистограмма становится неэффективным средством оценивания энтропии. Пусть, например, каждый пиксель является реализацией трехкомпонентного случайного вектора $(R_{pix}, G_{pix}, B_{pix})$, каждый из компонентов которого может принимать любое из 256 значений. Тогда всего будет существовать свыше 16 млн его возможных различных реализаций. Чтобы надежно оценить вероятность каждой из них, потребуется изображение весьма больших размеров. Вычислять энтропию для каждого из цветовых каналов по отдельности некорректно, так как они не являются статистически независимыми, т. е. $H(R_{pix}, G_{pix}, B_{pix}) < H(R_{pix}) + H(G_{pix}) + H(B_{pix})$. В действительности, взаимная информация между разными каналами весьма велика. Однако если воспользоваться анализом независимых компонент и перейти к такому случайному вектору (C_1, C_2, C_3), компоненты которого статистически независимы, то для оценивания энтропии $H(f_k)$ можно пользоваться гистограммами так же, как и при работе с полутоновыми изображениями.

Помимо длины описания содержания областей в основанной на теории информации целевой функции, задающей качество сегментации, присутствует еще одно слагаемое — длина описания $L(\delta G_k)$ границы области δG_k. Корректное определение ее значения подразумевает решение задачи сегментации границ с построением структурных элементов. Выполнение этой операции для каждой гипотезы сегментации изображения неэффективно с вычислительной точки зрения, в связи с чем в процессе выделения областей целесообразно использовать грубую оценку величины $L(\delta G_k)$. При ис-

пользовании цепного кодирования каждая точка контура содержит одно из N_{dir} возможных направлений на следующую точку (обычно используется N_{dir}, равное 4 или 8), поэтому при таком представлении контура $L(\delta G_k) = \|\delta G_k\| \log_2 N_{dir}$. Полная длина описания изображения f будет составлять

$$L(f) = \sum_k \left(\|G_k\| \cdot H(f_k) + N_{\text{int}} \log_2 N_{\text{int}} + \|\delta G_k\| \cdot \log_2 N_{dir} \right). \tag{3.11}$$

Приведем теперь один из возможных алгоритмов сегментации.

Алгоритм сегментации. Для нахождения областей $G_1, \ldots, G_d$, минимизирующих целевую функцию (3.11), был реализован следующий алгоритм последовательного слияния областей, начиная с некоторых исходных областей малого размера.

1. Разбить изображение на исходные области пикселей $K \times K$ (где K мало: $K \approx 3 \div 7$ пикселей). Этот шаг нужен для того, чтобы можно было осуществить оценку энтропии интенсивностей внутри областей.

2. Последовательно объединять все пары граничащих друг с другом областей, если это объединение приводит к уменьшению их длины описания. При этом не нужно пересчитывать все слагаемые суммы (3.11), достаточно лишь сравнить длину описания объединенной области с суммой длин описания двух отдельных областей. После того как процесс объединения областей остановился из-за отсутствия пар областей, объединение которых приводит к уменьшению целевой функции, перейти к следующему шагу.

3. Для каждого набора пикселей, соответствующего какой-либо исходной области $K \times K$, лежащей на границе некоторой области, полученной после объединения, принять решение о его отнесении к соседней области на основе критерия минимальности длины описания. После того как не останется таких групп пикселей, принадлежность которых следовало бы изменить, перейти к следующему шагу.

4. Аналогичную процедуру выполнить для отдельных пикселей, лежащих на границах областей.

Шаги 3 и 4 необходимы по двум причинам. Во-первых, при исходном разбиении изображения на квадратные области в них могли попасть пиксели, принадлежащие разным видимым поверхностям. Во-вторых, при начальном слиянии небольших областей используется лишь локальная информация, не гарантирующая достижения глобаль-

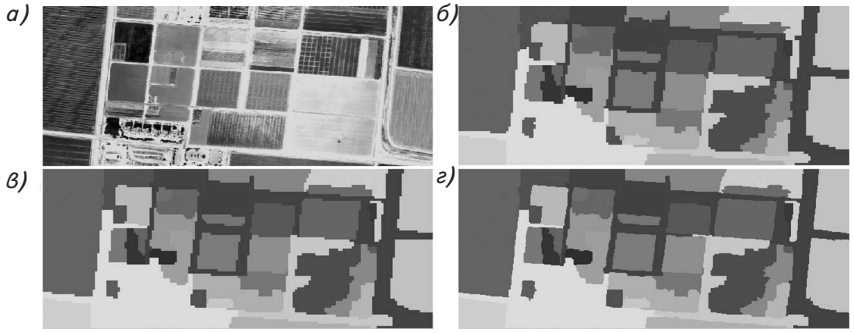

Рис. 3.10. Шаги алгоритма сегментации изображений на области: *а* — исходное изображение; *б* — результат попарного слияния исходных областей (квадраты 3×3 пикселя); *в* — результат третьего шага алгоритма; *г* — конечный результат сегментации после 4-го шага (попиксельной коррекции формы областей)

ного минимума общей длины описания. Шаг 3 может быть пропущен, так как он полностью заменяется шагом 4, но его выполнение позволяет повысить эффективность вычислений.

Основные шаги приведенного алгоритма показаны на рис. 3.10.

Для определения того, насколько корректно работает предложенный алгоритм, следует протестировать его на искусственно созданных изображениях, для которых истинная модель известна. Пример такого тестирования приведен на рис. 3.11, 3.12, из которых видно, что алгоритм действительно работает в соответствии с ожиданиями, несмотря на некоторые упрощения, сделанные при выводе критерия оптимальности (3.11). Отметим, что более сильные упрощения могут привести к неадекватному результату. Например, полное игнорирование слагаемого $L(\delta G_k)$ приводит к чрезмерной фрагментарности результатов сегментации (рис. 3.13, *а*–*в*).

В приведенном алгоритме сегментации использовались тривиальные регрессионные модели для описания содержания каждой из областей. Несмотря на то что такой алгоритм сравнительно неплохо способен различать текстуры (рис. 3.14), это все же является лишком сильным упрощением. Посмотрим, как можно расширить алгоритм за счет использования более сложных регрессионных моделей.

Расширение алгоритма. Описанный алгоритм может быть расширен посредством использования более сложных

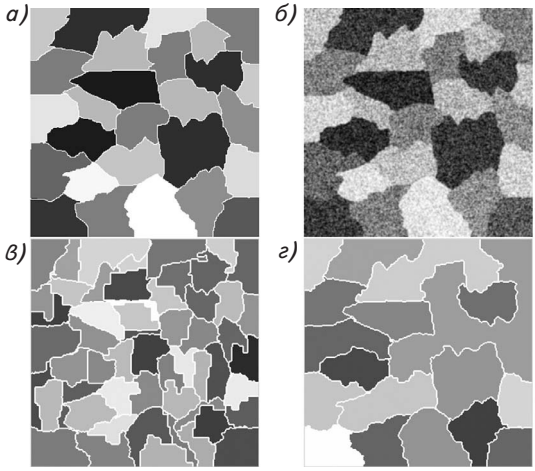

Рис. 3.11. Пример сегментации искусственного изображения: *а* — исходное изображение, состоящее из областей с постоянной интенсивностью с выделенными границами; *б* — зашумленное изображение, использованное для сегментации; *в* — недостаточное объединение областей; *г* — избыточное объединение областей. Объединение областей осуществлялось алгоритмом, описанным в тексте, но при использовании критерия среднеквадратичного отклонения. Результаты *в* и *г* получены при разных порогах, накладываемых на значения дисперсий интенсивностей в областях

регрессионных моделей для описания содержания областей. Пусть $g_k(x, y, \vec{w})$ — регрессионная модель, параметры которой оцениваются для k-й области изображения $f_k(x, y)$.

Рис. 3.12. Сегментация изображения, приведенного на рис. 3.11, *б*, при использовании критерия МДО и описанного в тексте алгоритма. Различные оттенки серого использованы для различения областей и не связаны с их содержанием. Результат соответствует истинному изображению рис. 3.11 с точностью до небольших отклонений границ областей от истинного положения из-за влияния шумов

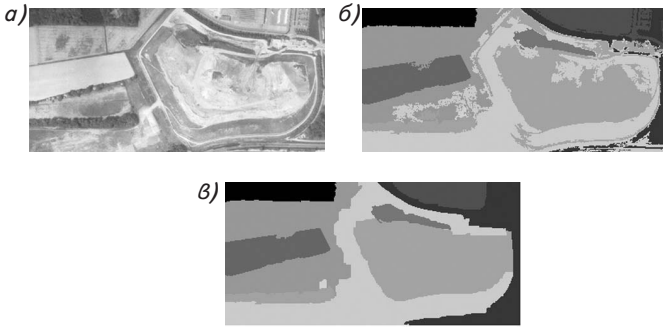

Рис. 3.13. Сегментация аэрокосмического изображения: *а* — исходное изображение; *б*, *в* — результат его сегментации без учета и с учетом длины описания границ соответственно

Для каждой области может быть решена задача регрессии: среди всех моделей (которые могут принадлежать разным параметрическим семействам) должна быть выбрана та, которая минимизирует длину описания невязок и длину описания параметров модели. В вышеописанном алгоритме, по сути, все интенсивности пикселей трактовались как невязки, а регрессионные модели имели ноль параметров. При использовании более сложных регрессионных моделей также имеет смысл строить гистограмму невязок и считать ее энтропию вместо того, чтобы оценивать их длину описания через логарифм среднеквадратичного отклонения, как это делается при решении классических регрессионных задач (см. п. 2.6).

Необходимость использовать регрессионные модели для описания содержания областей возникает по двум основ-

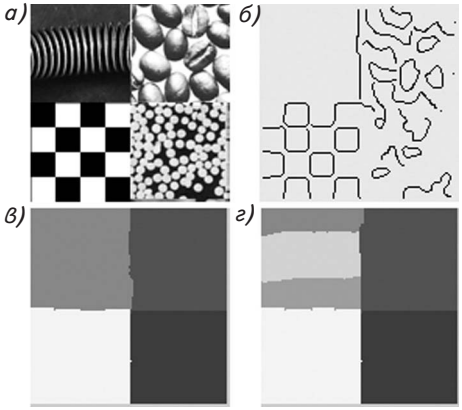

Рис. 3.14. Иллюстрация возможности различения текстур: *а* — изображение, составленное из четырех текстур; *б* — контуры, полученные с помощью фильтрации Дериша (как видно, эти контуры имеют мало общего с границами текстурных областей); *в* — результат сегментации с помощью критерия МДО; *г* — результат сегментации (процедура слияния областей была остановлена, когда осталось шесть областей)

ным причинам: из-за наличия пространственных взаимосвязей между пикселями (наличия текстуры) и из-за возможности неравномерного освещения сцены. Две эти причины требуют использования разных типов функциональных зависимостей. Привлечение периодических функций в качестве регрессионных моделей может помочь в описании текстуры (если нет сильных проективных искажений), а для учета неравномерности освещения следует привлекать монотонные функции.

В описанный выше алгоритм внесем следующую модификацию. При подсчете длины описания некоторой области на изображении будем вписывать модели

$$g(x, y, \vec{w}) = w_0 + w_1 x + w_2 y + w_3 x^2 + w_4 xy + w_5 y^2 \quad (3.12)$$

и вычислять энтропии невязок $r_k(x, y) = \left[f_k(x, y) - g_k(x, y, \vec{w}) \right]$, округленных до целочисленных значений (именно так представляются интенсивности). К длине описания невязок следует добавлять длину описания параметрической части, которую можно оценить как $L_{p,k} = \frac{n_{p,k}}{2} \log_2 \|G_k\|$, где $n_{p,k}$ — число параметров регрессионной модели, описывающей содержание k-й области.

Изменение алгоритма сегментации проявляется лишь в том, что при вычислении длины описания каждой области (в том числе и при принятии решения об их слиянии) одновременно определяется и оптимальная регрессионная модель.

Поскольку в качестве регрессионных моделей использовались непериодические функции, то улучшение результатов следует ожидать на изображениях с неравномерным освещением. Рассмотрим три примера. На рис. 3.15, *б* представлено искусственно созданное изображение с плавными изменениями яркости. Использование упрощенного алгоритма сегментации приводит к плохим результатам, что говорит о неполноте пространства гипотез. Описание содержания областей регрессионными моделями позволяет получить желаемые области (рис. 3.16).

На рис. 3.17 приведены результаты сегментации реальных изображений, по которым отчетливо видны преимущества расширенного алгоритма. Отсюда можно заключить, что дальнейшее расширение алгоритма сегментации за счет включения регрессионных моделей других типов (в частности, для описания текстур) должно привести к дальней-

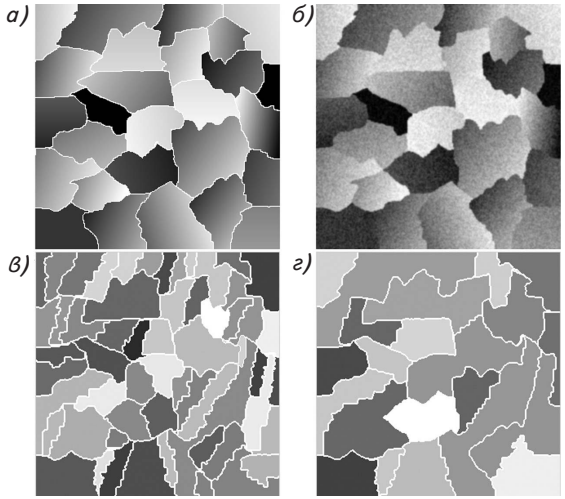

Рис. 3.15. Пример сегментации искусственного изображения: *а* — исходное изображение, состоящее из областей с интенсивностью, меняющейся по линейному закону; *б* — зашумленное изображение, использованное для сегментации; *в* — результат сегментации с критерием МДО без использования регрессионных моделей для описания регулярных изменений интенсивностей внутри областей; *г* — результат сегментации при использовании того же метода, что и в случае *в*, но с изменением порога объединения областей

шему улучшению результатов. Вероятно, при построении регрессионных моделей перспективно использовать вейвлеты, если судить по эффективности их применения в задачах сжатия изображений.

Рис. 3.16. Результат сегментации изображения, приведенного на рис. 3.15, *б*, усложненным алгоритмом, описывающим содержимое каждой области подходящей регрессионной моделью. Поскольку используется представление, адекватное сегментируемому изображению, и привлекается корректный принцип длины описания, результат сегментации оказывается правильным

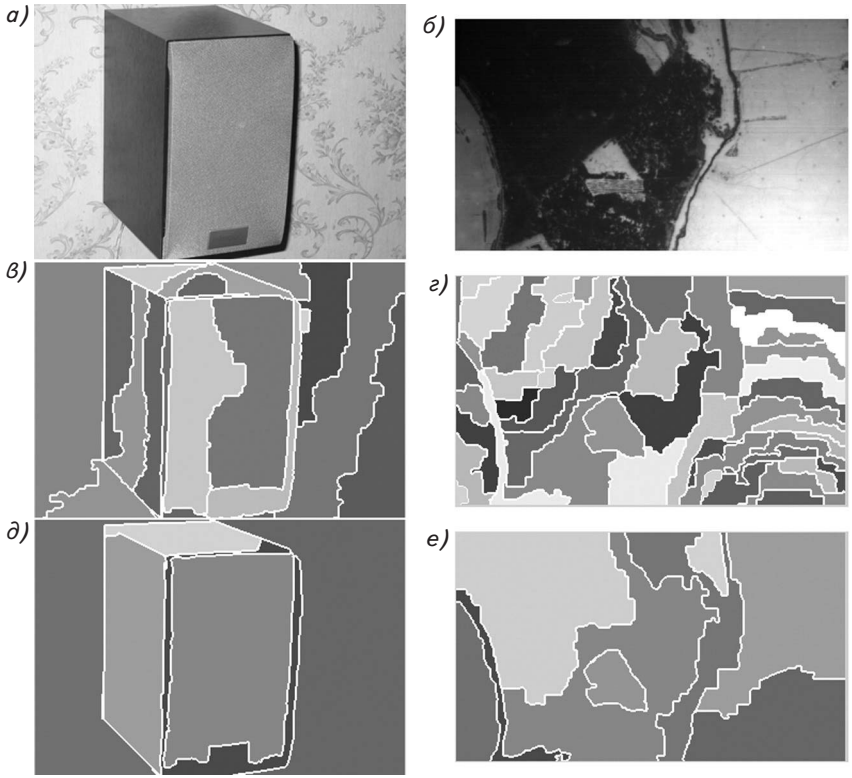

Рис. 3.17. Примеры реальных изображений двух типов, при сегментации которых необходимо использовать регрессионные модели для описания содержания областей: *а*, *б* — исходные изображения фрагмента трехмерной сцены и аэрокосмическое изображение заснеженного участка местности соответственно; *в*, *г* — результаты сегментации этих изображений без использования регрессионных моделей; *д*, *е* — результаты сегментации с использованием регрессионных моделей соответственно

Описание регулярных изменений яркости внутри областей также требует дополнительных исследований. Поскольку такие изменения вызваны неравномерностью освещения и изменениями ориентации поверхностей относительно наблюдателя и относительно источников освещения, то выявление регулярных изменений яркости видимых поверхностей может позволить восстановить информацию как об источниках освещения, так и о форме самих поверхностей. Однако для восстановления физических характеристик сцены необходимо привлекать не только эту, но и более высокоуровневую информацию, следующим этапом в извлечении которой является построение структурных элементов.

3.2.4. Построение структурных элементов на основе контурной информации

Границы областей, сформированных с помощью описанного выше алгоритма сегментации, могут рассматриваться как контуры, на основе которых строится промежуточное символьное представление изображений. Контуры, однако, могут быть получены и другими способами (см. п. 3.1.5). Рассмотрим задачу описания контуров как отдельную задачу, в которой контуры, выделенные каким-либо образом на изображении, являются входной информацией.

Будем считать, что каждый контур описывается отдельно от других, поэтому рассмотрим проблему анализа единственного контура δG, заданного последовательностью точек $\{(x_i, y_i)\}_{i=1}^{N}$, где $N = \|\delta G\|$ — общее число точек на данном контуре. Но такое задание контура не слишком информативно. Один из стандартных способов повысить информативность — представить контур в виде кривой, принадлежащей некоторому параметрическому семейству. Возникает вопрос о выборе этого семейства. Согласно пятому предположению Д. Марра о свойствах физического мира, границы видимых поверхностей почти всюду гладки, значит, контур целесообразно описывать кусочно-гладкой кривой, т. е. представлять как совокупность сегментов, для каждого из которых подбирается наиболее подходящая гладкая кривая (или регрессионная модель). Это классическая задача сегментации. Существуют и принципиально другие подходы к описанию контуров или областей на изображении (например, с помощью методов математической морфологии), которые находят применение в некоторых задачах компьютерного зрения. Мы, однако, для единообразия в решении различных задач ограничимся рассмотрением подхода на основе сегментации.

Каждый сегмент контура соотносится с непроизводным структурным элементом. Эти элементы могут быть различных типов в зависимости от того, кривой из какого параметрического семейства описан каждый из них. Дополнительным типом структурных элементов является угол (точка на контуре, разделяющая два сегмента) или соединение (если исходные контуры могут ветвиться). Совокупность всех видов кривых, используемых для описания сегментов, является алфавитом структурных элементов. Символы этого алфавита могут зависеть от предметной области. Например,

в задачах дактилоскопии, распознавания рукописного текста и некоторых биомедицинских приложениях могут использоваться различные структурные элементы. Тем не менее человек может научиться работать с любым из этих типов изображений. Определение общего алфавита структурных элементов для описания произвольных изображений — одна из центральных задач иконики.

При сегментации контуров с целью формирования структурных элементов возникают те же сложности, что и в общей задаче сегментации, а именно: выбор количества сегментов и выбор среди регрессионных моделей различной степени сложности для описания каждого сегмента. В задачах интерпретации изображений это проблемы выбора количества и типа структурных элементов.

Рассмотрим два крайних случая. Через N точек может быть проведена некоторая кривая, содержащая N свободных параметров, т. е. весь контур может быть представлен (с нулевой ошибкой) как один сегмент с соответствующей сложной моделью. Второй крайний случай — это разбиение контура на $N-1$ сегментов, каждый из которых состоит из двух точек, через которые проведена прямая (также с нулевой ошибкой). Оба эти варианта точно описывают контур, но они плохо применимы на практике, так как не выделяют на контуре значимой информации.

При создании практических решений часто руководствуются дополнительными эвристиками, не выводящимися из применяемого метода оценивания параметров. Иными словами, эти эвристики не обоснованы теоретически, а значит, весьма вероятно получение неоптимального решения. Одним из примеров таких эвристик является введение верхнего порога на среднеквадратичную ошибку (СКО). Если при описании сегмента контура с помощью отрезка прямой значение СКО оказывается больше заданного порога, то сегмент либо разбивается на два, либо описывается более сложной функцией (к примеру, дугой окружности). Универсального порога быть не может, следовательно, решение либо будет неадаптивным к содержанию изображения, либо будет требовать ручной настройки параметров для получения оптимальных результатов.

Воспользуемся принципом МДО для получения целевой функции, не требующей подобной ручной настройки параметров. Предположим, что есть отправитель и получатель сообщения. Пусть отправителю даны значения $\{(x_i, y_i)\}_{i=1}^{N}$,

описывающие контур δG. Эти значения необходимо передать получателю. Получателю координаты точек контура неизвестны, но он знает способ их представления, который используется отправителем (т. е. может произвести однозначное декодирование полученного сообщения).

Контур представляется в виде совокупности сегментов, каждый из которых описывается кривой из некоторого параметрического семейства. Чтобы отправитель мог точно восстановить форму контура, ему необходимо знать количество сегментов и для каждого из них знать тип структурного элемента, его параметры, а также неточности описания структурным элементом данного сегмента контура (невязки).

Рассмотрим стандартный случай, когда точки контура задаются с точностью до одного пикселя, а сам контур является восьмисвязным. Пусть у нас есть параметры кривой, описывающей сегмент контура. Передвигаясь по этой кривой, мы можем ставить точки на целочисленную сетку, восстанавливая контур. При постановке точки нужно учитывать для нее величину невязки, которая должна быть округлена до одного пикселя, а значит, есть возможность непосредственно считать энтропию невязок через их гистограмму. Значения коэффициентов кривой также могут задаваться не с абсолютной точностью, а так, чтобы их варьирование в некоторых пределах не приводило к смещению точек кривой на целочисленной сетке.

Рассмотрим j-й сегмент $\delta G^{(j)}$, состоящий из $\left\|\delta G^{(j)}\right\|$ точек. Пусть он описывается кривой с $n_p^{(j)}$ свободными параметрами $\vec{w}^{(j)}$, дающей невязки $r_i^{(j)}, i = 1, \ldots, \left\|\delta G^{(j)}\right\|$. Оценка длины описания сегмента состоит из следующих слагаемых:

• числа битов (обозначим через b), необходимых для обозначения типа структурного элемента (оценить эту величину для каждого типа элементов можно по ансамблю изображений исходя из принципа МДО, но здесь мы этот вопрос опускаем);

• длины описания параметров структурного элемента: $\frac{n_p^{(j)}}{2}\log_2\left\|\delta G^{(j)}\right\|$.

• длины описания невязок $r_i^{(j)}$ — величин отклонения структурного элемента от точек контура; считая невязки

независимыми отсчетами некоторой дискретной случайной величины $R^{(j)}$, эту длину описания можно посчитать как длину кода Хаффмана, оцененную как $\left\|\delta G^{(j)}\right\| H(R^{(j)})$, где $H(R^{(j)})$ — оценка (по гистограмме) энтропии величины $R^{(j)}$; помимо передачи закодированных невязок для декодирования также необходима таблица перекодировки, длину которой можно грубо оценить как $n_r^{(j)} \log_2 n_r^{(j)}$, где $n_r^{(j)}$ — количество различных величин невязок для данного сегмента.

Итак, общая длина описания будет равна

$$L = \sum_j L_j = \sum_j \left[b + \frac{n_p^{(j)}}{2} \log_2 \left\|\delta G^{(j)}\right\| + \left\|\delta G^{(j)}\right\| H(R^{(j)}) + n_r^{(j)} \log_2 n_r^{(j)} \right]. \tag{3.13}$$

В действительности значение L, вычисленное для каждого контура, необходимо подставить в уравнение (3.11) вместо слагаемых $\|\delta G_k\| \log_2 N_{dir}$ [именно так была грубо оценена длина описания границы k-й области $L(\delta G_k)$ в задаче сегментации изображения] и минимизировать полученную таким образом целевую функцию. Последовательное и независимое решение задачи сегментации и задачи построения структурного описания контуров — это приближение, которое приводит к потере части полезной информации, а именно так работает большинство систем интерпретации изображений. Пока воспользуемся этим приближением, т. е. рассмотрим задачу сегментации априорно заданных контуров на основе целевой функции (3.13), но в п. 3.5.4 вернемся к вопросу об уточнении результатов сегментации во время построения структурного описания.

В качестве алфавита структурных элементов используем отрезки прямых линий, дуги окружностей и произвольные кривые второго порядка. Кривые более высоких степеней не привлекаются в работах по структурному описанию изображений — опыт специалистов в области компьютерного зрения говорит о нецелесообразности их использования для описания структуры изображения. Можно привести и другие аргументы в пользу этого утверждения. Кривые третьего и более высоких порядков содержат большое число параметров, и с точки зрения принципа МДО они могут быть использованы для описания только достаточно длинных контуров с соответствующей формой, причем нет основания

считать, что существуют природные процессы, порождающие такие контуры. Выбор между алфавитами структурных элементов может осуществляться на основе принципа МДО: если добавление структурного элемента нового типа позволяет уменьшить длину описания изображений, усредненную по некоторой репрезентативной выборке, то расширение алфавита путем включения этого типа элементов является целесообразным.

Алгоритм построения структурных элементов полностью аналогичен алгоритму сегментации изображения, описанному выше, за тем исключением, что он применяется в одномерном случае. Этот алгоритм состоит из последовательного слияния соседних сегментов контура, которые изначально состояли из одной точки. При проверке, следует ли объединять два соседних сегмента, из них формируется сегмент контура, через точки которого проводятся прямая, окружность и произвольная кривая второго порядка. Для каждой из этих регрессионных моделей трех типов определяется гистограмма невязок, с помощью которой вычисляется их энтропия. Из регрессионных моделей выбирается та, которая дает минимальную длину описания (с учетом энтропии невязок, числа точек в сегменте и числа параметров в модели), т. е. одновременно с подсчетом длины описания определяется и тип структурного элемента. Длина описания объединенного сегмента сравнивается с суммой длин описания двух сегментов, из которых он был сформирован. Если объединение приводит к уменьшению длины описания, то результат слияния сегментов сохраняется, а соответствующая длина описания запоминается.

Рис. 3.18. Результаты сегментации границ областей, представленных на рис. 3.17 (*а*) и 3.10, *г* (*б*). Поскольку существуют точки, являющиеся граничными для трех областей, то при сегментации появляются Т-образные соединения, которые можно выделить в отдельный тип структурных элементов

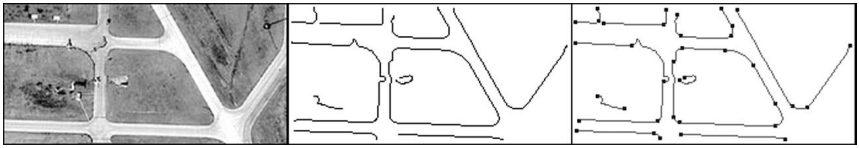

Рис. 3.19. Результат сегментации контуров, полученных с помощью фильтрации Дерише

После того как не останется пар сегментов, объединение которых приводит к уменьшению длины описания, выполняется следующий шаг алгоритма, аналогичный шагу 4 алгоритма сегментации изображений. На этом шаге проверяется возможность отнесения точек, краевых для данного сегмента, соседним сегментам.

На рис. 3.18, *а*, *б* и 3.19 представлены результаты применения алгоритма сегментации контуров для границ областей, построенных алгоритмом, описанным в п. 3.2.3, и для контуров, извлеченных с помощью фильтрации Дериша [243] соответственно. Структурные элементы, соответствующие этим сегментам контуров, не являются полностью не зависимыми друг от друга. Учет взаимосвязей между элементами может быть использован для формирования составных структурных элементов и для уменьшения общей длины описания за счет извлечения общей информации в описаниях отдельных структурных элементов.

Отметим, что алгоритмы сегментации изображений и контуров очень похожи (за исключением размерности данных) и могут быть реализованы в виде одной и той же вычислительной процедуры. Такой подход к сегментации можно использовать не только в анализе изображений, но и в других задачах, в которых имеются численные данные, распределенные на регулярной сетке. То, что в интерпретации изображений используются модально неспецифичные алгоритмы, весьма важно, поскольку позволяет надеяться на возможность создания библиотеки подобных алгоритмов, которая затем может быть использована для решения различных задач.

Сформированные структурные элементы далее подвергаются группированию с целью формирования составных структурных элементов. Этот процесс также может быть рассмотрен с информационной точки зрения.

3.2.5. Формирование составных структурных элементов

Составные структурные элементы являются еще более уникальными, чем непроизводные элементы и, тем более, чем точки контуров. В связи с этим составные элементы гораздо легче отождествить между собой на изображениях, что делает их привлекательными для распознавания и совмещения изображений или выявления изменений. В то же время эта уникальность накладывает и сильные ограничения на необходимую надежность обнаружения таких элементов.

Идея составных структурных элементов тесно связана с ранними работами по синтаксическому анализу изображений (несложно провести параллель между составными структурными элементами и нетерминальными символами в формальных грамматиках). Однако в синтаксическом подходе предполагалось, что непроизводные элементы на изображении выделяются безошибочно и их взаимное расположение строго детерминировано (и даже в рамках стохастических грамматик эти предположения ослаблялись лишь незначительно). Естественно, подобный подход оказался недостаточно гибким для изображений реальных сцен.

Более обоснованный подход разрабатывался Д. Марром в его информационной теории зрения [235]. К сожалению, он во многом остался лишь неформальным словесным описанием общих идей, касающихся того, каким образом должны группироваться структурные элементы. Здесь мы попробуем установить теоретическую основу для развития этих идей. Третье и четвертое предположения Д. Марра говорят о том, что структурные элементы могут обладать «внутренним» подобием и располагаться в пространстве таким образом, чтобы образовывать некоторые регулярные конфигурации. Группирование подобных структурных элементов используется на практике для формирования составных структурных элементов, однако строго понятие подобия не вводится.

Поскольку структурные элементы задаются набором (вектором) признаков, степень сходства определяется расстоянием в пространстве признаков между соответствующими векторами. Метрика в этом пространстве часто задается эвристически, т. е. третье предположение Д. Марра говорит о необходимости группирования элементов, но не указывает критерий этого группирования.

Группа структурных элементов не только в пространстве может образовывать регулярную конфигурацию, но и некоторый их признак (например, ориентация) может меняться регулярным образом, о чем говорит четвертое предположение Д. Марра. В связи с этим сравнения векторов признаков двух структурных элементов не всегда может быть достаточно, чтобы определить, следует ли их группировать (рис. 3.20, *а*, *б*).

Покажем возможность введения строгого критерия группирования структурных элементов на основе принципа МДО. Сходство элементов будем оценивать по взаимной информации между их описаниями. Регулярность в расположении или регулярное изменение каких-то других характеристик в группе структурных элементов также естественно оценивать по длине описания.

Учет взаимной информации в описании структурных элементов при их группировании может осуществляться по двум различным схемам. Представим, что есть несколько контуров, описывающихся прямыми линиями с примерно одинаковой ориентацией. И пусть эти элементы объединяются в группу. Если рассматривать их ориентации как отвлеченные числа, то для группы структурных элементов можно хранить среднее значение ориентации, а для каждого элемента в группе — отклонение его ориентации от среднего значения. Если разброс ориентаций небольшой, то такой способ представления приведет к уменьшению общей длины описания. Однако для этого необходимо, чтобы число объединяемых в группу элементов было не слишком мало.

Второй способ учета взаимной информации в описаниях элементов заключается в том, чтобы (в рамках данного примера) назначить всем элементам, объединяемым в груп-

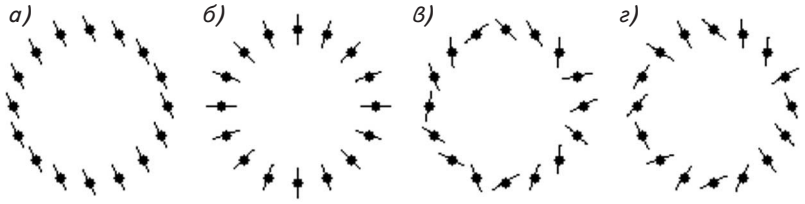

Рис. 3.20. Примеры влияния взаимной зависимости элементов на процесс их группирования: *а*, *б* — группы элементов, признаки которых меняются регулярным образом, но по различным функциональным зависимостям; *в*, *г* — группы элементов, не имеющие связи между своими признаками

пу, одинаковые ориентации. При этом отпадает необходимость описывать отклонения ориентаций элементов от среднего значения, но, поскольку меняются сами отрезки, ухудшается точность, с которой они аппроксимируют соответствующие сегменты контуров. Если увеличение энтропии невязок компенсируется уменьшением числа параметров, то может быть принято решение об объединении структурных элементов. В рамках данной схемы число объединяемых структурных элементов может быть небольшим. Например, две почти параллельные линии могут быть объединены с формированием составного элемента «две параллельные линии», т. е. будет принято решение, что два отрезка не просто похожи по ориентации, а действительно параллельны. Это, однако, произойдет только в том случае, если отклонение от параллельности вызвано случайными искажениями контуров, а не их систематическим расхождением.

Рассмотрим обе схемы выявления взаимной информации в структурных элементах.

Группирование элементов по их подобию и регулярности расположения. Пусть есть набор однотипных структурных элементов $\{(x_i, y_i, \vec{z}_i)\}_{i=1}^{N}$, где (x_i, y_i) — координаты i-го структурного элемента; $\vec{z}_i$ — вектор его дополнительных параметров (например, ориентация и длина для отрезка прямой, радиус для пятна и т. д.). В принципе, координаты элемента можно было бы включить в общий вектор параметров (объединив третье и четвертое предположения Д. Марра в одно), но мы этого делать не будем и для большей наглядности ограничимся одним дополнительным признаком со значением z_i для i-го элемента. Обобщение на случай нескольких признаков не составляет принципиальной трудности, если верно предположение, что эти признаки являются независимыми. Описываемый подход также можно обобщить и на группирование разнотипных структурных элементов, введя тип элемента как дополнительный компонент вектора признаков. При группировании разнотипных элементов имеет смысл объединять описание только тех признаков, которые имеют один и тот же смысл (например, координаты на плоскости), что делает группирование разнотипных элементов не столь эффективным, как группирование однотипных элементов (но не исключает возможности такого группирования, рис. 3.21), а также усложняет представление изображений.

Рис. 3.21. Иллюстрация к возможности группирования зрительной системой человека разнотипных элементов на основе регулярностей в их взаимном расположении, благодаря чему достигается возможность распознавания изображенного на рисунке объекта

Оценим длину описания структурных элементов в случае отсутствия группирования. Самый простой способ описания заключается в представлении каждого элемента вектором признаков (x_i, y_i, z_i) не зависимым от других элементов образом. Однако он не дает адекватной оценки длины описания. Вместо этого следует построить минимальное остовое дерево и хранить не абсолютные координаты (x_i, y_i) элементов, а приращения их координат относительно ближайшего элемента в остовом дереве [267, гл. 3].

Чтобы не строить остовое дерево (эта процедура весьма ресурсоемка), можно воспользоваться следующей грубой оценкой. Пусть для каждого структурного элемента определено расстояние d_i в плоскости изображения до ближайшего элемента. И пусть $d = \frac{1}{N}\sum_{i=1}^{N} d_i$ — среднее расстояние между ближайшими элементами. Величина $\Delta d = \left(\overline{(d-d_i)^2}\right)^{1/2} \Big/ \sqrt{2}$ является оценкой среднеквадратичного отклонения (по каждой из координат x, y) расстояния между ближайшими структурными элементами от среднего значения. Тогда для описания положения структурных элементов требуется примерно $2N \log_2 \Delta d$ бит информации. Это достаточно грубая оценка, но для реальных изображений она является удовлетворительной.

Для описания прочих параметров требуется $NH\left(\{z_i\}_{i=1}^{N}\right)$ бит информации, где энтропию $H\left(\{z_i\}_{i=1}^{N}\right)$ считаем возможным оценить через гистограмму значений параметра z, который рассматриваем как дискретную величину. Для описания таблицы перекодировки требуется дополнительно $n_z \log_2 n_z$ бит, где n_z — количество различных значений па-

раметра z. Общую длину описания набора структурных элементов, не разбитых на группы, можно оценить как

$$L_{xy}^{(0)} = 2N \log_2 \Delta d + NH\left(\{z_i\}_{i=1}^{N}\right) + n_z \log_2 n_z, \qquad (3.14)$$

причем на i-й структурный элемент приходится примерно $2\log_2 \Delta d - \log_2 P(z = z_i)$ бит.

Рассмотрим теперь длину описания упорядоченной группы из M структурных элементов $\{(x_i, y_i, z_i)\}_{i=1}^{M}$, являющихся подмножеством исходного набора. В отличие от предыдущего случая, здесь предполагается, что в расположении структурных элементов присутствует некоторая регулярность. Иными словами, исходя из информации о положении предыдущих элементов, можно предсказать положение следующих элементов. Будем разделять предсказание по направлению к следующему элементу и предсказание по расстоянию до него.

Простейший способ предсказания положения следующего элемента использует положения двух предыдущих элементов, т. е. вместо координат (x_i, y_i) нужно получить значения $(\Delta r_i, \Delta h_i)$ (рис. 3.22). Эти значения определены для $M - 2$ элементов, для первых же двух элементов необходимо информацию об их координатах представлять так же, как и в случае отсутствия группирования.

Тогда оценка длины описания координат структурных элементов в группе (состоящей не менее чем из трех элементов) будет

$$L_{xy} = 4\log_2 \Delta d + (M-2)\times$$
$$\times\left[\log_2\left(\frac{1}{M-2}\left(\sum_{i=1}^{M-1}\Delta r_i^2\right)^{1/2}\right) + \log_2\left(\frac{1}{M-2}\left(\sum_{i=1}^{M-1}\Delta h_i^2\right)^{1/2}\right)\right]. \qquad (3.15)$$

Очевидно, такая простая модель отдает предпочтение группированию структурных элементов, расположенных на

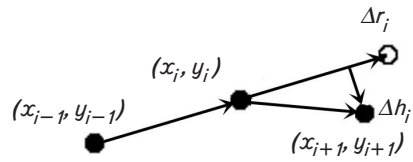

Рис. 3.22. Описание положения элемента через его отклонение $(\Delta r_i, \Delta h_i)$ от предсказанного положения

прямой линии с постоянным интервалом. В случае, когда элементы располагаются на окружности, значения $(\Delta r_i, \Delta h_i)$ будут отличны от нуля, но будут приблизительно одинаковы для любого i. Пусть Δr и Δh — средние значения ошибок. Вместо того чтобы описывать сами ошибки $(\Delta r_i, \Delta h_i)$, можно описывать их отклонения от средних значений $(\Delta r_i - \Delta r, \Delta h_i - \Delta h)$, но при этом для группы необходимо также описать и средние значения ошибок. Это аналогично тому, чтобы отдельно описать положения не первых двух, а первых трех точек. Тогда для такой модели можно выполнить следующую оценку длины описания:

$$L_{xy} = 6\log_2 \Delta d + (M-2)\times$$
$$\times\left\{\log_2\left[\frac{1}{M-2}\left(\sum_{i=1}^{M-1}(\Delta r_i - \Delta r)^2\right)^{1/2}\right] + \log_2\left[\frac{1}{M-2}\left(\sum_{i=1}^{M-1}(\Delta h_i - \Delta h)^2\right)^{1/2}\right]\right\}. \quad (3.16)$$

Определяя две длины описания, можно сделать выбор между расположением структурных элементов на прямой и на окружности (как это было в случае точек контура; разница лишь в том, что тогда расстояние между точками было фиксированным). Такой выбор можно делать отдельно для ошибок Δr и Δh.

Эту схему можно расширять и далее, делая представление более богатым. Также можно построить несколько иное представление, в котором модель кривой будет задаваться в явном виде, хотя принципиального изменения в результатах группирования это не вызовет. В представление следует включить и возможность описания пропущенных структурных элементов.

Рассмотрим теперь вторую составляющую, связанную с описанием дополнительных признаков структурных элементов.

Как и в случае совместного описания структурных элементов, эта длина описания выражается как $MH\left(\{z_i\}_{i=1}^{M}\right)$. Различие заключается в том, что здесь энтропия $H\left(\{z_i\}_{i=1}^{M}\right)$ определяется по выборке из M элементов, входящих в группу, а не по всей совокупности элементов. Если в группу объединяются элементы с совпадающими признаками, то данная энтропия будет равна нулю. Если производится по-

пытка объединить в группу случайные элементы, то энтропия значений признаков внутри группы будет соответствовать энтропии признаков по всей совокупности, а значит, выигрыша в длине описания не будет. Поскольку для каждой группы характерно свое распределение признаков структурных элементов, то помимо закодированных признаков необходимо учитывать и длину таблицы перекодировки, для представления которой необходимо примерно $n_z \log_2 n_z$ бит информации (здесь n_z — число различных значений признаков в данной группе). Тогда для описания признаков элементов внутри группы нужно $L_z = MH\left(\{z_i\}_{i=1}^{M}\right) + n_z \log_2 n_z$ бит информации.

Здесь также есть возможность дальнейшего расширения представления, которая заключается в том, чтобы предсказывать значения признаков z_i в зависимости от координат (x_i, y_i) или номера данного элемента i в упорядоченной группе. Человек способен обнаруживать подобные закономерности (см. рис. 3.20), но здесь мы приведем критерий без учета возможных взаимосвязей между дополнительными признаками структурных элементов.

Этот критерий, определяющий, является ли формирование данной группы выгодным или нет, имеет вид

$$\Delta L = \left(\left[H\left(\{z_i\}_{i=1}^{N}\right) - H\left(\{z_i\}_{i=1}^{M}\right)\right]M - n_z \log_2 n_z\right) + \\ + \left(2M \log_2 \Delta d - L_{xy}\right) \qquad (3.17)$$

и представляет собой выигрыш в длине описания, возникающий в результате формирования группы. Если он больше нуля, то группу формировать целесообразно, в противном случае — нет.

Рассмотрим теперь возможный алгоритм группирования. Очевидно, перебирать все возможные группы невозможно с вычислительной точки зрения, поэтому перебор различных вариантов группирования должен быть направленным. Для этого можно применить следующую схему:

- для каждого структурного элемента проверяется некоторое количество наиболее близких к нему элементов, с каждым из которых формируется пара, рассматриваемая в качестве начального фрагмента группы структурных элементов;
- для каждой такой пары предпринимается попытка ее расширения, для чего исследуется возможность продолже-

ния исходного отрезка в обе стороны (предпринимается попытка предсказать положение следующего и предыдущего структурных элементов в группе исходя из положения элементов данной пары). Выбирается лучшее продолжение [с точки зрения максимизации выражения (3.17)], и расширение продолжается до тех пор, пока добавление новых элементов увеличивает значение ΔL;

• из всех построенных групп, содержащих более трех элементов, выбирается группа, дающая максимальное значение ΔL (3.17); структурные элементы, входящие в эту группу, исключаются из рассмотрения, и для групп, содержащих исключенные элементы, заново выполняется предыдущий шаг.

Приведем два примера, показывающих работу алгоритма. На рис. 3.23 представлен результат поиска групп среди случайно расположенных структурных элементов (признаки элемента — координаты и размер, в связи с чем элементы трактуются как пятна). Поскольку среди случайных элементов не выделено групп, кроме реально присутствующей, то можно считать, что критерий на основе принципа МДО был выбран корректно (несмотря на некоторые упрощения) и алгоритм группирования работает правильно.

Другой тест был проведен на изображении реальной сцены (спутниковой фотографии поверхности Земли). В каче-

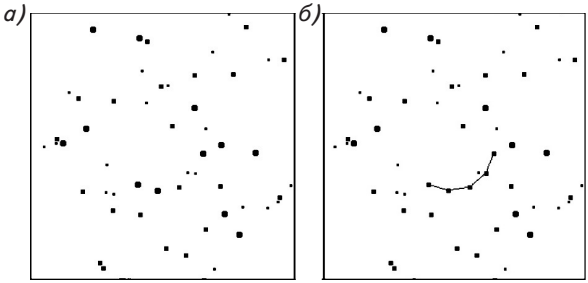

Рис. 3.23. Результат поиска групп среди искусственно сформированных структурных элементов (положения и размеры присвоены элементам случайным образом, за исключением пяти структурных элементов, образующих дугу окружности): *а* — элементам присвоены случайные размеры, и их группирование оказывается нерациональным с точки зрения длины описания (заметим, что и визуально трудно выделить эту группу); *б* — элементам присвоены одинаковые размеры, и группа успешно выделяется (случайно расположенные элементы не объединяются)

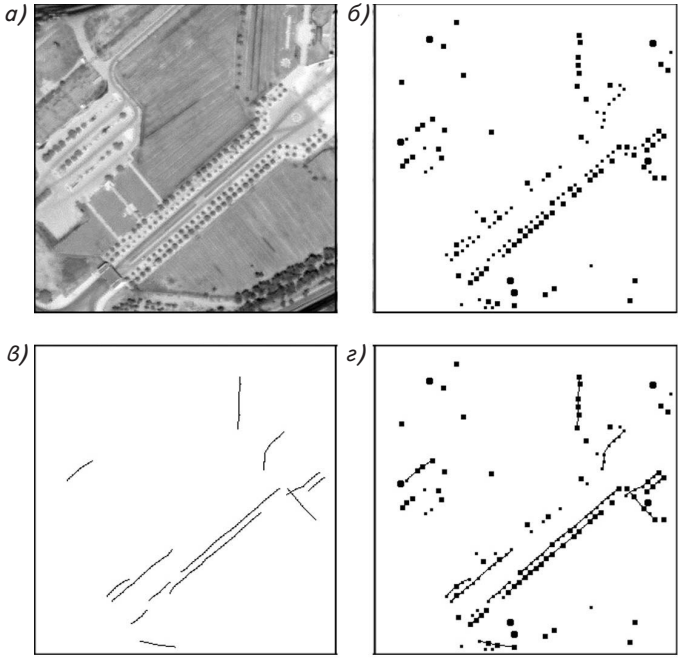

Рис. 3.24. Результат группирования структурных элементов, извлеченных из реального изображения: *а* — исходное изображение; *б* — пятна, выделенные простым алгоритмом детектирования; *в* — линии, показывающие построенные группы пятен; *г* — пятна, объединенные в группы (указаны соединительными линиями)

стве структурных элементов выступали небольшие пятна, соответствующие отдельно стоящим деревьям. На рис. 3.24 представлен результат группирования этих пятен. Несмотря на то что для их выделения применялся сравнительно простой и ненадежный алгоритм, группирование было осуществлено вполне успешно.

Формирование составных элементов. Группирование небольшого числа структурных элементов, параметры которых оказываются близкими, но не идентичными, оказывается неэффективным с точки зрения целевой функции (3.17). Это вовсе не является ошибкой. Действительно, вероятность случайно встретить, скажем, два элемента с близким значением некоторого признака довольно высока. Тем не менее объединение малого числа элементов возможно, если их параметры считать идентичными. Тогда число признаков при объединении элементов уменьшается (не нуж-

но хранить числа, отвечающие отклонениям признаков от среднего значения в группе). Однако, если отправитель включает в сообщение только информацию о средней ориентации для группы, для точного восстановления контуров получателю должны быть известны отклонения положений точек контура от положений, задаваемых структурными элементами с усредненными параметрами. Это означает, что в процессе такого группирования должна осуществляться коррекция самих структурных элементов. Для определения качества, с которым скорректированные элементы описывают контуры, необходимо обращаться к уровню построения структурных элементов.

Выше мы рассмотрели информационный критерий формирования структурных элементов на основе контуров. Поэтому здесь рассмотрим группирование этих же структурных элементов, так как необходим возврат на уровень ниже.

Способ формирования составных элементов в точности такой же, как и при сегментации контуров: при последовательном слиянии сегментов контура формировался тестовый сегмент, для которого определялся тип структурного элемента и оценивалась длина описания. При этом могли быть объединены два сегмента, которым до объединения соответствовали два отрезка, а после объединения — более длинный отрезок или дуга окружности. Здесь ситуация в точности такая же, но используются более сложные модели, и при объединении может потребоваться одновременное рассмотрение не пары, а нескольких структурных элементов, не обязательно находящихся на одном и том же контуре.

Для оценки как длины описания исходных элементов, подвергаемых объединению, так и полученного составного структурного элемента, может быть использована целевая функция (3.13), вернее, одно из ее слагаемых под знаком суммы. Рассмотрим процесс группирования на примере двух почти параллельных прямых. Параллельность означает, что при совместном описании двух прямых требуется на один параметр меньше, чем при их раздельном описании. Усреднив параметр ориентации, можно построить совместную гистограмму невязок для двух прямых и вычислить соответствующую энтропию. Если увеличение энтропии компенсируется уменьшением числа параметров, то такие прямые объединяются в группу и описываются как точно параллельные.

Аналогичным образом могут быть сформированы и другие геометрические примитивы, такие, как перпендикулярные прямые, квадраты, прямоугольники или параллелограммы (или их части в форме П), равнобедренные или равносторонние треугольники и т. д. Адекватный список может быть составлен постепенным добавлением геометрических примитивов в представление при условии, что это приводит к уменьшению средней длины описания для ансамбля изображений. Возможно, системой машинного зрения в процессе функционирования должно осуществляться автоматическое обучение более сложным геометрическим примитивам по мере работы с новыми изображениями. Однако здесь в качестве примера будет приведен лишь алгоритм для принятия решения об объединении структурных элементов на примере достаточно простых геометрических примитивов. Ограничимся такими составными элементами, как группы, состоящие из произвольного числа параллельных и перпендикулярных линий. Для формирования этих групп воспользуемся следующим алгоритмом.

Для каждого структурного элемента просматривается некоторая его окрестность и выбираются кандидаты для объединения в группу, которые являются либо почти параллельными, либо почти перпендикулярными данному элементу. Размер просматриваемой окрестности и допуск на параллельность являются параметрами алгоритма поиска, задающими компромисс между скоростью и точностью группирования. Далее каждая из этих гипотез проверяется, т. е. предпринимается попытка сформировать составной элемент (параллельные или перпендикулярные прямые). Если эта попытка оказывается удачной (в смысле уменьшения длины описания), то предпринимается попытка присоединить к данной группе другие элементы. Для параллельных и перпендикулярных элементов в группе хранится одна и та же ориентация, но для каждого элемента указывается, совпадает ли его ориентация с данной или отличается на 90°.

На рис. 3.25, *а*, *б* приведен пример выделения групп параллельных и перпендикулярных линий на аэрокосмическом снимке.

Такой алгоритм группирования может использоваться и для формирования составных структурных элементов на основе уже построенных групп элементов. К примеру, некоторые группы пятен, представленные на рис. 3.24, *г*, могут быть далее объединены как параллельные группы. Это

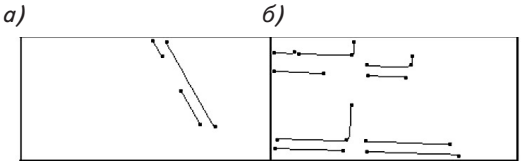

Рис. 3.25. Две группы параллельных (*а*) и перпендикулярных (*б*) линий, выделенных на основе сегментов контуров, представленных на рис. 3.19

приводит к идее иерархического группирования структурных элементов, позволяющего выявлять все более сложные взаимосвязи между ними. В рамках такого представления структура изображений будет описываться на довольно высоком уровне абстракции (см. рис. 3.21), от которого можно уже будет переходить к описанию трехмерной организации сцены и распознаванию сложных объектов.

На примере задачи группирования структурных элементов интересно было бы узнать, насколько согласуется перцептивное группирование, осуществляемое зрительной системой человека, с принципом МДО. В гл. 1 были приведе-

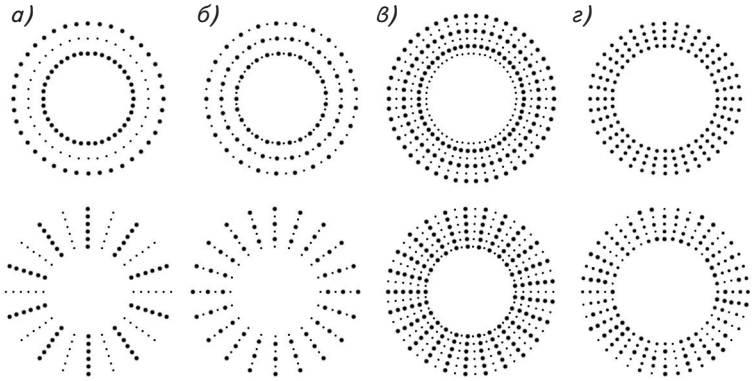

Рис. 3.26. Конкурирующие пространственные конфигурации: *а* — выбор между двумя способами группирования достаточно очевиден, потому что он поддерживается как взаимным расположением, так и размерами пятен; *б* — в группы объединяются пятна разных размеров, поскольку регулярности во взаимном расположении доминируют; *в* — выбор способа группирования опирается на сходство размеров пятен, так как по расположению нельзя отдать предпочтение одному из них; *г* — выбор способа группирования неоднозначен, и виден слабый эффект конкурирующих конфигураций

ны ссылки на работу, исследующую наличие этого соответствия на уровне простых структур [45]. Существуют широкие возможности по развитию подобных исследований.

В качестве одного из примеров можно привести изучение восприятия конкурирующих пространственных конфигураций. С информационной точки зрения такие конфигурации соответствуют ситуации, в которой описание структуры можно осуществить различными способами, причем длины соответствующих описаний примерно одинаковы. Значит, можно предсказать, какие конфигурации будут восприниматься человеком как конкурирующие, а какие будут восприниматься однозначно. На рис. 3.26 представлено несколько вариантов расположения пятен двух размеров. Видно, что при изменении расположения пятен меняется способ их группирования. Неоднозначность способа группирования создает впечатление конкурирующих пространственных конфигураций (характерный пример таких конфигураций приведен в работе [235, с. 63]). Уровень, когда конкуренция возникает, достаточно точно соответствует тому, что два способа группирования приводят к примерно одинаковой длине описания. К сожалению, детальное изучение вопросов психологии восприятия выходит далеко за рамки данной технической книги, а представленный здесь результат требует более детальной проверки. Тем не менее рискнем предположить, что использование принципа МДО может оказаться весьма продуктивным для предсказания и исследования различных феноменов зрительного восприятия человека.

3.2.6. Пример практического приложения: совмещение изображений

Сопоставление изображений, заключающееся в отождествлении на них идентичных элементов, является принципиальным для решения множества задач компьютерного зрения. Например, при распознавании объектов их изображения необходимо сравнить с изображениями некоторых эталонных объектов и из всех эталонов выбрать наилучший либо указать на отсутствие подходящего эталона. При выявлении изменений в результате сопоставления двух изображений нужно не только определить степень их сходства, но и ответить на вопрос о соответствии отдельных эле-

ментов пары изображений. В других задачах сопоставление сопровождается не выделением различий, выражающихся в наличии или отсутствии каких-то объектов на изображении, а измерением некоторых параметров объектов, таких как ориентация, положение или размеры. Таким образом, сопоставление изображений является неотъемлемым компонентом практически любой системы компьютерного зрения, что делает эту задачу крайне важной как в теоретическом, так и в практическом аспектах.

Очень часто анализируемые изображения оказываются полученными с разных ракурсов. Для сопоставления таких изображений возможно два принципиальных подхода: либо использовать представления изображений, инвариантные пространственному преобразованию, либо восстанавливать взаимное преобразование изображений в явном виде. Как правило, при построении инвариантного описания изображения (например, посредством метода статистических моментов) теряется и некоторая полезная информация. Более того, использование этого подхода затруднительно для задач выявления изменений или измерения параметров объектов; оно более оправданно в ограниченных задачах распознавания целей.

При явном восстановлении параметров взаимного пространственного преобразования задача сопоставления изображений оказывается сопряженной с задачей их совмещения, т. е. приведения изображений в единую систему координат. Совмещение изображений оказывается необходимым этапом при решении таких задач, как выявление изменений на серии снимков, синтез панорамных снимков, дополнение информации в одном снимке данными из другого снимка (это может быть полезно, если снимки получены в разных спектральных диапазонах или если нужно восстановить информацию в затененных и загороженных областях) и т. д.

Задача сопоставления изображений обычно осложняется тем, что смена ракурса является не единственной причиной различия изображений. В дополнение к этому на изображениях могут присутствовать различия, вызванные шумами, изменением освещения, сменой типа сенсора (часто возникает потребность в совмещении, к примеру, оптических и радиолокационных снимков), собственной изменчивостью объектов на сцене (например, уже упоминавшейся сезонно-суточной изменчивостью). Построение методов

сопоставления и совмещения, не чувствительных к различным типам изменчивости, является весьма актуальной задачей.

Проблеме совмещения изображений было посвящено множество работ, и для ее решения было предложено достаточно большое количество методов (см., например, [249, 321–327]), которые различаются по следующим компонентам [321]:

1) типу допустимого пространственного преобразования;

2) типу сопоставляемых элементов изображения (или типу используемого представления);

3) стратегии поиска оптимальных параметров преобразования, основанной на критерии качества и оптимизационном алгоритме.

Тип пространственного преобразования определяется природой совмещаемых изображений. Допустимое пространственное преобразование описывается одним из следующих способов: глобальным преобразованием [324, 328], задающим общее отображение всей площади одного изображения на другое, и полем смещений [326, 327], определяющим собственный сдвиг для каждой точки изображения. В качестве глобального преобразования может выступать преобразование из группы движения, преобразование подобия, аффинное или проективное преобразования и др. Поле смещений обычно используется в тех случаях, когда глобальное преобразование отсутствует, а сами смещения не слишком велики. Это характерно для задач стереозрения [326] и некоторых биомедицинских приложений [325, 329].

В то время как тип пространственного преобразования зависит от задачи, выбор представления изображений в достаточной степени произволен. Существуют методы сопоставления, использующие сами пиксели с соответствующими им значениями интенсивности [292, 329]; методы, осуществляющие поиск соответствия между точками контуров или краевыми точками [248, 324], различными структурными или геометрическими элементами [322, 323], а также между метками, обозначающими конкретные физические объекты [330]. Таким образом, в задаче совмещения изображений привлекаются все те представления, о которых шла речь выше. В случаях, когда совмещаемые изображения идентичны с точностью до взаимного пространственного преобразования, допустимо привлечение достаточно простых представлений. Однако в общем случае использова-

ние представлений, инвариантных условиям съемки, является ключевым для успешной работы алгоритмов совмещения. В п. 3.1 обосновывалось использование иерархических структурных представлений как обладающих требуемым свойством инвариантности, а в п. 3.2 приводилось описание теоретико-информационного подхода к построению представления такого типа. Рассмотренное выше представление действительно может быть успешно применено для решения задачи совмещения изображений [331, 332].

Использование принципа МДО в задаче совмещения не ограничивается лишь построением представления изображений на его основе. Этот принцип может помочь и в выборе третьей компоненты методов совмещения, а именно: при выводе критерия качества совмещения. Действительно, совмещение изображений можно рассматривать с точки зрения минимизации суммарной длины описания двух изображений [63, 267, 333]. Если описания изображений содержат одинаковые элементы, то (при условии, что эти элементы отождествлены) при совместном описании изображений совпадающие элементы могут описываться только один раз. Чем больше элементов в описаниях изображений отождествлено, тем меньше будет суммарная длина описания. В общем виде информационный критерий качества совмещения можно записать так:

$$L(f_1, f_2 \circ T) = L(f_1) + L(f_2 \circ T) - I(f_1, f_2 \circ T), \qquad (3.18)$$

где $T : G \to G$ — пространственное преобразование, действующее в области G, на которой задано изображение; $f_2 \circ T$ — преобразованное второе изображение; $L(f_1)$ — длина описания изображения, вычисление которой подробно разбиралось в п. 3.2; $I(f_1, f_2 \circ T)$ — количество взаимной информации в первом и преобразованном втором изображениях.

При использовании простейшего (пиксельного) представления изображений длина описания изображения может быть оценена через энтропию интенсивностей его пикселей [63, 334]. Несложно заметить, что именно такой случай рассматривался в п. 1.4.4 для одномерного сигнала и преобразования сдвига, где также отмечалось, что, согласно принципу МДО, при совместном описании двух наборов данных необходимо не только извлекать взаимную информацию из них, но и минимизировать длину описания каждого набора. В работах [266, 267] взаимную информацию неиерархических структурных описаний, совмещенных некоторым

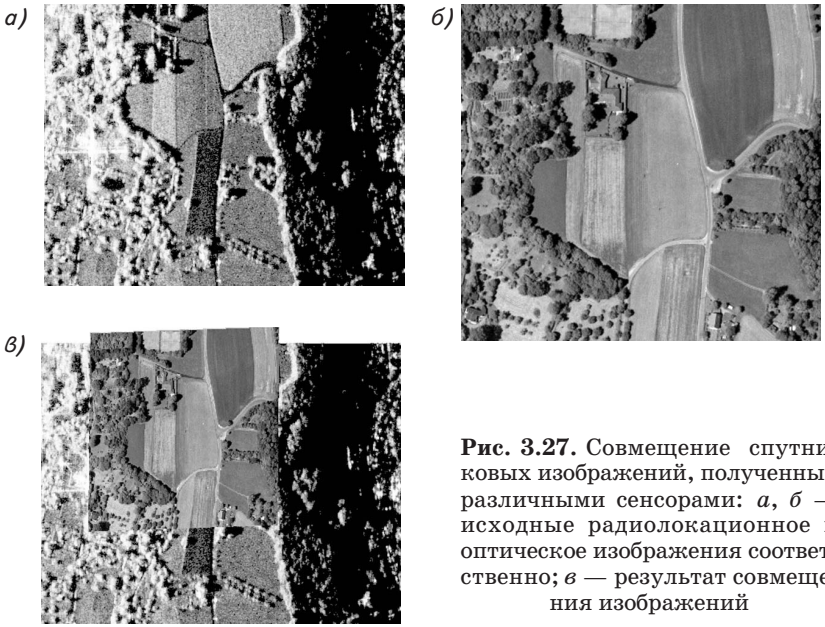

Рис. 3.27. Совмещение спутниковых изображений, полученных различными сенсорами: *а*, *б* — исходные радиолокационное и оптическое изображения соответственно; *в* — результат совмещения изображений

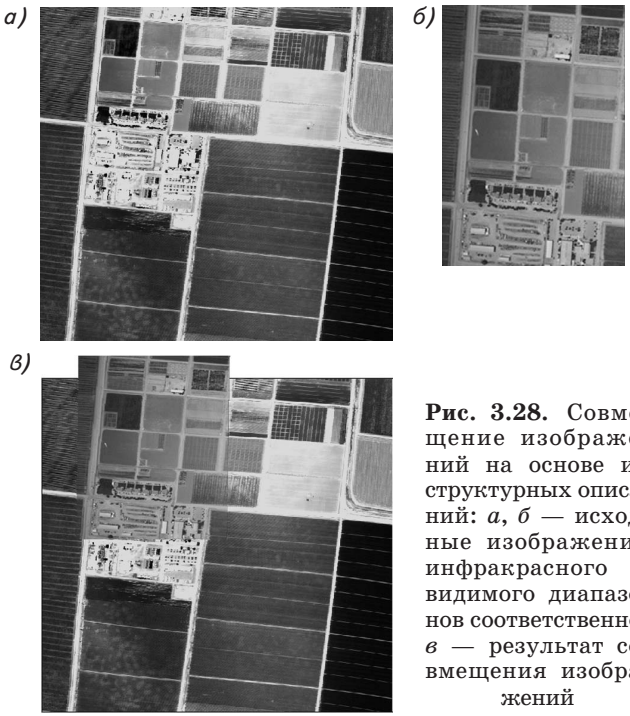

Рис. 3.28. Совмещение изображений на основе их структурных описаний: *а*, *б* — исходные изображения инфракрасного и видимого диапазонов соответственно; *в* — результат совмещения изображений

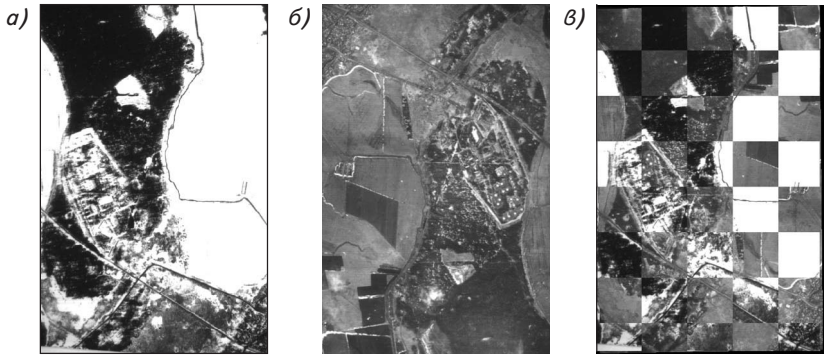

Рис. 3.29. Совмещение спутниковых изображений, полученных в разные сезоны: *а*, *б* — исходные изображения; *в* — результат совмещения изображений (в виде мозаики) после корректного структурного сопоставления

пространственным преобразованием, предлагается оценивать через оценивание длины описания минимального остового дерева (вопрос о способе построения структурных описаний в этих работах не обсуждается).

Значение $L(f_1, f_2 \circ T)$ можно вычислять и непосредственно тем же способом, что и длину описания одного изображения, просто рассматривая $\{f_1, f_2 \circ T\}$ как «цветное» изображение. Все алгоритмы построения структурного описания по-

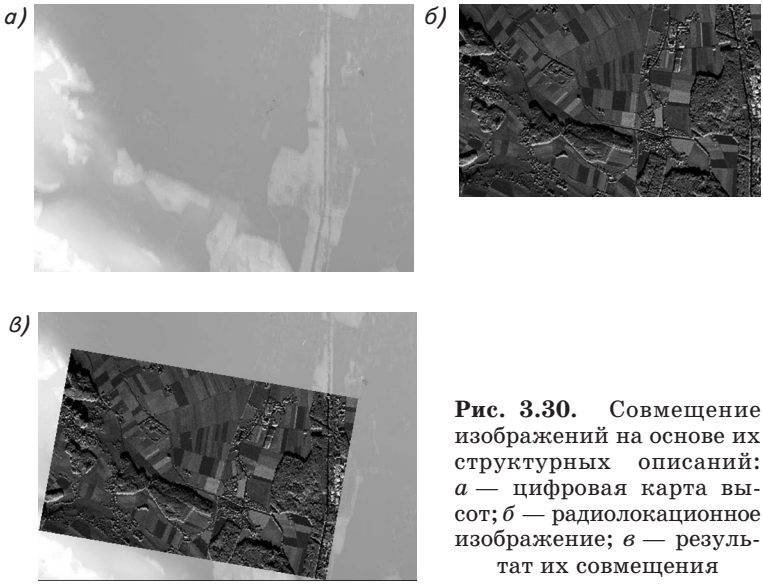

Рис. 3.30. Совмещение изображений на основе их структурных описаний: *а* — цифровая карта высот; *б* — радиолокационное изображение; *в* — результат их совмещения

лутонового изображения легко применить и для многокомпонентного изображения. Однако строить полное описание изображения $\{f_1, f_2 \circ T\}$ для каждой гипотезы пространственного преобразования T неэффективно с вычислительной точки зрения. В практических реализациях предпочтительнее привлекать некоторые приближенные оценки. Использование единого подхода к описанию и к совмещению изображений дает значительные удобства.

Разбор самих алгоритмов совмещения на основе иерархических структурных описаний выходит за рамки данной книги. Приведем лишь некоторые примеры результатов совмещения, выполненного этими алгоритмами (более подробно см. в работах [297, 331, 332]). На рис. 3.27–3.30 представлены результаты совмещения аэрокосмических изображений, имеющих существенные отличия, вызванные сменой типа сенсора или сезонными изменениями. Возможность совмещения в таких сложных случаях говорит о жизнеспособности описанного подхода.

3.2.7. Некоторые выводы относительно общей проблемы индукции

Мы рассмотрели некоторые аспекты проблемы интерпретации изображений и описали информационный подход к построению иерархического структурного представления изображений. Многие элементы этого представления были приведены в весьма упрощенной форме, так как мы вовсе не ставили целью описать законченную систему машинного зрения, а лишь хотели продемонстрировать возможность применения принципа МДО для сложной задачи анализа данных, сводимой к индуктивному выводу. Тем не менее мы надеемся, что приведенный выше материал будет полезен и специалистам в области компьютерного зрения. Основным результатом по отношению к иконике является возможность установления строгого критерия качества представлений, что дает возможность целенаправленного улучшения этих представлений, причем конкретные направления возможного улучшения были указаны.

Для теоретико-информационного подхода к индукции можно тоже сделать определенные выводы. Во-первых, на примере конкретной задачи мы могли убедиться, что использование универсального пространства гипотез, состоя-

щих из программ для универсальной машины Тьюринга, не может рассматриваться как общее и достаточное средство для решения любых задач индуктивного вывода. Вместо этого должно строиться представление, являющееся моделью предметной области и содержащее априорную информацию о ней. Для таких предметных областей, как зрительное восприятие, необходимый объем априорной информации, по-видимому, делает практически невозможным ее автоматическое накопление «с нуля».

Во-вторых, интерес представляет перенесение иерархического подхода к интерпретации изображений на общую проблему индуктивного вывода. В таком иерархическом подходе на нижних уровнях анализа делаются некоторые упрощающие предположения, касающиеся вида целевой функции, которые существенно уменьшают сложность задачи, но требуют последующей коррекции результатов на основе информации, полученной на следующих уровнях анализа. Далее мы увидим, что подобный иерархический подход с обратными связями характерен и для анализа речи.

Мы также не обсуждали представления изображений, основанные на знаниях, поскольку считаем, что информация о распознаваемых объектах (в том числе и соответствующие им лингвистические метки) не должна закладываться в систему априорно, а должна получаться машинной системой в процессе обучения путем анализа как зрительной, так и лингвистической информации. Естественный способ получения лингвистической информации заключается в использовании слуха. Проблема распознавания речи сама по себе является сложной проблемой, для которой может быть применен принцип МДО.

3.3. ТЕОРЕТИКО-ИНФОРМАЦИОННЫЙ ПОДХОД К МАШИННОМУ ВОСПРИЯТИЮ РЕЧИ

3.3.1. Проблема машинного слуха и распознавание речи

Слух в дополнение к зрению является сенсорной модальностью, активно исследуемой в ИИ. Другие чувства: осязание, обоняние и вкус (анализ химического состава), чувство равновесия — исследованы гораздо слабее. Еще менее исследованными (с точки зрения их компьютерного моделирования) яв-

ляются сенсорные модальности, получающие информацию от внутренних рецепторов (интероцепторов). Вероятно, это вызвано либо большей сложностью реализации технических устройств, выполняющих функции соответствующих органов чувств, либо меньшей практической значимостью.

Как правило, проблема слуха незаметно переводится в плоскость речевого общения. И вместо направления «компьютерный слух», аналогичного направлению «компьютерное зрение», мы имеем область «анализ и распознавание речи», которая вместе с вопросами синтеза речи (а также с задачами компрессии речи, имеющими более выраженную практическую направленность) входит в проблемную область речевых технологий. В системах распознавания речи для повышения надежности лингвистическая информация закладывается на весьма низкие уровни модуля интерпретации акустической информации. Это приводит к невозможности восприятия такими системами других звуков, кроме звуков человеческой речи (возможно, только на определенном языке). В этом смысле такие системы являются достаточно узкоспециализированными.

Их можно сравнить (и как мы увидим ниже, не без основания) с системами распознавания рукописного текста. Можно было бы сказать, что ситуация в области компьютерного слуха такова, как если бы большинство специалистов по компьютерному зрению занималось вопросами распознавания рукописного или печатного текста. Это сравнение, однако, не совсем справедливо, так как интерпретация произвольных звуковых сигналов является актуальной лишь в небольшом количестве приложений, в то время как желание разговаривать с компьютером на естественном языке давно преследует человека, не говоря уже о многих сугубо практических задачах, связанных с анализом речи. Более того, можно встретить немало работ и по автоматическому анализу музыкальных произведений. Однако никто не занимается разработкой системы анализа звуков, способной адекватно интерпретировать как речь, так и музыку с помощью одних и тех же (по крайней мере, на нижних уровнях анализа) средств.

Тем не менее именно универсальная система интерпретации звуковых сигналов потребовалась бы системе машинного обучения, снабженной соответствующей сенсорной модальностью. Здесь мы, однако, рассмотрим более частную проблему распознавания речи, поскольку она более детально

проработана. Кроме того, существуют работы по применению принципа МДО к этой проблеме.

Проблемы анализа речи не ограничиваются ее распознаванием. Речь содержит не только лингвистическую информацию, в ней также отражаются некоторые характеристики *диктора* (человека, речевой сигнал которого подвергается анализу). В связи с этим перед системой анализа речи могут стоять и такие задачи, как, например, идентификация личности по голосу или определение психофизического состояния диктора (находится ли он под воздействием алкоголя или в состоянии стресса и т. д.). Здесь мы ограничимся именно распознаванием речи, оставив в стороне другие проблемы ее анализа.

Сама задача распознавания речи (или более широкая задача понимания смысла речевого высказывания) может ставиться по-разному. Основным параметром здесь является объем *словаря* (лексикона), т. е. количество слов, которые могут быть предъявлены системе для распознавания. На одном конце спектра находятся системы с малыми словарями, содержащими единицы или десятки слов. Они, как правило, предназначены для решения конкретных практических проблем (таких, как голосовой набор номера или вербальное управление каким-либо устройством, например, управление функциями автомобиля с помощью голосовых команд). На другом конце спектра находятся системы с потенциально неограниченным словарем.

Другой важный параметр определяется тем, произносятся ли слова раздельно или на вход системе подается *слитная речь*. В последнем случае возникает дополнительная (и весьма сложная) задача — сегментация слитной речи на слова.

Поскольку речевой сигнал сильно зависит от особенностей голоса диктора, то еще одним важным параметром систем распознавания речи является то, какие ограничения накладываются на диктора. В простейшем случае это может быть единственный диктор с хорошей дикцией (или обученный пользователь), на голос которого настроена система. В наиболее сложных случаях на диктора никакие ограничения не накладываются (ни на пол, ни на возраст, ни даже на отсутствие у него акцента), а у системы какая-либо априорная информация о дикторе отсутствует.

Естественно, для систем с большим числом ограничений может быть предложено эффективное частное решение. В частности, для распознавания отдельных слов малого сло-

варя и одного или нескольких дикторов могут использоваться классические методы распознавания образов (распознавания с учителем), в которых слово как целостный образ задается фиксированным набором признаков. Обучение с учителем осуществляется по репрезентативной выборке речевых сигналов, для которых задаются соответствия со словами из интересующего лексикона и вычисляются векторы признаков. В таких системах принцип МДО применим постольку, поскольку он применим в методах распознавания образов (см. гл. 2). Рассмотрение столь ограниченной проблемы распознавания речи не даст нам нового материала о возможных применениях принципа МДО, поэтому мы обратимся к проблеме распознавания речи с наиболее широкой постановкой: дана слитная речь, содержащая слова из неограниченного лексикона и произнесенная неизвестным диктором.

3.3.2. Основные понятия в области распознавания речи

Человеческая речь (речевой поток) представляет собой непрерывную одномерную последовательность звуков. Звук может описываться как акустический сигнал через его волновые характеристики. Звуки речи, однако, не являются произвольными акустическими колебаниями и имеют вполне определенный источник — *артикуляционный аппарат* человека, состоящий из гортани, рта и носовой полости. Этим-то и отличается проблема распознавания речи от проблемы интерпретации произвольного акустического сигнала или произвольного изображения, в котором природа исходных данных гораздо более неопределенна.

Звуки речи могут быть отождествлены с конечным набором фонем. *Фонемы* — это неделимые звуковые единицы языка. Количество фонем зависит от языка, но, как правило, находится в пределах 30–50.

Ограниченность числа звуковых единиц языка (фонем) по сравнению с произвольными звуками «компенсируется» тем, что речь обладает сложной структурой, поскольку ее источником является человек. Структура же естественных звуковых сигналов более проста, хотя сами звуки разнообразнее. На фонетическом уровне проявлением структурности речи является то, что соседние фонемы не являются независимыми, а группируются в слова (точнее, в *словоформы*).

Зависимость фонем очевидна, так как число словоформ данного языка существенно меньше, чем число всевозможных комбинаций фонем. С позиции алгоритмической теории информации, последовательности фонем в речи как индивидуальные цепочки символов не являются случайными (см. п. 1.5.5).

Смысл фонемы как неделимой звуковой единицы языка заключается в том, что звуковой сигнал, соответствующий некоторой словоформе, отличается от звукового сигнала, соответствующего другой словоформе, по тому, из каких фонем сконструированы данные словоформы, т. е. по их фонетическим транскрипциям. Это значит, что для распознавания словоформ *достаточно* использовать цепочки фонем, абстрагируясь от более низкоуровневой акустической информации (и понижая тем самым размерность пространства признаков).

В связи с этим в системах распознавания речи с неограниченным словарем обычно присутствует подсистема предварительного анализа, переводящая речевой поток в цепочки фонем. Подобное представление можно сравнить с промежуточным символьным представлением в системах интерпретации изображений. Именно в этой подсистеме в наибольшей степени должна учитываться акустическая природа исходных данных. В ходе дальнейшего анализа можно в значительной мере абстрагироваться от того, каким путем получено символьное описание.

Связь между фонемами и буквами, используемыми в языке, не является взаимно однозначной. Точный перевод цепочек фонем в орфографический текст (а именно он обычно является выходом из систем распознавания речи) невозможен без распознавания словоформ. Итак, в системах распознавания речи присутствуют три основных уровня: речевой сигнал, цепочки фонем, последовательность слов. В системах понимания речи добавляется еще один уровень — уровень смысла текста (аналогичный семантическому уровню в системах понимания изображений). Эти же уровни свойственны и системам синтеза речи, но преобразования в них идут в обратном порядке [216, с. 95]. Между этими крупными уровнями находятся подуровни, о которых будет сказано ниже. Преобразование «текст—смысл» практически никак не связано со способом получения текста, поэтому здесь оно рассматриваться не будет (сильно ограниченная проблема смысла будет рассмотрена в п. 3.4).

Наличие смысла в тексте, в частности, означает, что последовательность слов, образующих текст, не является случайной. Хотя в системах распознавания речи смысл текста не восстанавливается, но учет статистических взаимосвязей между словами позволяет повысить эффективность распознавания. Такой учет осуществляется с помощью так называемых «моделей языка» (обычно крайне простых), среди которых наиболее популярными являются скрытые марковские модели или очень близкие к ним по смыслу модели на основе N-грамм. Последние будут кратко рассмотрены в п. 3.3.5.

Кроме фонетической структуры речи в задачах анализа речи важным является выявление *просодической структуры*, формирующейся с помощью интонации, пауз или выделения слов акцентом. Существуют и приемы, позволяющие передавать просодическую составляющую в письменной речи. Обилие таких приемов характеризует художественные тексты, в особенности поэзию. Учет просодической структуры чрезвычайно важен для достижения понимания смысла речевого высказывания и определения мотивов диктора, по которым им было сформировано данное сообщение. Например, одна и та же фраза в зависимости от интонации может превратиться в приказ, просьбу, вопрос и т. д., а понимание фраз, содержащих аллегории или гиперболы, вряд ли может быть достигнуто на основе только синтаксического анализа.

3.3.3. Распознавание фонем по различительным признакам

В систему распознавания речи речевой поток поступает в виде акустического сигнала (рис. 3.31). При этом возникает проблема нахождения соответствия между характеристиками этого сигнала и фонемами (рис. 3.32), цепочки которых образуют слова. Поскольку источником звуков речи является артикуляционный аппарат человека, то естественно связать акустические характеристики фонем с их артикуляционными характеристиками, т. е. с местом и способом их образования в речевом аппарате (подробнее см. [216, с. 97]). Для установления такой связи строится модель речевого аппарата человека как акустической системы [216, с. 99]. На основе этой модели может быть определена фор-

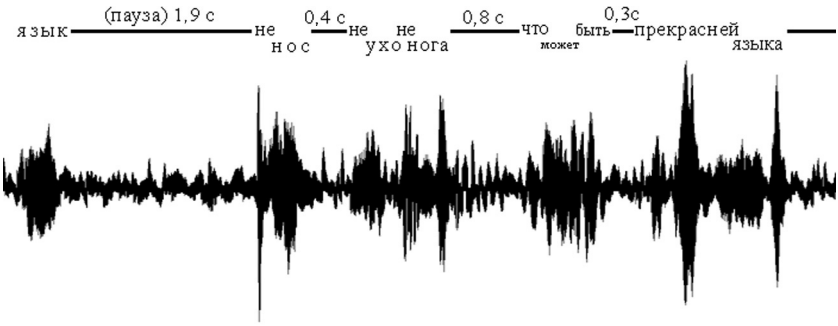

Рис. 3.31. Пример реализации речевого сигнала продолжительностью 8 с (отрывок из произведения К. Арбенина «Монолог о языке») на фоне сложных шумов (запись, выполненная на 44 кГц, т. е. сигнал содержит около 350 тыс. отсчетов; видно, что многие слова в слитной речи не разделяются паузами)

ма акустического сигнала при конкретной конфигурации системы. В свою очередь, конфигурация голосового тракта при произнесении конкретной фонемы может быть определена с помощью методов экспериментальной фонетики. В результате акустическая модель позволяет связать некоторую фонему с соответствующей ей идеализированной формой сигнала. К сожалению, помимо того, что сама модель является определенным упрощением голосового тракта, одной фонеме не соответствует единственная конфигурация акустической системы.

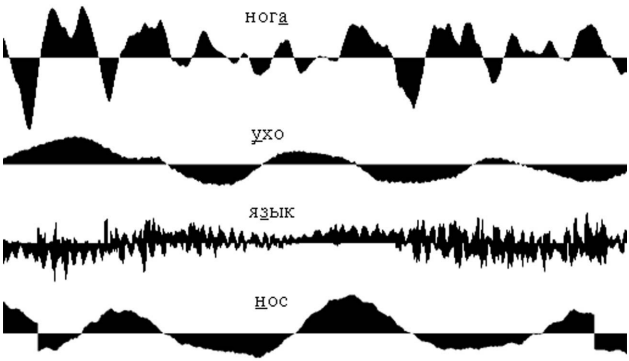

Рис. 3.32. Фрагменты реализации акустического сигнала продолжительностью приблизительно по 20 мс, соответствующие звучанию четырех фонем. Согласные звуки, в отличие от гласных, плохо описываются гармоническими функциями: звук «з» имеет характер шума, звук «н» содержит резкие переходы, которые дают широкий спектр

В частности, это вызвано эффектом *коартикуляции*, который влечет за собой зависимость реализации данной фонемы в виде акустического сигнала от контекста. Приведем естественное объяснение эффекта коартикуляции. Пусть «имеется четкое соответствие между фонемой, которую говорящий собирается произнести, и сигналами к мотонейронам мускулатуры голосового канала, получаемыми от мозга. Реальные физические движения в голосовом канале, которые и определяют произнесенный звук, будут зависеть от конфигурации голосового канала, которая была перед тем, как нейронами получены новые сигналы. Также важно и то, что не все группы мускулов одинаково быстро начинают участвовать в формировании определенного звукового сигнала. Поэтому действия этих групп мускулов, участвующих в формировании последовательно произносимых фонем, могут на самом деле осуществляться отчасти параллельно. Поскольку произнесенный звук является результатом прохождения воздуха по всему голосовому каналу, то акустический сигнал, соответствующий определенной фонеме, зависит также от фонем, сформированных как до, так и после этой фонемы» [52, с. 424].

Эффект коартикуляции приводит к формированию разнообразных оттенков фонем — аллофонов, точнее, *комбинаторных аллофонов*. Аллофоны другого типа — *позиционные аллофоны* — «обусловлены положением фонемы в слове или фразе по отношению к ударному слогу, концу и началу слова и т. д.» [216, с. 98].

Это говорит о том, что между уровнем акустического сигнала и уровнем фонем должен быть, по крайней мере, один промежуточный уровень — уровень различительных (информативных) признаков фонем. Тогда задача распознавания фонем разбивается на две: 1) построение детектора различительных признаков фонем в речевом потоке; 2) распознавание фонем на основе признаков.

2-я задача является классической задачей распознавания образов. В системах распознавания речи, как правило, применяется метод обучения с учителем: используется обучающая выборка, содержащая речевой сигнал и согласованные с ним правильные фонемы. После вычисления признаков получается набор соответствий: вектор признаков — фонема, который может быть использован для формирования классов. Для этого могут привлекаться, например, модели гауссовых смесей, что обосновано наличием аллофо-

нов (компонента смеси соотносится с аллофоном). Естественно, используются и другие методы распознавания образов.

Для решения 1-й задачи — определения признаков фонем — должны привлекаться более специализированные методы. Акустическая модель предсказывает наличие резонансных частот у голосового тракта, что приводит к формантному методу анализа (см. [216, с. 100]). Упрощенно, *формантой* называется энергетический максимум в определенной полосе частот в определенном временном интервале. Помимо частоты форманта характеризуется шириной полосы, в которой обнаруживается всплеск энергии, и амплитудой. Каждая фонема может быть представлена набором формант (обычно ограничиваются несколькими первыми формантами) и описана вектором признаков, составленным из параметров этих формант. Параметры, однако, не полностью детерминированы фонемой. К примеру, частота второй форманты у согласных зависит от того, какая гласная за ними следует в слоге [216, с. 101]. Амплитуда же форманта не является постоянной, и ее изменение имеет характерный временной профиль в зависимости от способа образования фонемы.

Близким по сути описанному выше способу анализа речевого сигнала является спектральный анализ. В ходе его также строится спектрограмма — зависимость спектра сигнала от момента времени, для чего выполняется преобразование Фурье в локальных временных окнах. При этом, однако, не происходит в явном виде обращения к акустической модели речевого аппарата человека и не осуществляется восстановление параметров этой модели, а используются более абстрактные спектральные признаки, в пространстве которых осуществляется классическое распознавание.

Однако чисто спектральное представление оказывается не вполне достаточным для описания всего разнообразия звуков речи. В частности, простая акустическая модель, которая приводит к формантному или спектральному анализу речевого сигнала, достаточно точно описывает лишь процесс порождения гласных звуков и гораздо хуже — многих согласных звуков (см. рис. 3.32), которые в то же время являются существенно информативнее. Оценивание спектра сигнала с использованием временных окон с фиксированным для всех частот размером также не вполне коррект-

но: во-первых, затруднительно оценивать гармоники с периодом, большим выбранного размера окна, и, во-вторых, высокочастотные гармоники усредняются по многим своим периодам, что при нестационарном сигнале дает плохую оценку локального спектра.

Выполненное в локальном окне преобразование Фурье может быть обобщено вейвлет-преобразованием, в котором не обязательно используется именно тригонометрический базис, а в случае его использования размер окна оказывается связанным с частотой. Опишем общую идею подхода распознавания фонем, тесно связанного с вейвлет-преобразованием, поэтому начнем с его формального определения.

Вейвлет-преобразование функции $x(t)$ есть ее скалярное произведение с базисными функциями специального вида (рис. 3.33). Эти базисные функции задаются через масштабирование и сдвиг некоторой функции-прототипа $\psi(t)$:

$$\psi_{\alpha,\tau}(t) = \frac{1}{\sqrt{\alpha}}\psi\left(\frac{t-\tau}{\alpha}\right), \qquad (3.19)$$

где α, τ — соответствующие параметры масштабирования и сдвига.

Тогда непрерывное вейвлет-преобразование определяется как

$$\tilde{x}(\tau,\alpha) = \int x(t)\psi_{\alpha,\tau}(t)dt = \frac{1}{\sqrt{\alpha}}\int x(t)\psi\left(\frac{t-\tau}{\alpha}\right)dt. \qquad (3.20)$$

Для расположения точек на регулярной сетке дискретное преобразование будет осуществляться через суммирование для масштаба $\alpha = 2^n$ и сдвига $\tau = m\Delta t$, где Δt — шаг сетки. Набор функций для различных натуральных n и целых m образует базис.

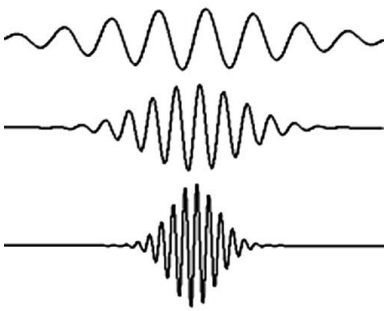

Рис. 3.33. Пример базисных функций для вейвлет-преобразования, полученных в результате масштабирования единственной функции-прототипа, локализованной одновременно во времени и частотной области

Само по себе разложение сигнала по базисным вейвлетам для наших целей является немногим интереснее преобразования Фурье или разложения в ряд Тейлора. Напомним, что когда шла речь об аппроксимации набора точек функцией из некоторого семейства, то сложность функции не должна была превышать количества содержащейся в исходных данных информации (см. п. 2.6). Учет всех коэффициентов вейвлет-преобразования не приводит к уменьшению длины описания сигнала. Но, как оказывается, лишь немногие коэффициенты вейвлет-преобразования являются заметно отличными от нуля. При этом для разных фонем значимыми оказываются разные члены разложения, что делает возможным использование их параметров в качестве характеристических признаков фонем (в частности, на основе вейвлетов может производиться формантный анализ) [335].

Использование значимых компонентов вейвлет-преобразования оказывается эффективным и для сжатия речи (см. [336] в качестве применения адаптивных вейвлетов для сжатия). Более того, в этой области данный подход был развит в алгоритм наилучшего базиса [337]. Суть этого алгоритма заключается в том, что для каждого участка речевого потока выбирается собственный базис (основанный на разных функциях-прототипах), который наиболее эффективен для сжатия.

Эффективность алгоритма наилучшего базиса связана с тем, что речь является нестационарным процессом и содержит участки, порожденные различными способами (в смысле артикуляции). Использование вейвлет-преобразований с различными базисами позволяет описывать как периодические колебания, так и локальные события в потоке речи (см. на рис. 3.32 форму сигнала, соответствующего фонеме «н» в слове «нос»). Неудивительно, что алгоритм наилучшего базиса оказался весьма полезным и для распознавания и анализа речи [335].

Несложно дать интерпретацию алгоритма наилучшего базиса с точки зрения построения модели на основе принципа МДО. Пусть для некоторого фрагмента речевого сигнала выполняется (дискретное) вейвлет-преобразование и выбирается несколько наиболее значимых коэффициентов. После их извлечения уровень сигнала существенно понижается, и остаток можно интерпретировать как шум (это классическая задача регрессии, рассмотренная в п. 2.6, однако

со специфическими базисными функциями). Разделение же речевого сигнала на фрагменты, каждый из которых описывается собственной регрессионной моделью (в данном случае составленной из различных базисных функций), — это не что иное, как задача сегментации, которой мы также коснулись в п. 2.6.

Рассмотрение выделения характеристических признаков фонем как задачи сегментации с позиций принципа МДО позволяет уйти от дискретного вейвлет-преобразования, требующего вычисления значения всех его коэффициентов для фиксированных значений α и τ и не допускающего использования нескольких базисов для одного фрагмента сигнала. Тогда можно предложить следующую схему анализа. Итеративно находим параметры вейвлетов, после извлечения которых из сигнала энтропия последнего максимально уменьшается (заметим, что параметры могут принимать не только значения $\alpha = 2^n$ и $\tau = m\Delta t$, но и любые промежуточные, что позволяет эффективнее, в смысле длины описания, моделировать сигнал). Выполняем эту процедуру до тех пор, пока добавление нового вейвлета приводит к уменьшению длины описания (параметры α и τ самого вейвлета, а также номер базиса, к которому он принадлежит, необходимо описывать, что должно компенсироваться уменьшением энтропии сигнала, полученного после вычитания вейвлета).

При использовании алгоритма, аналогичного алгоритму наилучшего базиса, должна быть определена библиотека базисов (или функций-прототипов). Выбор функций-прототипов — эмпирическая задача. Возможность выбора различных базисов для анализа делает этот подход расширяемым на звуки не только человеческой речи. Вполне возможно, что может быть сформирована библиотека базисов, которая будет достаточной для адекватного описания весьма разнообразных естественных звуков, а также звуков речи. Нам больше импонирует именно такой подход к выделению признаков, не обращающийся непосредственно к информации о речевом тракте человека, по крайней мере, на самых ранних уровнях анализа акустической информации. Помимо возможности его расширения на произвольные звуки (т. е. возможности построения на его основе системы машинного слуха общего назначения) он обосновывается тем, что и у самого человека нижние уровни интерпретации акустической информации сформировались за-

долго до возникновения речи (а значит, и потребности в ее понимании).

Итак, параметры извлеченных вейвлетов могут рассматриваться как признаки фонем для дальнейшего распознавания. В действительности, эти параметры содержат не только информацию о фонетическом составе речевого сообщения, но также и о дикторе [335] (информация об индивидуальных характеристиках голосовых связок, тесно связанных, в частности, с полом и возрастом говорящего, содержится в области низких частот). В связи с этим могут быть поставлены различные задачи обучения с учителем (например, распознавания слов или идентификации диктора), которые, как мы видели в п. 2.3, могут также решаться с использованием принципа МДО. Вместо того чтобы хранить сами параметры вейвлетов (признаки), можно хранить номера классов, каждому из которых соответствует один или несколько эталонных векторов признаков и отклонение этих параметров от эталонных значений.

В действительности анализ должен быть более сложным. Как мы отмечали выше, из-за зависимости произнесения фонемы от контекста возникают аллофоны, поэтому между параметрами соседних вейвлетов будет существовать определенная статистическая зависимость. Если эту зависимость не учитывать, то классы, соответствующие одной фонеме, будут весьма широкими и могут пересекаться с классами других фонем. Чтобы этого избежать, необходимо корректировать значения вейвлет-признаков с учетом контекста. Это требует более изощренной схемы кодирования (и введения еще одного промежуточного представления речевого сигнала), которая также приводит и к уменьшению длины описания, но ее рассмотрение выходит за рамки данной книги.

3.3.4. Распознавание слов по цепочкам символов

В результате работы нижнего уровня системы анализа речи строится символьное описание сигнала, представляющее собой цепочки фонем. Если бы полученные на предыдущем уровне анализа последовательности фонем в точности соответствовали фонетическим транскрипциям слов, присутствующих в лексиконе системы распознавания, то задача распознавания слов была бы тривиальной и заклю-

чалась бы в просмотре таблицы соответствия. Однако фонетическая транскрипция одного и того же слова может содержать пропущенные, лишние или замененные фонемы по сравнению с некоторой канонической транскрипцией, что связано со следующими двумя причинами [338]:

1) неидеальной работой подсистемы распознавания фонем, вызванной зашумленностью речевого сигнала, неполнотой обучающей выборки, недостатками выбранной системы характеристических признаков фонем, плохим учетом эффекта коартикуляции и т. д.;

2) ошибочным (отличным от канонического варианта) произношением слова, обусловленным наличием акцента или диалектом языка, либо, напротив, слишком беглым произнесением родной речи.

Еще одним препятствием при переводе последовательностей фонем в орфографический текст является то, что в слитной речи могут отсутствовать паузы между словами (см. рис. 3.31). Для человека, привыкшего к родной речи, ее слитность оказывается малозаметной, но для иностранца беглая речь носителя языка часто оказывается сложнее для восприятия, чем речь другого иностранца. В качестве другой иллюстрации к проблеме разделения слитной речи на слова может выступать затрудненность чтения «слитного» орфографического текста:

обратитевниманиенасколькологчебылобыпрочитатьэтоттекстеслибысловавнембылиразделенынробеламизадачаусложнилаьбыещебольшееслибывданномтекстеприсутствовалиошибки!

Из-за отклонения фонетических транскрипций слов от их словарных версий выделение границ слов оказывается действительно трудной задачей.

Давайте рассмотрим, как эту задачу можно решить, руководствуясь принципом МДО. Предположим сначала, что сегментация на слова осуществлена, поэтому рассмотрим цепочку фонем, соответствующую одному слову.

Будем считать, что эту цепочку нужно передать между двумя информационными системами (отправителем и получателем сообщения), при этом необходимо выбрать такое представление, в котором эта цепочка кодировалась наиболее компактно. Будем считать, что как отправитель, так и получатель сообщения имеют в своем распоряжении

словари с фонетическими транскрипциями отдельных слов. Исходная цепочка отличается от канонической версии слова, но можно надеяться, что это отличие не слишком сильное. Последнее означает, что при передаче цепочки фонем выгоднее (с точки зрения длины описания) передать код, обозначающий наиболее подходящее слово, и дополнить его информацией об отклонениях в фонемах от канонической транскрипции. Приходим к уже привычной схеме кодирования, содержащей две части: описание модели (в данном случае — слова) и описание отклонения реальных данных от модели. Согласно принципу МДО, следует выбрать то слово, которое дает минимум суммарной длины описания.

Пусть A — набор символов, соответствующих фонемам данного языка, а $\Gamma \subset A^*$ — словарь (набор канонических транскрипций слов, составленных из фонем, входящих во множество A; для одного слова в словаре может присутствовать несколько его альтернативных реализаций), имеющийся у системы распознавания. И пусть $\alpha = a_1 a_2 \ldots a_M$, $a_i \in A$ — исходная цепочка фонем, которая может и не принадлежать словарю Γ, а $\beta = b_1 b_2 \ldots b_{M'}$, $b_i \in A$ — слово ($\beta \in \Gamma$), принимаемое в качестве гипотезы для «объяснения» цепочки α. Тогда наиболее подходящим словом будет

$$\beta_{MDL} = \arg\min_{\beta \in \Gamma} \left(L(\beta) + L(\alpha \mid \beta) \right). \qquad (3.21)$$

Прежде чем переходить к определению слагаемых в данной сумме, отметим, что тот факт, что множество A — это множество фонем, не имеет принципиального значения для дальнейшего рассмотрения. В равной степени это может быть, например, множеством букв письменной речи, тогда как α может являться цепочкой распознанных рукописных или печатных символов.

Рассмотрим длину описания $L(\beta)$ гипотезы о некотором слове β. Она определяется как минус логарифм безусловной вероятности того, что случайно взятое из потока речи слово окажется именно этим словом. Естественно, это значение будет отличаться от числа фонем, используемых для озвучивания данного слова, хотя определенная взаимосвязь между этими двумя величинами имеется. Подобные априорные вероятности встречи различных слов могут быть определены на основе простого подсчета частот встречаемости слов в текстовом корпусе (например, подборке художественных текстов). Если целью является создание специа-

лизированной системы распознавания речи, предназначенной для работы в определенной предметной области, то тексты, использующиеся в качестве обучающей выборки, должны быть подобраны с учетом специфики задачи, и в результате будут получены другие частоты слов.

Оценивание слагаемого $L(\alpha \mid \beta)$ заметно сложнее. Для его определения необходимо представление, в рамках которого описывались бы различия двух цепочек фонем α и β. В работе [338] (см. также [339, 340]) перевод одной цепочки в другую предлагается осуществлять набором операций вставки, замены и удаления фонем. Для этого используется следующая схема кодирования.

1. Сначала описываются вставки в первую цепочку фонем пустых символов.

2. Затем удаления, замены и вставки символов представляются единообразно как замены. Они могут быть различены по тому, что вставки описываются как замены пустых символов значимыми, удаления — как замены значимых символов на пустые, а обычные замены — как замены значимых символов на значимые.

В качестве полезного дополнения можно ввести такие замены вида $aa \to a$ и $a \to aa$, характерные как для устной, так и для письменной речи, но мы их опустим для простоты изложения.

Поскольку здесь не приводилось описания правил фонетической транскрипции и списка фонем, то для наглядности воспользуемся примером с орфографическим текстом. Пусть на вход системы распознавания подана последовательность символов «обберация». В качестве двух гипотез рассмотрим слова «аберрация» и «операция» (рис. 3.34).

Определим необходимое количество информации для описания вставок пустых символов. Сначала необходимо передать количество вставляемых пустых символов N_e, для чего требуется $-\log_2 P(N_e)$ бит информации [априорная вероятность $P(N_e)$ того, что требуется вставить N_e пустых символов, определяется из обучающей выборки]. Пустой сим-

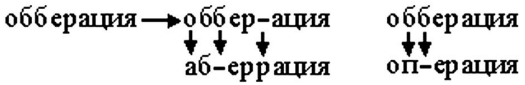

Рис. 3.34. Пример перевода цепочек фонем друг в друга путем добавления пустых символов с последующей попарной заменой символов

вол может быть вставлен в любую из $M + 1$ позиций, включая вставку до и после всех символов исходной строки. Набор из N_e пустых символов в строку длины M можно вставить $C_{M+1}^{N_e}$ способами, для чего требуется $\log_2 C_{M+1}^{N_e}$ бит. В п. 2.3.6 мы приводили асимптотическую оценку этой величины, однако из-за того, что в данной задаче значения N_e и M малы, эта оценка оказывается весьма грубой. Пусть после вставки в исходную строку пустых символов получается строка $\alpha' = a'_1 a'_2 \ldots a'_{M+N_e}$.

Определим теперь необходимое количество информации для описания замен символов. Можно предложить следующие две схемы кодирования замен. В первой схеме описываются все замены, включая замену символа на самого себя. Во второй схеме описываются только замены символов на другие символы, но при этом для каждой замены требуется указывать позицию заменяемого символа, равно как и общее число замен. С точки зрения распознавания слов обе схемы почти эквивалентны. Вторая схема удобнее для решения практических задач компрессии, но мы воспользуемся первой схемой как более единообразной. После выполнения данной операции каждый символ a'_i заменяется символом b'_i, которые образуют строку $\beta' = b'_1 b'_2 \ldots b'_{M+N_e}$, которая после удаления пустых символов должна превращаться в строку β.

Итак, необходимо описать $M + N_e$ замен символов $a'_i \to b'_i$, причем замены различных символов не равновероятны. На каждую такую замену приходится $-\log_2 P(b'_i \mid a'_i)$ бит информации, где $P(b'_i \mid a'_i)$ — вероятность того, что истинный символ в строке b'_i при условии, что в исходной строке на этом месте присутствует символ a'_i. Эти вероятности определяются в процессе распознавания с учителем по набору исходных данных, для которых известны правильные результаты распознавания. В нормальной ситуации вероятности замены символа самим собой должны быть максимальны и должны описываться малым количеством информации.

Заметим, что вероятности $P(b'_i \mid a'_i)$ могут существенно различаться для разных способов получения исходных строк символов. Например, вероятность $P(\text{п}|\text{б})$ относительно высока для распознавания речи (звуки «п» и «б» достаточно похожи и могут быть спутаны, особенно при беглой или зашумленной речи) и относительно низка при визуальном распознавании текста, поскольку по написанию соответствующие буквы различаются весьма сильно, а орфографическая ошиб-

ка замены «б» на «п» достаточно редкая. В то же время вероятность $P(а|о)$ сравнительно велика в обоих случаях.

Итак, общая длина описания строки α и слова-гипотезы β будет

$$L = -\log_2 P(\beta) + \log_2 C_{M+1}^{N_e} - \sum_{i=1}^{M+N_e} \log_2 P(b_i' \mid a_i'). \quad (3.22)$$

Для ее определения необходимо найти оптимальный набор операций над символами, переводящий цепочку α в слово β. Это может быть сделано с помощью алгоритма дистанции редактирования, представленного в работе [341]. К сожалению, его описание выходит за рамки данной книги.

Воспользовавшись этим алгоритмом для каждого предполагаемого слова и оценив соответствующую длину описания (3.22), можно выбрать слово, наилучшим образом соответствующее исходным данным.

С практической точки зрения для данной цепочки символов нерационально вычислять длину описания по формуле (3.22) для каждого слова из словаря, поскольку это требует поиска наилучшего способа перевода одной цепочки символов в другую. В то же время многие слова не будут иметь ничего общего с исходными данными и могут быть сразу отброшены. Для осуществления этой операции может быть выполнена операция группирования слов данного языка по их фонемному (или буквенному) составу в несколько (возможно, частично перекрывающихся) классов [338]. Тогда распознавание цепочки фонем начинается с ее отнесения к одному из таких классов, после чего распознавание ведется только среди слов выбранного класса. Существуют и другие приемы оптимизации методов распознавания слов, которых мы касаться не будем.

Более принципиальным является вопрос об определении границ слов в слитной речи, который мы пока опустили, предположив, что на вход системы распознавания подается цепочка фонем, соответствующих единственному слову.

3.3.5. Выделение границ слов и модели языка на основе N-грамм

С помощью указанного выше критерия можно выбирать для данной цепочки фонем наиболее подходящее слово. Однако для слитной речи, не разделенной на слова, возника-

ет дополнительная неопределенность в положении границ слов, которую необходимо устранить. Наиболее прямое решение этой проблемы заключалось бы в переборе всех возможных способов разбиения цепочки фонем на фрагменты с определением для каждого разбиения суммарной длины описания всех фрагментов:

$$L = \sum_{k=1}^{K} \left[-\log_2 P(\beta_k) + \log_2 C_{N+1}^{N_e} - \sum_{i=1}^{N+N_e} \log_2 P(b_i' \mid a_i') \right], \quad (3.23)$$

где K — общее количество слов в разбиении; значения N, N_e, b_i', a_i' зависят от рассматриваемого слова β_k.

Очевидно, такое решение очень ресурсоемкое. Другая крайность заключается в том, чтобы начать анализировать цепочку фонем сначала и распознавать по одному слову, перебирая различные варианты конца слова и выбирая из них лучший. После того как первое слово распознано и положение его конца определено, можно это положение принять за начало следующего слова и продолжить для него анализ аналогичным образом. Такая схема сегментации с одновременным распознаванием является наиболее быстрой, но не гарантирует (и в большинстве случаев не достигает) оптимального распознавания.

Действительно, представим, что распознаваемый текст начинается с «выходканцлерайз...» Альтернативные варианты выделения первого слова будут: «вы|ходканцлерайз...», «выход|канцлерайз...», «выходка|нцлерайз...»

Поскольку они описывают различный фрагмент исходных данных, то сравнение их длин описания не вполне корректно, хотя по этому критерию их и можно отделить от прочих вариантов выделения первого слова. Выбор как наиболее короткого, так и наиболее длинного слова не приведет к правильному результату. При наличии ошибок (особенно при распознавании фонетического текста) ситуация осложнится еще больше. Таким образом, необходим некоторый промежуточный вариант анализа, не вовлекающий полный перебор и не принимающий чисто локальных решений.

Один из возможных способов такого анализа заключается в том, чтобы осуществлять распознавание слов, начиная сначала, но сохранять не единственную лучшую гипотезу, а некоторое их ограниченное количество, отбрасывая наименее перспективные гипотезы в смысле их длины опи-

сания. Каждую гипотезу о том, какие слова составляют некоторую начальную подстроку исходной цепочки фонем, можно оценить с помощью целевой функции (3.23). Тогда будут исключаться как локально плохие гипотезы (такие, как «выхо|дканцлерaиз...»), так и гипотезы, резко ухудшающие длину описания при дальнейшем распознавании и сегментации (такие, как «выходка|нцлерaиз...»). Подобный анализ удобен также и тем, что речевой сигнал приходит на вход системы распознавания постепенно, поэтому в данном случае система будет способна осуществлять распознавание по мере поступления, не дожидаясь для начала работы всего речевого сообщения.

Сформулированный метод анализа включает в себя множество таких технических деталей, как выбор достаточного числа одновременно рассматриваемых гипотез или ограничения, накладываемого на длину описания, при котором гипотеза исключается из рассмотрения, порядок просмотра гипотез, специальный выбор среди гипотез, в которых конец последнего распознанного слова совпадает, что делает их почти идентичными с точки зрения последующей сегментации, и т. д. Все эти вопросы хотя и весьма важны для реализации системы распознавания речи, но являются недостаточно универсальными в смысле оптимизации перебора и не имеют прямого отношения к целевой функции, основанной на принципе МДО, поэтому мы их опустим.

Более важным является то, что целевая функция для последовательности слов, введенная просто как сумма длин описаний отдельных сегментов, является недостаточной для надежного распознавания. В приведенном выше примере вариант сегментации «вы|ход|канцлер|а|из...» оказывается не хуже правильного варианта. Подобным образом при сегментации входной строки «отделкадров» следующие варианты оказываются почти эквивалентными: «от|дел|кадров», «отдел|кадров», «отделка|дров», поскольку все состоят из словарных слов. Несмотря на это, человек вполне способен определить правильный (вернее, наиболее вероятный) вариант. Неоднозначности проявляются еще сильнее, когда возможны ошибки в фонемах или буквах, составляющих входные цепочки. Вспомним само по себе неоднозначное слово «обберация». Но и эта неоднозначность может быть снята, если рассматривать слово не изолированно, а в контексте. Сравните, например, «обберация объектива» и «хирургическая обберация».

Очевидно, причина заметной ограниченности способности системы распознавания, построенной на основе целевой функции (3.23), по сравнению со способностями человека заключается в том, что здесь слова рассматриваются как статистически не зависимые. Это видно по тому, что в формулу (3.23) входят слагаемые $-\log_2 P(\beta_k)$. Также понятно, что слова, организованные в осмысленные предложения, не являются не зависимыми друг от друга.

Таким образом, при определении длины описания распознанного фрагмента речевого сообщения в формуле (3.23) сумма $-\sum_{k=1}^{K} \log_2 P(\beta_k)$ является плохой оценкой истинной длины описания цепочки слов $-\log_2 P(\beta_1, \beta_2, \ldots, \beta_K)$. Установление распределений совместных вероятностей даже для небольшого числа K является крайне проблематичным. Использование вероятностей является чисто байесовским подходом, и здесь отчетливо видны его недостатки: если вероятностный подход применяется при выборе модели, то его становится весьма затруднительно применить при выборе метамодели (а распределение вероятностей $P(\beta_1, \beta_2, \ldots, \beta_K)$ является именно метамоделью по отношению к задаче распознавания отдельных слов, т. е. является моделью языка). Такая ситуация приводит к необходимости использовать другие модели языка или применять более грубые оценки совместных вероятностей появления слов.

Использование более сложных моделей языка связано с разработкой систем понимания речи, а проблема смысла здесь не рассматривается. Для распознавания же речи весьма эффективными оказываются достаточно простые модели языка, основанные на подсчетах N-грамм слов или на скрытых марковских моделях. Под *N-граммой* понимается просто цепочка из N символов (например, слов, фонем и т. д.).

Эти методы используют оценку вероятности $P(\beta_1, \beta_2, \ldots, \beta_K)$, доступную для вычисления. Представим эту вероятность в виде суммы условных вероятностей

$$P(\beta_1, \beta_2, \ldots, \beta_K) = P(\beta_1) + P(\beta_2 \mid \beta_1) + \ldots$$
$$\ldots + P(\beta_K \mid \beta_{K-1}, \beta_{K-2}, \ldots, \beta_1). \qquad (3.24)$$

Считая, что некоторое слово зависит только от $N-1$ предыдущих слов (вместе с самим словом они образуют N-грамму), можно использовать следующую аппроксимацию условных вероятностей:

$$P(\beta_k \mid \beta_{k-1}, \beta_{k-2}, \ldots, \beta_1) \approx P(\beta_k \mid \beta_{k-1}, \beta_{k-2}, \ldots, \beta_{k-N+1}). \quad (3.25)$$

Если подставить условные вероятности $P(\beta_k \mid \beta_{k-1}, \beta_{k-2}, \ldots, \beta_{k-N+1})$ вместо безусловных вероятностей $P(\beta_k)$ в выражение (3.23), то система распознавания станет в определенной степени учитывать контекст, в котором находятся слова. При этом сам метод сегментации и распознавания слов останется абсолютно без изменений. Сложность же заключается в получении оценок условных вероятностей: при словаре в 50 000 слов даже число биграмм становится свыше миллиарда, т. е. даже при использовании десятка тысяч книг в качестве обучающей выборки надежно оценить частоты биграмм слов удастся лишь для части пар слов, образующих устойчивые словосочетания; остальные же пары либо не встретятся вовсе, либо встретятся малое число раз, что даст очень грубую оценку вероятности. Непосредственное определение частот триграмм слов становится практически неосуществимым, хотя определение вероятностей наиболее устойчивых сочетаний из трех и даже более слов остается все еще доступным.

Тем не менее даже использование биграмм слов позволяет заметно повысить надежность системы распознавания. В частности, неоднозначность выбора между вариантами сегментации «от|дел|кадров», «отдел|кадров», «отделка|дров» разрешается, а в случае строк «обберация объектива» и «хирургическая обберация» слово «обберация» может быть правильно (и различным образом) распознано на основе того, рядом с каким словом оно появилось. Несмотря на такое улучшение, модели на основе N-грамм слов являются очень слабыми в качестве моделей языка. В частности, одно и то же по смыслу сообщение, соответствующее фразе: «Лишь большое количество специальных терминов в данном тексте не позволит Смиту перевести его», — можно передать с помощью нескольких миллионов различных фраз ([133, с. 13] со ссылкой на [342, с. 179–190]), что означает наличие взаимосвязей между словами в тексте, являющихся существенно сложнее, чем простое повышение вероятности совместного появления групп слов. Этот факт, кстати, означает, что и преобразование «текст—смысл» можно также рассматривать с точки зрения принципа МДО, разделяя смысл фразы и ее конкретную реализацию в виде последовательности слов. К сожалению, на настоящий момент работы, исследующие данный вопрос, отсутствуют, по-

скольку пока еще сложно указать адекватное представление для описания смысла фразы и тем более текста.

Одна из возможностей использования зависимостей между всеми словами, входящими в предложение, заключается в подсчете частот N-грамм не самих слов, а частей речи, к которым они относятся. Поскольку количество различных частей речи существенно меньше, чем число слов, для них имеется возможность оценивать частоты N-грамм для бóльших значений N. Это требует внесения в систему анализа речи дополнительной лингвистической информации, на основе которой можно было бы распознавать части речи, к которым относятся те или иные слова. Эту лингвистическую информацию сложно извлечь с помощью простого статистического обучения на основе текстового корпуса, поэтому она используется далеко не во всех системах распознавания. Тем не менее если она доступна, то может быть использована для повышения эффективности распознавания, причем в рамках принципа МДО (см., например, [338]). Решению задачи разделения слитного текста на слова при использовании принципа МДО также посвящена статья [343].

3.3.6. Выделение устойчивых сочетаний фонем

Очень часто одному и тому же слову может соответствовать несколько его форм, выражающихся различными цепочками фонем или букв. К примеру, слово «длина» в данном тексте встречается в таких формах, как «длиной», «длину», «длине» и т. д. С точки зрения рассмотренного выше метода распознавания это различные слова, каждое из которых должно быть представлено в словаре. Для языков, выразительность которых заключается в синтаксисе предложений, а не в морфологии слов, словарь оказывается приемлемых размеров, даже если в него включены различные формы слов. Но для языков с более богатой морфологией (преимущественно агглютинативного и флективного типов) количество словоформ может существенно превосходить количество слов, но словоформы образуются достаточно похожим образом, например, добавлением или модификацией приставок, суффиксов и окончаний. Для русского языка помещение в словарь существительных в разных падежах и числах, прилагательных в разных падежах, числах

и родах и т. д. увеличит его размер на порядок (например, полная парадигма прилагательного может включать от 24 до 29 форм). В таких языках, как английский, для тех же целей используются предлоги и артикли, которые пишутся отдельно от слова, значение которого модифицируется.

В связи с этим возникает потребность во введении дополнительного уровня между фонемами или буквами и словами. Можно предложить несколько отправных точек для разработки такого уровня. Во-первых, можно обратить внимание на то, что минимальной артикуляционной единицей является слог, т. е. слова при их произнесении конструируются из слогов. Во-вторых, можно воспользоваться изученными в лингвистике правилами словообразования. Действительно, с морфологической точки зрения все формы слов конструируются из *морфов* — минимальных значимых частей, выделяемых в словоформах. И, в-третьих, можно применить статистическое обучение для автоматического выделения часто встречающихся цепочек фонем или букв и из них сформировать словарь частей слов.

Хотя выделение слогов может использоваться в системах распознавания речи (см., например, [344, 345]), но делается это скорее для улучшения качества распознавания отдельных фонем, а не для описания структуры слов. Данные морфологии могут оказаться значительно полезнее. Перечень морфов для конкретного языка является доступным и может быть использован вместо словаря в алгоритме, рассмотренном в п. 3.3.5.

Морфы являются менее уникальными, чем словоформы (особенно это касается не корневых, а аффиксальных морфов, играющих служебную, словообразовательную роль). В связи с этим при разделении цепочки символов на морфы возможна существенная неопределенность, для устранения которой необходимо использовать взаимосвязи между морфами. Эти взаимосвязи можно выразить через частоты N-грамм морфов, как это делалось выше для описания взаимосвязей между словами. Однако вместо этого можно обратиться к данным морфонологии, изучающей формальные закономерности сочетаемости морфов (вернее, *морфем*, которые играют в словах ту же конструктивную роль, что и морфы играют в словоформах; но мы на этом различии останавливаться не будем). В п. 3.3.5 мы упоминали, что можно использовать N-граммы частей речи в дополнение к N-граммам слов. Также и здесь можно использовать

N-граммы типов морфов (префиксальные, суффиксальные, корневые, флексийные и т. д.), частоты которых можно определить для бóльших значений N, чем частоты N-грамм самих морфов.

При использовании разнообразных данных морфологии возникает проблема сделать их пригодными для использования в системах распознавания. Также использование большого объема лингвистической информации, относящейся к конкретному языку, кажется слишком частным решением. В связи с этим рассмотрим подход к формированию лексикона частей слов на основе машинного обучения.

Представим, что у нас имеются частоты встречаемости всех словоформ языка (например, русского). И пусть есть большой текстовый корпус. На уровне отдельных слов эффективным будет такое кодирование, что каждой словоформе соответствует код, длина которого определяется по вероятности соответствующей словоформы (например, код Хаффмана). Перекодирование отдельных морфов или, тем более, отдельных букв будет заметно менее выгодным. Еще эффективнее будет кодирование, учитывающее высоковероятные N-граммы словоформ.

Заметим, однако, что такое сжатие будет целесообразным только для действительно больших текстовых корпусов. К примеру, роман «Анна Каренина», состоящий чуть больше, чем из 1,3 млн букв, содержит около 35 тысяч различных словоформ общей длиной около 275 тысяч букв. Число же биграмм словоформ будет еще больше, т. е. таблица кодов для словоформ имеет заметный размер даже по отношению к несжатому тексту, а перечень всех встречающихся биграмм не будет уступать ему.

Интересно также и то, что в таком большом тексте, как роман «Анна Каренина», разнообразие встречаемых словоформ весьма ограничено и далеко не исчерпывает всего многообразия языка даже для таких обычных слов, как, например, слово «луна». В данном романе оно встречается только в форме «луна» и «луну». В то же время в произведении «Мастер и Маргарита» к этим словоформам также добавляются такие, как «луны», «луне», «луною», «луной». Большинство же биграмм словоформ просто уникально для одного текста.

Все это наглядно демонстрирует информационную избыточность таблицы словоформ. Мы сейчас обсуждаем не

практические вопросы сжатия текста, а проблему выделения морфов. Сжимаемость таблицы словоформ означает возможность их разложения и компактного представления в виде совокупности морфов, количество которых заметно меньше количества всевозможных словоформ.

Итак, можно сформулировать задачу построения лексикона морфов как задачу оптимального кодирования списка словоформ языка. Отметим, что в этой задаче не должны учитываться частоты встречаемости словоформ, и каждая форма слова входит в список единственный раз: мы разделили задачу сжатия текста на уровне целых слов, использующую «модель языка» (где и учитывается вероятность слов или их N-грамм), и задачу сжатия «модели языка» путем выявления его морфологии.

Пусть дан список словоформ $\Gamma = \{\alpha_i\}$, каждая из которых представляет собой цепочку символов $\alpha_i = a_{i,1}a_{i,2}\ldots a_{i,M_i}$, $a_{i,j} \in \mathrm{A}$. В рамках такого представления оптимальное кодирование осуществляется на уровне символов в соответствии с их частотой встречаемости в словаре Γ. Пусть $n_\Gamma(a)$ — это количество появлений символа a во всех строках $\alpha_i \in \Gamma$, а $N_\Gamma(\mathrm{A})$ — общая длина всех строк словаря Γ, представленных символами из алфавита A. Тогда длина описания словаря Γ оптимально перекодированными символами алфавита A будет

$$L(\Gamma \mid \mathrm{A}) = N_\Gamma(\mathrm{A}) \log_2 N_\Gamma(\mathrm{A}) - \sum_{a \in \mathrm{A}} n_\Gamma(a) \log_2 n_\Gamma(a). \quad (3.26)$$

Это выражение несложно получить, подставив в формулу энтропии вместо вероятностей $P(a)$ их оценки через частоты $n_\Gamma(a)/N_\Gamma(\mathrm{A})$.

Формула (3.26) может быть использована для любого алфавита символов A, но для разных алфавитов длина описания $L(\Gamma|\mathrm{A})$ будет различна. Возможность использования разных алфавитов и необходимость выбора между ними заставляет вместо длины описания $L(\Gamma|\mathrm{A})$ в качестве целевой функции использовать длину описания

$$L(\Gamma, \mathrm{A}) = L(\Gamma \mid \mathrm{A}) + L(\mathrm{A}), \quad (3.27)$$

где $L(\mathrm{A})$ — длина описания алфавита с соответствующей ему таблицей перекодировки. Эта величина зависит от способа введения новых символов и будет рассмотрена ниже.

Для выделения морфов требуется ввести такой набор дополнительных символов, при записи словоформ через

которые длина описания (3.27) будет минимальна. Эта задача нетривиальна. Мы рассмотрим следующую простую схему, позволяющую получить интересные, хотя и неоптимальные результаты.

Предположим, что мы объединяем цепочку из двух символов $a_i a_j$ и добавляем новый символ b, заменяя все вхождение $a_i a_j$ в словаре на b. При этом увеличивается число символов в алфавите и, возможно, их энтропия, но уменьшается длина некоторых словоформ. Пусть число появлений подстроки $a_i a_j$ во всех строках $\alpha \in \Gamma$ равно $n_\Gamma(a_i a_j)$. Тогда несложно вывести формулу, определяющую изменение длины описания $L\left(\Gamma, \mathrm{A} \cup \left\{b = a_i a_{ij}\right\}\right) - L\left(\Gamma, \mathrm{A}\right)$:

$$\Delta L_{ij} = \left(N_\Gamma(\mathrm{A}) - n_\Gamma(a_i a_j)\right)\log_2\left(N_\Gamma(\mathrm{A}) - n_\Gamma(a_i a_j)\right) -$$
$$- N_\Gamma(\mathrm{A})\log_2 N_\Gamma(\mathrm{A}) + n_\Gamma(a_i)\log_2 n_\Gamma(a_i) + n_\Gamma(a_j)\log_2 n_\Gamma(a_j) -$$
$$- n_\Gamma(a_i a_j)\log_2 n_\Gamma(a_i a_j) -$$
$$- \left(n_\Gamma(a_i) - n_\Gamma(a_i a_j)\right)\log_2\left(n_\Gamma(a_i) - n_\Gamma(a_i a_j)\right) -$$
$$- \left(n_\Gamma(a_i) - n_\Gamma(a_i a_j)\right)\log_2\left(n_\Gamma(a_j) - n_\Gamma(a_i a_j)\right) + d_{ij}. \tag{3.28}$$

Здесь d_{ij} — длина описания нового символа, требующего указания двух символов $a_i a_j$, из которых он составляется с указанием оптимального кода, что требует примерно

$$d_{ij} = -\log_2 \frac{n_\Gamma(a_i)}{N_\Gamma(\mathrm{A}) - n_\Gamma(a_i a_j)} - \log_2 \frac{n_\Gamma(a_j)}{N_\Gamma(\mathrm{A}) - n_\Gamma(a_i a_j)} -$$
$$- \log_2 \frac{n_\Gamma(a_i a_j)}{N_\Gamma(\mathrm{A}) - n_\Gamma(a_i a_j)} \text{ бит.} \tag{3.29}$$

После введения нового символа число вхождений как каждого из символов a_i и a_j, так и общее число символов, из которых составлены словоформы, уменьшается на $n_\Gamma(a_i a_j)$. Отметим также, что для определения ΔL_{ij} не требуется вычисление энтропии символов для старого и нового алфавитов, присутствующее в формуле (3.26).

Величина ΔL_{ij} показывает изменения длины описания при введении нового символа. Если ее значение положительно, то введение нового символа нерационально. Пусть теперь ищется пара $a_i a_j$, дающая минимальное (отрицательное)

значение ΔL_{ij}. После замены вхождений этой пары на новый символ можно повторять данную процедуру уже для расширенного алфавита до тех пор, пока находится пара символов, формирование на основе которой нового символа приводит к уменьшению общей длины описания.

Мы применили этот алгоритм к списку словоформ, присутствовавших в текстах нескольких классических произведений. В результате первыми были объединены такие сочетания букв, как «ст», «пр», «по», «ся», «сь», «ть», «ль», «ой», «вы», «ющ», «ны», «ла», «го», «ра» и др. Далее на основе новых символов стали выделяться в качестве наиболее выгодных следующие сочетания: «ться», «лся», «раз», «про», «при», «вши», «лась», «лись», «ого», «пере» и т. д., а также более редкие двухбуквенные сочетания — «не», «ре», «па», «та», «от», «со» и др. Видно, что многие из них являются приставками, суффиксами, окончаниями или их устойчивыми конкатенациями, как, например, цепочки «лись», «лась» или «лся», регулярно встречающиеся в глаголах прошедшего времени. Дальнейшая работа алгоритма привела к формированию более длинных цепочек букв, из которых конструируются словоформы. В конечном итоге каждое слово оказалось иерархически декомпозировано на все более маленькие части (рис. 3.35).

Как видно из рис. 3.35, в результате могут формироваться элементы, соответствующие как частям реальных морфов («пе», «ре», «го» и т. д.), так и их конкатенациям («вшись», «нена», «ного» и др.). Уровень морфов как бы размывается, захватывая наиболее вероятные N-граммы как букв, так и морфов. Можно было бы попытаться избавиться от подобной иерархичности, если бы целью было автоматическое выявление морфологии языка. Для задачи распознавания речи это является излишним, так как такая иерархичность отражает реальные закономерности в бук-

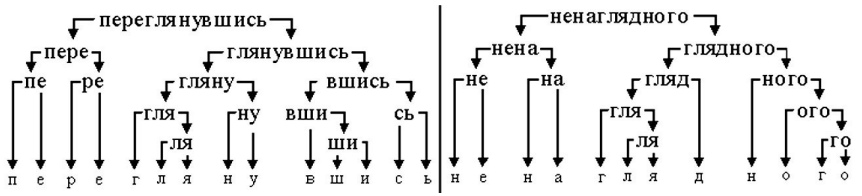

Рис. 3.35. Примеры иерархической декомпозиции словоформ на части, автоматически выявленные в ходе алгоритма последовательного слияния цепочек букв

венном представлении слов. Сами же морфы этим закономерностям далеко не всегда следуют. Например, нетривиально объединить в одну парадигму словоформы: «ем», «ешь», «едят», «ели» и т. д., не опираясь на их семантику.

Описанный алгоритм поиска не является оптимальным, так как решение об объединении символов, которое является локально оптимальным, впоследствии может оказаться не самым лучшим. Например, может быть принято решение об объединении букв «н» и «о» и лишь затем на их основе сформировать цепочки «ной», «ное», «ною».

Вместо последовательного слияния может использоваться и другой алгоритм. Мы предполагали, что слова записываются как цепочки букв, которые и составляют исходный алфавит. Вместо этого можно в качестве исходного алфавита взять сами слова и осуществлять их последовательное разбиение на части, как это предлагается в работе [345]. Не будем обсуждать преимущества и недостатки обоих методов, поскольку, вероятно, наилучшим решением будет их некое совместное использование.

Помимо того, что морфы являются структурными единицами, из которых конструируются словоформы, и поэтому могут быть выделены как устойчивые сочетания букв (фонем), некоторые морфы также связаны с грамматическими категориями, к которым относятся слова. Например, морфы «ся», «сь», «те» встречаются в глаголах, «ющ», «ящ» — в причастиях. Морфы связаны не только с частями речи, но и с падежом, числом, родом или временем, в котором используется данное слово. Наличие взаимной информации в морфе и грамматической категории позволяет использовать их совместно (по морфам, присутствующим в словоформе, можно предсказывать грамматическую категорию слова, а также осуществлять предсказание в обратную сторону). В частности, если в текстовом корпусе для каждого слова указана его часть речи, то эту информацию можно использовать для повышения качества выделения морфов. Принцип МДО позволяет корректным образом одновременно учитывать как повторяемость морфов в словоформах, так и соответствие некоторых морфов частям речи [346]. Однако на рассматриваемом здесь уровне анализа информация о частях речи недоступна.

Выделенные в результате автоматического обучения части слов оказываются весьма полезными при распознавании речи для языков с богатой морфологией [345]. Оценив

вероятности их совместного появления, можно надежно распознавать даже словоформы, не вошедшие в словарь системы анализа речи, поскольку предпочтение будет отдаваться цепочкам букв (или фонем — в зависимости от обучающих данных), наиболее согласованных с морфологией языка. Например, слово «разуверенность» не является общепринятым словом, но вполне может встретиться в речи для обозначения результата разуверения. Ему не будет соответствовать никакая запись в словаре системы распознавания, в связи с чем оно либо должно быть заменено на наиболее близкий (и неправильный) вариант, либо оставлено как цепочка отдельных символов (и будет наверняка содержать ошибки на уровне фонем). Распознавание же этого слова на уровне морфов и статистических взаимосвязей их появления позволит получить орфографический текст без ошибок и облегчить нахождение начала следующего слова.

Интересно отметить, что сходный анализ ведется на подсознательном уровне и человеком, изучающим язык, что хорошо видно по словотворчеству детей. Так, ребенок в возрасте от двух до пяти лет овладевает всеми тонкостями морфологии языка, его приставками, суффиксами и флексиями и, не зная всех устоявшихся словоформ, может как конструировать, так и правильно понимать смысл слов, не встречающихся во «взрослом» языке, но удовлетворяющих общим закономерностям словообразования. К примеру, в книге «От двух до пяти» К. И. Чуковский пишет [347, с. 107]: «Услыхав от какого-то мальчика, будто *лошада копытнула* его, я при первом удобном случае ввернул эти слова в разговор с моей маленькой дочерью. Девочка не только сразу поняла их, но даже не догадалась, что их нет в языке. Эти слова показались ей совершенно нормальными».

В действительности, анализ, проводимый мозгом малолетнего ребенка, гораздо более глубокий, чем простое выделение морфем (мы сейчас не говорим о других осуществляемых им видах лингвистического анализа), и опирается он не только на лингвистическую информацию. Приведем еще одну цитату из книги К. И. Чуковского [там же, с. 109]: «Хотя ребенок и не мог бы ответить, почему он называет почтальона *почтаником*, эта реконструкция слова свидетельствует, что для него практически вполне ощутима роль старорусского суффикса *ник*, который характеризует человека главным образом по его профессиональной работе — пожар*ник*, физкультур*ник*, сапож*ник*, печ*ник*. Называя по-

чтальона *почтаником*, ребенок включил свой неологизм в разряд этих слов и поступил вполне правильно, потому что если тот, кто работает в саду, есть садовник, то работающий на почте есть и вправду *почтаник*. Пусть взрослые смеются над почтаником. Ребенок не виноват, что в грамматике не соблюдается строгая логика». Отсюда и из других подобных примеров (см. [347]) видно, что мозгом ребенка осуществляется выделение взаимосвязей не только между лингвистическими объектами, но и понятиями, имеющими другую природу. Использование нелингвистической информации может помочь и для упорядочения и формирования различных лингвистических категорий.

Сделаем еще одно замечание. Как мы отмечали в п. 3.3.5, использование N-грамм слов или фонем сталкивается с проблемой размерности уже при сравнительно малых значениях N. В то же время взаимозависимость элементов речи гораздо обширнее. В чисто статистическом подходе приходится выбирать некоторое (эмпирически заданное) количество наиболее вероятных N-грамм, причем максимальное N фиксировано. Вместо того чтобы принять вероятность ни разу не встретившейся N-граммы равной нулю, эта вероятность считается через вероятности $(N-1)$-грамм. В алгоритме построения лексикона морфов мы использовали информационный критерий для определения, стоит ли вводить новый символ для обозначения цепочки букв, т. е. их N-граммы. Длина кода, соответствующего этому символу, использовалась вместо вероятности N-граммы. При этом могли формироваться цепочки произвольной длины в последовательности, учитывающей их вклад в уменьшение длины описания. Если какая-то цепочка букв не была объединена, то длина описания этой цепочки будет автоматически получаться как сумма длин описаний ее подцепочек; причем подцепочки будут выбраны однозначно по принципу МДО. Это соответствует вычислению вероятностей N-грамм, не вошедших в список наиболее часто встречаемых.

Таким образом, в информационном подходе естественным образом разрешаются проблемы, возникающие в статистическом подходе и требующие там введения дополнительных эвристик. Это неудивительно с учетом того, что в статистическом подходе не принимается во внимание сложность модели (описание вероятностей N-грамм). Минимизация длины описания может использоваться вместо любого применения N-грамм: для фонем, морфов или слов.

3.3.7. Ограничения рассмотренных методов машинного восприятия

Мы рассмотрели проблему распознавания речи и убедились, что каждый его этап может быть выполнен под руководством целевой функции, составленной в соответствии с принципом МДО. В дополнение к цитировавшимся выше работам можно привести и другие статьи, в которых принцип МДО используется для задач анализа речи, например, [348–351]. Как и в случае интерпретации изображений, анализ речи оказывается целесообразно проводить с использованием иерархии представлений, включающей уровень акустического сигнала, различительных признаков фонем, фонем, морфов, словоформ и словосочетаний. Можно заметить, что на каждом уровне оказывается полезным использовать более высокоуровневую и контекстную информацию, например, качество распознавания фонем или слов может быть повышено, если учитывать взаимную информацию между ними и соседними элементами того же уровня или длину описания элемента следующего уровня, в который входит данный элемент. Подробно этот вопрос рассмотрен в п. 3.5.

Как при анализе изображений, так и при распознавании речи использовались методы обучения с учителем или представления задавались в явном виде. Например, для обучения подсистемы распознавания фонем требовалось предварительное обучение по записям речевого сигнала, согласованного с правильными фонемными последовательностями, а для сегментации изображений — задание семейств регрессионных моделей. Единственная попытка применить обучение без учителя заключалась в автоматическом определении лексикона морфов (см. п. 3.3.6).

Машинная система, построенная в таком стиле, не сможет самостоятельно обучаться новому языку — она будет требовать для этого участия людей-экспертов. Подобная система распознавания речи не сможет адекватно интерпретировать даже простейшие звуки, не являющиеся звуками речи (звонок в дверь или гудок поезда), не говоря уже про более сложные неречевые акустические сигналы (например, музыку). Для создания системы машинного обучения общего назначения это недопустимо. При разработке подсистемы зрения использование большого объема априорной информации можно считать менее сильным ограничением, полагая, что если от системы может потребоваться научить-

ся понимать новый язык, то научиться видеть «новый мир» — вряд ли. Тем не менее и система зрения должна обладать определенной гибкостью, в частности, желательно, чтобы она могла использоваться для разных типов сенсоров (например, получающих изображения в разных диапазонах длин волн) без принципиальной доработки.

Еще одна проблема, не рассмотренная здесь, — проблема смысла. При разговоре об интерпретации изображений мы ограничились анализом промежуточного символьного представления и не затронули семантический уровень, а при обсуждении речевых технологий ограничились системами распознавания речи, а не ее понимания. Мы это сделали намеренно, поскольку вопрос понимания семантики выходит за рамки одной модальности. Действительно, под пониманием изображений обычно подразумевается возможность их описания на естественном языке, а под пониманием речевого сообщения — возможность соотнести описанную в нем ситуацию с объектами реального мира (например, зрительно представить описанную сцену). Понимание смысла предъявляемой информации — это способность представить ее разными способами, перевести из одного представления в другое (например, чтобы убедиться, понимает ли человек некоторое высказывание, можно его попросить переформулировать это высказывание в других терминах). Классические системы понимания (изображений или речи), которые в процессе функционирования оперируют только с одним типом апостериорной информации, а вся информация другого типа в них закладывается априорно, кажутся нам неперспективными для создания систем машинного обучения общего назначения.

Ниже будет рассмотрен пример подхода, не требующего введения высокоуровневой информации априорно, а извлекающего эту информацию на основе анализа данных, полученных от сенсоров различных модальностей. Говоря более конкретно, мы опишем в общих чертах подход к проблеме машинного обучения, разрабатываемый в Media Laboratory Массачусетсского Технологического Института. В рамках данного подхода машинной системой осуществляется выделение из слитной речи слов, заранее ей неизвестных, и их связывание со зрительными образами. В результате обучения система оказывается способна назвать визуально предъявленный ей предмет и по названию предмета выбрать его из некоторой совокупности других пред-

метов. Таким образом, можно утверждать, что подобная система действительно понимает смысл слов, соответствующих некоторым предметам, причем перечень предметов и слов, распознаванию которых система может научиться, не предопределен, а само обучение осуществляется на основе естественных данных (аудиовидеозапись игры нянь с младенцами), подготовка которых не требует участия специалистов. Хотя сама система использует существенные упрощения, базовые принципы, заложенные в нее, являются очень сильными.

3.4. ФОРМИРОВАНИЕ ЛИНГВИСТИЧЕСКИХ ЕДИНИЦ, ОСНОВАННЫХ НА СЕМАНТИКЕ, НА ПРИМЕРЕ СИСТЕМЫ CELL

3.4.1. Проблема смысла референтных выражений

Многие вопросы, рассматривающиеся в философии, имеют свое отражение в области искусственного интеллекта. С одной стороны, это создает определенную методологическую базу для построения вычислительных моделей когнитивных процессов, а с другой — позволяет на практике проверять адекватность философских концепций. Мы рассмотрели проблему индуктивного вывода, а здесь кратко остановимся на проблеме смысла языковых выражений.

Для ответа на такие вопросы, как «Что такое смысл высказывания?» или «Какие предложения являются осмысленными, а какие — бессмысленными?», строятся семантические теории языка. В аналитической философской традиции, включающей, в частности, неопозитивизм и современные логико-философские школы, производится попытка построения формальной системы наподобие логического исчисления, которая бы описывала семантику некоторого естественного языка, т. е. определяла бы смысл (или его отсутствие) любого предложения на основе только лингвистической информации. Тем самым производится абсолютизация семантики языка, рассмотрение языка как сложившейся, застывшей структуры, не зависящей от носителей языка [352, с. 17, 18]. Такая методологическая предпосылка ведет к невозможности объяснения феномена обучения языку и возможности передачи нового знания с помощью языка.

Эти тенденции нашли свое отражение и в исследованиях в области ИИ. К примеру, семантические сети, используемые для представления знаний, описывают смысл некоторого понятия (необязательно, но, как правило, маркирующегося некоторой лингвистической конструкцией) через его взаимосвязи с другими понятиями. С нашей точки зрения, такая сеть не содержит семантику, несмотря на свое название. Эта парадигма, используемая в экспертных системах или в системах машинного восприятия, основанных на знаниях, осложняет развитие данных отраслей. Указанные выше философские проблемы здесь представляются как проблемы крайней ограниченности возможности подобных систем учиться на собственном опыте. Они оказываются неспособными к самостоятельному формированию новых понятий и включению их в свою базу знаний. Опора на готовое знание при проектировании таких систем полностью соответствует исследованию сложившегося языка в аналитической философии, в которой мыслительные структуры сводятся к языковым [352, с. 118].

Учет феномена усвоения языка и необходимость построения обучающихся систем заставляет рассматривать проблему смысла как часть общей гносеологической проблемы, включающей также и нелингвистическую составляющую. Именно такое философское рассмотрение проводится в книге [352]. В ней автор раскрывает неразрешимость проблемы смысла языковых выражений вне *концептуальной системы* носителя языка как системы его мнений и знаний о мире, отражающих его познавательный опыт на доязыковом и языковом этапах и уровнях, и не сводимой к какой бы то ни было лингвистической сущности [352, с. 12]. Исходно понятие концептуальной системы было введено Куайном для представления языка как состоящего из предложений, расположенных на разных уровнях этой системы, начиная от периферии и кончая внутренней, центральной, наиболее удаленной от контактов с физическим миром частью. Периферийные предложения системы, или *предложения наблюдения* (такие, как «Это красное», «Идет дождь» и т. п.), представляют собой точки соприкосновения системы с физической реальностью [352, с. 92]. «С предложений наблюдения, согласно Куайну, начинается усвоение естественного языка. Эти предложения являются отправными точками и научной теории. От них путем определенных языковых и поведенческих манипуляций, основы-

вающихся на аналогии и индукции и на подкреплении, выделении правильных, адекватных реакций, совершается переход» к предложениям, относящимся к центральной части концептуальной системы [352, с. 94]. В этой работе указываются недостатки такой концепции (в частности, в ней есть возможность иметь дело только с наблюдательными феноменами) и осуществляется ее расширение. Мы не будем подробно на этом останавливаться, так как в описываемой ниже вычислительной модели рассматривается лишь начальный этап формирования концептуальной системы и усвоения языка, опирающийся всецело на наблюдательные феномены и связанный с проблемой *референции*.

Возможность указания или референции на объекты мира принципиальна для построения речевых высказываний о свойствах и отношениях этих объектов. В естественном языке функция референции «отводится *сингулярным терминам: собственным именам и определенным дескрипциям*. В случае собственных имен указание на объект, или референт, осуществляется посредством его *называния* (например, „Аристотель", „Афродита"), в случае определенных дескрипций — посредством его *описания* („ученик Платона и учитель Александра Македонского", „богиня любви"). За кажущейся простотой этих вещей стоит, однако, одна из наиболее глубоких, фундаментальных проблем философии языка, заключающаяся в выяснении семантического статуса сингулярных терминов, или проблема *их осмысленности*. Она состоит, в частности, в поиске ответа на вопрос о том, имеют ли *собственные имена смысл*, т. е. являются ли они осмысленными, аналогично предикатам — прилагательным, общим существительным, глаголам и построенным из них определенным дескрипциям» [352, с. 122].

Возможность указания на объект мира требует способности выделить этот объект в окружающем мире и распознать его на основе нелингвистической информации, что также является когнитивным актом (или результатом индуктивного вывода). Именно благодаря связыванию нелингвистической информации об объекте с соответствующим ему сингулярным термином последний оказывается вовлеченным в подлинно семантические отношения с миром, т. е. обретает смысл. Результат такого связывания — *концепт*, т. е. элемент концептуальной системы.

Смысл терминов оказывается субъективным, поскольку при образовании концептов каждый раз используется несколько

различная информация: разные носители языка обладают различным знанием о мире. Овладение языком, как и вся познавательная деятельность, идет постепенно: при построении новых концептов используется информация об уже построенных. Роль языка заключается в том, что, будучи вовлеченным в процесс построения концептуальной системы, он значительно облегчает выход за непосредственный опыт и формирование концептов, содержащих абстрактные понятия.

Очевидно, проблемы с автоматическим обучением, возникающие в экспертных системах, связаны с тем, что в них не производится последовательное построение субъективной концептуальной системы. Вместо этого в них закладывается объективное «знание» предметной области, однако лишенное семантики, что делает крайне затруднительным автоматическое построение новых концептов, согласованных с уже имеющимися знаниями, добавление которых не нарушало бы целостность концептуальной системы.

Иной подход к проблеме знания и смысла развивается в работах [288–290; 353–354] в Media Laboratory Массачусетсского Технологического Института (есть и другие группы исследователей, придерживающиеся сходных подходов, см., например, работу Р. Брукса [355]). На наш взгляд, этот подход относится к классическим способам представления знаний (таким, как семантические сети или фреймы), как цитированная выше доктрина концептуальных систем относится к аналитической традиции в философии (не отвергая ее, но расширяя). В работе [206] подробно рассматривается вопрос построения концептов, непосредственно связанных с данными наблюдений, для чего, по сути, производится попытка моделирования начальных этапов процесса усвоения языка младенцами. При этом указывается важность экстралингвистической информации в этом процессе. Воплощением этой компьютерной модели стала система CELL (Cross-channel Early Lexical Learning), которая впоследствии той же группой исследователей была развита в системы Describer, Newt, Ripley, Fuse.

3.4.2. Общая архитектура системы CELL

Представим себе машинную систему, оснащенную сенсорами, возможно различных модальностей. В описываемом подходе предполагается, что информация, поступающая от сенсо-

ров, разделяется на каналы, являющиеся либо лингвистическими, либо семантическими (контекстными). Введение информационных каналов в дополнение к сенсорным модальностям вызвано тем, что как лингвистическая, так и семантическая информация может поступать от сенсоров любой модальности.

К примеру, лингвистический канал может содержать извлеченную из акустического сигнала речь или извлеченные из визуального сигнала жесты, движение губ или письменную речь. Тактильный сигнал также может нести для человека лингвистическую информацию, о чем говорит опыт обучения слепоглухонемых детей (см., например, [356, с. 53–58]). В то же время любая из этих сенсорных модальностей может нести и семантическую информацию. Например, семантический канал может содержать описание формы, извлеченное из визуального или тактильного сигнала, описание цвета, движения и т. д.

Человек способен самостоятельно выделять лингвистический канал (младенец, естественно, заранее не знает, что какие-то поступающие сигналы являются лингвистическими). Но автоматическое разделение каналов в описываемой работе не рассматривается, а полагается, что корректное разделение каналов заложено в систему априорно. Видимо, это связано не с какими-то принципиальными сложностями, а с соображениями переноса разработанной системы на практические задачи. Действительно, как мы увидим ниже, методы работы с каналами в большой степени симметричны, и обозначение одного как лингвистического, а второго как семантического относится, скорее, к интерпретации работы системы разработчиком. Отметим, что и младенцы не различают услышанные слова как особый вид сенсорных данных, а использование ими самими речи — как особый вид деятельности [357, с. 192–193]. Это обусловливает мистицизм, присущий детскому мышлению: имя предмета (как часть сенсорного образа, связанного с этим предметом) не отличается от самого предмета, представленного другой частью (например, зрительной) сенсорного образа. Отсутствие такого различения вызывает попытку изменять реальность с помощью слов (с этим связаны магия, шаманство в примитивных обществах [357, с. 120–121]), т. е. различение слова и дела, предмета и его названия происходит на более поздних этапах развития индивида, чем моделируемый в обсуждаемой работе начальный этап усвоения языка. Вернемся, однако, к описанию технической системы.

В этой системе в соответствующих каналах лингвистическая или семантическая информация кодируется посредством системы признаков, процедура выделения которых считается данной априорно. Для поступающих по каждому каналу векторов признаков можно решить задачу их группирования и сформировать классы образов. Если речь идет о семантических признаках, то подобные классы соответствуют некоторым *семантическим категориям*, а если о лингвистических — *лингвистическим единицам*. В процессе обучения без учителя (см. п. 2.4), как правило, строятся классы образов, описывающиеся эталонным образом, прототипом и величиной допустимых отклонений от прототипа, при которых образ все еще относится к данному классу. Именно так в системе CELL и задаются как семантические категории, так и лингвистические единицы.

Лексемой в данной работе называется концепт, включающий в себя некую лингвистическую единицу, ассоциированную с некоторой семантической категорией. Сложность заключается в том, что требуется не просто связать лингвистическую единицу и семантическую категорию, но одновременно с этим и построить соответствующие им классы образов, поскольку они не даются системе априорно.

Формирование классов образов для каждого из каналов в отдельности можно было бы осуществлять в стиле обучения без учителя. Однако богатство поступающей информации столь велико, что надежно выделять классы затруднительно. Таким образом, следует решать задачи построения классов и их связывания одновременно. Одна из основных идей заключается в том, что использование лингвистической информации позволит надежнее выделять релевантные семантические категории, а использование семантической информации (контекста) — надежнее вести обработку лингвистической информации (выделять и осуществлять классификацию отдельных лингвистических единиц из потока слитной речи). Посмотрим, как это было сделано в описываемой работе.

Для семантического и лингвистического каналов вводится понятие S- и L-событий. Событие — это последовательность выделенных признаков в интервале времени, в котором наблюдается активность в соответствующем канале. Для лингвистического канала это соответствует цепочке лингвистических признаков (пока не конкретизируем, каких именно), возможно, разделенных паузами, обособляющи-

ми речевое высказывание. Например, при устном вводе L-событие соответствует цепочке произнесенных вслух слов и может быть обнаружено детектором речь/молчание. S-событие соответствует заметным изменениям в сенсорном канале, например достаточно быстрому движению. Критерий значимости изменений в канале не вводится и предлагается определять его эвристически для каждого отдельного канала.

Итак, пусть в каждом канале выделены события как цепочки признаков. Как L-, так и S-событие может включать образы, относящиеся к различным классам. Такие протяженные во времени события необходимо разбить на некоторые более короткие временные интервалы, сегменты. Для лингвистических событий эти временные интервалы определяют гипотетические границы лингвистических единиц. Если используется несколько лингвистических каналов (например, фонемный и движение губ), то гипотезы об интервалах могут взаимно проверяться и делаться более достоверными. В семантических каналах также необходимо ввести разделение на сегменты. Границы сегментов соответствуют резкому (во времени) изменению скорости, контраста, цвета и т. д. Это означало бы изменение распознаваемых семантических категорий, если бы они были уже сформированы. В связи с этим вводится понятие L- и S-подсобытий как подпоследовательностей в цепочках признаков, составляющих события.

Речь, обращенная к младенцам, обычно относится к мгновенному контексту. Значит, при овладении лексикой является принципиальным одновременное появление лингвистических единиц и соответствующих им семантических категорий, представленных в каналах L- и S-подсобытиями. Поскольку в описываемой работе моделируется восприятие младенцев и используются записи игры с ними, то делается предположение, что L- и S-подсобытия, относящиеся к одному и тому же объекту, присутствуют в перекрывающихся во времени L- и S-событиях. Когда такие перекрывающиеся события встречаются, то формируется единое LS-событие, которое и подлежит дальнейшему анализу. Это событие помещается в кратковременную память, являющуюся FIFO-буфером (очередью). Размер этого буфера в системе CELL был взят 7 ± 2 LS-событий. Помимо ссылки на моделирование человеческого восприятия другим аргументом является экономия вычислительных ре-

сурсов, поскольку поиск в кратковременной памяти повторяющихся событий является исчерпывающим и требует экспоненциального времени.

После помещения LS-событий в буфер требуется выделить подсобытия, в частности, необходимо осуществить сегментацию слитной речи на отдельные слова при неизвестном лексиконе. При этом используется предположение, что в помещенных в буфер LS-событиях значимые подсобытия неоднократно повторяются, поэтому их можно выделить как наиболее длинные, повторяющиеся (с определенной точностью) цепочки признаков. Идея поиска повторяющихся подсобытий основывается на том, что речь, обращенная к младенцам, является чрезвычайно избыточной. Если младенцу дают в руки какую-то игрушку, то ее название повторяется много раз («посмотри, какой мячик; мячик круглый, ...»). Поскольку подсобытия повторяются неточно, то, чтобы их искать, вводится (в качестве дополнительной априорной информации) метрика в каждом L- и S-каналах. Пары повторяющихся и совместно встречающихся L- и S-подсобытий помещаются в память следующего уровня, имеющую значительно больший объем. Если во время обнаружения следующих LS-событий появляются похожие (в смысле близости в пространстве признаков) значимые пары подсобытий, то на их основе строится лексема (концепт), состоящая из эталонной семантической категории, эталонной лингвистической единицы и радиусов в пространствах признаков, определяющих границы концепта. Это аналогично инкрементному обучению без учителя.

Принятие решения о формировании лексемы осуществляется на основе взаимной информации, содержащейся в соответствующих L- и S-подсобытиях. Пусть имеется набор пар совместно встречающихся подсобытий (L_i, S_i). Тогда для каждой из них можно оценить безусловные вероятности $P(L)$ и $P(S)$ их появления, а также вероятность совместного появления $P(L, S)$. Взаимная информация (см. п. 1.3.2) $\log_2 P(L, S) - [\log_2 P(L) + \log_2 P(S)]$ будет показывать оценку выигрыша в длине совместного описания L- и S-подсобытий по сравнению с их независимым описанием. Суть уменьшения длины описания состоит не в том, что векторы признаков, соответствующих L- и S-подсобытиям, как-то связаны, а в том, что сами подсобытия появляются совместно, т. е. появление одного события может быть предсказано при появлении другого подсобытия.

Если оцененная взаимная информация превосходит некоторый порог, то концепт формируется. В процессе функционирования системы, формирования ею концептов рассматриваются в промежуточной памяти только такие события, которые не распознаются как сформированные лексемы.

3.4.3. Реализация зрительной и акустической подсистем в системе CELL

Один из возможных вариантов реализации описываемой системы использует аудио- и видеовход. Мы приводим ее краткое описание, чтобы читатель получил представление о закладываемых в конкретную реализацию ограничениях. Более подробная информация приведена в оригинальных работах. В этой реализации аудиоинформация получается с микрофона, на который записывается естественная, слитная речь воспитателя, обращенная к младенцу. В качестве семантических каналов используются каналы, содержащие информацию о форме и цвете объекта, присутствующего в поле зрения камеры.

Ставятся задачи выявления лингвистических единиц и семантических категорий формы и цвета и их связывания в концепт таким образом, чтобы в результате обучения система могла называть предъявленные ей объекты, а также, наоборот, по названию объекта выбирать его из заданного набора. Таким образом, в рамках построенной концептуальной системы реализуется функция референции на доступные непосредственному наблюдению объекты.

Для выделения лингвистических и семантических признаков необходимы модально-специфичные процедуры, которые считаются заданными априорно. Опишем, как ведется выделение семантических признаков в двух каналах: формы и цвета.

Трехмерная форма объекта в явном виде не восстанавливается. Вместо этого используются сгруппированные описания плоских форм, которые получаются в результате съемки объекта с разных ракурсов. На каждом таком снимке осуществляется разделение на объект и фон, при котором используется предположение о том, что фон является сравнительно однородным (однотонная скатерть, простыня и т. д.), на который помещен крупный предмет близко

к центру кадра. Все пиксели, которые не относятся к фону, помечаются как принадлежащие к объектам. Находятся связные области из таких пикселей, и из этих областей выделяется наибольшая, расположенная ближе к центру кадра. Также рассматривается случай, когда подходящий объект отсутствует (установлением порога на размер и положение) или не полностью попадает в кадр. Далее анализируются форма и цвет выделенной таким образом области, предположительно соответствующей изображению объекта.

Для представления цвета используется двухцветная гистограмма 8×8 ячеек нормализованных цветов (r, g), инвариантная к смене интенсивности освещения. Поскольку цвета каждого пикселя нормированы на его уровень яркости, то третий цвет дополнительной информации не несет и в представлении не используется.

Описание формы тоже осуществляется с помощью двумерной гистограммы. Для ее построения находятся точки, лежащие на границе выделенной области. Для них определяются наклоны касательных, проведенных к границе. Также для каждой пары точек границы определяется нормализованное расстояние. По этим двум параметрам строится гистограмма, которая и описывает форму области. Это представление инвариантно к масштабу и повороту объекта в плоскости изображения.

Как видно, представление изображений весьма простое и опирается на сильные предположения. В нем не выделяется значимая информация, а часть ее теряется. Например, нормирование цвета каждого пикселя на его яркость делает неразличимой шахматную доску от белого листа бумаги, а само представление остается неинвариантным к освещению сцены светом, отличным от белого. Еще больше упрощения касаются канала, содержащего информацию о форме: на основе использованных в системе CELL признаков формы хорошо различимы лишь объекты простых форм. Также не описываются другие аспекты изображения, например, текстура поверхности. Такие упрощения, однако, не умаляют значимость работы в области моделирования процесса построения концептов.

Заметно большее внимание в работе уделено лингвистическому каналу. В реализации предполагалось, что система обучения обладает информацией о фонетической структуре (английского) языка до начала обретения знания слов.

Извлечение лингвистических признаков разбивается на два этапа. Первый этап анализа — спектральный, на котором выделяются спектральные признаки и подавляются гармоники, не относящиеся к речи. На основе этих признаков ведется классическое распознавание образов для заданных классов, соответствующих фонемам. Для этого используется RNN (рекуррентные нейронные сети). Мы не останавливаемся на их описании, поскольку использование именно этого аппарата здесь не принципиально. Также мы не останавливаемся более детально и на вопросе выделения спектральных признаков, так как оно ведется в классических традициях анализа речи.

Извлеченные лингвистические и семантические признаки далее используются для формирования лингвистических единиц и семантических категорий и для их связывания в концепты, как было описано в п. 3.4.2.

3.4.4. Основные результаты тестирования системы CELL

Система CELL, оснащенная акустическими и видеосенсорами, обучалась по записям игры взрослых с младенцами примерно десятимесячного возраста [352, гл. 5]. Для обучения использовались записи игры разных взрослых с младенцами с привлечением различных объектов (игрушек, мячей, обуви и т. д.). Одна запись относилась к одному предмету. Уклон в работе был сделан на анализ речи: речь была естественной, слитной, могло встречаться много лишних слов, не относящихся к предъявляемому объекту, а взрослым не давалось указания учить младенцев словам. В то же время сегментация изображения была существенно облегчена, присутствовал и хорошо выделялся лишь единственный объект, в отличие от множества лишних слов. Тем не менее изображения были также реальные.

Для оценки качества работы системы измерялись точность сегментации слитной речи на слова, процент правильного выделения лингвистических единиц и процент их правильного связывания с соответствующими объектами, что проверялось по тому, правильное название визуально предъявленного объекта или нет давалось впоследствии системой. Здесь есть определенная неточность: связывание лингвистических единиц происходит не с объектом, а с се-

мантической категорией. К примеру, слово «мяч» ею будет связываться с категорией круглой формы, и мяч будет неотличим от другого круглого объекта (того же цвета, если также используется канал цвета).

В качестве опорной системы для сравнения использовалась система, опирающаяся только на акустическую информацию. Она пыталась осуществить сегментацию слитной речи на слова и построить лингвистические единицы, не связанные с семантическими категориями, для чего производился поиск наиболее часто повторяющихся фрагментов речи, как и в исходной системе CELL.

Как качество сегментации, так и процент правильно выделенных лингвистических единиц, оказался заметно выше в случае привлечения семантической информации, чем при использовании только лингвистической информации — цепочек выделенных из акустического сигнала фонем. В частности, процент правильно построенных лингвистических единиц (т. е. единиц, соответствующих словам языка) в первом случае составил около 70 %, во втором случае — лишь около 30 %. Это означает, что присутствие контекстной, семантической, информации заметно облегчает усвоение языка. Также можно сделать и симметричное утверждение: лингвистическое подкрепление облегчает формирование семантических категорий, даже столь непосредственно связанных с чувственным опытом. Доля же правильного связывания лингвистических единиц и семантических категорий в системе CELL составила около 60 %. При обучении без учителя и при использовании простых представлений изображений, а также сравнительно небольших обучающих выборок это является весьма хорошим результатом.

В работах [206, 358] описываются и практические приложения, в частности, возможность построения адаптивных интерфейсов на основе принципов, развитых и проверенных в системе CELL. В существующей парадигме построения интерфейсов пользователь должен адаптироваться к интерфейсу, предложенному разработчиком, имея в лучшем случае ограниченные возможности его ручной настройки. В парадигме адаптивных интерфейсов сам интерфейс автоматически адаптируется к конкретному пользователю, выявляя в процессе обучения взаимосвязи между (устными) командами, имеющими смысл лингвистических единиц, и соответствующими им (с точки зрения пользователя) действиями, имеющими смысл семантических категорий.

3.4.5. Дальнейшее развитие системы CELL

Система CELL является наиболее ранней и на данный момент имеет ряд расширений. Это системы Describer [289, 359] и Newt [354], робот Ripley [353, 360, 361], система Fuse [290] и др. [288, 362].

В системах Describer и Newt семантическая основа подводится не только под отдельные слова, но и под целые референтные выражения, описывающие пространственные отношения между объектами, такие как «большой синий квадрат справа от красного прямоугольника». Система Describer учится по примерам смоделированных сцен, сопровождающихся выполненными людьми описаниями, по которым можно выбрать нужный объект. Система Newt учится по изображениям реальных сцен, сопровождающихся устным описанием того объекта, на который система указывает лазерной указкой (на сцене может присутствовать несколько объектов, помещенных на однородный фон). Системы Describer и Newt отличаются в том, что в первой осуществляется описание выбранного объекта на сцене, во второй — выбор объекта по его словесному описанию. Помимо проблемы связывания слов с семантическими категориями здесь возникают также две дополнительные проблемы: разделение слов на классы (например, слова «желтый», «розовый», «зеленый» должны принадлежать одному классу, а «над», «слева», «рядом» — к другому классу) и определение порядка слов в предложении (обучение синтаксису). Выделение классов слов рассматривается в качестве необходимого этапа перед выявлением синтаксиса. Без знания синтаксиса невозможно различить такие выражения, как «the box next to the ball» и «the ball next to the box». В русском языке выражения будут различимы из-за разных словоформ, соответствующих разным падежам, но это вовсе не снимает полностью проблему обучения синтаксису, а переводит ее часть в проблему обучения морфологии языка, которая в цитируемых работах не рассматривается.

В роботе Ripley подход получил дальнейшее развитие по нескольким направлениям. Во-первых, была расширена сенсорная система робота за счет добавления в нее проприоцепторной подсистемы (у животных проприоцепторы располагаются в суставах, связках и мышцах), чувства гравитации и тактильной подсистемы (чувства осязания). Во-вторых, добавлена система эффекторов: у робота появилась

возможность совершать движения и брать предметы манипулятором. И, в-третьих, была добавлена ментальная модель окружающего физического мира, в которой описываются пространственные положения обнаруженных роботом объектов. Расширение системы сенсоров и эффекторов дает роботу возможность в процессе обучения устанавливать семантическую опору для таких слов, как «легкий», «тяжелый», «дотрагиваться», «поднимать», «давать» и т. д. Ментальная модель физического пространства не только позволяет осуществлять в нем навигацию, но и позволяет понимать различия в таких фразах, как «мяч слева от меня» или «мяч слева от тебя». Для этого в ментальной модели реализована возможность смены точки зрения, что осуществляется с помощью библиотеки 3D-моделирования OpenGL, на основе которой и строится воображение робота. Таким образом, слова «мой» или «твой» находят свою семантическую опору в ментальной операции переноса точки зрения. Другая возможность переноса точки зрения — ее перенос во времени, для чего оказывается необходимым вводить память, которая бы хранила историю состояний мира. Обращение к такой памяти дает семантическую основу для понимания времен глаголов.

В системе Fuse основное внимание уделяется вопросам взаимодействия зрительной и акустической подсистем на нижнем уровне, т. е. влиянию речи на управление вниманием зрительной подсистемы и влиянию визуальной информации на улучшение сегментации и распознавания слов и понимания речи. На этих вопросах мы остановимся в п. 3.5.6.

3.4.6. Нерешенные проблемы автоматического построения концептуальных систем

Отличительной чертой системы CELL, как и ее расширений, является обучение без учителя по естественным, необработанным данным. Как мы видели на примерах различных подзадач задачи распознавания речи, в использующиеся на практике системы закладывается большое количество априорных данных, которые подготавливаются вручную. Система, подобная CELL, сама извлекает сведения о среде, что делает ее гораздо более универсальной. Конечно, необходимость обучения перед эксплуатацией далеко не

всегда удобна с практической точки зрения, однако такое обучение не задерживает начало эксплуатации, а автоматизирует процесс проектирования. Именно такие адаптивные самообучающиеся системы в будущем будут играть все большую роль как при создании человеко-машинных интерфейсов, так и во многих других областях.

Вторая важная особенность данной системы заключается в совместном использовании лингвистической и семантической информации в процессе построения концептов, что противопоставляется символическому подходу к семантике [363]. Можно сказать, что у такой системы, если она выделит лингвистическую единицу «мяч» и свяжет ее с соответствующим сенсорным образом, действительно есть некоторое понимание слова «мяч». В семантических сетях такое понимание сингулярных терминов отсутствует; оно выражается через их связи с другими терминами, также лишенными индивидуального смысла и не опирающимися на семантические категории. Эта же особенность автоматического построения концептов отличает систему CELL и от систем интерпретации изображений, основанных на знаниях, в которых связывание элементов представления изображений и подсистемы, представляющей знание о предметной области, производится вручную. Системы понимания изображений и речи, на наш взгляд, должны строиться в духе системы CELL, именно поэтому мы их не рассматривали в предыдущих параграфах, посвященных каждой из модальностей в отдельности.

Но, несмотря на свои достоинства, система CELL моделирует лишь один из аспектов начального обучения младенцев и имеет множество упрощений и ограничений, необходимость снятия которых открывает интересные направления для исследования. Рассмотрим некоторые из них.

Усложнение представлений сенсорной информации. Привлеченное представление изображений было весьма упрощенным и неинформативным по сравнению с уже разработанными в иконике, что компенсировалось ограничениями, наложенными на организацию сцены. При этом не все объекты, визуально различимые человеком, могут быть различены в рамках указанного представления. Оно также не отражает пространственные отношения между объектами, описываемые наречиями: «между», «снаружи», «внутри», «слева» и т. д. Последнее ограничение частично преодолено в системах Describer и Newt, но и в них представление

изображений остается весьма упрощенным, что отмечают и сами авторы подхода [360]. Интересной задачей является включение в систему, подобную CELL, богатого иерархического структурного представления изображений, в котором выделялась бы релевантная информация на изображении и которое допускало бы более широкий класс семантических категорий. Интересно было бы также проследить отражение различных семантических категорий, выражаемых в рамках такого представления, в лингвистических конструкциях и возможность их связывания в концепты, а также рассмотреть и симметричную проблему, которую можно сформулировать следующим образом: как нужно расширить существующие представления изображений, чтобы они предоставляли максимально широкую семантическую базу для используемых человеком лингвистических конструкций?

Представление речевого сигнала может быть также расширено посредством учета структурных связей между различными лингвистическими объектами разных уровней (фонемами, морфами, словоформами и т. д.). В системе Describer сделан первый шаг к такому расширению с помощью *N*-грамм слов. В конечном итоге хотелось бы, чтобы система могла овладеть грамматикой языка, но это невозможно без дальнейшего расширения семантической базы. В частности, в п. 3.3.6 мы указывали на затруднительность автоматического выявления морфов только на основе лингвистической информации, поскольку разные морфы имеют различную семантическую нагрузку.

Расширение семантической базы. В системе CELL моделируется начальный этап формирования лишь той парадигматической подсистемы языка, которая обеспечивает номинативную функцию речи. Иначе говоря, система способна обучиться лишь существительным, обозначающим конкретные предметы. Более поздняя система Describer способна также различать семантические категории пространственных отношений между предметами («над», «слева») и отдельные признаки предметов («красный», «зеленый», «узкий» «широкий»). Робот Ripley способен понимать другие признаки предметов («легкий», «тяжелый») за счет расширения системы сенсоров, а также некоторые глаголы («дотрагиваться», «опускать») за счет введения системы эффекторов. Однако вопрос формирования системы предикативной (глагольной) функции языка, а также синтагматиче-

ской (объединение слов в связные высказывания) и просодической его систем как отдельных систем оставлен нерассмотренным. Глаголы, предложения (как целостные объекты) и интонационно-мелодическая сторона речевого высказывания не имеют семантической опоры в (статических) зрительных образах. Значит, для построения концептуальной системы, способной включать эти элементы, необходима более широкая семантическая база.

Здесь уместно обратиться к нейропсихологическим данным. Оказывается, что при локализованных повреждениях мозга разные структуры речи (просодическая, синтагматическая, парадигматическая и т. д.) могут нарушаться независимо [133]. Оказывается также, что нарушения номинативной (именной) функции речи тесно связаны с сенсорными расстройствами [133, с. 111, 112], а предикативной (глагольной) — с повреждениями премоторных систем [133, с. 78, 81, 143]. Можно с уверенностью предположить, что глаголы, в отличие от существительных, имеют семантическую опору не в рецепторах, а в эффекторах. Это в начальной форме используется в роботе Ripley, хотя явное разделение функций языка и их связи с принципиально разной семантической основой не проводится. Подобные вопросы, на наш взгляд, являются крайне важными, поскольку позволяют определить компоненты, необходимые для создания ИИ (к примеру, ответить на вопрос: может ли интеллект быть пассивным наблюдателем, т. е. быть лишенным эффекторов?).

В связи с этим отметим еще один момент, касающийся естественного интеллекта. Как известно (см., например, [356, с. 55–65; 357, с. 101, 102]), мышление примитивных людей тесно связано с движениями собственного тела, особенно с движениями рук. Например, индейцы «умеют думать руками, как современный человек иногда может думать вслух» [356, с. 62]. Опора в таком мышлении именно на «ручные понятия» связана с разнообразием движений, выполняемых руками при изготовлении различных вещей (действие при этом направляется на конкретный предмет). Танец в первобытных обществах также тесно связан с процессом мышления [356, с. 62]. По словам Эйзенштейна, на таком уровне развития человека «двигательный акт есть одновременно акт мышления, а мысль — одновременно пространственное действие» [356, с. 62]. Если принять, что предметы, над которыми выполняются действия, и сами двигательные акты через сенсорную и эффекторную системы

дают семантическую опору для лингвистических единиц, то механизмы, связывающие движение, язык и мышление, становятся несколько более понятными. Также обратим внимание читателя и на то, что формирование внутренней речи и ее последующее свертывание до уровня мысли происходит не сразу [133, с. 7–9] и некоторое время мышление осуществляется вслух, причем вовсе не в целях коммуникации [357, с. 139, 140]. Такая речь, однако, не тождественна мышлению, так как скрытыми остаются те ментальные образы, которые сопровождают произнесенные вслух слова.

Еще одним важным источником информации для живого существа являются интероцепторы («внутренние рецепторы»), сообщающие мозгу о состоянии организма и отдельных органов и позволяющие поддерживать гомеостаз. В работе [207, с. 236] приводится положение, согласно которому действие внутренних механизмов приспособления нами переживается как эмоции. В то же время, как мы отмечали в п. 3.3.1, в чисто практических целях для определения психофизического или эмоционального состояния человека используется анализ просодической структуры его речи. Видимо, интероцепторы дают семантическую опору для просодической структуры речи. Следовательно, для получения соответствующей семантической опоры в машинную систему необходимо ввести собственные эмоции. В таком аспекте проблема просодии в литературе еще не освещалась, равно как и проблема мотивации при формировании речевого высказывания (поскольку мотивация также может нарушаться при локальных повреждениях мозга, в результате которых остаются незатронутыми другие речевые функции, то следует полагать, что проблема мотивации должна исследоваться дополнительно к исследованию других компонентов речи).

Итак, для понимания языка машинной системой оказывается необходимым снабжение ее рецепторами (сенсорами), эффекторами, возможностью оценки качества собственного состояния и рядом других компонентов. Все эти компоненты находят свое отражение в языковых структурах. Более того, их включение принципиально необходимо и в систему ИИ. Действительно, даже программу, играющую в шахматы, можно представить как имеющую сенсоры, по которым подается информация о текущей игровой ситуации, эффекторы, с помощью которых совершаются ходы, и целевую функцию, оценивающую текущую ситуацию. Од-

нако, в отличие от системы CELL, шахматные программы не строят новые концепты, ограничены вложенными в них сведениями и фиксированной целевой функцией.

Обсуждение всех компонентов, необходимых для создания семантической базы, достаточной для понимания языка, выходит далеко за рамки данной книги. Вероятно, для создания системы, обладающей необходимой семантической базой, потребуется объединение усилий специалистов разных областей.

Однако требование наличия сенсорной, эффекторной и других «телесных» подсистем вовсе не говорит о том, что они должны реализовываться в стиле воплощенного подхода Брукса (о котором мы упоминали в п. 3.1.1). Не менее интересной, чем реализация зрительной, слуховой или тактильной модальности, является реализация искусственных модальностей, действующих в цифровых мирах. Примером такой модальности может служить «кодовая модальность», предназначенная для понимания и осмысленного порождения компьютерных программ (в конечном итоге это позволило бы системе анализировать и улучшать собственный код).

Добавление возможности формирования абстрактных концептов. Как было отмечено выше, система CELL способна связывать лингвистические единицы лишь с семантическими категориями, доступными непосредственному восприятию. То же самое относится к системе Describer и роботу Ripley. Это самый начальный этап формирования концептуальной системы. Добавление новых видов сенсоров или эффекторов не позволит получить семантические категории, которые бы соответствовали многим терминам естественного языка. Даже такому слову, как «мебель», не соответствует какая-то семантическая категория в смысле системы CELL. Тем более таким системам недоступен смысл фраз (за исключением простых референтных фраз), предложений или текстов. Это говорит о том, что сами семантические категории и лингвистические единицы должны быть помещены в некоторую концептуальную систему, служащую для представления знаний субъекта о мире. При этом должна быть возможность построения новых концептов с использованием информации об уже имеющихся, так что концепты окажутся связанными различными зависимостями.

Здесь как раз могут принести большую пользу результаты, полученные в области представления знаний с помощью

семантических сетей, фреймов и т. д. Их можно использовать как подсказку, какая именно система концептов должна строиться в процессе обучения. Однако вопрос вызывает не только представление знаний, но и процесс его автоматического построения. Формирование абстрактных концептов в чем-то должно быть похоже на формирование концептов в системе CELL. К примеру, формирование общих понятий, по своей сути, является распознаванием образов, выделением общего (взаимной информации) в частных понятиях. К сожалению, многие феномены этим не объясняются, например, возможность усвоения понятия, данного в определении в стиле толкового словаря.

Процесс образования удаленных от периферии концептов интересен и сам по себе, особенно в свете разнородности семантической базы. Иерархически связанные между собой элементарные действия дают концептуальный аппарат для планирования поведения. Периферийные цели, заложенные в человека природой, вплетаются (не без участия языка и общества) в общую концептуальную систему, формируя более абстрактные цели и систему жизненных ценностей. И так далее. Проследить эти процессы было бы, бесспорно, очень интересно, и не только для того, чтобы попытаться их смоделировать, но и чтобы лучше понять самих себя.

Использование принципа МДО при построении концептуальной системы. В системе CELL (равно как в Describer, Newt и Ripley) используется довольно много эвристически введенных элементов: порогов для принятия решения о формировании семантических категорий, лингвистических единиц, концептов, метрик в пространствах признаков и т. д. Как процесс интерпретации сенсорных сигналов, так и весь процесс образования концептов, на наш взгляд, можно было бы рассмотреть как индуктивный вывод с позиции принципа МДО, что позволило бы сделать анализ более строгим, а результаты более надежными.

Для отдельных сенсорных модальностей такая возможность была показана ранее. Индуктивный вывод может использоваться и для формирования моторных понятий, а также для связывания двигательных актов с классами сенсорных образов (что на нижнем уровне абстракции соответствует образованию условных рефлексов) или с лингвистическими единицами. К примеру, в работе [312] принцип МДО используется в задаче описания перемещения манипулятора робота.

Связывание семантических категорий и лингвистических единиц в системе CELL осуществляется на основе взаимной информации, что можно непосредственно трактовать с точки зрения длины описания.

Конечно, основной целью при этом вовсе не является собственно сжатие данных. Действительно, уже сейчас вполне доступны жесткие диски, позволяющие хранить несколько месяцев видео, сжатое на пиксельном уровне с незначительными потерями информации. Специализированные хранилища данных располагают ресурсами для хранения такого видео в течение сотен лет. Для хранения абстрагированной до структурных представлений видеоинформации они являются практически безграничными. Нет оснований считать, что объем человеческой памяти меньше, что подтверждают и исследования феномена эйдетической памяти [357, с. 83–85]. Но представим, что в памяти сохраняются необработанные данные. Тогда, чтобы извлечь из памяти нужную информацию, потребовалось бы заново «просматривать» все ее содержимое. Таким образом, проблема заключается не столько в объеме памяти, сколько в доступе к ней. Причем, как показывают нейропсихологические исследования различных нарушений памяти при повреждении мозга [364], проблема доступа к памяти является ключевой, и при повреждениях мозга страдают в первую очередь именно различные механизмы доступа к памяти, а не ее содержимое.

В процессе оптимизации длины описания выделяется наиболее значимая информация, по которой и осуществляется индексация содержимого памяти. Вопросы организации памяти в системах машинного обучения с инкрементным построением концептуальных систем на данный момент мало изучены, и мы вынуждены их опустить. Однако можно надеяться, что в будущем изучение этих вопросов поможет лучше понять и ряд феноменов, связанных с человеческой памятью, например, эффект утраты эйдетических способностей в процессе культурного развития человека, а также использование гипноза для доступа к информации, которую сознательно человек вспомнить не может. Все это связано с тем, что культурный человек использует осмысленные понятия для доступа к памяти, поэтому неосмысленная сенсорная информация оказывается недоступной через подобные понятия, хотя это и не означает, что она не сохраняется в памяти. К этому же типу относится тот факт, что

человек практически не помнит первые годы своей жизни. Вероятно, это также связано с тем, что в этот период жизни еще не были сформированы концепты, через которые человек в дальнейшем обращается к памяти.

С позиций принципа МДО можно также рассмотреть и вопрос о порядке выявления закономерностей в сенсорных данных.

Первыми обнаруживаются простейшие закономерности: повторение сегментов данных в одном канале или их совместное появление в нескольких каналах. В случае, если перебор идет от простейших или наиболее коротких (как априорно наиболее вероятных) моделей к более сложным, то первыми будут выделяться именно такие закономерности, как простое повторение сегментов данных. Для компенсации априорной неопределенности простых моделей требуется меньшее количество апостериорной информации, и обучение ведется быстрее. В этом смысле интуитивное речевое поведение взрослых по отношению к младенцам является близким к оптимальному (но, видимо, оно может быть целенаправленно улучшено).

Таким образом, поиск повторяющихся сегментов данных или их взаимных появлений может быть обоснован не только с точки зрения моделирования обучения младенцев, как это делалось в оригинальных работах по системе CELL, а с той точки зрения, что такие закономерности являются наиболее вероятными априори. Если бы они не находились, то в процессе поиска регулярностей следовало бы перейти к более сложным моделям.

Принцип МДО можно рассматривать в качестве фундаментальной основы для построения целевых функций, которые служат для управления как процессом интерпретации сенсорной информации в каждой модальности, так и процессом построения концептуальной системы.

Однако помимо целевой функции не менее важным является алгоритм ее оптимизации. На наш взгляд, и здесь могут быть найдены общие принципы, которые позволят находить приближенные решения сложных задач за малое (полиномиальное) время. Эти принципы могут быть прослежены наиболее явно в концепции *адаптивного резонанса* (АРТ, адаптивного резонанса теория; ART, adaptive resonance theory), к рассмотрению которой мы сейчас и перейдем.

3.5. ИЕРАРХИЧЕСКИЕ ПРЕДСТАВЛЕНИЯ, НЕПОЛНАЯ ДЕКОМПОЗИЦИЯ ЗАДАЧ И АДАПТИВНЫЙ РЕЗОНАНС

3.5.1. Введение иерархичности при решении NP-полных задач

Как отмечалось выше, интерпретация сенсорной информации осуществляется иерархически и для каждой сенсорной модальности может быть выделен собственный набор уровней. Чем вызвана такая иерархичность представлений? Конечно, можно было бы отнести ее на тот счет, что иерархичность присуща самому анализируемому сигналу как отражение иерархического строения источника этого сигнала. Но на этот вопрос можно взглянуть и с другой стороны. Можно утверждать, что иерархичность вводится информационной системой для субоптимального решения NP-полных задач за полиномиальное время (детерминированным образом). Продемонстрируем это утверждение на примере классической NP-полной задачи коммивояжера.

В этой задаче даны набор городов и матрица расстояний между ними. Необходимо составить наикратчайший маршрут, двигаясь по которому, можно посетить все города, побывав в каждом по одному разу. Неизвестны алгоритмы, точно решающие эту задачу быстрее, чем за экспоненциальное (в зависимости от числа городов) время. Обозначим через $f(n)$ число операций, необходимых для решения задачи коммивояжера для n городов некоторым алгоритмом.

Видно, что в постановке задачи никакой иерархичности исходно не предполагается. Введем ее. Разобьем все города на группы, скажем, по 10 городов. Группы городов, в свою очередь, объединим в сверхгруппы, каждая из которых содержит по 10 групп, и так далее. Пусть всего будет N городов. Решим задачу коммивояжера для каждой из $N/10$ групп, для чего потребуется $Nf(10)/10$ операций. В каждой группе будет некоторый начальный и конечный город. Передвижение между группами возможно только по этим «точкам входа» в группы, и можно определить расстояние между каждой группой. Таким образом, группа городов начинает выступать в качестве отдельного сверхгорода, и можно поставить задачу коммивояжера для этих сверхгородов. Для каждой сверхгруппы решим задачу коммивояжера. Для каждой сверхгруппы, содержащей 10 сверхгородов, по-

требуется $f(10)$ операций, а всего их $N/100$. И так далее. Всего нам потребуется для полного решения задачи $Nf(10) \times \times [1/10 + 1/100 + ...]$ операций, т. е. время решения задачи коммивояжера будет линейно по числу городов!

Естественно, такое решение будет приближенным. Также необходимо ответить на вопрос, какое разделение городов на группы будет являться предпочтительным для получения более точного решения. Очевидно, оно должно как можно лучше описывать присущую распределению городов иерархичность (например, города могут располагаться на разных континентах). Мы, однако, не ставим целью предъявить метод решения задачи коммивояжера, поэтому данный вопрос опустим.

Описанный алгоритм может дать и очень плохое решение или даже привести к невозможности решения задачи. В рамках примера с городами, расположенными на разных континентах, невозможность решения возникнет, если путь между городами одного континента будет заканчиваться в городе, в котором не будет аэропорта или порта, чтобы попасть на другой континент. Очевидно, решать задачу в отдельности для разных групп не вполне корректно. Это было бы корректно, если бы группы были полностью разделены, тогда задача коммивояжера могла бы быть факторизована. Здесь же подзадачи остаются связанными, поэтому при объединении решений отдельных подзадач может потребоваться эти решения скорректировать. Возможный механизм коррекции мы описывать не будем, а просто отметим, что коррекция решений нижнего уровня в процессе решения задачи на следующем уровне оказывается необходимой для избежания грубых ошибок и для общего улучшения результата.

Как мы видим, отсутствие иерархичности в исходных данных вовсе не мешает эту иерархичность вводить. Конечно, чем более сильные (контекстные) связи остаются между группами, тем грубее будет результат такого подхода, но обычно это приемлемая плата за уход от экспоненциальной сложности вычислений.

Итак, использование иерархических представлений данных может рассматриваться как способ декомпозиции NP-полной задачи на подзадачи, но поскольку подзадачи не являются независимыми, то их раздельное решение дает грубое приближение, которое следует улучшать. Зная нулевое приближение к решению каждой из подзадач, можно

эти решения скорректировать. Прием коррекции хорошо известен в задачах интерпретации сенсорной информации, и называется он адаптивным резонансом.

3.5.2. Понятие адаптивного резонанса

В п. 3.1.1 был приведен пример проявления эффекта перцептивной готовности (см. рис. 3.1). Один и тот же символ в зависимости от контекста может интерпретироваться совершенно по-разному, причем у человека, как правило, и мысли не возникает, что этот символ может иметь другую интерпретацию. Аналогичный эффект имеет место и для слуха. В эксперименте, описанном в работе [365], человеку предъявляется на слух фраза, в которой первая буква первого слова зашумлена, так что это может быть любое слово из нескольких вариантов. В зависимости от продолжения фразы это первое слово воспринимается по-разному, несмотря на то что во всех случаях оно может быть одним и тем же, причем часто человек не осознает осуществляемого им выбора. «?eel is on the...» — так звучит эта фраза. В зависимости от того, каким из слов: «orange» (апельсин), «wagon» (вагон), «shoe» (туфля) — она заканчивается, первое слово воспринимается (носителями языка) как «peel» (кожура), «wheel» (колесо) и «heel» (каблук) соответственно. Несложно придумать аналогичные примеры и для других языков.

Можно было бы предположить следующий механизм. Пусть с предыдущего уровня на следующий уровень подается несколько гипотез. В рамках предыдущего примера это все возможные слова, заканчивающиеся на «eel». На следующем уровне происходит построение более высокоуровневой модели (например, описывается вся фраза целиком), в рамках которой и делается выбор между разными гипотезами предыдущего уровня. При использовании такого механизма поток информации будет однонаправленным — снизу вверх (именно такой подход мы применили в п. 3.3.5 для сегментации речи на слова). Использование такого механизма хотя и возможно, но наталкивается на некоторые трудности. Снизу вверх может подаваться лишь ограниченное число гипотез (если передавать все гипотезы, то мы вернемся к полному перебору, и в разделении на уровни не будет смысла, что хорошо видно на задаче коммивояжера),

причем среди этих гипотез обязательно должна присутствовать истинная. Человек же для объяснения элемента данных на нижнем уровне, используя контекст, может выбрать ту гипотезу, которая вне контекста была бы гораздо менее вероятна, чем некоторая другая гипотеза. Таким образом, в гипотезы, подаваемые на следующий уровень, может потребоваться включить и маловероятные гипотезы. Как же тогда ограничить число гипотез? Это можно сделать, только используя контекстную информацию.

Гроссберг предлагает не ограничиваться рассмотрением прямых связей между уровнями и ввести обратные связи. Тогда гипотезы, рассматриваемые на верхних уровнях, будут влиять на работу нижних уровней. Это влияние может проявляться как в указании на необходимость генерации новой гипотезы, так и в поддержании или подавлении уже сгенерированных гипотез. Например, если человек слышит «?орт», то на уровне фонем для первой фонемы существует несколько десятков гипотез, так что из них может быть выбрано несколько лучших, которые будут поданы на уровень слов. На этом уровне может быть предложено несколько гипотез слов, например, «торт», «порт», «борт»... При генерации этих гипотез будет осуществлено обращение на уровень ниже в целях добавления букв «т», «п», «б» в качестве гипотез нижнего уровня, если они еще не были рассмотрены, и установления степени их соответствия имеющимся данным. Если гипотезы, рассматриваемые на разных уровнях, поддерживают друг друга, то они «вступают в резонанс», подавляя другие гипотезы.

Идея адаптивного резонанса позволила объяснить Гроссбергу некоторые особенности человеческого восприятия, например, задержку в осознании сенсорной информации по сравнению со временем, требуемым для прохождения сигнала по зрительному или слуховому тракту. Это время, необходимое для установления резонанса.

Прояснение вопросов человеческого восприятия с помощью концепции адаптивного резонанса важно и интересно (например, интересен вопрос соотнесения силы априорного ожидания с апостериорной информацией: в крайнем случае, когда крайне сильно возбуждение, распространяемое с верхних уровней вниз, человек видит или слышит лишь то, что хочет, или то, к чему готов). Мы, однако, ограничимся здесь проблемами использования концепции адаптивного резонанса в системах машинного восприятия.

Общая идея адаптивного резонанса не говорит о том, как именно нужно организовывать связь между уровнями, в частности, как должна осуществляться поддержка гипотезами одного уровня гипотез другого уровня. Самим Гроссбергом эта проблема решается в рамках теории адаптивного резонанса, в которой рассматривается искусственная нейронная сеть определенного вида. Эта теория легла в основу целой серии работ, посвященных исследованию адаптивного резонанса, но, к сожалению, эти исследования не выходят за границы нейросетевой парадигмы. Такое рассмотрение в значительной мере сужает общую идею адаптивного резонанса. Попробуем взглянуть на проблему с другой стороны. Нас будет преимущественно интересовать вопрос сравнения гипотез в рамках иерархических представлений. В связи с этим рассмотрим адаптивный резонанс в рамках задачи индуктивного вывода с позиций принципа МДО и покажем, что адаптивный резонанс может управляться информационной целевой функцией.

3.5.3. Теоретико-информационная интерпретация адаптивного резонанса

Пусть исходными данными для индуктивного вывода служат данные D. Рассмотрим двухуровневое пространство гипотез $\mathrm{H}_1 \times \mathrm{H}_2$: данные D сначала объясняются некоторой гипотезой $h_1 \in \mathrm{H}_1$, которая, в свою очередь, объясняется некоторой гипотезой следующего уровня $h_2 \in \mathrm{H}_2$. Например, гипотеза h_1 может быть гипотезой о контурах на изображении, а h_2 — гипотезой о структурных элементах, построенных на основе этих контуров. Целевая функция, описывающая качество двухуровневого представления в целом, будет: $L_{12} = L(D \mid h_1) + L(h_1 \mid h_2) + L(h_2)$. Перебор всех возможных пар $(h_1,\ h_2) \in \mathrm{H}_1 \times \mathrm{H}_2$ позволит найти абсолютный минимум этой целевой функции, но это полный перебор, который нас не устраивает.

Вместо этого можно воспользоваться оценкой $L_1 = L(D|h_1) + L(h_1)$, где $L(h_1) > L(h_1|h_2) + L(h_2)$, и, минимизируя ее, найти некоторую лучшую гипотезу h_1^*. Далее, минимизируя $L_{12} = L(D \mid h_1^*) + L(h_1^* \mid h_2) + L(h_2)$ по $h_2 \in \mathrm{H}_2$, можно найти лучшую гипотезу h_2^*. В результате нам потребуется перебирать не $\|\mathrm{H}_1\| \cdot \|\mathrm{H}_2\|$, а $\|\mathrm{H}_1\| + \|\mathrm{H}_2\|$ гипотез, но пара

(h_1^*, h_2^*), вероятно, не будет давать абсолютный минимум функции L_{12}. Возможно, будет существовать пара (h_1', h_2'), такая, что $L(h_1^* \mid h_2^*) + L(h_2^*) >> L(h_1' \mid h_2') + L(h_2')$, и соотношение $L(D \mid h_1^*) + L(h_1^*) < L(D \mid h_1') + L(h_1')$ поменяет знак после подстановки точного значения $L(h_1|h_2) + L(h_2)$ вместо оценки $L(h_1)$. Чтобы такое случалось как можно реже, необходимо выполнение условия $\left|L(D \mid h_1') - L(D \mid h_1'')\right| >> \left|L(h_1') - L(h_1'')\right|$ для как можно большего числа пар гипотез h_1', h_1''. Если это условие выполняется, то выбор между гипотезами h_1' и h_1'' не зависит от выбора гипотезы следующего уровня. Заметим, что при анализе сенсорной информации это условие, как правило, выполняется. Например, с контурного уровня, как и с уровня фонем, наверх подается очень мало информации по сравнению с объемом исходных данных, но выбор гипотезы первого уровня не является однозначным.

Пусть на нижнем уровне был сформирован набор лучших гипотез $h_{1,i}^*$, а на верхнем уровне для них были найдены гипотезы $h_{2,i}^*$, минимизирующие L_{12} при фиксированных $h_{1,i}^*$. Если бы не было обратных связей, то на этом стоило бы остановиться и передать набор $h_{2,i}^*$ на следующий уровень. При наличии обратных связей для каждой гипотезы $h_{2,i}^*$ можно произвести минимизацию величины $L_{12} = L(D \mid h_1) + L(h_1 \mid h_2^*) + L(h_2^*)$ по $h_1 \in \mathrm{H}_1$ и получить новую гипотезу $h_{1,i}^{**}$, которая, вполне вероятно, будет отличаться от предыдущей гипотезы $h_{1,i}^*$. Вместе с тем улучшится и качество гипотезы $h_{2,i}^*$, так что она может стать предпочтительнее некоторой другой гипотезы $h_{2,i}^*$, т. е. ей будет оказана поддержка с нижнего уровня. Поиск в пространстве H_1 может вестись вблизи гипотезы $h_{1,i}^*$, тогда гипотеза $h_{1,i}^{**}$ будет являться результатом ее коррекции. Этот поиск может также вестись вблизи некоторого прототипа гипотезы $h_{2,i}^*$, например, если это гипотеза — структурный элемент, то его прототип на нижнем уровне — соответствующий идеальный контур. После коррекции гипотез нижнего уровня можно опять вернуться на верхний уровень и уточнить гипотезы $h_{2,i}^*$.

Такой итеративный процесс будет сходиться, так как длина описания будет монотонно убывать с номером итерации и будет ограничена снизу нулем. Это аналог адаптивного резонанса: в паре низкоуровневая и высокоуровневая гипотезы корректируют друг друга, улучшая совокупное качество описания данных. Другие пары также подвержены такому процессу, но могут в результате дать более

низкое качество. Заметим, что такое взаимное улучшение похоже на алгоритм ожидания-максимизации (см. п. 2.4.4).

Описанная общая схема требует в каждом конкретном случае уточнения механизма генерации новых и коррекции уже имеющихся гипотез, поскольку полный перебор даже на каждом уровне в отдельности вычислительно неэффективен. Рассмотрим возможные реализации этой схемы в разных задачах интерпретации сенсорной информации, на примере которых и поясним ее функционирование.

3.5.4. Адаптивный резонанс при интерпретации изображений

При обсуждении проблемы интерпретации изображений были рассмотрены переходы между следующими уровнями: пиксельным уровнем, контурным уровнем, уровнем непроизводных структурных элементов, уровнем составных структурных элементов или их групп. Гипотезы первого уровня строятся при переходе между пиксельным и контурным уровнями, гипотезы второго уровня — при переходе между контурным и структурным уровнями, третьего уровня — при переходе между уровнем структурных элементов и их групп. Таким образом, адаптивный резонанс может проявляться в коррекции контуров при построении структурных элементов и в коррекции структурных элементов в процессе их группирования. Если бы рассматривали восстановление трехмерной геометрии сцены или распознавание объектов на ней, то добавились бы дополнительные уровни, от которых вниз также могут идти обратные связи.

Рассмотрим механизм коррекции контуров при построении структурных элементов. В п. 3.2.3 был описан алгоритм сегментации изображения, в результате которого формировалось контурное описание изображения. Этот алгоритм управлялся целевой функцией (3.11), которую мы здесь представим как

$$L(image) = L(D \mid h_1) + L(h_1) = L(image \mid contours) + L(contours). \tag{3.30}$$

Заметим, что длина описания $L(contours)$ вычислялась через длину контуров

$$L(contours) = \sum_k \left(\| \delta G_k \| \log_2 N_{dir} \right), \tag{3.31}$$

т. е. контуры на этом уровне представлялись в несжатом виде, и эта оценка является верхней оценкой длины описания контуров.

Далее единственная лучшая (найденная) гипотеза о контурах, присутствующих на изображении, передавалась алгоритму построения структурных элементов, который минимизирует целевую функцию (3.13):

$$L(contours) = L(h_1 \mid h_2) + L(h_2) = \\ = L(contours \mid primitives) + L(primitives). \qquad (3.32)$$

Построение единственного набора контуров, который при проведении дальнейшего анализа остается неизменным, используется в подавляющем большинстве систем компьютерного зрения. Как мы теперь видим, это ведет к невозможности нахождения глобального оптимума целевой функции, описывающей качество функционирования системы в целом. Построение структурных элементов на основе этих контуров ведет к дальнейшему отклонению от оптимума, а вовсе не к разрешению неопределенности, возникшей на контурном уровне. В результате выполнения нескольких таких переходов надежность системы может катастрофически упасть. Заметим, что при построении контуров у человека может использоваться информация, полученная с верхних (вплоть до сознания) уровней, в частности, экспериментально установлено, что стимуляция определенной части коры головного мозга вызывает изменение в рецептивных полях ганглиозных клеток сетчатки (выполняющих аналог дифференцирования) [207, с. 107].

Рассмотрим возможный механизм коррекции контуров при построении структурных элементов. В нашем случае контур является границей между двумя областями на изображении. При коррекции контура происходит перенос точек из одной области в другую, соответственно меняется энтропия интенсивностей пикселей внутри этих областей. При этом длина описания $L(image|contours)$ увеличивается, а $L(contours|primitives)$ — уменьшается. Последнее происходит, если при перемещении пикселей из одной области в другую граница приближается к модели контура, задаваемой структурным элементом.

Попытаемся теперь перенести каждую граничную точку из текущей области в соседнюю область. Если это вызывает уменьшение общей длины описания

$$L(image) = L(image \mid contours) +$$
$$+ L(contours \mid primitives) + L(primitives), \qquad (3.33)$$

то такой перенос следует выполнить. При попытке переноса каких-либо точек следует также пересчитать и параметры соответствующего структурного элемента, т. е. в результате этого процесса изменятся и сами элементы. Процедуру коррекции следует итеративно повторять, пока она приводит к уменьшению общей длины описания.

После этой процедуры контуры вовсе не будут точно повторять структурные элементы, а лишь приблизятся к ним в тех местах, где это адекватно содержимому изображения. Сами структурные элементы также окажутся несколько иными, чем при их построении на основе первоначальных контуров (рис. 3.36, *а–г*).

Более важным, однако, является использование коррекции контуров в процессе построения структурных элементов. К примеру, если возникает сомнение, следует ли объединять два сегмента контура в один, необходимо для каждой из двух гипотез произвести коррекцию контуров и выбрать тот результат, который даст меньшую длину описания, просуммированную на двух уровнях абстракции. Такое адаптивное обращение к нижнему уровню в процессе построения структурных элементов, а не после него, позволяет разрешить возникающие на этом уровне неопределенности и заметно улучшить структурное описание (рис. 3.37).

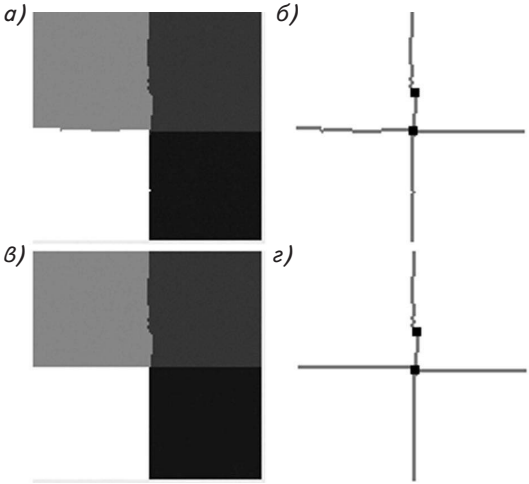

Рис. 3.36. Коррекция результатов выделения границ областей на изображении, приведенном на рис. 3.14, *а*, в результате сегментации контуров: *а*, *б* — исходный результат сегментации изображения и границ областей; *в*, *г* — результаты сегментации после коррекции. Попиксельное смещение контуров, уменьшающее длину описания на двух уровнях, приводит к коррекции положения контуров и параметров структурных элементов, описывающих эти контуры

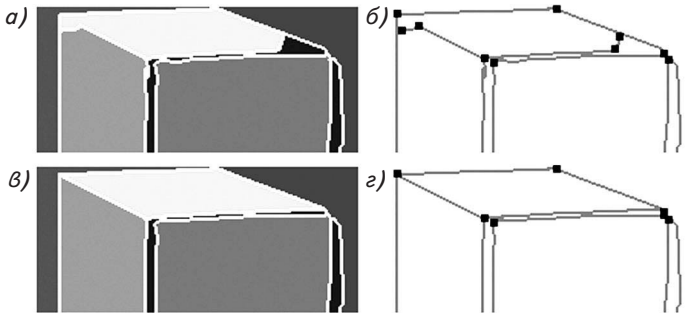

Рис. 3.37. Пример адаптивной коррекции контуров в процессе построения структурных элементов (использован фрагмент изображения, приведенный на рис. 3.17): *а*, *б* — исходный результат сегментации изображения и границ областей; *в*, *г* — результаты сегментации с использованием коррекции. При принятии решения о слиянии двух сегментов контуров производится коррекция положения точек контуров до и после слияния, что позволяет точнее оценить приоритетную гипотезу

Возможности повышения качества строящихся непроизводных структурных элементов путем введения обратных связей этим не ограничиваются. Да и сами описанные выше алгоритмы коррекции требуют уточнения. Мы также не рассмотрели возможность передачи с контурного уровня на уровень структурных элементов нескольких альтернативных гипотез сегментации изображения. Все эти вопросы требуют дополнительных исследований. Однако представляется, что суть подхода проиллюстрирована в достаточной степени.

Похожий прием введения обратных связей может использоваться и на следующей паре уровней: уровне непроизводных элементов и уровне их групп. Неявно он уже использовался нами в алгоритме построения составных элементов (см. п. 3.2.5). Действительно, решение о построении составного структурного элемента принималось не только на основе параметров непроизводных элементов, но и на основе контурной информации. При объединении двух отрезков прямых в элемент «параллельные прямые» происходила коррекция самих структурных элементов (т. е. формировалась новая гипотеза на уровне непроизводных элементов на основе информации, поступившей со следующего уровня). При этом вычислялась заново энтропия невязок, с которыми скорректированные элементы описывают кон-

туры. В случае одновременного функционирования всех уровней следовало бы также для этой новой гипотезы осуществить и коррекцию контуров, что дало бы еще более точный результат, но эта возможность не проверялась.

Другая непроверенная возможность связана с поиском пропущенных элементов в результате анализа регулярностей в расположении выделенных элементов. Действительно, если при объединении некоторых элементов в группу обнаруживается, что описание этой группы было бы компактнее, если бы был добавлен еще один элемент, который отсутствует, то можно было бы этот элемент ввести в предположение, что рядом с ним проходит необнаруженный контур. Далее следовало бы проверить гипотезу о наличии этого контура на нижнем уровне и, если бы она нашла подкрепление, модифицировать результат сегментации изображения. Информационный критерий, связывающий все уровни работы системы, позволяет корректно определять величину подкрепления, оказываемого гипотезе на разных уровнях, а не вводить это подкрепление эвристически.

Все те же механизмы введения обратных связей могут быть сформированы вплоть до семантического уровня описания сцены. Если бы это удалось сделать, то можно было бы построить очень надежную систему распознавания объектов по их изображениям. Такая возможность продемонстрирована в книге [366] для частной задачи — распознавания рукописного текста: описывается система, использующая мягкое вероятностное распознавание (с предыдущих уровней системы на следующие подается не одна, а несколько гипотез) и включающая обратные связи между уровнями. Верхние уровни системы являются модально-неспецифичными и мало чем отличаются от верхних уровней систем распознавания речи.

3.5.5. Адаптивный резонанс в анализе речи

Благодаря одномерности звукового сигнала и меньшей априорной неопределенности, присущей речи, введение обратных и горизонтальных связей в многоуровневых системах распознавания речи заметно легче, чем в системах машинного зрения, и часто эти связи явно или неявно присутствуют.

В п. 3.3 мы выделили следующие уровни анализа речевого сигнала: уровень акустического сигнала, различитель-

ных признаков, фонем, морфов, словоформ, словосочетаний и фраз. На каждой паре переходов между тремя соседними уровнями может быть реализован механизм адаптивного резонанса.

Рассмотрим первые три уровня. Различительные признаки (в п. 3.3.3 мы рассматривали в качестве таковых коэффициенты вейвлетов) выявляются путем решения регрессионных задач, в которых не учитывается связь между самими признаками. Можно считать, что минимизируется целевая функция: $L(signal \mid features) + L(features)$, где слагаемое $L(features)$ оценивается грубо, по числу параметров регрессионных моделей. Естественно, извлеченные признаки могут быть определены неточно в силу зашумленности акустического сигнала. Далее, при распознавании фонем, для каждого конкретного набора признаков выдвигаются различные гипотезы. В классическом распознавании образов признаки считаются фиксированными и минимизируется целевая функция $L(features \mid phomenes) + L(phomenes)$.

Для организации адаптивного резонанса при рассмотрении каждой фонемы в качестве гипотезы можно обращаться на уровень ниже и корректировать различительные признаки, так чтобы они лучше соответствовали данной фонеме (при этом уменьшается $L(features \mid phomenes)$) за счет незначительного ухудшения длины $L(signal \mid features)$. Если фонема представляется моделью гауссовой смеси в пространстве признаков, то попытка коррекции различительных признаков заключается в смещении вектора признаков по направлению к эталонному образу одной из компонент смеси (как правило, описывающей аллофон). Изменение значений различительных признаков означает изменение параметров вейвлетов, а потому изменяется энтропия невязок, с которыми данный набор вейвлетов описывает сигнал. К сожалению, такая простая схема в реализации оказывается существенно сложнее. Тем не менее любая ее реализация, приводящая к уменьшению длины описания, позволит повысить надежность распознавания фонем и надежность системы распознавания речи в целом. Как правило, в существующих системах (за исключением реализаций, использующих нейронную сеть типа АРТ) это не делается вовсе. Однако ситуация несколько меняется на следующем уровне.

На уровне фонем при их распознавании вовсе не используется оценка величины $L(features)$ через частотность оди-

ночных фонем. Вместо этого используются частоты N-грамм: вероятность появления каждой последующей фонемы вычисляется с учетом нескольких уже распознанных перед ней фонем. Иными словами, привлекаются горизонтальные связи на этом уровне без обращения к следующему уровню — уровню морфов. Благодаря малому числу фонем такие горизонтальные связи вводятся весьма просто. Вероятно, их введение было бы полезно и при интерпретации изображений, однако там из-за большого разнообразия данных ввести такие связи сравнительно несложно только при сильных ограничениях (например, в задачах распознавания текста). Как это сделать в общем случае, пока остается неясным.

На уровне морфов и словоформ также используются горизонтальные связи через привлечение моделей N-грамм. Однако обратные связи вводятся весьма редко. В частности, в п. 3.3.4 мы рассмотрели способ распознавания слов по цепочкам символов (фонем или букв), в котором возможные ошибки в символах описывались на уровне самих символов (некоторая замена символов $a \to b$ кодировалась в соответствии с вероятностью этой замены). При этом не делалось различий между ошибкой подсистемы распознавания фонем и реальной ошибкой диктора. Последняя действительно должна описываться через подстановку фонем, но первая может быть исправлена, если воспользоваться обратными связями между уровнями. Ошибка системы распознавания, вызванная шумами, связана с реальной неопределенностью в выборе фонемы, так что изменение фонемы не должно привести к сильному ухудшению качества описания различительных признаков, но избавит от необходимости описывать подстановку фонем. Такое исправление, в свою очередь, улучшит качество слова, в рамках которого такое исправление фонем имело место, и оно будет выбрано с большей вероятностью среди прочих гипотез. Естественно, в этом процессе также может использоваться информационная целевая функция, объединяющая соответствующие уровни.

На следующем уровне — уровне цепочек слов — мы не касались моделей, более содержательных, чем N-граммы, так что организацию адаптивного резонанса с этим уровнем мы вынуждены опустить. Однако напомним, что использование и простых горизонтальных связей с помощью моделей N-грамм позволяет заметно повысить надежность распознавания

слов, особенно при необходимости выполнения сегментации речи на слова (см. п. 3.3.5).

Итак, в системах распознавания речи организация обратных связей в целях минимизации общей целевой функции также возможна. Кроме того, здесь существует широко используемая возможность организации горизонтальных связей, которые позволяют на каждом уровне получать более корректные оценки общей целевой функции, не обращаясь к следующему уровню. Благодаря тому что, начиная с уровня фонем, пространство гипотез является перечислимым, здесь возможно с каждого предыдущего уровня на следующий подавать не одну локально лучшую гипотезу, а большее их количество, что уменьшает вероятность пропустить глобально лучшую гипотезу. Такие возможности делают распознавание речи привлекательным с точки зрения исследования механизмов взаимодействия между уровнями. Далее эти механизмы должны быть перенесены и на другие системы анализа информации.

3.5.6. Использование обратных связей при совместной интерпретации аудио- и видеоинформации

Выше мы обсудили вопросы введения обратных связей при интерпретации изображении и распознавании речи. Была отмечена возможность введения обратных связей с верхних уровней систем интерпретации соответствующих модальностей. Более того, как показывают нейрофизиологические данные, «все входные системы организма непосредственно контролируются центральной нервной системой» [207, с. 104], причем контроль осуществляется над рецепторами в двигательной системе, кожными рецепторами, слуховым, обонятельным и зрительным восприятием [207, с. 106]. Напомним, что стимуляция соответствующей части коры головного мозга вызывает изменение в рецептивных полях ганглиозных клеток. Через центральную нервную систему осуществляется и организация влияния одной модальности на другую. Например, экспериментально проверено, что незрительные стимулы, такие как звуковые щелчки, вызывают отклик в зрительном нерве у кошек [207, с. 106]. Другой интересный опыт провел Мертон (см. [207, с. 107] и там же ссылку на оригинальную работу). Он парализовал мышцы

собственного глаза, а затем попытался двигать глазами. Оказалось, что видимое изображение мира «прыгало» в том же направлении, куда он (сознательно) пытался переместить свой взгляд. В частности, это означает, что информация, полученная от одной сенсорной модальности, может использоваться для улучшения результатов интерпретации информации в рамках другой сенсорной (или эффекторной) модальности. Существуют вычислительные модели, полностью подтверждающие это положение.

Опишем кратко уже упоминавшуюся в п. 3.4.5 систему Fuse [290], в которой осуществляется влияние сенсорных подсистем друг на друга через их верхние уровни, хотя в этой системе и не используется информационный подход.

В системе Fuse ставится следующая задача: по словесному описанию найти объект на сцене, который наилучшим образом соответствует смыслу описания. Эта задача аналогична задаче, решавшейся в системе Newt (см. п. 3.4.5). Сами системы тоже во многом похожи, в частности, в них используются практически одинаковые представления сенсорной информации. Но, в отличие от системы Newt, в системе Fuse основное внимание уделяется совместному использованию аудио- и видеоинформации на ранних (не заключительных) этапах их интерпретации.

В результате зрительного анализа система строит представление сцены, которое используется для того, чтобы предсказать возможные цепочки слов (гипотезы верхнего уровня в подсистеме понимания речи), которыми человек может описать тот или иной объект на сцене. В свою очередь, результаты работы подсистемы понимания речи влияют на процесс зрительного анализа: по мере распознавания слов в цепочке происходит динамическая фокусировка зрительного внимания на возможные реферируемые во фразе объекты. Управление вниманием и предсказание вероятных фраз взаимно улучшают друг друга, так что процесс с высокой вероятностью сходится к согласованным интерпретациям сигнала в каждой модальности. Несложно увидеть аналогию между таким итеративным взаимным улучшением результатов интерпретации и механизмом адаптивного резонанса между гипотезами в различных подсистемах.

Эта работа противопоставляется наивной форме модульного подхода (часто используемого при разработке программного обеспечения, проектировании роботов или интеллектуальных систем), в которой каждый модуль представ-

ляется в виде черного ящика, доступ к внутренней структуре которого закрыт из других модулей. Выше мы уже показали неправомерность жесткого разбиения на модули при интерпретации сенсорной информации в рамках одной модальности. В этой работе на конкретных практических примерах демонстрируется также ошибочность полностью не зависимой разработки подсистем зрения, слуха, управления движением и т. д.

Рассмотрим чуть подробнее функционирование системы Fuse, отталкиваясь от верхнего уровня подсистемы понимания речи. Пусть x — исходный акустический сигнал; α — цепочка слов (гипотеза верхнего уровня). Тогда выбор наилучшей гипотезы осуществляется следующим образом:

$$\alpha^* = \arg\max_{\alpha} \left(P(x \mid \alpha) P(\alpha) \right), \qquad (3.34)$$

где $P(x \mid \alpha)$ — правдоподобие сигнала в рамках данной гипотезы (в действительности это правдоподобие вычисляется через набор промежуточных уровней — см. п. 3.3); $P(\alpha)$ — априорная (безусловная) вероятность встретить данную цепочку слов (в данной предметной области). При классическом способе построения системы распознавания речи вероятности $P(\alpha)$ задаются статической моделью языка, определяемой, например, с помощью вероятностей N-грамм слов (см. п. 3.3.5), оцененных на этапе предварительного обучения по текстовому корпусу.

В работе [290] предлагается использовать динамическую модель языка, в которой вероятности $P(\alpha)$ определяются исходя из зрительного контекста. Как только построено описание сцены, по нему можно оценить вероятность встретить то или иное описание каждого объекта на сцене. Выдвигать гипотезы о каждом возможном описании каждого возможного объекта на сцене проблематично, поэтому можно использовать более общие ограничения, накладываемые на модель языка зрительным контекстом. Динамическая модель языка может содержать вероятности не только N-грамм слов, но и отдельных слов. Эти вероятности и можно оценить, используя распознанные объекты сцены. Например, если система «видит», что на сцене нет ни одного красного объекта, то вероятность встретить слово «красный» для нее будет существенно занижена.

Здесь уместно вспомнить интересный эксперимент на эффект перцептивной готовности, в котором испытуемому

предъявляется изображение слов, обозначающих цвет («желтый», «зеленый» и т. д.), причем сами слова написаны каким-либо другим цветом. Далее человеку предлагается назвать цвет, которым написано то или иное слово. Интересен процент ошибок при назывании цвета, а также задержка, возникающая при попытке назвать соответствующий цвет. Читатель может самостоятельно поставить такой эксперимент.

Итак, после получения начального описания сцены могут быть модифицированы априорные вероятности слов, что ведет к улучшению результатов сегментации цепочек фонем на слова и результатов распознавания слов. В результате распознавания слов фразы выдвигаются гипотезы о том, какие объекты на сцене могут быть описаны этой фразой. Происходит фокусировка зрительного внимания на эти объекты, которая заключается в том, что на следующей итерации именно эти объекты используются для динамической генерации модели языка. Уточненная модель языка может быть далее использована для более надежного анализа речи. На второй итерации уже есть возможность не только оценить вероятности отдельных слов, но и их N-грамм, что приводит к улучшению результатов распознавания слов во фразе. Новая гипотеза о содержании фразы может использоваться для более четкой фокусировки внимания. И так далее, пока в фокусе внимания не останется лишь один объект, задающий модель языка, в которую укладывается хорошо распознанная фраза.

Для краткости изложения мы описали общую схему работы системы лишь для одной гипотезы о содержании фразы. В действительности может одновременно рассматриваться несколько альтернативных гипотез, среди которых будет выбрана одна лучшая, вступающая в наибольший резонанс со зрительным контекстом.

Результаты, полученные в описанной работе на основе стохастических гипотез, несложно перенести на информационный подход. В системе Fuse обратные связи ограничены лишь несколькими верхними уровнями в подсистеме распознавания речи (они не доходят до уровня распознавания фонем) и практически полностью отсутствуют в зрительной подсистеме (описание сцены остается неизменным, в частности, система не может обнаружить «не замеченный» ею объект, о наличии которого сообщается во фразе; осуществляется лишь фокусировка внимания на обнару-

женные объекты). Интересной задачей, решение которой позволило бы повысить надежность системы, было бы добавление в систему Fuse недостающих обратных связей (о способах их введения говорилось выше).

Несмотря на ряд упрощений, система Fuse ставит важную общую проблему использования контекста при анализе новой информации. Хотя в ней и рассматривается лишь вопрос привлечения текущего зрительного контекста для улучшения качества распознавания речи, но подразумевается возможность использования контекста в самом широком смысле. Такой общий контекст создается не только текущей информацией, поступающей от органов чувств, но и ранее полученной информацией, и он очень широко используется человеком для надежной и быстрой интерпретации новых данных, для преодоления неопределенности, содержащейся в этих данных. К сожалению, в большинстве современных систем машинного обучения проблема контекста не изучается, поэтому на данный момент отсутствует достаточный эмпирический материал для освещения этой проблемы.

3.5.7. Концепция метасистемных переходов

Выше было показано, что иерархичность — это универсальное средство, вводящееся при решении сложных задач. Использование иерархического подхода не ограничивается индуктивным выводом в целях интерпретации сенсорной информации. При решении любой сложной задачи человек всегда пытается разбить ее на подзадачи. Зачастую решение одних подзадач зависит от решения других. Раздельное их решение ведет к неточностям. Чем сложнее исходная задача, чем больше уровней в ее разбиении и чем сильнее остаются связи у подзадач, тем грубее будет решение, полученное путем непосредственного объединения решений подзадач. Для задач реального мира данное упрощение оказывается неприемлемым, и его необходимо преодолеть, что осуществляется с помощью механизма адаптивного резонанса, позволяющего учесть остаточные связи между задачами. Этот механизм оказывается практически столь же универсальным приемом, как введение самой иерархичности. Для реализации всего потенциала механизма адаптивного резонанса необходима его проблемно-независимая форма-

лизация, которая может быть осуществлена на базе информационного подхода.

Совмещение двух концепций: адаптивного резонанса и принципа минимальной длины описания — может превратить их в очень мощный инструмент. Проблема «идеального» индуктивного вывода, с которой мы столкнулись в гл. 1, заключалась в том, что поиск наилучшей модели — это NP-полная задача, время решения которой экспоненциально зависит от сложности модели. Можно ли, используя иерархический подход и механизм адаптивного резонанса, разработать алгоритм поиска субоптимальной по длине описания программы для машины Тьюринга, время работы которого будет полиномиальным? К сожалению, есть один принципиальный момент, который мешает дать однозначный ответ на этот вопрос. Дело в том, что при решении NP-полных задач (коммивояжера или интерпретации сенсорной информации) уровни иерархии задавались вручную и зависели от задачи. Иерархическая декомпозиция произвольной задачи на подзадачи сама по себе является NP-полной задачей, ничуть не более простой, чем исходная задача. Иными словами, не следует надеяться создать простую, универсальную систему решения задач. Тем не менее концепция иерархичности и адаптивного резонанса является весьма продуктивной, но она сама оказывается лишь одним «уровнем» в проблеме моделирования. Следующий шаг — определить механизмы формирования иерархии, которые бы стали следующим уровнем.

В ходе эволюции строились очень сложные иерархические системы, решающие разнообразные задачи (в том числе и анализа сенсорной информации различных модальностей и в различных средах). Это дает материал для анализа в целях выделения принципов построения следующего уровня иерархии. Такой анализ был осуществлен В. Ф. Турчиным в книге «Феномен науки: кибернетический подход к эволюции» [367], в которой он развивает концепцию метасистемных переходов как механизма образования следующего уровня иерархии. Метасистемой называется система, состоящая из подсистемы X, управляющей многими однородными подсистемами A_1, A_2,..., A_n. Метасистемный переход от системы A к системе X заключается в многократном дублировании системы A и образовании метауровня X (рис. 3.38). При этом подсистемы A_i не являются тождественными, а специализируются путем отбора. В книге при-

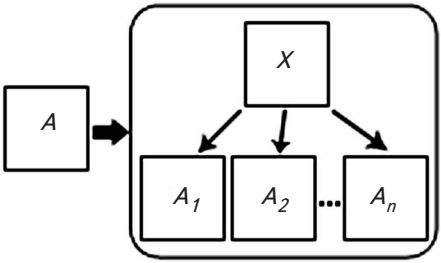

Рис. 3.38. Общая схема метасистемного перехода по Турчину

водится следующая цепочка наиболее значимых метасистемных переходов в процессе эволюции:

- управление положением = движение;
- управление движением = раздражимость;
- управление раздражимостью = сложный рефлекс;
- управление сложными рефлексами = ассоциирование;
- управление ассоциированием = мышление;
- управление мышлением = культура.

Более подробно механизмы метасистемного перехода рассмотрены в гл. 3 книги Турчина, но, к сожалению, они недостаточно проработаны для их воплощения в виде вычислительной модели. Тем не менее от концепции метасистемных переходов можно в будущем отталкиваться при решении поставленной проблемы автоматического формирования иерархий.

Кроме того, эта концепция легла в основу суперкомпилятора Рефал (см. http://refal.net/). Мы не будем углубляться в идею суперкомпиляции (см., например, [368]), а лишь проведем аналогию с интерпретацией изображений. Если иерархическое разбиение программ на подпрограммы можно сравнить с подходом с переменной разрешающей способностью на пиксельном уровне, то идею суперкомпиляции — с иерархичностью по уровням абстракции (пиксельному, контурному, структурному). Так что интересной задачей для исследования является адаптация идей суперкомпиляции к проблеме вычисления алгоритмической сложности.

В концепции метасистемных переходов не прорабатывается возможность коррекции архитектуры нижних уровней после построения верхних уровней, т. е. адаптивный резонанс можно ввести не только в функционирование иерархической системы, но и в процесс ее построения. Иерархичность в природе универсальна на всех уровнях, начиная с клеток (можно начать цепочку и с элементарных частиц), кончая (на данный момент) наукой и культурой. С момента

Большого Взрыва шел процесс образования новых уровней организации материи. Если этот процесс однонаправленный, то строящиеся системы не оптимальны (к примеру, нейроны возникли задолго до возникновения разума; почему же они должны быть оптимальным аппаратным средством для решения интеллектуальных задач?). Наличие обратных связей между различными уровнями организации материи подразумевает коррекцию ранее сформированных уровней на основе результатов, полученных на поздних уровнях. Являются ли работы по генетике (генетической модификации биологических видов), химии (создание новых химических соединений, не встречающихся в природе) или ядерной физике (создание новых элементов) примерами таких обратных связей? Если да, то насколько глубоко эти обратные связи могут распространяться? Можно ли рассматривать процесс создания искусственного интеллекта как процесс установления резонанса между различными уровнями организации материи, а человеческий интеллект — как нулевое приближение к нему? Однако здесь мы вступаем в область спекулятивных суждений, так что оставим эти вопросы без ответа.

3.6. ЗАКЛЮЧЕНИЕ

В этой главе были описаны некоторые возможности применения принципа МДО при решении задач машинного восприятия, которые нами рассматривались в качестве частного случая индуктивного вывода. Особенность этих задач заключается в большом объеме данных, служащих основой для индуктивного вывода, и в большой сложности источника, порождающего эти данные. Благодаря этой особенности становится совершенно ясно, что умозрительные универсальные методы индуктивного вывода, дающие идеальное решение за экспоненциальное время, для решения этих задач применены быть не могут. Это, казалось бы, должно привести к тому, что для каждой отдельной задачи необходимо разрабатывать частные методы, более нигде не применимые. К счастью, это оказывается не совсем так. Хотя в методах интерпретации сенсорной информации различных модальностей и должна присутствовать модально-специфическая априорная информация, сами методы имеют много общего. Общность методов выражается в том, что они

базируются на очень похожих алгоритмах распознавания, группирования, регрессии и сегментации, что делает из этих алгоритмов полезный инструментарий для решения различных задач индуктивного вывода.

Этим общность методов анализа не ограничивается. Оказывается, что системы индуктивного вывода на основе сенсорной информации любой модальности строятся иерархически, что позволяет получать приемлемое по качеству решение за полиномиальное время. Благодаря механизму адаптивного резонанса, связывающего разные уровни иерархии, работа системы оказывается гораздо более робастной. Привлечение иерархических представлений с обратными связями в естественных системах анализа информации оказывается настолько широким, что их всегда следует иметь в виду при решении любой сколько-нибудь сложной задачи индуктивного вывода, особенно если это решение опирается на принцип МДО.

С другой стороны, и привлечение принципа МДО может оказаться полезным в машинном восприятии. Мы показали, что его можно применить для интерпретации сенсорной информации разных модальностей. При этом, как утверждают многие исследователи, повышается качество работы систем машинного восприятия по сравнению с классическими подходами. Одно из преимуществ информационного подхода заключается в том, что разные уровни анализа разных модальностей связываются единой целевой функцией, не важно, дискретные или непрерывные представления там используются. Это позволяет строго описать эффект адаптивного резонанса через задание общей для нескольких уровней целевой функции, выражающей длину описания. Такая информационная трактовка адаптивного резонанса позволяет корректно воплощать данный механизм в технических системах.

Альтернативным подходом для решения задач индукции является байесовский подход, но при его привлечении возникают трудности с получением априорных вероятностей моделей и с вычислением правдоподобия для нетривиальных стохастических моделей. При использовании информационного подхода эти проблемы решаются при конструктивном задании представления предметной области, что также дает и критерий выбора между представлениями, а значит, позволяет сделать исследования в области компьютерного восприятия более целенаправленными.

В результате функционирования обучающейся системы машинного восприятия должна строиться концептуальная система, состоящая из взаимосвязанных понятий, основанных на семантике. Мышление осуществляется построенными понятиями, но работа системы восприятия еще не мышление. В следующей главе мы рассмотрим возможность применения принципа МДО к проблемам, решение которых традиционно относилось к прерогативе интеллекта.

Глава 4

ВЫСОКОУРОВНЕВЫЕ ЗАДАЧИ ИНДУКТИВНОГО ВЫВОДА

4.1. ПРОБЛЕМА ИНДУКТИВНОГО ВЫВОДА СИМВОЛЬНЫХ ПРЕДСТАВЛЕНИЙ

В предыдущей главе в общих чертах была описана схема начального этапа построения концептуальной системы, состоящей из дискретных элементов — понятий и взаимосвязей между ними. Формирование абстрактных концептов возможно на основе только семантической информации, однако наличие лингвистического канала заметно облегчает этот процесс, слова выступают своего рода «центрами кристаллизации» в концептуальной системе. При этом лингвистическая информация передается посредством тех же сенсорных каналов, что и семантическая информация. Для младенца слова играют роль обычных сигналов, поступающих из физического мира. Отличие этих сигналов от прочих состоит в том, что они входят в другую систему, обусловленную человеческим социумом.

Вторую сигнальную систему обычно называют одним из наиболее принципиальных (наряду, например, со способностью изготавливать орудия труда, которая, хотя и есть у животных, но в зачаточном состоянии) отличий человека от животного. Иными словами, давно уже общепринятым стал тезис, согласно которому мышление человека отличается от мышления животного способностью оперировать знаками, система которых образует язык. Хотя многие животные также используют язык для обмена информацией (причем эта способность в процессе эволюции появилась очень рано, например, весьма сложный язык есть у муравьев или пчел), но язык животных является преимущественно врожденным (а значит, изменяющимся в ходе биологической эволюции, т. е. очень медленно) и узкоспециализированным. Даже у высших животных способности к языку весьма ограничены (хотя они и оказались заметно выше, чем предполагалось ранее). Поскольку у человека есть стремление отделять себя от мира животных и признавать наличие интеллекта только за собой, то неудивительно, что главенствующим направлением в искусственном интеллекте долгое время было исследование того, что отличает че-

ловека от животного, а именно языка, или, вернее, развитой способности к манипулированию знаками. Наибольшее же проявление интеллектуальных способностей у человека усматривалось в решении абстрактных задач, например, в игре в шахматы или в доказательстве математических теорем. Творческие способности, носящие невербальный характер, отделялись от интеллектуальных способностей, ассоциировавшихся с «логическим мышлением».

Использование знаков, бесспорно, является ключевым моментом в развитии человеческого мышления. Язык является как средством коммуникации, так и средством моделирования внешнего мира [367, гл. 7], что, в частности, позволяет манипулировать объектами, недоступными в физическом мире. Классическим примером моделирования посредством манипуляции со знаками является счет на пальцах: маленький ребенок, который не умеет считать в уме, может определить количество человек, оставшихся в комнате, загибая и разгибая пальцы, когда люди входят в комнату и выходят из нее соответственно. Палец, соотнесенный с человеком, выступает в роли знака, примитивной модели. Постепенно манипуляции с материальными знаками редуцируются до использования их ментальных образов (например, мышление ребенка вслух сворачивается до внутренней речи), хотя многие символьные операции человек продолжает выполнять при использовании внешних средств (простой пример — умножение в столбик). Таким образом, преобразование знаков, действительно, имеет прямое отношение к мышлению. И хотя на уровне сознания человек может оперировать не только словами, а вербальное мышление — лишь «вершина айсберга» (на уровне сознания обрабатывается порядка $10–10^2$ бит/с, в то время как весь мозг обрабатывает не мене 10^{10} бит/с), но исследование этого верхнего уровня было, бесспорно, полезным для развития области искусственного интеллекта и имело много практических приложений (хотя некоторые исследователи [74, с. 802] и утверждают, что «было ошибкой начинать работы в сфере ИИ с реализации высокоуровневых процессов рассуждений в моделируемом разуме»).

Суть данного направления исследований была выражена Ньюэллом и Саймоном в гипотезе физической символьной системы [369], согласно которой для достижения интеллектуального поведения системой необходимо и достаточно, чтобы эта система выполняла преобразование сим-

вольной информации (отметим, что здесь использующиеся понятия необходимости и достаточности не обладают математической строгостью, так как множество интеллектуальных систем полагается подмножеством множества физических символьных систем). По сути, данная гипотеза говорит не более того, что искусственный интеллект может и должен быть реализован на физическом воплощении универсальной машины Тьюринга (или ее эквивалента). Однако акцент делается на символьные операции, хотя ту же гипотезу можно было бы выдвинуть и относительно количественных операций.

В этом контексте интересно вспомнить о неполноте формализованной арифметики. Суть формализации некоторой научной теории (в рамках аксиоматического метода) заключается в ее сведении к конечному набору операций над символами. Оказывается, что относительно количественных операций, являющихся предметом арифметики, существуют утверждения, истинность или ложность которых не может быть установлена в рамках формализованной арифметики.

Существование неразрешимых проблем хотя и не говорит о слабости аксиоматического метода (который также можно назвать синтаксическим), но имеет большое значение для математической методологии. Особенно ярко это значение видно на примере понятия действительного числа, в котором сходятся два противоположных подхода познания и осмысления в математике: абстрактная алгебра и топология [370, с. 24–41]. Так, топология отталкивается от понятия непрерывности как базового понятия и лишь в процессе конкретизации объектов своего рассмотрения придает им некую структуру, в то время как в алгебре непрерывность вводится лишь в самом конце конкретизации и довольно искусственным образом [370, с. 34]. При этом многие теоремы в математике могут рассматриваться с обеих позиций, причем в рамках топологического подхода эти теоремы могут иметь тривиальные доказательства, а в рамках алгебраического — очень сложные, и наоборот.

В п. 3.4.1 мы уже сталкивались со случаем, когда отвлеченные философские проблемы имели непосредственное отражение в практических задачах. Также и упомянутые сейчас теоретические проблемы математики имеют весьма прямую связь с рассматриваемыми в данной книге практическими методами анализа данных. В частности, не пред-

ставляет трудности провести параллель между топологическими и дискриминантными методами и между алгебраическими и структурными методами распознавания образов. Действительно, отправной точкой дискриминантных методов является понятие непрерывности, в то время как для символьных методов непрерывность вводится очень тяжело (см. п. 4.5.6). Области применения у этих двух классов методов также оказываются различными.

Различие алгебраического и топологического подходов в математике дает намек на принципиальную необходимость использования нескольких разных типов представлений и в машинном обучении. Параллели можно продолжить и дальше, приняв связь между областями математики и типами данных (булевыми, целочисленными, с плавающей точкой, строковыми, массивами и др.) в языках программирования. Хотя все вычислительные операции можно представить только через операции над строками, либо только через арифметические операции, либо только через логические операции, эти типы данных разделены не зря. Можно утверждать, что они создают фундамент различного рода представлений. Вернемся, однако, к символьным представлениям.

Исследование символьных систем привело к трем важнейшим принципам в методологии искусственного интеллекта [74, с. 782]:

1) символьные представления являются основным средством для описания мира (представления знаний);

2) перебор вариантов в рамках символьных представлений (поиск в пространстве состояний) является моделью мыслительных процессов;

3) интеллектуальность символьной системы не зависит от средств ее реализации (отвлеченность когнитивной архитектуры).

Первые два принципа связаны с исследованием символьных *представлений* информации и алгоритмов *поиска* соответственно. Результаты, полученные по символьным представлениям, имеют широкое применение при проектировании экспертных систем и естественно-языковых (ЕЯ) систем. Исследование алгоритмов поиска наиболее интенсивно велось в направлении эвристического программирования, без которого было невозможно обойтись в игровых задачах, автоматическом доказательстве теорем, а также при разработке машин логического вывода, являющихся вто-

рым (наряду с подсистемой представления знаний) основным компонентом экспертных систем.

Обращение к методам эвристического программирования было бы весьма плодотворным для наших целей, поскольку в этих методах глубоко проработана проблема поиска в условиях комбинаторного взрыва. Время поиска в пространстве состояний, порождаемом практически любой нетривиальной задачей, оказывается экспоненциально зависящим от размерности задачи, в связи с чем необходимо отказаться от просмотра всего пространства состояний (и, возможно, от получения оптимального решения) в целях сокращения перебора. Общие приемы сокращения перебора получили название *эвристик*.

Многие эвристики связаны с *инвариантными преобразованиями* в пространстве состояний. Например, для игры в крестики-нолики на «бесконечном» поле игровое состояние инвариантно по отношению к параллельному переносу, вращению на угол, кратный 90°, и зеркальному отражению. В частности, это приводит к тому, что положение первого «крестика» не имеет значения, а из всех возможных положений следующего «нолика» можно рассматривать менее четверти. Без учета этих инвариантов сделать даже первый ход было бы нетривиально: пришлось бы перебирать все возможные положения первого «крестика» и для каждого из них анализировать все возможные положения «ноликов». Другого рода эвристики опираются на *эвристические меры* качества, приписываемые каждой точке в пространстве состояний. Например, в игре в крестики-нолики в такую эвристическую меру в качестве составной части могут входить количество и длины прямых открытых цепочек из «крестиков» («ноликов»). Тогда алгоритм поиска будет в первую очередь просматривать такие ходы, которые ставят «крестик» или «нолик» невдалеке от уже имеющихся, что резко повысит эффективность перебора. В существующих же шахматных программах эвристическая оценка качества позиции крайне сложна и включает не только подсчет количества и силы фигур, но и оценку позиционного преимущества.

Инварианты и эвристическая мера связаны со спецификой конкретной задачи (или предметной области в экспертных системах). Существуют и некоторые общие эвристики, или метаправила (см., например, [209, с. 116–118]). Примером такой эвристики может служить правило, согласно ко-

торому требуется в первую очередь рассматривать наиболее новые данные. Однако такие эвристики не формализованы, а их применение также зависит от конкретной задачи.

Как поиск инвариантов, так и конструирование эвристической меры в большинстве случаев осуществляется человеком, формализующим экспертные знания о конкретной проблеме. Особый интерес представляют методы (к сожалению, недостаточно разработанные) автоматического построения эвристик на основе опыта решения простых задач и использования этих эвристик при решении сложных задач. Автоматическое построение эвристик — очень сложная задача. Подумайте, например, как автоматически установить индифферентность первого хода в игре в крестики-нолики на большом поле и увеличение значимости этого хода при уменьшении размеров поля или как автоматически сформировать меру качества шахматной позиции (причем не зная силы фигур). Программы EURISKO [371] и LEX [74, с. 388–391] являются примерами программ, способных учиться новым эвристикам, хотя их возможности весьма ограничены.

В задачах индуктивного вывода также широко применяется поиск в пространстве состояний (гипотез), но вместо некоторой эвристической меры может использоваться длина описания. В связи с этим представляют интерес сами алгоритмы обхода пространства состояний. Не подвергая анализу общеизвестные алгоритмы поиска (такие, как поиск в глубину или ширину, процедура альфа-бета-отсечения, минимакса и т. д.), укажем, однако, на возможность использования методов эвристического программирования в задачах индуктивного вывода. Существует и другая возможность: использовать методы индуктивного вывода в эвристическом программировании. Разработка хорошей эвристики — эмпирическая проблема [74, с. 158]. Поиск эвристик на основе опыта тоже может рассматриваться как индуктивный вывод. Для этого необходимо описать пространство эвристик (или способ их представления), определить, что является наблюдательными данными, и сформировать информационный критерий качества эвристики.

В целях иллюстрации применения принципа МДО в этой книге рассматриваются не общие проблемы искусственного интеллекта, а лишь те его задачи, которые сводятся к индуктивному выводу. К сожалению, эвристическое программирование преимущественно применялось к задачам де-

дуктивного вывода, а автоматическое построение эвристик изучено мало. На этих задачах затруднительно одновременно обсуждать механизмы поиска и демонстрировать использование принципа МДО, поэтому нам не удастся в должной мере осветить результаты, достигнутые в области эвристического программирования. Однако остается еще одна чисто индуктивная задача — построение символьного представления по данным наблюдений, что является дальнейшим продолжением процесса интерпретации сенсорной информации. Исследователями отмечается [74, с. 785], что «прямым следствием бедной семантики является то, что методология поиска в традиционном ИИ рассматривает лишь предварительно интерпретированные состояния и их контексты. Это означает, что создатель программы ИИ связывает с используемыми символами семантический смысл». В рамках концептуальной системы хаос сенсорных ощущений переводится в упорядоченные цепочки символов. Далее оказывается необходимым осуществлять индукцию по цепочкам символов, что является связующим звеном между машинным восприятием и экспертными системами. И здесь принцип МДО находит широкое применение.

4.2. ФОРМАЛЬНЫЕ ГРАММАТИКИ

4.2.1. Историческая справка

Проблемы автоматического обучения языку уже были затронуты в гл. 3. Но там мы ограничились вопросами обучения простым сингулярным терминам и не рассмотрели такие важные проблемы, как формирование общих понятий и обучение грамматике языка. Слова составляют предложения не произвольным образом, а по определенным правилам. Чтобы выявить закономерности в расположении слов в предложении, необходимо определиться с представлением, в рамках которого можно было бы эти закономерности описывать.

Эти вопросы являются центральными для изучения в лингвистике — науке о структуре естественных языков. Именно исходя из потребностей лингвистики, Ноамом Хомским [372–374] в середине прошлого века была разработана теория формальных грамматик, которая стала одним из основных разделов математической лингвистики. В отли-

чие от традиционной лингвистики, математическая лингвистика опирается на формальные математические модели при исследовании структуры естественных и искусственных языков. Такой подход стал актуальным, когда эвристические методы традиционной лингвистики оказались недостаточными для постановки и решения проблем автоматического перевода и машинного понимания текста, а также для удовлетворительного анализа начавших развиваться искусственных языков, в том числе и языков программирования.

Итак, представление структуры естественных языков может основываться на аппарате формальных грамматик. Однако этим их применение далеко не ограничивается. Возникновение и развитие формальных грамматик представляет собой интересный феномен в науке. Предложенная (исходя из нужд одной науки) теория формальных грамматик содержала в себе настолько фундаментальные идеи, что оказалась не менее значимой для других областей знаний, весьма далеких, на первый взгляд, от лингвистики. В первую очередь речь идет о теории алгоритмов и теории автоматов. Оказалось, что наиболее широкий класс формальных грамматик эквивалентен машине Тьюринга. При этом формальные грамматики предоставляют удобные средства и для задания более слабых моделей алгоритмов (вплоть до модели конечного автомата, о чем будет сказано ниже). Нам это тем более интересно, что, перенеся понятие алгоритмической сложности с концепции машины Тьюринга на концепцию формальных грамматик и воспользовавшись менее мощными представлениями, можно надеяться получить такие методы поиска минимальных моделей, которые помогают избежать проблемы комбинаторного взрыва.

Теория формальных грамматик, позволив перекинуть мост между исследованиями структуры языков и теорией алгоритмов, стала основой при разработке трансляторов и компиляторов, а затем стала использоваться и в автоматическом программировании.

И наконец, формальные грамматики оказались применимыми в распознавании образов — на их базе были развиты синтаксические методы распознавания. Это лишний раз подчеркивает общность формальных грамматик, их способность описывать структуру любых, не только лингвистических, объектов. При использовании в машинном обучении формальные грамматики могут рассматриваться как средство задания произвольных языков представления инфор-

Таблица 4.1

Основные области применения теории формальных грамматик

Область применения	Классические вопросы	Вопросы ИИ
Лингвистика	Описание структуры естественных языков	Автоматический перевод, понимание текстов
Искусственные языки	Модули синтаксического разбора в трансляторах и компиляторах	Автоматическое написание и понимание программ
Теория алгоритмов	Исследование типов языков и классификация алгоритмов	Символьные представления в задачах машинного обучения
Теория автоматов	Классификация динамических систем; расширение модели конечных автоматов	Символьные представления в задачах управления и принятия решений
Распознавание образов	Синтаксический подход к распознаванию образов	

мации или пространств моделей. Возможность конструирования символьных представлений на основе формальных грамматик заставляет обратить на них наше внимание в рамках темы данной книги.

В табл. 4.1 приведены основные области применения теории формальных грамматик.

Благодаря столь разнообразным применениям формальные грамматики исследовались весьма активно и стали самостоятельным предметом изучения, абстрагированным от конкретных приложений. В теории формальных грамматик получено множество результатов. Даже простое перечисление их займет много места и времени. Приведем лишь основные понятия и ограничимся иллюстрацией применимости принципа МДО в грамматическом выводе. Более детальное изложение теории формальных грамматик и ее применения можно найти в работах [121, 375].

4.2.2. Основные определения

Как уже отмечалось, формальные грамматики были призваны математически строго описать структуру языков. Каждое предложение некоторого языка можно считать цепочкой символов. В зависимости от уровня рассмотрения для

естественного языка в качестве таких символов можно принимать буквы, морфемы, слова и т. д. Наличие структуры в предложениях (словах) языка означает присутствие некоторых закономерностей в расположении составляющих их символов. Таким образом, формальные грамматики предназначены для описания закономерностей строения предложений конечной совокупностью однозначно определенных правил. В зависимости от формы этих правил обычно выделяют:

• *порождающие грамматики*, состоящие из множества правил построения любых допустимых в данном языке предложений; процесс построения предложения также определяет его структуру;

• *распознающие грамматики*, состоящие из совокупности правил, которые путем анализа структуры некоторого предложения позволяют определить, является ли оно допустимым в данном языке или нет;

• *преобразующие,* или *трансформационные, грамматики,* содержащие правила преобразования любых корректно построенных предложений в другие допустимые предложения с учетом их структуры (такие преобразования могут быть необходимы, например, для получения синонимичных предложений).

Более подробно рассмотрим более изученные порождающие грамматики. Введем для них основные понятия.

Под *алфавитом*, как и раньше, будем подразумевать некоторое конечное множество, элементы которого будем называть *символами*.

Предложением в данном алфавите будем называть произвольную (конечную) цепочку символов этого алфавита.

Языком над данным алфавитом будем называть произвольное (возможно, бесконечное) множество предложений в этом алфавите.

Отметим, что вместо термина «алфавит» в литературе может использоваться термин «словарь» (а вместо термина «символ» — термин «слово»). В то же время для обозначения цепочки символов вместо термина «предложение» также может использоваться термин «слово» (к примеру, в п. 1.3 цепочку кодовых символов мы называли «кодовым словом», как принято в теории информации). Чтобы избежать путаницы, мы выбрали наиболее удаленные по смыслу понятия «символ» и «предложение».

Порождающей грамматикой будем называть четверку $G = (V_T, V_N, P, S)$, где V_T — алфавит *терминальных* (основ-

ных) *символов*; V_N — алфавит *нетерминальных* (вспомогательных) *символов*, причем $V_N \cap V_T = \varnothing$; $V_N \cup V_T = V$ — алфавит *грамматики* G; P — множество *правил подстановки* (или продукций); S — *начальный* (корневой) *символ*, $S \in V_N$.

Здесь мы воспользовались стандартными обозначениями элементов грамматики. Также принято для обозначения терминальных символов использовать строчные буквы латинского алфавита — a, b, c, ...; для обозначения нетерминальных символов — прописные буквы A, B, C, ..., а для предложений в алфавите V — строчные буквы греческого алфавита α, β, γ, ... Пустую цепочку, не содержащую символов, обычно обозначают через Λ.

Как и ранее, через V^* будем обозначать все возможные цепочки символов (предложения) алфавита V. Через V^+ обозначим множество $V^* \backslash \{\Lambda\}$. Тогда правило подстановки (элемент множества P) будет иметь вид: $\alpha \to \beta$, где $\alpha \in V^+$, $\beta \in V^*$, причем хотя бы один символ предложения α должен быть нетерминальным. Это правило говорит о том, что в любом предложении вхождение цепочки символов α *может быть* заменено цепочкой β: $\left(\forall \gamma, \delta \in V^*\right) \gamma\alpha\delta \Rightarrow \gamma\beta\delta$. Символ $\Rightarrow$, в отличие от символа $\to$, который применяется при записи правил вывода, используется для обозначения возможности вывода одного предложения из другого в результате применения некоторого количества (одного или нескольких) правил грамматики. Вывод в грамматике начинается с корневого символа S и заключается в последовательном применении правил подстановки.

Предложение называется *терминальным*, если оно состоит только из терминальных символов. Поскольку в любом правиле вывода слева стоит цепочка, содержащая, по крайней мере, один нетерминальный символ, то к терминальному предложению не могут быть применены никакие правила вывода. Могут существовать и нетерминальные предложения, вывод из которых не может быть продолжен.

Через $\Gamma(G)$ обозначим язык, порождаемый грамматикой G и содержащий все терминальные предложения (и только их), выводимые из начального символа S, т. е.

$$\Gamma(G) = \left\{\alpha \middle| \left(\alpha \in V_T^*\right) \& \left(S \Rightarrow \alpha\right)\right\}. \qquad (4.1)$$

Рассмотрим пример. Пусть даны алфавиты $V_T = \{a, b\}$, $V_N = \{S, A, B\}$ и правила подстановки $P = \{S \to AS, S \to SB,$

$S \to \Lambda, A \to ab, B \to ba\}$. В грамматике с такими компонентами могут быть выведены любые цепочки вида $(ab)^n(ba)^m$. Приведем одну из бесконечного множества возможных последовательностей подстановок:

$$S \Rightarrow AS \Rightarrow AAS \Rightarrow AASB \Rightarrow AAB \Rightarrow abAB \Rightarrow ababB \Rightarrow ababba.$$

На основе этого простого примера сделаем ряд наблюдений.

• На каждом шаге вывода может существовать выбор из нескольких возможностей применения грамматических правил. Таким образом, грамматика не задает детерминированный алгоритм порождения некоторой цепочки символов, в отличие, например, от машины Тьюринга, последовательность операций которой предопределена содержимым входной ленты. Грамматика не указывает последовательность действий по порождению цепочек символов, а устанавливает ограничения на возможные действия (и, как следствие, на порождаемые цепочки). В то же время каждая порождаемая цепочка описывается в грамматике последовательностью правил подстановки, соответствующей генеративной модели набора символьных данных. В этом и заключается смысл грамматики как символьного представления, задающего пространство структурированных объектов.

• Конечный набор правил может порождать язык, содержащий бесконечное количество предложений.

• Различные грамматики могут порождать одинаковые языки (такие грамматики называются *слабоэквивалентными*). Например, грамматика, состоящая из элементов $V_T = \{a, b\}$, $V_N = \{S\}$, $P = \{S \to abS, S \to Sba, S \to \Lambda\}$, порождает тот же язык, что и грамматика, рассмотренная выше.

В приведенном примере нетерминальные символы использовались для обозначения определенных цепочек терминальных символов. Нетерминальные символы могут использоваться также для обозначения некоторого подмножества терминальных символов. Например, если даны алфавиты терминальных символов $V_T = \{a, b, c, d\}$ и нетерминальных $V_N = \{S, A, B\}$ и существуют правила подстановки $A \to a, A \to b$ и $B \to c, B \to d$, то нетерминальные символы A и B будут обозначать соответствующие классы терминальных символов $\{a, b\}$ и $\{c, d\}$. В более сложных случаях одни нетерминальные символы могут обозначать классы других нетерминальных символов или их цепочек, а также

смешанные классы и цепочки, что приведет к сложным структурам предложений.

Итак, терминальные символы представляют собой непроизводные элементы, из которых формируются языковые объекты, а нетерминальные символы являются различного рода обобщениями этих непроизводных элементов, описывая их классы или сконструированные из них цепочки.

Рассмотрим пример простой грамматики, описывающей небольшую часть естественного языка. Пусть алфавит V_T содержит следующие терминальные символы (слова): *быстрый, быстрые, медленный, медленные, большой, большие, компьютер, компьютеры, калькулятор, калькуляторы, число, числа, величина, величины, умножает, умножают, складывает, складывают*. И пусть алфавит V_N содержит следующие нетерминальные символы: S — понятие правильно построенного предложения; C_1 — существительное единственного числа; C_2 — существительное множественного числа; $П_1$ — прилагательное единственного числа; $П_2$ — прилагательное множественного числа; $Г_1$ — глагол единственного числа; $Г_2$ — глагол множественного числа; C_1', C_2' — группы существительного; $Г_1'$, $Г_2'$ — группы глагола. И пусть есть следующий набор правил:

$$S \to C_1'Г_1';\ \ S \to C_2'Г_2';$$

$$C_1' \to П_1C_1;\ \ C_1' \to C_1;\ \ C_2' \to П_2C_2;\ \ C_2' \to C_2;$$

$$Г_1' \to Г_1C_1';\ \ Г_1' \to Г_1C_2';\ \ Г_1' \to Г_1;\ \ Г_2' \to Г_2C_1';\ \ Г_2' \to Г_2C_2';$$

$$Г_2' \to Г_2;$$

$$C_1 \to \textit{компьютер};\ C_1 \to \textit{калькулятор};\ C_1 \to \textit{число};\ C_1 \to \textit{величина};$$

$$C_2 \to \textit{компьютеры};\ C_2 \to \textit{калькуляторы};\ C_2 \to \textit{числа};\ C_2 \to \textit{величины};$$

$$Г_1 \to \textit{умножает};\ Г_1 \to \textit{складывает};$$

$$Г_2 \to \textit{умножают};\ Г_2 \to \textit{складывают};$$

$$П_1 \to \textit{быстрый};\ П_1 \to \textit{медленный};\ П_1 \to \textit{большой};$$

$$П_2 \to \textit{быстрые};\ П_2 \to \textit{медленные};\ П_2 \to \textit{большие}.$$

Такая грамматика порождает некоторые предложения русского языка, имеющие сходную синтаксическую структуру. Пример допустимого предложения представлен на рис. 4.1 с соответствующим *структурным деревом*. Не-

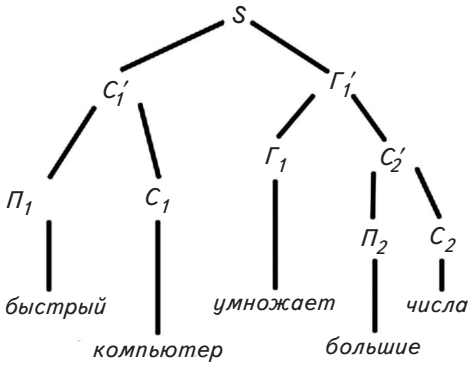

Рис. 4.1. Структурное дерево фразы «быстрый компьютер умножает большие числа», порожденной с помощью некоторой простой формальной грамматики

терминальные символы здесь обозначают части речи и синтаксические типы. В равной степени нетерминальные символы могут обозначать общие понятия, устойчивые словосочетания, парадигмы слов (как обобщения словоформ) и т. д. Таким образом, порождающие грамматики задают представления, которые могут использоваться при конструировании концептуальной системы, вернее, некоторых из тех ее аспектов, которые были опущены в гл. 3 настоящей книги.

Хотя приведенная в примере грамматика порождает только правильно построенные предложения, не все из них оказываются вполне осмысленными. В частности, допустимым является такое предложение: «Медленные величины складывают большой калькулятор». Иными словами, грамматические правила не описывают семантику языка. В теории формальных грамматик семантическая структура лексических единиц разработана наиболее слабо [133, с. 17]. Попытка разрешить проблему семантики предпринимается в рамках трансформационных грамматик, в которых задаются такие формальные правила преобразования предложений, которые сохраняют смысл предложения. Однако, как было отмечено в гл. 3, проблема смысла неразрешима только на основе лингвистической информации. В связи с этим не следует искать решения этой проблемы в самих формальных грамматиках и не нужно рассматривать неразрешимость проблемы в качестве недостатка этой теории. Вместо этого следует изучать возможность построения сенсорных представлений в качестве семантической опоры для лингвистических объектов (возможно, представленных с помощью формальных грамматик).

Вернемся, однако, к самой теории формальных грамматик. Введем следующее понятие. Некоторая грамматика G

называется *однозначной*, если для каждого слова из порождаемого ею языка $\alpha \in \Gamma(G)$ верно, что все возможные варианты его вывода имеют идентичные структурные деревья. Вместо сравнения структурных деревьев двух выводов можно воспользоваться следующим приемом: заменить правила $\alpha \to \beta$ правилами $\alpha \to (\beta)$, где скобки выполняют роль служебных символов и не учитываются при выводе. Если в результате двух выводов формируются цепочки символов, совпадающие вплоть до расстановки скобок, то такие выводы не считаются существенно различными. Грамматика называется неоднозначной, если хотя бы для одной цепочки $\alpha \in \Gamma(G)$ имеются существенно различные выводы.

Часто подчеркивается неоднозначность естественных языков: в них существуют предложения, имеющие несколько альтернативных синтаксических структур при одинаковом написании. Классическим примером неоднозначного предложения для английского языка является предложение «They are flying planes» («Это летящие самолеты» или «Они летят на самолетах»). Несмотря на такую неоднозначность, обычно ограничиваются рассмотрением однозначных грамматик, так как они существенно проще в использовании.

Выражения естественного языка характеризуются не только неоднозначностью — в значительно большей степени им присуща избыточность: одна и та же мысль может быть выражена многими способами. В п. 3.3.5 мы уже приводили пример предложения из работы [133, с. 13]: «Лишь большое количество специальных терминов в данном тексте не позволит Смиту перевести его». Существует несколько миллионов предложений, соответствующих данному предложению по смыслу. Процесс формирования высказывания может характеризоваться как развертывание мысли во внешнюю речь. Для понимания высказывания, напротив, необходимо свернуть (или сжать) это высказывание. Иными словами, необходимо разделить независимые факторы: смысл высказывания и его грамматическую структуру. Часто в лингвистических исследованиях вводится глубинная грамматическая структура, представляющая собой промежуточный уровень между уровнем мысли и уровнем поверхностной грамматической структуры (уровнем организации развернутого речевого высказывания). Глубинные грамматические структуры высказывания строятся на основе малого числа правил, которые практически не связаны с конкретным естественным языком, но все еще являются модально-специфичными.

Вернемся к рассмотренному примеру формальной грамматики и обратим внимание на использованные в нем нетерминальные символы. Например, символ Γ_2 обозначал глагол множественного числа. Видно, что каждое слово описывается целым набором признаков (таких, как часть речи, род, лицо, число и т. д.). Всевозможных комбинаций этих признаков может быть очень много, так что при том простом способе задания правил вывода, который мы использовали в нашем примере, описание грамматики может получиться очень громоздким. Для описания сложных закономерностей в структуре предложений необходимо использовать правила более сложного вида. В зависимости от вида используемых правил грамматики обладают разной выразительной силой и могут быть разделены на несколько типов.

4.2.3. Типы формальных грамматик

Грамматики с правилами подстановки произвольного вида оказываются неудобными в применении на практике. Вместо них используются грамматики некоторых специальных типов. В зависимости от формы правил постановки принято выделять четыре типа грамматик, предложенных Н. Хомским (впоследствии были выделены промежуточные подтипы). Эти типы грамматик вводятся путем наложения все более сильных ограничений на вид правил подстановки.

Неограниченные грамматики (грамматики типа 0) — характеризуются правилами подстановки вида $\alpha \to \beta$, где на цепочки $\alpha \in V^+$, $\beta \in V^*$ не накладывается никаких ограничений.

Грамматики непосредственно составляющих (НС-грамматики, контекстно-зависимые грамматики, грамматики типа 1) — состоят из правил вида $\gamma A \delta \to \gamma \beta \delta$, где $A \in V_N$; $\gamma, \delta \in V^*$; $\beta \in V^+$. Наложение ограничений на вид правил делает языки, порождаемые НС-грамматиками, более узкими, чем языки, порождаемые грамматиками типа 0. Условие $\beta \in V^+$ (или $\beta \neq \Lambda$) делает НС-грамматики *неукорачивающими:* в них в процессе вывода применение любого правила приводит к тому, что цепочка увеличивает или, по крайней мере, не уменьшает свою длину.

Контекстно-свободные грамматики (КС-грамматики, бесконтекстные грамматики, грамматики типа 2) — состоят из продукций вида $A \to \beta$, где $A \in V_N$; $\beta \in V^+$. Посколь-

ку в НС-грамматиках возможно выполнение условия $\gamma = \delta = \Lambda$, очевидно, что КС-грамматики являются частным случаем НС-грамматик, в которых при замене нетерминального символа выбор подставляемой цепочки может производиться в зависимости от контекста.

Регулярные грамматики (автоматные грамматики, грамматики типа 3) — содержат правила вида $A \to aB$ и $A \to b$, где $A, B \in V_N$; $a, b \in V_T$, т. е. в правилах подстановки фигурируют лишь отдельные символы, а не их цепочки. В связи с этим количество различных правил в грамматиках этого типа ограничивается размером алфавитов. В грамматиках типов 0–2 такого ограничения нет: при фиксированных алфавитах количество правил может быть любым.

Отметим, что в грамматиках типов 1–3 на каждом шаге подстановки заменяется лишь один нетерминальный символ. Это позволяет представлять структуру предложения в виде дерева, подобного тому, которое представлено на рис. 4.1. Некорневые узлы и листья этого дерева называют *составляющими*, а узлы, являющиеся непосредственными потомками некоторого узла, — его *непосредственными составляющими*. Отсюда и происходит название НС-грамматик. Например, на рис. 4.1 C_1' и Γ_1' (группа существительного и группа глагола) являются непосредственными составляющими S (предложения).

В таком представлении предложения в форме структурного дерева не фиксируется последовательность правил вывода, за исключением того, что узлы должны рассматриваться позже своих родителей. В случае наличия правил, с помощью которых производится замена нескольких символов, структурное дерево невозможно. В то же время в неукорачивающей грамматике правило вывода, содержащее замену нескольких символов, может быть преобразовано в некоторое количество правил, в каждом из которых заменяется лишь один символ. В частности, правило вида $AB \to BA$ может быть реализовано на основе правил НС-грамматик вида $\gamma A \delta \to \gamma \beta \delta$ с добавлением четырех новых нетерминальных символов A_1, A_2, B_1, B_2 и с введением правил подстановки: $AB \to AB_1$; $AB_1 \to A_1B_1$; $A_1B_1 \to BB_1$; $BB_1 \to BA$.

Введение ограничений на правила вывода не обязательно приводит к сужению класса порождаемых языков, но с большей вероятностью делает описание некоторых регулярных структур более громоздким. Для теоретического анализа свойств грамматик обычно удобнее сделать правила

максимально простыми и однотипными, если это не приводит к сужению класса порождаемых языков. На практике, при построении конкретной грамматики, добавление нового вида правил, даже не приводящее к расширению класса порождаемых языков, может позволить получить существенно более компактную грамматику (с меньшим числом грамматических категорий и правил вывода), но с более сложными процедурами синтаксического разбора (см. п. 4.2.5).

Классы языков, порождаемых грамматиками типов 0–3, не совпадают [375, с. 124]. Наиболее широкими, естественно, являются языки класса 0. Интересно, что для любого языка класса 0 существует допускающая его машина Тьюринга (язык допускается некоторой машиной Тьюринга, если она останавливается за конечное число шагов, получив на вход любую из цепочек языка). И наоборот, если язык допускается некоторой машиной Тьюринга, то он является языком типа 0. Поскольку проблема останова произвольной машины Тьюринга неразрешима, следует ожидать, что и многие проблемы, связанные с неограниченными грамматиками, являются алгоритмически неразрешимыми. В частности, неразрешима проблема определения того, порождается ли некоторая цепочка некоторой неограниченной грамматикой в произвольном случае (в работе [375, с. 146] приведен ряд проблем теории грамматик с указанием того, в рамках грамматик каких типов данные проблемы алгоритмически разрешимы).

Языки типа 3 совпадают с множеством языков, допустимых автоматами гораздо более частного, чем машина Тьюринга, вида — конечными автоматами. Для автоматных грамматик многие проблемы имеют весьма простые решения, но сами грамматики обладают чрезмерно меньшей выразительной силой. Некий компромисс предоставляют НС- и КС-грамматики, которым и было посвящено большинство исследований. Для любой грамматики типа 1 или 2, как и для грамматики типа 3, может быть построен алгоритм, позволяющий для любой цепочки определить, порождается ли она данной грамматикой. Это принципиально отличает данные грамматики от неограниченных грамматик. Наибольший интерес представляют КС-грамматики, поскольку они достаточно выразительны для того, чтобы с их помощью исследовать языки программирования и, в определенной степени, — естественные языки; при этом для КС-грамматик многие проблемы имеют более простые

решения, чем для НС-грамматик (подробнее преимущества КС-грамматик см. в работах [375, с. 107, 124, 125]).

Если конкретная формальная грамматика задает представление информации, то типы грамматик задают пространства представлений. Возможность манипуляции такими пространствами путем простого разрешения или запрещения правил подстановки определенного вида делает привлекательным применение формальных грамматик в задачах машинного обучения.

Существуют различные типы грамматик, состоящие из правил вывода специфического вида, но совпадающие с одним из четырех упомянутых выше типов по множеству порождаемых языков. Были предложены и типы грамматик, порождающих промежуточные классы языков. Например, линейные грамматики, являющиеся такими КС-грамматиками, у которых все правила вывода в правой части имеют не более чем один нетерминальный символ, порождают множество языков, более узкое, чем множество, порождаемое всеми КС-грамматиками, но более широкое, чем множество, порождаемое только автоматными грамматиками. Выделено большое количество (несколько десятков, см., например, [375, с. 110, 122]) типов грамматик. Рассмотрение каждой из них выходит далеко за рамки данной книги. Ограничимся лишь упоминанием тех фактов, которые нам пригодятся для иллюстрации применения принципа МДО к грамматическому выводу.

4.2.4. Стохастические грамматики

Неограниченные грамматики в некотором смысле эквивалентны машинам Тьюринга, т. е. дают возможность построения универсальных представлений информации, однако зачастую бывает весьма полезно расширить формализм порождающих грамматик. Обратим внимание, что естественный язык допускает различные способы построения предложений, но одни из них могут использоваться существенно чаще, чем другие. В связи с этим каждому правилу вывода можно приписать некоторую вероятность, с которой оно применяется. Это приводит к понятию стохастических грамматик.

Стохастической грамматикой называется четверка $G = (V_T, V_N, P_S, S)$, где V_T, V_N — терминальный и нетерминальный алфавиты; S — начальный символ; P_S — множе-

ство стохастических правил подстановки, имеющих вид $\alpha \xrightarrow{P(\beta|\alpha)} \beta$, где $P(\beta \mid \alpha)$ — вероятность замены цепочки α на цепочку β (а не на другую цепочку).

Имеются следующие ограничения на вероятности: $0 < P(\beta \mid \alpha) \le 1$ (равенство вероятности нулю равносильно отсутствию соответствующего правила подстановки в грамматике) и $\sum_{\beta} P(\beta \mid \alpha) = 1$. В зависимости от формы правил стохастические грамматики также могут принадлежать типам 0–3, и различия между типами обычных формальных грамматик сохраняются и для стохастических грамматик.

Вообще говоря, существует более широкое определение стохастических грамматик, в котором вероятность применения некоторого правила зависит от того, какие правила в процессе вывода были применены до них. Такие грамматики применяются нечасто, и мы их рассматривать не будем.

Полагая применение правил независимым, можно определить вероятность осуществления некоторого процесса вывода через произведение вероятностей использованных правил. Поскольку используются условные вероятности $P(\beta \mid \alpha)$, то полагается, что на каждом шаге вывода выбор осуществляется только между правилами, в левой части которых стоит цепочка α. В то же время может быть применено некоторое правило и к какой-то другой подцепочке текущей цепочки. Таким образом, вероятность вывода как произведение вероятностей его отдельных правил имеет смысл безусловной вероятности, только если вывод не зависит от последовательности применения правил.

Вероятность некоторого вывода (вернее, структурного дерева) может быть принята за вероятность цепочки, являющейся результатом этого вывода. Это будет верно тогда, когда грамматика является однозначной, т. е. данной цепочке может соответствовать только одно структурное дерево. В противном случае определение вероятностей предложений порождаемого грамматикой языка $\alpha \in \Gamma(G)$ оказывается гораздо более проблематичным. Это является одной из причин, по которой зачастую ограничиваются рассмотрением однозначных грамматик.

Стохастические грамматики задают представления цепочек символов с распределением вероятностей по цепочкам, что отличает такие грамматики от обычных порождающих грамматик, задающих лишь представления с жесткими ограниче-

ниями на множества описываемых в их рамках цепочек. Посмотрим на конкретных примерах, как стохастические грамматики задают распределение априорных вероятностей.

Примеры. Пусть дана грамматика G: $V_T = \{a, b\}$; $V_N = \{S\}$; $P = \{S \to 0S;\ S \to 1S;\ S \to \Lambda\}$. В скобках отметим, что эта грамматика не является НС-грамматикой, так как не является неукорачивающей (в ней присутствует правило $S \to \Lambda$). Тем не менее для наглядности здесь и далее в примерах будут использоваться правила такого типа, так как они очень удобны для прерывания вывода. Заданная нами грамматика порождает все битовые строки произвольной длины. Зададим распределение вероятностей для цепочек языка $\Gamma(G)$, определив вероятности для правил вывода.

1. $P(0 \mid S) = P(1 \mid S) = P(\Lambda \mid S) = 1/3$. Тогда все цепочки длины n являются равновероятными и порождаются с вероятностью $3^{-(n+1)}$, так как вывод цепочки длины n осуществляется посредством $n + 1$ правил подстановки. Такая вероятность может показаться странной. Действительно, почему вероятность не равна, к примеру, 2^{-n}? Напомним, что при введении алгоритмической вероятности (см. п. 1.6.1) вероятность программы для машины Тьюринга длины $l(\alpha)$ принималась равной $P(\alpha) = 2^{-l(\alpha)}$. Заметим, однако, что для того, чтобы вероятности оказались нормированными, строки должны были представлять собой некий префиксный код, т. е. рассматривалось лишь подмножество множества всех битовых строк. В данном же случае возможны любые цепочки. Последовательность примененных правил вывода для порождения цепочки является ее описанием. Это описание неявно включает также информацию о длине цепочки. Полученные вероятности являются нормированными. Действительно, поскольку число различных цепочек длины n равно 2^{-n}, то вероятность получить в процессе вывода произвольную цепочку равна $\sum_{n=0}^{\infty} 2^n \cdot 3^{-(n+1)} = \frac{1}{3} \sum_{n=0}^{\infty} \left(\frac{2}{3}\right)^n = 1$. Это говорит о том, что вероятности цепочек языка $\Gamma(G)$ в случае описанной стохастической грамматики рассчитаны правильно.

2. $P(0 \mid S) = P(1 \mid S) = 0,49;\ P(\Lambda \mid S) = 0,02$. При таких вероятностях для правил вывода вероятность для каждой цепочки длины n будет равна $0,49^n \cdot 0,02$. Вероятность также зависит только от длины цепочки, но не от ее содержания, однако эта зависимость отличается от зависимости в примере 1. В частности, вероятность пустой цепочки здесь равна $0,02$,

в то время как в примере 1 — 1/3. Вероятность получения произвольной цепочки длины n будет $2^n \cdot 0,49^n \cdot 0,02 \sim 0,98^n$, т. е. зависимость гораздо более пологая, чем зависимость $(2/3)^n$ в примере 1. Итак, в стохастической грамматике существует возможность устанавливать зависимость вероятностей предложений языка $\Gamma(G)$ от их длины, изменяя вероятности правил вывода.

3. $P(0 \mid S) = 0,125;\ P(1 \mid S) = 0,375;\ P(\Lambda \mid S) = 0,5$. Нетрудно убедиться, что в такой стохастической грамматике наиболее вероятными будут цепочки, содержащие три четверти единиц и одну четверть нулей. Итак, стохастические грамматики позволяют устанавливать вероятности предложений языка в зависимости от входящих в эти предложения символов.

4. Рассмотрим другую стохастическую грамматику G: $V_T = \{a, b\}$; $V_N = \{A, S\}$; $P = \{S \to AS;\ S \to \Lambda;\ A \to 011111;\ A \to 0;\ A \to 1\}$; $P(AS \mid S) = P(\Lambda \mid S) = 0,5;\ P(011111 \mid A) = P(0 \mid A) = P(1 \mid A) = 1/3$. Данная грамматика является неоднозначной, но для нашего примера это несущественно. Важно то, что этой грамматикой могут порождаться любые цепочки, но с большей вероятностью будут порождаться предложения, содержащие три четверти единиц и одну четверть нулей. Другими словами, по содержанию символов в предложениях эта грамматика не отличается от рассмотренной в примере 3. Однако эта грамматика будет с большей вероятностью порождать цепочки, содержащие шаблон «011111».

Итак, с помощью стохастических грамматик можно в компактной форме задавать распределения априорных вероятностей на множестве цепочек символов. Для определения вероятности некоторой цепочки в рамках данной грамматики необходимо найти последовательность правил вывода, применение которых приводит к порождению цепочки. Это является задачей синтаксического разбора — классической задачей в теории формальных грамматик. Другой важной задачей является восстановление грамматики по обучающей выборке предложений. Сначала мы кратко рассмотрим первую задачу.

4.2.5. Синтаксический разбор

Представления данных, которые могут основываться, в частности, на формальных грамматиках, могут быть применены при решении двух задач: синтеза и анализа. Порож-

дающие грамматики задают правила построения символьных конструкций, определяя целое их множество (возможно, бесконечное). Использование формальных грамматик при решении задачи синтеза требует выбора из всего множества предложений языка некоторого одного предложения. Такой выбор подразумевает наличие некоего внешнего по отношению к формальной грамматике критерия, определяемого целью, которая преследуется при генерации конкретного предложения. В частности, при формировании речевого высказывания исходным является мотив [133, с. 33]. Грамматика языка определяет лишь форму, а не содержание мысли. Содержание же речевого высказывания зависит от мотива, т. е. от решаемой задачи, и не может рассматриваться вне этой задачи. Например, грамматика, описывающая синтаксис некоторого языка программирования, позволяет лишь избегать ошибок компиляции, но написание конкретной программы зависит от ее предназначения. Итак, синтез новых объектов может осуществляться с помощью формальных грамматик, но выходит за их рамки.

Внутренней для формальных грамматик является проблема анализа, или проблема *синтаксического (грамматического) разбора* (parsing). Она заключается в том, чтобы для данного предложения установить, может ли оно быть порождено данной грамматикой и, если может, определить его структуру.

Определение структуры необходимо для машинного понимания высказываний естественного языка или для автоматического перевода. В последнем случае должны быть определены две грамматики, связанные общей глубинной синтаксической структурой предложений. Полученное дерево разбора предложения в грамматике исходного языка задает направление вывода в грамматике объектного языка, что полностью (если не учитывать возможность неоднозначности предложения) снимает проблему выбора из всего множества предложений языка.

Ответ на вопрос о возможности порождения некоторого предложения данной грамматикой важен для распознавания образов. В формальной грамматике с каждым порождаемым ею предложением связывается структура этого предложения посредством ограниченного количества правил. Считая предложения языка символьными описаниями неких объектов, грамматику следует рассматривать в качестве класса объектов со сходной структурой. Тогда син-

таксический разбор соответствует процедуре классификации образов. В п. 2.3.7 мы указывали на некоторые ограничения, возникающие в дискриминантном подходе к распознаванию образов, и приводили пример с разделением классов целых и дробных чисел, которое вряд ли может быть осуществлено в рамках дискриминантного подхода, но легко может быть реализовано в рамках синтаксического подхода. Однако не следует надеяться на создание универсального синтаксического метода распознавания. В частности, при работе с данными количественной природы синтаксические методы будут слишком неэффективны. Отметим, что даже выполнение классификации образов при использовании грамматик достаточно общего вида оказывается нетривиальной задачей.

Задачу синтаксического разбора можно рассматривать, абстрагируясь от конкретного приложения. Как уже отмечалось, для произвольной неограниченной грамматики проблема синтаксического разбора неразрешима, так что использование грамматик этого типа затруднительно. Для грамматик типов 1–3 алгоритмы синтаксического разбора существуют. Однако синтаксический разбор для НС-грамматик, хотя и разрешим, является NP-полной задачей, что ограничивает их применение. Существуют две общие стратегии разбора: сверху вниз и снизу вверх.

При разборе сверху вниз берется начальный символ S и осуществляется перебор возможных последовательностей правил подстановки. Как только порождена искомая цепочка, перебор останавливается.

При разборе снизу вверх преобразуется цепочка, разбор которой нужно осуществить. Правила подстановки при этом инвертируются. В ходе перебора возможных последовательностей инвертированных правил, примененных к исходной цепочке, производится попытка получить начальный символ S.

Оба варианта ненаправленного перебора одинаково неэффективны. Эффективность может быть повышена введением некоторых эвристик, направляющих перебор. Но и в случае направленного перебора задача о возможности порождения данной цепочки неограниченной грамматикой остается неразрешимой, что тесно связано с неразрешимостью проблемы останова. В приведенных стратегиях перебора проблема останова также не решается и для грамматик типов 1–3. Эту сложность можно преодолеть для не-

укорачивающих грамматик: достаточно останавливать перебор для таких промежуточных цепочек вывода, которые имеют длину, превышающую длину разбираемой цепочки (отдельно следует учитывать возможность циклических подстановок, например, $A \to B$, $B \to A$, но и эта проблема может быть решена, что является одним из удобств использования неукорачивающих грамматик).

Приведем пример направленного разбора. Пусть из S в результате применения правил подстановки необходимо получить цепочку $a_1, \ldots, a_n$. Будем искать все правила подстановки вида $\alpha \to a_1\beta$, т. е. такие правила, применение которых может быть причиной появления символа a_1 в начале цепочки. Каждая цепочка α имеет вид $X\alpha'$, где X — некоторый символ. Для каждой цепочки поставим подзадачу: найти все правила, которые в результате могут дать символ X. Получим множество символов, которые могут в процессе вывода оказаться самыми левыми и позволят получить символ a_1 на первой позиции при окончании вывода. Далее ищем все правила вывода $S \to \beta$, где первый символ цепочки β принадлежит найденному множеству. Таким образом, получаем ограничение на первый шаг вывода при переборе сверху вниз. Заметим, что эти ограничения будут корректными только в том случае, если грамматика является неукорачивающей (в частности, если она не содержит правил подстановки, содержащих в правой части пустую цепочку). Далее сходным образом определяются ограничения на правила, применяемые на втором шаге вывода.

Рассмотрим конкретный пример. Пусть дана грамматика G: $V_T = \{a, b, c\}$; $V_N = \{S, A, B, C\}$;

$$P = \begin{Bmatrix} S \to Aab, & S \to aBc, & S \to aCc, & S \to Cac \\ A \to cbB, & A \to aaa, & A \to cC & \\ B \to bbA, & B \to bcC, & C \to ccc, & C \to ba \end{Bmatrix},$$

и пусть требуется осуществить синтаксический разбор цепочки $cbaab$.

• Символ «c» стоит в правой части самым левым в правилах: $A \to cbB$, $A \to cC$, $C \to ccc$.

• Символы «A» и «C» стоят самыми левыми в правых частях правил: $S \to Aab, S \to Cac$.

• На первом шаге вывода из четырех правил могут использоваться только два правила: $S \to Aab, S \to Cac$. Множество правил подстановки, приводящих к символу «c» на

самой левой позиции: $A \to cbB, A \to cC, C \to ccc$, т. е. имеем три возможности продолжения вывода: $S \Rightarrow cbBab, S \Rightarrow cCab$, $S \Rightarrow cccac$.

• Рассматривается возможность вывода второго символа «b». Цепочка $cbBab$ его уже содержит. Цепочка $cccac$ является терминальной и не совпадает с искомой. Для цепочки $cCab$ исследуется возможность получения символа «b» в левой позиции при выводе из начального символа C. Это возможно только при применении правила $C \to ba$.

• Для следующего шага вывода рассматриваются две исходные цепочки: $cbBab$ и $cbaab$. Поскольку вторая цепочка является искомой, разбор останавливается (несложно также убедиться, что для цепочки $cbBab$ получить на третьей позиции символ «a» невозможно).

• Таким образом, решение будет: $S \Rightarrow Aab \Rightarrow cCab \Rightarrow cbaab$.

При направленном разборе происходит также порождение всех вариантов разбора, но ограничения, накладываемые данными (разбираемой цепочкой), используются не только при окончательном сравнении, но и при отсечении вариантов, которые заведомо не ведут к успеху. В рассмотренном примере на первом шаге это были замены $S \to aBc$, $S \to aCc$. При ненаправленном переборе они также были бы рассмотрены и на их основе были бы порождены многие другие предложения, что делает направленный перебор гораздо эффективнее.

Существуют и упрощенные эвристики, применимые для НС-грамматик, например, из перебора можно сразу исключить все правила подстановки, содержащие в правой части терминальные символы, не принадлежащие искомой цепочке. Напротив, могут использоваться и весьма изощренные методы разбора ([375, с. 134–143]). Все проблемы при направленном переборе не решаются. Использование дополнительных способов ограничения перебора позволяет повысить эффективность синтаксического разбора, но все равно он сводится к перебору вариантов.

Принципиальное повышение эффективности достигается при вводе дальнейших ограничений на вид грамматических правил. Для КС-грамматик определенного вида и автоматных грамматик существуют хорошо известные беспереборные процедуры синтаксического разбора.

Поскольку синтаксический разбор может соответствовать процедуре классификации, а некая формальная грамматика — описывать класс образов, следует ожидать, что для

формальных грамматик может быть поставлена задача, аналогичная задаче распознавания образов. Это задача грамматического вывода или восстановления грамматик по обучающей выборке предложений.

4.3. ГРАММАТИЧЕСКИЙ ВЫВОД

4.3.1. Основные определения и постановка задачи

При практическом использовании формальных грамматик для структурного описания естественных и искусственных языков, классов образов или неких источников символьной информации может оказаться полезным автоматическое определение соответствующей грамматики по множеству примеров предложений (цепочек символов). Обучение грамматике на основе примеров предложений называют *восстановлением грамматики* (или *грамматическим выводом*).

Задача грамматического вывода возникает в синтаксическом распознавании образов, а также во многих других приложениях формальных грамматик. В частности, при использовании формальных грамматик для описания структуры естественных языков (для систем речевого общения, машинного перевода, поисковых систем и т. д.) грамматика конструируется в основном вручную, на основе имеющихся лингвистических данных. Использование автоматических методов грамматического вывода может уменьшить стоимость построения грамматик, избавиться от субъективности, сделать более легкой адаптацию системы обработки предложений естественного языка в новую область [376]. Для нас задача восстановления грамматик интересна тем, что она является задачей индуктивного вывода, характерной для символьных методов машинного обучения, и привлечение принципа МДО позволяет улучшить ее решение.

Для формального описания задачи грамматического вывода введем следующие определения.

Информационной последовательностью $I(\Gamma)$ языка Γ будем называть последовательность цепочек, каждая из которых принадлежит одному из множеств $\{\alpha^+ \mid \alpha^+ \in \Gamma\}$ или $\{\alpha^- \mid \alpha^- \in V_T^* \setminus \Gamma\}$ с указанием того, к какому именно множеству принадлежит та или иная цепочка. Последовательность цепочек языка $\{\alpha^+ \mid \alpha^+ \in \Gamma\}$ будем называть *положительной*

информационной последовательностью $I^+(\Gamma)$, а последовательность цепочек из дополнения языка $\{\alpha^- \mid \alpha^- \in V_T^* \setminus \Gamma\}$ — *отрицательной информационной последовательностью* $I^-(\Gamma)$.

Информационная последовательность $I(\Gamma)$ языка Γ называется *полной*, если $I^+(\Gamma)$ содержит все цепочки языка Γ, а $I^-(\Gamma)$ содержит все цепочки, не принадлежащие языку Γ.

Грамматика G *согласована* с грамматикой G_0, если порождаемые ими языки совпадают, т. е. $\Gamma(G) = \Gamma(G_0)$.

Пусть $C = \{G_i\}$ — класс грамматик. Класс языков $\Gamma(C) = \{\Gamma(G) \mid G \in C\}$ называется *идентифицируемым*, если для любой грамматики $G \in C$ и любой полной информационной последовательности $I(\Gamma(G))$ существует некоторое число N и алгоритм, который бы, получая на входе подпоследовательность $I(\Gamma(G))$, содержащую не менее N цепочек, на выходе давал бы грамматику, согласованную с грамматикой G.

Наряду с понятием информационной последовательности используется понятие образца (или выборки) языка Γ. *Образцом* S_t языка Γ будем называть последовательность цепочек $\{\alpha_i\}_{i=1}^t$, для каждой цепочки которой известно, принадлежит ли она языку Γ или его дополнению $V_T^* \setminus \Gamma$. *Положительным образцом* будем называть множество $S_t^+ = S_t \cap \Gamma$, а *отрицательным образцом* — множество $S_t^- = S_t \cap (V_T^* \setminus \Gamma)$.

Грамматика G называется *совместимой* с образцом S_t, если она порождает все положительные примеры этого образца и не порождает ни одного отрицательного примера.

Структурно-полный образец $S_t(\Gamma(G))$ языка $\Gamma(G)$ — это образец, содержащий такие цепочки, при построении которых каждое правило подстановки грамматики G использовалось хотя бы по одному разу.

Структурная полнота образца является необходимым условием возможности восстановления всех правил грамматики, в то время как полнота информационной последовательности может выступать в качестве достаточного условия идентифицируемости некоторых классов языков. На практике структурная полнота образца достигается существенно легче, чем полнота информационной последовательности, но обоснование структурной полноты также возможно далеко не во всех случаях (чтобы убедиться в этом, попробуйте составить структурно полный образец русского языка).

При решении задачи грамматического вывода различают *текстуальное (текстовое) представление*, при котором

имеются лишь положительные примеры, и *информаторное представление*, при котором есть как положительные, так и отрицательные примеры.

Обычно язык является гораздо более узким, чем его дополнение $V_T^* \setminus \Gamma$. Как правило, алгоритмическая сложность предложения $\alpha \in \Gamma$, имеющего определенную структуру, заметно меньше его длины, в то время как алгоритмическая сложность цепочек $\beta \in V_T^* \setminus \Gamma$ близка к их длине, т. е. большинство цепочек, не вошедших в компактно описанный язык Г, являются случайными (см. п. 1.5.5 об индивидуальной случайности бинарной строки). Если отрицательные примеры не подбираются каким-то специальным образом, то они будут в большинстве своем чисто случайными цепочками. Такой отрицательный образец не несет практически никакой полезной информации о восстанавливаемой грамматике. Отрицательные примеры могут быть полезны, только если они и сами обладают некоторой структурой, не описывающейся искомой грамматикой. Такие «хорошие» отрицательные примеры могут возникнуть, когда есть *информатор* — устройство, среда или человек, который относительно каждой цепочки, поданной на вход, сообщает, порождается ли эта цепочка искомой грамматикой или нет.

Информатор потенциально реализует полную информационную последовательность и применяется следующим образом. Машинная система, обучающаяся грамматике, использует гипотезу о грамматике для генерации предложений языка, и информатор указывает, являются ли построенные машинной системой предложения правильными или нет. Если машинная система ошибается и генерирует отрицательные примеры, то эти отрицательные примеры не будут случайными, а будут иметь структуру, определяемую текущей гипотезой о грамматике (т. е. эта структура будет содержать лишь незначительные отклонения от требуемой структуры предложения).

«Хорошие» отрицательные примеры могут строиться и самим информатором (учителем). Как правило, такие отрицательные примеры сопровождаются положительными примерами и должны помочь в выявлении некоего (желательно одного) грамматического правила. Именно такого рода отрицательные примеры используются при обучении детей (например: «Черепахи не умеют *ни* летать, *ни* прыгать» — правильно, «Черепахи не умеют *не* летать, *не* прыгать» — неправильно).

Чаще всего отрицательные примеры используются лишь в качестве ограничений, привлекаемых для отсева грамматик в процессе поиска. Методы конструктивного использования отрицательных примеров (для уточнения отдельных правил) исследованы мало.

Еще одна возможность появления неслучайных отрицательных примеров может возникнуть в синтаксическом распознавании образов, если положительные примеры относятся к одному классу образов, а отрицательные — к другим классам. Здесь, однако, не обосновано введение жесткого ограничения на то, что цепочки, относящиеся к одному классу, не могут порождаться грамматикой другого класса (классы могут пересекаться). В связи с этим отрицательные примеры могут использоваться не для вывода грамматики данного класса образов, а для вывода грамматик других классов. В этом случае для каждого класса используется собственное текстуальное представление.

Таким образом, информаторное представление наиболее характерно для случая, когда машинная система может порождать предложения в рамках текущей гипотезы о грамматике и проверять правильность этих предложений у информатора. Из-за малой изученности конструктивного использования такой информации мы также большее внимание уделим текстуальному представлению. Формирование положительных образцов большого объема более доступно: достаточно взять текстовый корпус, включающий грамотно написанные литературные произведения.

Существует два варианта постановки задачи восстановления грамматик.

В первой формулировке (на которую мы будем ссылаться как на *проблему согласования*) предполагается, что есть некоторая истинная грамматика G_0, и требуется по информационной последовательности построить такую грамматику G, которая была бы согласована с грамматикой G_0, т. е. $\Gamma(G) = \Gamma(G_0)$.

Во второй формулировке (на которую мы будем ссылаться как на проблему грамматического вывода) считается, что по образцу S_t необходимо построить такую грамматику G, которая бы порождала все цепочки положительного образца S_t^+ (и, возможно, бесконечное множество других цепочек) и не порождала цепочки отрицательного образца S_t^- (и, возможно, бесконечное множество других цепочек), т. е. была бы совместима с этим образцом.

Обычно выделяют два класса алгоритмов восстановления грамматик: перечислением и индукцией. Оба класса алгоритмов имеют определенную связь с двумя приведенными формулировками задачи восстановления грамматик.

4.3.2. Восстановление грамматик перечислением

При формулировании проблемы согласования грамматик предполагается существование некой «истинной» грамматики. Целью является нахождение грамматики, согласованной с этой «истинной» грамматикой. Требование согласованности является настолько сильным, что вызывает сомнение возможность достижения этой цели в достаточно общем случае. Действительно, для любой фиксированной (конечной) информационной последовательности всегда существует грамматика ad hoc, которая совместима с этой последовательностью, но которая вряд ли согласована с истинной грамматикой. Возникает вопрос: при каких ограничениях проблема согласования разрешима?

В первых теоретических исследованиях, проведенных Голдом и Фелдманом [377, 378], предполагалось, что даны *полные* информационные последовательности (либо только положительная, либо положительная и отрицательная).

Требование полноты позволяет строить теоремы о существовании алгоритмов восстановления грамматик при различных условиях и давать их формальные доказательства. К сожалению, полных информационных последовательностей на практике не встречается. Доказательства существования алгоритмов обычно неконструктивны (либо в них строятся алгоритмы, непригодные из-за проблемы комбинаторного взрыва). Такие теоремы полезны тем, что с их помощью можно выяснить, для решения каких вариантов задачи грамматического вывода алгоритмы не стоит и искать.

Нас, однако, эти результаты будут мало интересовать. Поясним причину такого, немного пренебрежительного, отношения к теоретическим результатам на основе следующего примера. Известно [121, с. 223], что не существует алгоритма, который бы решал проблему согласования для любой грамматики из класса регулярных грамматик при текстуальном представлении. Более того, если рассматривается класс грамматик, включающий все конечные грамматики и лишь одну бесконечную грамматику, то для этой

грамматики проблема согласования также не может быть решена на основе текстуального представления. Несмотря на такой «суровый» приговор, на практике приходится решать проблемы восстановления грамматик на основе текстуального представления, даже используя неполную положительную информационную последовательность (при этом восстановление «истинной» грамматики, конечно, не гарантируется).

Мы не приводим формулировок конкретных теорем и, тем более, их доказательств, так как в дальнейшем нам эта информация не потребуется (см. работы [377, 378]). Больший интерес для нас представляют сами алгоритмы восстановления грамматик. С упомянутыми теоретическими работами тесно связаны алгоритмы восстановления грамматик перечислением.

При восстановлении грамматик перечислением осуществляется перебор (перечисление) всех грамматик из некоторого заданного класса до тех пор, пока не будет найдена грамматика, согласованная с информационной последовательностью языка. Естественно, для бесконечных языков на практике не может использоваться полная информационная последовательность. В связи с этим используется некоторая конечная ее подпоследовательность. Если находится несколько грамматик, совместимых с такой подпоследовательностью, то размер подпоследовательности увеличивается и происходит дальнейший отсев грамматик. При таком полном переборе «истинная» грамматика находится гарантированно, коль скоро рассматривается идентифицируемый класс языков.

Такое свойство алгоритмов восстановления грамматик не может не являться привлекательным. Однако на практике в полной мере встает проблема комбинаторного взрыва при перечислении грамматик. В связи с этим многие работы по восстановлению грамматик перечислением нацелены на оптимизацию перебора. Одним из базовых приемов является установление наиболее приемлемого порядка перечисления грамматик. Таковым оказывается *оккамовское перечисление* [52, с. 182], при котором грамматики рассматриваются в порядке возрастания сложности.

Эффективность оккамовского перечисления может быть обоснована не только на основе эвристических соображений, но и строго, если обратиться к понятию алгоритмической вероятности. Действительно, при таком перечислении грам-

матики будут рассматриваться в порядке убывания их априорных вероятностей, т. е. «истинная» грамматика будет встречена с большей вероятностью раньше, чем при произвольном перечислении грамматик.

Однако зачастую при применении оккамовского перечисления используется нестрогое понятие сложности. Различают внутреннюю сложность грамматики (которая может задаваться как, например, число нетерминальных символов или число правил вывода) и сложность вывода информационной последовательности (которая может задаваться как, например, суммарное или максимальное число подстановок, использованных при выводе примеров предложений из информационной последовательности).

Следующим шагом в сторону индуктивных методов грамматического вывода является использование сложности грамматики не просто для установления порядка перечисления грамматик, но и в качестве критерия выбора между грамматиками, совместимыми с данной конечной информационной последовательностью. Здесь оказывается принципиальным адекватное задание сложности грамматики. В частности, возникает проблема выбора между минимизацией внутренней сложности и минимизацией сложности вывода. Эти проблемы успешно решаются в индуктивном подходе к грамматическому выводу. Прежде чем перейти к нему, сделаем небольшое отступление.

Отличительной чертой восстановления грамматик индукцией является отсутствие требования к нахождению «истинного» (точного) решения. Хотя алгоритмы восстановления грамматик перечислением и индукцией могут и не содержать фундаментальных различий, эти два подхода принципиально различны с методологической точки зрения. В первом случае отправной точкой служит вопрос о существовании точного решения проблемы при тех или иных ограничениях. Отметим, что проверка удовлетворения этим ограничениям на практике недостижима (понятие идентифицируемости относится к классу языков; но как достоверно определить, к какому классу принадлежит неизвестный язык?). Во втором случае базовым является понятие модели, которая (при описании с ее помощью некоторого феномена реального мира) признается принципиально неточной или имеющей вероятностный характер.

Говоря о практическом применении алгоритмов, мы имеем в виду не чисто прагматический аспект их применения,

а методологический аспект: алгоритм является моделью информационных процессов. Как и любая физическая модель или теория, которая может быть математически корректной, но при этом может не иметь никакого отношения к реальности, алгоритм (не как математическая абстракция, а как модель «физических процессов» — а именно этим становится алгоритм при «погруженности» в физический мир) может быть формально правильным, но не применимым к реальным данным.

4.3.3. Эвристические процедуры грамматического вывода

Рассмотрим задачу грамматического вывода. В этой задаче не обязательно строить грамматику, согласованную с некоторой истинной грамматикой, а достаточно вывести грамматику, совместимую с данным (фиксированным) образцом языка $S_t = \{\alpha_i\}_{i=1}^{t}$. Если для восстановления грамматик перечислением наличие информатора крайне желательно, то при грамматическом выводе чаще ограничиваются текстуальным представлением. Мы также будем рассматривать в основном положительный образец S_t^+.

Для текстуального представления имеем два очевидных тривиальных решения: ad hoc- и «беспорядочную» (promiscuous) грамматики. Первая грамматика допускает только данные цепочки и содержит t правил $\{S \to \alpha_i\}_{i=1}^{t}$, вторая допускает вообще все цепочки (например, содержит правила вида $\{S \to a_i S \mid a_i \in V_T\}; S \to \Lambda$).

Если бы требовалось просто построить грамматику, которая бы порождала данные цепочки, то эта задача решалась бы тривиально. Но такие решения нас интуитивно не устраивают. Читателю сейчас уже должно быть очевидным, как следует расширить постановку задачи грамматического вывода, чтобы решение этой задачи давало результат, который ожидается исходя из здравого смысла. Предположим, однако, что нам неизвестен этот путь, и попробуем разобраться, что же мы хотим от грамматического вывода.

Обратим внимание, что предложенные тривиальные решения обладают следующими особенностями. Первое решение дает грамматику, не порождающую возможные отрицательные предложения, но при этом ни один новый положительный пример, не вошедший в образец, также порож-

даться не будет. Второе решение дает грамматику, порождающую любой новый положительный пример, но также порождающую и любое отрицательное предложение.

Здесь в завуалированном виде присутствует проблема обобщения. Казалось бы, нам нужна любая грамматика, удовлетворяющая набору цепочек. Но зачем нужна эта грамматика? Произвольная грамматика, просто порождающая данные положительные цепочки и не порождающая отрицательных цепочек, будет бесполезна (если информационная последовательность не является полной). Естественно, хотелось бы, чтобы грамматика *предсказывала*, какие новые цепочки возможны, а какие не должны порождаться. В приведенных тривиальных грамматиках отсутствует обобщение и, как следствие, теряется предсказательная сила.

Обучение грамматике по неполной информационной последовательности (или образцу языка) — это типичная задача индуктивного вывода. В приведенной формулировке задачи восстановления грамматики в явном виде не определены два компонента индуктивного вывода — пространство гипотез и критерий рациональности.

Пространство гипотез обычно определяется в каждом конкретном методе восстановления грамматик путем выбора типа грамматики и введением ограничений на вид правил подстановки. Меру качества грамматики-гипотезы «определяют таким образом, чтобы она давала в некотором смысле удовлетворительный результат» [121, с. 217]. Эвристическое определение меры качества грамматики сводится, как правило, к неформальному (и неточному) штрафованию сложности грамматики. Сейчас читателю уже должно быть очевидно, что предсказательная сила будет с наибольшей вероятностью максимальна у самой простой (в смысле количества информации) грамматики. Но прежде чем переходить к рассмотрению теоретико-информационного подхода к восстановлению грамматик индукцией, опишем некоторые классические эвристические методы грамматического вывода.

В отличие от методов восстановления грамматик перечислением в этих методах не осуществляется перебор грамматик, а производится постепенное упрощение грамматики, совместимой с данным образцом языка. Отличие методов грамматического вывода заключается в тех эвристических правилах упрощения, которые используются в том или ином методе.

Рассмотрим типичный эвристический метод грамматического вывода на следующем классическом примере положительного образца [52, с. 194; 120, с. 372; 121, с. 236]:

$$S_7 = \{caaab, bbaab, caab, bbab, cab, bbb, cb\}.$$

На *первом шаге* формируются нетерминальные символы и правила таким образом, чтобы они порождали исходные цепочки. Для первой цепочки можно ввести такие правила:

$$caaab : S \to cA_1,\ A_1 \to aA_2,\ A_2 \to aA_3,\ A_3 \to aA_4,\ A_4 \to b.$$

На основе приведенных правил может быть сформирована цепочка *caaabb*, и только она. Аналогичным образом вводятся правила подстановки для следующих цепочек. Для второй цепочки это будут

$$bbaab : S \to bA_5,\ A_5 \to bA_6,\ A_6 \to aA_7,\ A_7 \to aA_7,\ A_8 \to b.$$

Для третьей цепочки первое правило должно было бы иметь вид $S \to cA_7$, но уже на первом шаге эвристического алгоритма может производиться упрощение грамматики. Это упрощение заключается в том, что вместо введения нового нетерминального символа и нового правила подстановки используется уже введенный символ A_1 с соответствующим правилом $S \to cA_1$. Тогда для порождения третьей цепочки потребуются правила:

$$caab : S \to cA_1,\ A_1 \to aA_2,\ A_2 \to aA_3,\ A_3 \to b,$$

из которых новым будет только правило $A_3 \to b$.

Для оставшихся цепочек формируются следующие правила:

$$bbab : S \to bA_5,\ A_5 \to bA_6,\ A_6 \to aA_7,\ A_7 \to b;$$

$$cab : S \to cA_1,\ A_1 \to aA_2,\ A_2 \to b;$$

$$bbb : S \to bA_5,\ A_5 \to bA_6,\ A_6 \to b;$$

$$cb : S \to cA_1,\ A_1 \to b.$$

С учетом повторяющихся правил получаем следующую систему правил:

$$S \to cA_1,\ S \to bA_5;\quad A_1 \to aA_2,\ A_1 \to b;\quad A_2 \to aA_3,\ A_2 \to b;$$
$$A_3 \to aA_4,\ A_3 \to b;\quad A_4 \to b;\quad A_5 \to bA_6;$$
$$A_6 \to aA_7,\ A_6 \to b;\quad A_7 \to aA_7,\ A_7 \to b;\quad A_8 \to b.$$

На *втором шаге* алгоритма осуществляется редукция (упрощение) сложной грамматики на основе некоторых эвристических правил. Рассмотрим такое правило: объединить (отождествить) такие нетерминальные символы A и B, которые входят в правила подстановки вида $A \to \alpha$, $B \to \alpha$. Под объединением символов A и B понимается такая процедура, при которой вводится новый нетерминальный символ C, и все вхождения символов A и B заменяются этим символом. После объединения оба правила: $A \to \alpha$, $B \to \alpha$ приобретают вид $C \to \alpha$, т. е. число правил уменьшается (возможно также, что и некоторые другие правила после замены каждого из символов A и B на символ C будут тождественными).

Поскольку в построенной грамматике имеются правила подстановки

$$A_1 \to b,\ A_2 \to b,\ A_3 \to b,\ A_4 \to b,\ A_6 \to b,\ A_7 \to b,\ A_8 \to b,$$

то символы $A_1, A_2, A_3, A_4, A_6, A_7, A_8$ объединяются. Заменим все их вхождения символом C. Обычно в процессе грамматического вывода объединение символов осуществляется попарно, но здесь для сокращения записи мы воспользуемся объединением сразу семи символов. После такого объединения в грамматике останутся следующие правила:

$$S \to cC,\ S \to bA_5;\quad C \to aC,\ C \to b;\quad A_5 \to bC\,.$$

В данном случае полученный результат является окончательным, но вполне могло оказаться так, что объединение некоторых нетерминальных символов привело бы к возможности объединения новых нетерминальных символов и процесс редукции грамматики продолжился бы.

Сделаем некоторые замечания по данному примеру.

Рассмотренный грамматический вывод начинается с грамматики ad hoc, совместимой с заданным образцом и порождающей наиболее узкий язык, содержащий только предложения образца. На каждом шаге редукции грамматики порождаемый язык расширяется (или, по крайней мере, не сужается), т. е. происходит постепенное обобщение на основе образца языка. В случае попарного объединения нетерминальных символов смысл этого обобщения прост: классы символов или грамматические категории, которые обозначаются некоторыми нетерминальными символами, объединяются с целью формирования более общих классов и катего-

рий. При этом в процессе вывода текущая грамматика все время остается совместимой с образцом. Эти моменты (использование грамматики ad hoc в качестве нулевого приближения и ее постепенное упрощение с сохранением совместимости с образцом) характерны для большинства эвристических алгоритмов грамматического вывода. Если на каждом шаге вывода производится обобщение, то возникает вопрос о правомерности такого обобщения при выполнении того или иного эвристического условия. Этот вопрос в конечном счете сводится к выработке корректного критерия рациональности в индуктивном выводе.

Нетрудно заметить сходство алгоритмов сегментации, в которых происходили последовательное объединение сегментов (первоначальные сегменты включали по одной точке) и объединения нетерминальных символов. Можно также провести аналогию между таким процессом «улучшения» грамматики и градиентным спуском: в обоих случаях не просматривается все пространство поиска, а улучшенное решение получается из текущего решения в результате небольшой модификации.

При градиентном спуске происходит оптимизация конкретной целевой функции, в то время как в приведенном алгоритме редукция грамматики осуществляется после того, как будет выполнено условие для применения эвристического правила упрощения грамматики. Однако введение критерия качества грамматики оказывается целесообразным и в таких эвристических методах. Этот критерий качества может оказаться полезным, так как на каждом шаге грамматического вывода может существовать несколько способов редукции грамматики (число способов может быть достаточно велико, если используется много эвристических приемов упрощения грамматики). Тогда критерий качества позволяет сделать выбор между этими способами. Как и в случае восстановления грамматик перечислением, критерий качества в эвристических процедурах грамматического вывода является, как правило, нестрогой оценкой сложности грамматики. Очевидно, выбор адекватного критерия должен позволить улучшить эффективность вывода.

Еще одним аспектом эвристических методов грамматического вывода, требующим детального рассмотрения, является выбор операций по преобразованию грамматики в процессе ее редукции. Мы привели пример одной такой операции — объединение двух нетерминальных символов

при выполнении достаточно жесткого условия (если есть правила вида $A \to \alpha$, $B \to \alpha$, то символы A и B следует объединить). Могут быть использованы и другие эвристические правила. Например, применяется такая эвристика [120, с. 374]: если есть правила $A \to ab$, $B \to aC$ и $C \to b$, то нетерминальный символ A может быть заменен символом C, а правило $A \to ab$ удалено. Использование разных наборов эвристик приводит к тому, что путь грамматического вывода оказывается различным (для приведенного образца языка можно сравнить различные выводы [52, с. 195; 120, с. 374–376; 121, с. 237]). При этом нет гарантии, что разные пути вывода приведут к одному и тому же конечному результату — как и в случае градиентного спуска, алгоритм вывода может «застрять» в локальном минимуме. Выбор ограниченного числа эвристических приемов оказывается ненадежным.

Обратим внимание также и на следующее. Исходная грамматика, строящаяся в приведенном примере, является очень частным видом грамматик, а именно автоматной грамматикой. При применении любого правила объединения символов тип грамматики не меняется (если быть более точным, то грамматика может из нерекурсивной превратиться в рекурсивную). Возникает необходимость введения таких операций по преобразованию грамматик, которые бы позволяли расширять тип грамматик (или приводили бы к правилам подстановки нового вида). Такой является, например, операция конструирования. При ее выполнении на основе правил вида $A \to \alpha B$ и $B \to \beta C$ формируется правило вида $A \to \alpha\beta C$; старые правила удаляются только в том случае, если при порождении образца языка они всегда применялись вместе.

К изменению вида правил подстановки приводит еще одна операция — удаление нетерминального символа: если символ A входит в левую часть только одного правила подстановки вида $A \to \alpha$, то все вхождения этого символа в других правилах могут быть заменены цепочкой α, а правило $A \to \alpha$ удалено. В нашем примере удаление символа A_5 приведет к системе правил $S \to cC$, $S \to bbC$, $C \to aC$, $C \to b$. При применении операций конструирования и удаления порождаемый язык не расширяется; операция конструирования позволяет уменьшить сложность вывода, в то время как операция удаления — внутреннюю сложность грамматики, что еще раз ставит вопрос о введении адекватного критерия рациональности для грамматического вывода. Приведенных

операций все еще не хватает, чтобы выйти за рамки КС-грамматик (из-за этого и из-за сложности задачи синтаксического разбора для НС-грамматик при решении задачи грамматического вывода часто ограничиваются восстановлением КС-грамматик).

Итак, эвристические методы восстановления грамматик индукцией являются гораздо более эффективными с вычислительной точки зрения, чем методы восстановления грамматик перечислением. С другой стороны, эти методы не имеют строгого обоснования, так что при их использовании получение не только оптимального, но и сколько-нибудь приемлемого решения не гарантировано. Обоснованию подлежит как критерий оптимальности гипотез, так и алгоритм поиска в пространстве гипотез (эвристические операции преобразования грамматик неявно задают пространство гипотез, которое при восстановлении грамматик перечислением описывалось в явном виде). Выше мы отмечали возможность использования методов эвристического программирования в задачах индуктивного вывода. Рассмотренный здесь «жадный» алгоритм поиска является одним из классических алгоритмов в эвристическом программировании. Могут использоваться и алгоритмы более глубокого исследования пространства поиска.

4.3.4. Байесовский вывод стохастических грамматик

Корректный критерий качества грамматики может быть найден в рамках байесовского подхода. Здесь критерий качества некоторой грамматики G по отношению к данному образцу S_t — это ее апостериорная вероятность $P(G \mid S_t) \sim P(G)P(S_t \mid G)$.

Считая предложения в образце независимыми, имеем

$$P(S_t \mid G) = \prod_{i=1}^{t} P(\alpha_i \mid G). \qquad (4.2)$$

Для обычных порождающих грамматик значения $P(\alpha_i \mid G)$ являются неопределенными, однако в рамках статистического подхода естественно привлечь стохастические грамматики. Вероятность порождения цепочки α_i однозначной стохастической грамматикой G определяется через произведение вероятностей правил подстановки, использующихся при выводе этой цепочки.

Априорные вероятности грамматик $P(G)$ в рамках чисто статистических методов не определяются. Для их задания привлекается внутренняя сложность грамматики, определенная тем или иным образом. Отметим, что правдоподобие $P(S_t \mid G)$ связано со сложностью вывода образца языка в рамках данной грамматики. Таким образом, в байесовском подходе к восстановлению грамматик разрешается дилемма — минимизировать ли внутреннюю сложность грамматики или сложность вывода.

Апостериорная вероятность как критерий качества грамматики может использоваться при восстановлении грамматик как перечислением, так и индукцией. И в том и в другом случае алгоритмы могут быть легко модифицированы с целью использования этого критерия качества, но для этого необходимо для правил вывода уметь назначать вероятности. Это легко делается на основе подсчетов числа применений каждого из правил при выводе предложений образца. При этом максимизируется правдоподобие образца.

Пример. Рассмотрим уже знакомый нам образец языка

$$S_7 = \{caaab, bbaab, caab, bbab, cab, bbb, cb\}$$

и грамматику, содержащую правила

$$S \to cC,\ S \to bA,\quad C \to aC,\ C \to b,\quad A \to bC.$$

Запишем вывод каждой цепочки образца:

$$S \overset{S\to cC}{\Rightarrow} cC \overset{C\to aC}{\Rightarrow} caC \overset{C\to aC}{\Rightarrow} caaC \overset{C\to aC}{\Rightarrow} caaaC \overset{C\to b}{\Rightarrow} caaab;$$

$$S \overset{S\to bA}{\Rightarrow} bA \overset{A\to bC}{\Rightarrow} bbC \overset{C\to aC}{\Rightarrow} bbaC \overset{C\to aC}{\Rightarrow} bbaaC \overset{C\to b}{\Rightarrow} bbaab;$$

$$S \overset{S\to cC}{\Rightarrow} cC \overset{C\to aC}{\Rightarrow} caC \overset{C\to aC}{\Rightarrow} caaC \overset{C\to b}{\Rightarrow} caab;$$

$$S \overset{S\to bA}{\Rightarrow} bA \overset{A\to bC}{\Rightarrow} bbC \overset{C\to aC}{\Rightarrow} bbaC \overset{C\to b}{\Rightarrow} bbab;$$

$$S \overset{S\to cC}{\Rightarrow} cC \overset{C\to aC}{\Rightarrow} caC \overset{C\to b}{\Rightarrow} cab;$$

$$S \overset{S\to bA}{\Rightarrow} bA \overset{A\to bC}{\Rightarrow} bbC \overset{C\to b}{\Rightarrow} bbb;$$

$$S \overset{S\to cC}{\Rightarrow} cC \overset{C\to b}{\Rightarrow} cb.$$

Далее произведем подсчеты применений правил:

$$S \to cC : 4;\ S \to bA : 3;\ C \to aC : 9, C \to b : 7;\ A \to bC : 3.$$

Отсюда получим условные вероятности:

$$P(cC \mid S) = 4/7;\ P(bA \mid S) = 3/7;\ P(aC \mid C) = 9/16;$$
$$P(b \mid C) = 7/16;\ P(bC \mid A) = 1.$$

Теперь несложно определить правдоподобие образца языка в рамках данной грамматики и сравнить его с правдоподобием образца в рамках других грамматик. Очевидно, правдоподобие данных в рамках грамматики ad hoc будет заметно выше. В связи с этим без учета априорных вероятностей грамматик обойтись невозможно.

При обсуждении байесовского подхода к восстановлению грамматик перечислением нельзя не отметить следующий прием. При выполнении перечисления грамматик лучшая найденная грамматика G_0, совместимая с образцом, накладывает ограничение $q = P(G_0)P(S_t \mid G_0)$ на априорные вероятности грамматик, которые имеет смысл рассматривать далее. Действительно, так как для любой грамматики верно $P(G)P(S_t \mid G) < P(G)$, то грамматика с априорной вероятностью $P(G) < q$ будет заведомо хуже уже найденной грамматики. В случае перечисления грамматик в порядке убывания их априорных вероятностей этот прием позволяет установить надлежащий критерий остановки перебора.

Отметим, что апостериорная вероятность гипотезы как критерий ее качества включает лишь вероятности вывода положительных примеров, но никак не учитывает отрицательные примеры. При индуктивном подходе критерий качества полагается достаточным для выбора оптимальной гипотезы. В связи с этим возникает вопрос [52, с. 189]: как корректно и эффективно включить в байесовский вывод информацию об отрицательных примерах, доступных при информаторном представлении? Другими словами, наряду с проблемой априорных вероятностей в байесовском подходе к грамматическому выводу возникают и другие трудности. Все эти трудности могут найти разрешение в теоретико-информационном подходе.

4.3.5. Теоретико-информационный подход к грамматическому выводу

Концепция формальных грамматик Н. Хомского возникла примерно в то же время, что и концепция алгоритмической вероятности Р. Соломонова. Эти две концепции разви-

вались параллельно и оказывали мощное влияние друг на друга. Так, по заявлению самого Р. Соломонова [17], формальные грамматики оказались идеальным средством для осуществления индукции, дававшим представление информации, гораздо более удобное для выявления различного рода регулярностей, чем те представления, которые он использовал до этого. Хотя сам Р. Соломонов гораздо больше интересовался проблемами обучения и индукции, чем проблемами лингвистики, он внес определенный вклад и в развитие формальных грамматик, особенно в исследование проблемы восстановления грамматик. Не удивительно поэтому, что при применении принципа МДО в формальных грамматиках вспоминают Р. Соломонова как основоположника этого подхода (по-другому обстоит дело, например, в области статистического анализа данных, где с принципом МДО больше ассоциируется имя Риссанена, «популяризовавшего» этот принцип в статистическом сообществе).

Рассмотрим применение принципа МДО для грамматического вывода сначала в случае текстового представления. Информационная последовательность или образец языка являются исходными данными D. Грамматика выступает в роли генеративной модели этих данных. Чтобы описать конкретные данные с помощью такой модели, необходимо для каждого предложения указать последовательность применения правил вывода.

Представим, что есть отправитель и получатель сообщения, содержащего закодированный образец языка. Вместо того чтобы передавать сами предложения, передаются порождающие их последовательности грамматических правил (как описание данных в рамках модели). Будем считать, что в генеративной модели при порождении предложений применение последующих правил не зависит от того, какие правила были применены до этого. Тогда каждое грамматическое правило кодируется с помощью уникального кодового слова некоторой длины. Выбор кодовых слов произволен. С точки зрения принципа МДО этот выбор должен осуществляться таким образом, чтобы минимизировать общую длину сообщения.

Рассмотрим пример

$$S_8 = \{aaaaaaaba, aabbbbbbaba, aabbbbaba, aaaba,$$
$$bbbbbbbbaba, bbbbbbaba, bbbbaba, bbaba\}$$

и рассмотрим в качестве модели грамматику, включающую правила

$$G_1: \begin{array}{l} \Pi_{S1}: S \to A;\ \Pi_{S2}: S \to B;\ \Pi_{A1}: A \to aaA;\ \Pi_{A2}: A \to bbB;\ \Pi_{A3}: A \to aaC; \\ \Pi_{B1}: B \to bbB;\ \Pi_{B2}: B \to bbC;\ \Pi_C: C \to aba. \end{array}$$

Вместо передачи последовательности S_8 отправитель передает цепочки правил:

$$\Pi_{S1}\Pi_{A1}\Pi_{A1}\Pi_{A3}\Pi_C;\ \Pi_{S1}\Pi_{A1}\Pi_{A2}\Pi_{B1}\Pi_{B2}\Pi_C;$$
$$\Pi_{S1}\Pi_{A1}\Pi_{A2}\Pi_{B2}\Pi_C;\ \Pi_{S1}\Pi_{A3}\Pi_C;$$
$$\Pi_{S2}\Pi_{B1}\Pi_{B1}\Pi_{B1}\Pi_{B2}\Pi_C;\ \Pi_{S2}\Pi_{B1}\Pi_{B1}\Pi_{B2}\Pi_C;$$
$$\Pi_{S2}\Pi_{B1}\Pi_{B2}\Pi_C;\Pi_{S2}\Pi_{B2}\Pi_C.$$

Поскольку в данном примере в цепочке всегда присутствует только один нетерминальный символ (если вывод еще не окончен), то на каждом шаге вывода могут быть применены только те правила, у которых текущий нетерминальный символ стоит в левой части. Значит, для каждой группы правил можно использовать собственный префиксный код. Пусть

$$\Pi_{S1} = 0;\quad \Pi_{S2} = 1;\quad \Pi_{A1} = 0;\quad \Pi_{A2} = 10;\quad \Pi_{A3} = 11;$$
$$\Pi_{B1} = 0;\ \Pi_{B2} = 1;\ \Pi_C = \lambda.$$

Тогда закодированные цепочки правил будут иметь вид

$$00011, 001001,\ 00101,\ 011,\ 10001,\ 1001,\ 101,\ 11,$$

что заметно короче, чем S_8. По каждому коду можно однозначно восстановить цепочку символов, зная, что вывод начинается с начального символа S.

Отметим, что использование таких кодов, как, например, $\Pi_{A1} = 10;\ \Pi_{A2} = 0;\ \Pi_{A3} = 11$, привело бы к большей длине сообщения. Минимизация длины передаваемого сообщения посредством выбора оптимальных кодовых слов для правил подстановки, очевидно, соответствует построению кодов Хаффмана для каждой группы правил. Таким образом, при привлечении принципа МДО естественным образом возникает эквивалент стохастических грамматик (каждое правило кодируется кодом, длина которого зависит от частоты применения правила).

Сам код при этом строить не обязательно — достаточно воспользоваться оценкой длины кодового слова как ми-

нус логарифм от частоты его встречаемости в сообщении. Для правила $\alpha \to \beta$ это величина $P(\beta \mid \alpha)$. Тогда длину описания некоторого предложения α посредством грамматики G можно оценить как $-\log_2 P(\alpha \mid G)$. Отметим, что разделение предложений, присутствующих в сообщении, осуществляется автоматически: как только новое считанное правило приводит к терминальному предложению, то следующее правило трактуется как начало следующего вывода.

Суммарная длина описания образца S_t в рамках грамматики G будет $P(S_t \mid G) = \prod_{i=1}^{t} P(\alpha_i \mid G)$. Этот результат уже был получен для байесовского подхода. Однако теоретико-информационный подход позволяет лучше понять некоторые моменты. Во-первых, становится ясным, почему при грамматическом выводе стохастические грамматики использовать предпочтительнее. Выше уже отмечалось, что при применении теоретико-информационного подхода к восстановлению грамматик вероятности применения правил возникают с необходимостью, так как передается не последовательность самих правил, применение которых дает искомое предложение, а их коды, длина которых соответствует частоте встречаемости правил. Длинное правило может применяться гораздо чаще, чем короткое, и код его будет короче. Становится понятно, почему рассматриваются условные вероятности — достаточно передавать лишь информацию о том, какое правило на текущем этапе следует выбрать среди правил, которые могут быть применены (если на данном шаге вывода применимо лишь одно правило, то его можно вообще не описывать).

Во-вторых, несколько расплывчатое определение условных вероятностей правил становится более четким через длину кода: правила должны кодироваться так, чтобы по цепочке этих кодов можно было бы однозначно восстановить исходную цепочку. Если грамматика допускает построение структурных деревьев и последовательность, в которой заменяются нетерминальные символы, не имеет значения, то можно считать, что всегда заменяется самый левый нетерминальный символ. Если же такое ограничение не выполняется, то схема кодирования должна быть усложнена (например, помимо закодированного правила может оказаться необходимым передавать место в цепочке, к которому это правило применяется в процессе вывода).

Отметим, что приведенная грамматика G_1 не является однозначной. Например, цепочку *bbbbaba* можно породить следующими последовательностями правил вывода: $\Pi_{S1}\Pi_{A2}\Pi_{B2}\Pi_C$ и $\Pi_{S2}\Pi_{B2}\Pi_{B2}\Pi_C$. Это не мешает применять описанную выше схему кодирования, просто в рамках этой схемы неоднозначные грамматики из-за избыточности описания процесса порождения предложений оказываются менее эффективными, чем однозначные. Такое вольное оперирование способом подсчета вероятностей, на первый взгляд, кажется некорректным. Однако еще раз подчеркнем, что индуктивный вывод всегда осуществляется в рамках некоторого представления, которое выбирается в определенном смысле произвольно: выбор представления является метазадачей по отношению к конкретной задаче индуктивного вывода, и можно говорить не о корректности того или иного представления, а о его эффективности при решении класса задач.

И, в-третьих, в рамках теоретико-информационного подхода проще штрафовать сложность грамматики, связанную с ее априорной вероятностью. Действительно, чтобы получатель сообщения был способен расшифровать закодированный образец языка, он также должен получить и описание грамматики, с помощью которой этот образец закодирован.

Описание грамматики должно включать:

- таблицу кодов для терминальных и нетерминальных символов;
- описание правил подстановки с помощью кодов символов;
- коды правил подстановок, использующиеся для описания информационной последовательности.

Коды символов формируются в соответствии с частотой использования каждого из символов в правилах вывода. Например, в рассмотренном выше примере символы встречаются следующее число раз: S — 2 раза, A — 5, B — 5, C — 3, a — 6 и b — 7 раз. Символ «$\rightarrow$» не кодируется, так как в КС-грамматике левая часть правила подстановки всегда состоит из одного символа (однако для большей точности следовало бы учитывать необходимость кодирования длины каждого правила). Остается определенный произвол в способе описания правил грамматики, например, вместо записи вида $A \rightarrow \alpha_1$, $A \rightarrow \alpha_2$, $A \rightarrow \alpha_3$, ... можно воспользоваться записью $A \rightarrow \{\alpha_1 \mid \alpha_2 \mid \alpha_3 ...\}$. В соответствии с частотами символов строится код Хаффмана, определяются дли-

ны таблицы кодов символов и описания правил подстановки. Для простоты можно считать все символы равновероятными, тогда длину описания собственно грамматики можно оценить как $L(G) \approx \log_2 \left(|V_T| + |V_N|\right) L_P$, где L_P — суммарное число символов, использованных при описании правил.

В нашем примере $L_P = 28$; $|V_T| + |V_N| = 6$; $L(G) \approx 72$ бита, в то время как длина описания самой информационной последовательности в рамках данной грамматики равна $L(S_8 \mid G) = 33$ бита. Если не учитывать слагаемые второго порядка малости (длины описания таблиц перекодировки символов и правил подстановки), то суммарную длину описания можно грубо оценить как $L(S_8 \mid G) + L(G) \approx 105$ бит.

Рассмотрим несколько альтернативных грамматик.

• Грамматика ad hoc будет содержать восемь равновероятных правил $S \to \alpha_i$. Каждое из восьми предложений будет кодироваться тремя битами, т. е. $L(S_8 \mid G) = 24$ бита. С помощью трех символов (двух терминальных и одного начального) записываются восемь правил, содержащих 74 символа, т. е. $L(G) \approx 74 \log_2 3 \approx 117$ бит. $L(S_8 \mid G) + L(G) \approx 141$.

• «Беспорядочная» грамматика содержит три правила — $S \to aS$, $S \to bS$, $S \to \lambda$, которые полагаются равновероятными. Для их описания требуется $L(G) \approx 7 \log_2 3 \approx 11$ бит. Предложение длины n порождается путем применения $n + 1$ таких правил, т. е. $L(S_8 \mid G) \approx 74 \log_2 3 \approx 117$ бит; $L(S_8 \mid G) + L(G) \approx 128$.

• Ad hoc- и «беспорядочная» грамматики практически совпадают по результирующей длине описания, но в первом случае основной вклад вносится длиной описания грамматики, а во втором — длиной описания исходных примеров в рамках грамматики. Эти грамматики представляют два крайних случая: чрезмерная конкретизация и чрезмерное обобщение. Наилучшая грамматика должна быть компромиссом между этими двумя случаями.

• Приведенная в качестве примера грамматика может быть изменена без изменения языка, порождаемого этой грамматикой. В частности, могут быть удалены символ C и правило $C \to aba$, тогда изменятся два других правила: $\Pi_{A3} : A \to aaaba$, $\Pi_{B2} : B \to bbaba$. В данном случае значение $L(S_t \mid G)$ не изменяется, но изменяется $L(G)$. Если грубо оценивать $L(G) \approx \log_2 \left(|V_T| + |V_N|\right) L_P$, то величина $|V_T| + |V_N|$ при этом уменьшается, а множитель L_P может как уменьшаться, так и увеличиваться в зависимости от того, насколько часто удаляемый символ использовался в других пра-

вилах. В данном случае удаление символа C приводит к уменьшению длины описания грамматики. Казалось бы, с точки зрения порождения языка символ C был бесполезен, так как он всегда заменялся одной и той же цепочкой символов, т. е. не выполнял никакой конструктивной функции. Однако если представить, что во многих правилах встречается одна и та же (достаточно длинная) цепочка символов, то отведение под эту цепочку нового нетерминального символа приведет не только к уменьшению длины описания, но и к формированию новой грамматической категории, которой можно будет оперировать на более абстрактном уровне.

• Для приведенного примера можно предложить и более общую грамматику, порождающую более широкий язык. Примером ее может служить грамматика G_2: $S \to aaS$, $S \to bbS$, $S \to aba$. Для нее $L(G) \approx \log_2 \left(|V_T| + |V_N|\right) L_P \approx 12 \log_2 3 \approx 19$. Поскольку $P(aaS \mid S) = 6/29$; $P(bbS \mid S) = 15/29$; $P(aba \mid S) = 8/29$, то $L(S_8 \mid G) \approx 43$ бита. Хотя длина $L(S_8 \mid G)$ несколько увеличилась, за счет значительного уменьшения длины $L(G)$ эта грамматика оказывается предпочтительнее рассмотренной исходной. Увеличение длины $L(S_t \mid G)$ обычно свидетельствует о расширении языка. Действительно, если первоначально порождался язык $(aa)^n (bb)^m aba$, то теперь в языке цепочки aa и bb могут произвольно чередоваться; $L(S_8 \mid G) + L(G) \approx 62$.

• Рассмотрим еще одну грамматику G_3: $S \to aaS$, $S \to B$, $B \to bbB$, $B \to aba$. $L(G) \approx 14 \log_2 4 = 28$. Для нее частоты правил будут: $P(aaS \mid S) = 6/14$; $P(B \mid S) = 8/14$; $P(bbB \mid B) = 15/23$; $P(aba \mid B) = 8/23$. Тогда $L(S_8 \mid G) \approx 35$ бит; $L(S_8 \mid G) + L(G) \approx 63$. При учете длины описания таблицы перекодировки для правил вывода эта грамматика сравнивается с предыдущей грамматикой (с точностью до целых битов). В то же время эти две грамматики порождают весьма разные языки. Достоверный выбор между ними может быть сделан только путем расширения информационной последовательности.

• Добавим в образец языка цепочку $aaaabbaba$. Для грамматики G_2 вероятности правил станут $P(aaS \mid S) = 8/33$; $P(bbS \mid S) = 16/33$; $P(aba \mid S) = 9/33$, а $L(S_9 \mid G_2) \approx 50$. Вероятности правил грамматики G_3 изменятся таким образом: $P(aaS \mid S) = 8/17$; $P(B \mid S) = 9/17$; $P(bbB \mid B) = 16/25$; $P(aba \mid B) = 9/25$, а $L(S_9 \mid G_3) \approx 40{,}5$. Таким образом, грамматика G_3 станет несколько предпочтительнее, чем G_2. Различие меж-

ду ними будет и дальше увеличиваться при добавлении новых предложений в образец языка, коль скоро совместимость этих грамматик с образцом не будет нарушена.

Отметим, что использование формальных грамматик для представления информации само по себе задает некоторое распределение априорных вероятностей в пространстве гипотез. Между моделями G_2 и G_3, описанными в рамках какого-то другого формализма, могла бы быть заметная разница. В рамках самих формальных грамматик также обычно рассматриваются лишь узкие классы грамматик, что «смещает» априорное распределение вероятностей. В частности, здесь рассматривались лишь КС-грамматики. Для грамматик других типов необходимо использовать другие схемы кодирования, и, возможно, будет существовать грамматика, которая будет эффективнее всех рассмотренных. В этом смысле полученные результаты индуктивного вывода нельзя абсолютизировать — их всегда нужно рассматривать в контексте более общей задачи. К счастью, при применении формальных грамматик обучающая выборка обычно достаточно большая (хотя могут быть и исключения, например, при исследовании мертвых языков), т. е. для истинной грамматики длина описания $L(S_t \mid G)$ заметно превышает собственную сложность грамматики $L(G)$, в задании которой присутствует наибольший произвол.

Теоретико-информационный подход удобен для всякого рода расширений исходного представления. Важным на практике расширением постановки задачи грамматического вывода является допущение возможности того, что предложения образца языка содержат ошибки. Если для предложений с ошибками будет строиться формальная грамматика, то она также будет позволять порождать предложения с ошибками. Однако можно ввести представление, в котором ошибки будут описываться отдельно. Как и в п. 3.3.4, можно ввести ошибки удаления, замещения, вставки и перестановки символов. Тогда по каналу связи вместо описания исходных предложений будет передаваться описание исправленных предложений (в виде последовательности грамматических правил), сопровождающихся описанием допущенных в них ошибок. Это позволит существенно упростить модель (грамматику), так как от нее будет требоваться порождение только правильно построенных предложений с четкой структурой. Мы не будем подробно разбирать этот вариант задачи грамматического вывода, а лишь заметим,

что в рамках байесовского подхода подсчет апостериорной вероятности грамматики оказался бы весьма запутанным, в то время как в рамках теоретико-информационного подхода определить, как именно следует вычислять длину описания, существенно проще.

Принцип МДО позволяет ввести критерий качества грамматики при данном образце языка. Этот критерий может использоваться как при восстановлении грамматик перечислением, так и в эвристических процедурах грамматического вывода. В последнем случае эвристические правила по преобразованию грамматик могут быть заменены более строгим анализом. Например, в п. 4.3.3 были упомянуты такие условия слияния нетерминальных символов:

• если есть продукции вида $A \to \alpha$, $B \to \alpha$, то символы A и B должны быть объединены;

• если есть продукции вида $A \to ab$, $B \to aC$ и $C \to b$, то символы A и C должны быть объединены.

Эти и другие условия можно заменить одним:

• если в результате слияния нетерминальных символов уменьшается общая длина описания $L(S_t \mid G) + L(G)$, то эти символы должны быть объединены.

Простое применение эвристических правил (при достаточном их разнообразии) без проверки критерия качества получающейся при этом грамматики может приводить к чрезмерному обобщению. Представим, например, грамматику, содержащую (помимо прочих) продукции: $A \to$ *до*, $A \to$ *от*, $A \to$ *перед* и $C \to$ *до*, $C \to$ *ми*, $C \to$ *ля*. Поскольку имеются продукции $A \to$ *до* и $C \to$ *до*, то символы A и C в каком-либо эвристическом методе грамматического вывода могут быть объединены. Однако видно, что символы A и C могут трактоваться как различные категории, которые лишь частично перекрываются по включенным в них цепочкам символов. При решении вопроса об объединении нетерминальных символов на основе длины описания учитываются все правила, в которые эти символы входят, а также учитывается то, сколько раз эти правила использовались при порождении образца языка.

Помимо операции объединения нетерминальных символов могут использоваться и другие операции преобразования грамматик. Выше мы отмечали возможность использования таких операций, как удаление символа и конструирование. Эвристические правила, устанавливающие условия применения той или иной операции, могут быть существен-

но ослаблены (полное их удаление на практике приводит к существенному замедлению процедуры поиска оптимальной грамматики), если окончательное решение о применении правила принимается на основе критерия длины описания. Вопрос о выборе самих операций все еще остается открытым.

Итак, при восстановлении грамматик индукцией исходно конструируется некоторая ad hoc- или «беспорядочная» грамматика, которая заведомо совместима с образцом языка. Далее производится последовательное улучшение этой грамматики: на каждом шаге выбирается такая операция, применение которой дает наибольший прирост критерия качества. Как и при решении задачи сегментации, грамматический вывод можно сначала осуществлять для узкого пространства моделей, т. е. типа грамматик (например, для автоматных), а затем производить постепенное расширение пространства моделей (осуществлять добавление правил продукции нового вида), улучшая ранее полученное решение.

Мы описали лишь два класса процедур поиска при грамматическом выводе: поиск перечислением грамматик и поиск «жадным» алгоритмом. На данный момент весьма популярны альтернативные подходы на основе имитации отжига, эволюционного программирования, генетических алгоритмов (см., например, [376, 379–382]) и других стохастических оптимизационных методов. К сожалению, описание этих методов поиска выходит за рамки данной книги.

4.3.6. Некоторые замечания о восстановлении грамматик при информаторном представлении

Как уже отмечалось, в байесовском подходе возникает некоторая сложность при учете отрицательных примеров в критерии качества грамматики. В рамках теоретико-информационного подхода появление отрицательных примеров тут же ставит вопрос о способе их представления (или о *модели источника отрицательных примеров*, которая должна зависеть от типа источника). Поскольку в «данных наблюдений» (образце языка) присутствует информация о том, является ли пример положительным или отрицательным, то эту информацию тоже нужно «объяснить» в рамках модели.

Возвращаясь к рассуждениям с точки зрения передачи информации между отправителем и получателем сообщения, приходим к выводу, что должны передаваться как положительные, так и отрицательные примеры, а вместе с ними — информация о том, какие примеры являются положительными, а какие — отрицательными. Передаваемые отрицательные примеры должны быть закодированы. Это может быть сделано, например, путем построения для них собственной грамматики. Если априорно известно, что отрицательные примеры являются чисто случайными, то грамматика для них будет «беспорядочной» и ее восстановление производить не нужно.

Здесь уместно провести аналогию с распознаванием образов. Если есть две грамматики («положительная» и «отрицательная»), то это соответствует ситуации с двумя классами образов. Поскольку в образце языка указано, какие предложения являются положительными, а какие — отрицательными, то задача восстановления этих двух грамматик соответствует распознаванию образов, а не группированию. После восстановления двух грамматик каждый пример может классифицироваться на основе того, в рамках какой грамматики его описание будет более коротким. Это позволит определять, является ли предложение правильно построенным или нет. Не составляет труда сформировать информационную целевую функцию для этой задачи: отправитель должен закодировать описание двух грамматик, образец языка, описанный с их помощью, а также ошибки классификации (эти же компоненты целевой функции вводились в п. 2.3.6).

При таком подходе «положительной» грамматике не запрещается генерировать отрицательные примеры, если это помогает сделать грамматику проще. Установление того, правильно ли построено предложение, осуществляется в результате выполнения классификации.

Описание отрицательных примеров с помощью собственной грамматики имеет смысл только тогда, когда эта грамматика никак не связана с грамматикой, восстанавливаемой на основе положительных примеров. Если же они как-то связаны (а именно этот случай более интересен), то их описания должны иметь нечто общее, т. е. они должны обладать положительной взаимной информацией. Таким образом, вместо двух грамматик имеет смысл строить одну грамматику, содержащую правила трех типов: положительные,

отрицательные и нейтральные. Можно ограничиться только двумя типами правил: положительным и отрицательным (если при выводе предложения используется хотя бы одно отрицательное правило, то порожденное предложение считается отрицательным). При этом представление грамматик немного расширяется, но по последовательности правил автоматически определяется, является ли генерируемое предложение положительным или отрицательным (а предложений в образце обычно существенно больше, чем правил в грамматике), так что эту информацию дополнительно передавать не нужно. Здесь, опять же, не представляет трудности сформировать целевую функцию на основе длины описания. Конечно, при этом нерешенным остается вопрос об алгоритме восстановления таких грамматик.

Рассмотрим следующий пример, не требующий подсчета значений целевой функции. Возьмем уже знакомый положительный образец языка $S_7 = \{caaab, bbaab, caab, bbab, cab, bbb, cb\}$ и грамматику, содержащую следующие правила: $S \to cC$, $S \to bC$, $C \to aC$, $C \to b$, $C \to bC$. На основе этой грамматики может быть порождена цепочка $bbbab$, о которой информатор сообщает, что она построена неверно. На основе анализа других имеющихся образцов может быть обнаружено, что

$$bbab^{+}S \Rightarrow bC \Rightarrow bbC \Rightarrow bbaC\ldots;$$

$$bbb^{+}S \Rightarrow bC \Rightarrow bbC \Rightarrow bbb;$$

$$bbbab^{-}S \Rightarrow bC \Rightarrow bbC \Rightarrow bbbC\ldots,$$

т. е., вероятно, правило $C \to bC$ в последнем случае было применено некорректно. Поскольку это правило применялось при порождении других положительных образцов, то оно не может быть просто объявлено отрицательным, а должно быть разделено на два правила. Для этого требуется ввести новый нетерминальный символ A и правило $A \to bC$, которым следует заменить все вхождения правила $C \to bC$. Поскольку правило $C \to bC$ следует после правила $S \to bC$, то необходимо ввести еще одно правило — $S \to bA$. В итоге правила грамматики приобретают уже знакомый вид: $S \to cC$, $S \to bA$, $C \to aC$, $C \to b$, $A \to bC$ (правило $S \to bC$ может быть исключено, так как после введения правила $S \to bA$ оно больше нигде не используется). Эти положительные правила дополняются одним отрицательным правилом — $C \to bC$, которое не должно применяться.

В приведенном примере потребовалась операция расщепления нетерминального символа. Она оказалась успешной благодаря тому, что осуществлялась вручную. При автоматическом расщеплении нетерминального символа могут возникнуть определенные трудности, потребуется достаточно сложный алгоритм, осуществляющий дублирование всех правил, в которые входит расщепляемый символ (с заменой этого символа новым символом), с последующим отсевом всех лишних правил. Поскольку такая операция будет соответствовать сужению порождаемой грамматики, то при таком отсеве будет необходимо следить за тем, чтобы грамматика продолжала оставаться совместимой с образцом языка.

В данном примере можно проще учесть отрицательный пример, если разрешены контекстно-зависимые правила. Тогда достаточно ввести отрицательное правило вида $bbC \rightarrow bbbC$ (грамматика при этом получается не только типа 2, но и неоднозначной). Такой способ крайне удобен для ввода исключений из правил, но не позволяет произвести корректную конкретизацию грамматики. Естественно, все возможные ситуации при появлении новых отрицательных примеров не ограничиваются вышерассмотренной.

На приведенном примере видно, что использование отрицательных примеров в случае информатора может оказаться весьма сложной задачей. Хотя принцип МДО и не дает непосредственного решения этой проблемы, он позволяет ее корректно поставить. К сожалению, более глубокое исследование информаторного представления выходит за рамки данной книги. Мы привели краткое описание этой проблемы, чтобы продемонстрировать силу теоретико-информационного подхода в методологическом плане: он позволяет корректно строить критерии качества (т. е. ставить оптимизационную задачу) в таких запутанных случаях, как информаторное представление, когда непонятно, как выражать апостериорную вероятность гипотезы (и что считать гипотезой и для каких данных) в случае байесовского подхода.

4.4. ПРИЛОЖЕНИЯ МЕТОДОВ ВОССТАНОВЛЕНИЯ ГРАММАТИК НА ОСНОВЕ ПРИНЦИПА МДО В АНАЛИЗЕ ЕСТЕСТВЕННЫХ ЯЗЫКОВ

4.4.1. Краткое сравнение формальных грамматик с моделями языка на основе *N*-грамм

При рассмотрении проблем распознавания речи мы обращались к понятию *N*-грамм, на основе частот которых производилось описание закономерностей в чередовании фонем, морфов, словоформ и т. д. *N*-граммы являются весьма популярными в качестве основы для «моделей языка» при построении практических систем анализа речи и распознавания рукописного или печатного текста. При этом они явно малопригодны для решения многих других задач вычислительной лингвистики, таких как машинный перевод, автоматическое реферирование текстов, задач, требующих элементов понимания языка или более глубокого знания его структуры. В то же время на более полное описание структуры языка претендуют формальные грамматики. Посмотрим, какая существует взаимосвязь между *N*-граммами и формальными грамматиками.

Существенным отличием *N*-грамм от формальных грамматик является то, что первые представляют дескриптивную модель, а вторые — генеративную. Этим и вызвано несколько большее удобство использования *N*-грамм в задачах интерпретации лингвистических объектов. Однако формальные грамматики являются гораздо более мощным средством, а *N*-граммы можно представить как их частный случай.

Действительно, эквивалентной модели *N*-грамм будет следующая стохастическая грамматика:

$$V_T,\ V_N = \{S\},$$

$$P_S = \left\{ a_1 a_2 \ldots a_{n-1} S \xrightarrow{P(a_n | a_1 a_2 \ldots a_{n-1})} a_1 a_2 \ldots a_{n-1} a_n S \mid a_i \in V_T, n = 1, \ldots, N \right\}. \quad (4.3)$$

Отметим, что в этой грамматике нет дополнительных нетерминальных символов (обозначающих некоторые грамматические категории или классы цепочек символов), т. е., по сути, какое-либо обобщение (в индуктивном смысле) от-

сутствует. Это также хорошо видно из того, что грамматика, соответствующая модели N-грамм, получается крайне громоздкой, если в нее включить все возможные N-граммы.

Заметим, что такая стохастическая грамматика

$$V_T,\ V_N = \{S\},$$

$$P_S = \left\{ S \xrightarrow{P(a_1 a_2 \ldots a_n | S)} a_1 a_2 \ldots a_n S \mid a_i \in V_T,\ n = 1,\ \ldots,\ N \right\} \tag{4.4}$$

не будет являться вполне корректным эквивалентом модели N-грамм. Действительно, представим себе, что есть язык с двумя символами $V_T = \{a, b\}$, вероятности биграмм — $P(aa) = P(bb) = 0{,}5$, $P(ab) = P(ba) = 0$, а вероятности отдельных символов — $P(a) = P(b) = 0{,}5$. Первой грамматикой вида (4.3), содержащей правила $aS \to aaS$ и $bS \to bbS$, будут порождаться цепочки только из символа a или только из символа b. Вторая же грамматика, содержащая правила $S \to aaS$ и $S \to bbS$, может генерировать также такие цепочки: $aabbaaaabbaabbbb$. Очевидно, вероятности, рассчитанные для биграмм ab и ba на основе таких цепочек, будут отличны от нуля, хотя порождаемый язык содержит только цепочки из чередующихся биграмм aa и bb. Это говорит о том, что N-грамме не соответствует цепочка из N символов как отдельный объект. Вместо этого вероятность N-граммы говорит о вероятности появления некоторого символа вслед за уже существующими $N - 1$ символами. В связи с этим модели на основе N-грамм часто отождествляют с моделями на основе марковских цепей.

Отметим также, что частоты N-грамм фонем, морфем или слов можно посчитать, только когда они (фонемы и т. д.) выделены как отдельные объекты. Если, к примеру, не дан априори список морфем, то мы не можем посчитать частоты их N-грамм. Если речь является слитной и неизвестны границы слов, то их N-граммы также нельзя посчитать и отдельно приходится решать сегментацию слитной речи на слова. Несложно представить себе расширение грамматики (4.4), позволяющей одновременно описывать морфемы, слова и фразы. Для этого необходимо добавить правила вида $S \xrightarrow{P(A_i | S)} A_1 A_2 \ldots A_n S$, где нетерминальные символы A_i могут соответствовать языковым структурам разного уровня (например, если A_i — слово, то его структура

описывается через другие нетерминальные символы — морфемы: $A_i \to B_1 B_2 \ldots B_m$, которые, в свою очередь, уже выражаются через терминальные символы: $B_j \to a_1 a_2 \ldots a_l, a_k \in V_T$).

Восстановление подобной грамматики позволит одновременно определять морфемы, слова и словосочетания, выделяя вместе с тем в слитном тексте границы этих объектов. Безусловно, эта задача является весьма сложной с вычислительной точки зрения. Здесь мы, однако, хотим лишь показать преимущество формальных грамматик в качестве представления информации.

Часто в дополнение к N-граммам самих символов привлекаются N-граммы классов, к которым эти символы относятся (например, определяются N-граммы частей речи). При этом классы обычно конструируются вручную. Формальные грамматики позволяют описать и грамматические классы, соотнеся с ними некоторые нетерминальные символы, и устойчивые N-граммы этих классов, что даст описание структуры предложений. Для задания класса символов $\{X_i \mid X_i \in V\}$ необходимо ввести новый нетерминальный символ Y и совокупность правил $\{Y \to X_i\}$. При восстановлении формальных грамматик соответствующие классы можно надеяться определять автоматически.

Отметим, что формальные грамматики позволяют описывать «грамматические классы», включающие символы разных уровней (фонемы или буквы, морфемы, слова и т. д.), а также цепочки, состоящие из терминальных и нетерминальных символов различных уровней. Нужно ли вводить такое достаточно жесткое разделение на уровни, приписываемое естественному языку, в формальные грамматики априори или же оно может быть получено автоматически, при восстановлении грамматики более общего вида, допускающего смешанные классы и цепочки? Отметим также возможность последовательного (не одновременного) выявления морфем, слов, фраз, а также их классов, возможно, с коррекцией результатов более ранних этапов после выполнения более поздних этапов анализа. Видимо, выделение различных уровней лингвистических объектов необходимо, по крайней мере, в целях построения эффективных процедур восстановления грамматик.

Итак, формальные грамматики могут использоваться для существенного расширения моделей языков на основе N-грамм. Рассмотрим несколько примеров того, как это осуществляется.

4.4.2. Обучение фразам

Рассмотрим сначала задачу обучения фразам, имеющую непосредственное отношение к проблеме восстановления синтагматической структуры речи. Эта задача может быть сформулирована как задача восстановления грамматик вида (4.4). Воспользуемся уже привычной схемой: сначала сконструируем некоторую «беспорядочную», или ad hoc-грамматику, а затем будем ее улучшать (в смысле длины описания) путем последовательных преобразований множества правил вывода.

Поскольку в грамматиках вида (4.4) отсутствуют дополнительные нетерминальные символы, то и исходная грамматика должна быть соответствующего вида. Для образца языка $S_t = \{\alpha_i\}_{i=1}^{t}$ в качестве исходной можно выбрать грамматику ad hoc, содержащую правила $P = \{S \to \alpha_i\}_{i=1}^{t}$, а далее разбивать эти правила на правила вида $S \to \alpha_i' S$, $S \to \alpha_i'' : \alpha_i' \alpha_i'' = \alpha_i$. Другой способ восстановления грамматики заключается в том, чтобы начинать с «беспорядочной» грамматики, содержащей правила $P = \{S \to aS \mid a \in V_T\}$, и выполнять операцию конструирования: из двух правил $S \to \alpha S$ и $S \to \beta S$ формировать правило $S \to \alpha\beta S$. Операция конструирования может быть выполнена для любых двух правил, однако она будет иметь смысл только тогда, когда результирующее правило $S \to \alpha\beta S$ может быть использовано много раз при порождении данного текстового корпуса (иными словами, сочетание слов $\alpha\beta$ встречается достаточно часто в тексте).

По существу, второй вариант поиска фраз ничем не отличается от теоретико-информационного метода выделения устойчивых сочетаний фонем, рассмотренного в п. 3.3.6. Действительно, выполнение операции конструирования соответствует введению нового символа в алфавит морфов. В п. 3.3.6 мы приводили ссылку на работу, в которой выявление морфов осуществляется путем последовательной декомпозиции слов, что соответствовало бы первому из приведенных выше вариантов грамматического вывода, начинающегося с грамматики ad hoc. Сходными у этих методов будут и целевые функции. Несущественное различие будет состоять в слагаемом, соответствующем длине описания модели (в одном случае — грамматики, в другом случае — алфавита или словаря), так как используются несколько различные представления информации. Более значимые отличия будут в вычислительной эффективности: без введения

дополнительных эвристик поиск наилучшей пары правил для выполнения операции конструирования может быть весьма ресурсоемок.

На практике для решения задачи обучения фразам формальные грамматики практически не используются; вместо этого понятие фразы вводится априорно в явном виде. Тем не менее рассмотрение этой задачи как задачи восстановления грамматик позволяет определить те ограничения, которые используются здесь для представления информации. По форме записи правил грамматики (4.4) можно заключить, что при решении задачи обучения фразам абсолютно не учитываются контекст фразы, а также категории слов, составляющих фразу. Может быть поставлена задача выявления структуры фраз, для чего слова в текстовом корпусе заменяются символами, обозначающими соответствующие словам части речи (соотнесение слов и соответствующих им частей речи обычно выполняется вручную). В остальном задача выявления структуры фраз и предложений ничем не отличается от задачи обучения фразам, и для ее решения также может привлекаться принцип МДО [383–386].

Мы реализовали алгоритм обучения фразам на основе принципа МДО и протестировали его на небольшом текстовом корпусе, включающем несколько классических художественных произведений (общее число символов — 4,5 млн). При этом в качестве терминальных символов использовались словоформы (разные формы одного и того же слова считались различными). Различные частицы, союзы, предлоги и т. д. из рассмотрения не исключались (использовалась минимальная информация о языке).

Ниже приводится несколько цепочек слов, полученных в результате операций конструирования, которые были выполнены первыми и принесли наибольший выигрыш в длине описания:

«может быть» ($n_1 = 994, n_2 = 1326, n_{12} = 522$)
«как будто» ($n_1 = 6171, n_2 = 615, n_{12} = 546$)
«о том» ($n_1 = 2339, n_2 = 868, n_{12} = 454$)
«к нему» ($n_1 = 4110, n_2 = 371, n_{12} = 369$)
«потому что» ($n_1 = 909, n_2 = 12\,577, n_{12} = 588$)
«тотчас же» ($n_1 = 492, n_2 = 3555, n_{12} = 347$)
«то что» ($n_1 = 2879, n_2 = 11\,989, n_{12} = 729$)
«у него» ($n_1 = 2571, n_2 = 1410, n_{12} = 366$)
«об этом» ($n_1 = 558, n_2 = 684, n_{12} = 227$)
«с ним» ($n_1 = 7791, n_2 = 778, n_{12} = 386$)

Здесь n_1 и n_2 — число вхождений в текстовый корпус первого и второго слова соответственно; n_{12} — число вхождений цепочки этих слов.

Также на первых шагах алгоритма были получены такие цепочки, как «несмотря на», «как бы», «если бы» и т. д. Помимо таких шаблонов в качестве устойчивых сочетаний слов были выделены имена героев взятых литературных произведений, например, «Сергей Иванович», «Катерина Ивановна».

Интересен большой выигрыш в длине описания от введения таких сочетаний слов, как «к нему», «у него», «с ним». Выгодность введения правила подстановки, порождающего форму некоторого слова (или словосочетание) вместе с примыкающим предлогом, еще более явно видна на следующих примерах:

«в тайне» ($n_2 = 4$, $n_{12} = 4$)
«в частности» ($n_2 = 4$, $n_{12} = 4$)
«в [здравом рассудке]» ($n_2 = 4$, $n_{12} = 4$)
«в [трех шагах]» ($n_2 = 4$, $n_{12} = 4$)
«в [первые минуты]» ($n_2 = 4$, $n_{12} = 4$)
«на холме» ($n_2 = 7$, $n_{12} = 4$)
«на западе» ($n_2 = 3$, $n_{12} = 3$)
«на морозе» ($n_2 = 3$, $n_{12} = 3$)
«на левом» ($n_2 = 7$, $n_{12} = 4$)
«в левом» ($n_2 = 7$, $n_{12} = 3$)
n («в») = 15 593, n («на») = 9675

Здесь выигрыш от операции конструирования незначительный, но устойчиво положительный. Видно, что словоформы «западе» или «морозе» (а также многие другие) в нашем текстовом корпусе встречаются только после предлога «на», а словоформы «тайне», «частности» — только после предлога «в». Словоформа «левом» встретилась только после предлогов «на» и «в», в связи с чем на ее основе образовались два различных правила, порождающие сразу цепочки «на левом» и «в левом», а исходный символ «левом» исчез из текстового корпуса как самостоятельный элемент. Если бы рассматривался слитный текст, не разделенный на слова, то было бы сложно отличить приставки от предлогов (между ними есть определенная разница, в частности, предлоги связаны с падежами слов, в то время как приставки могут быть присоединены к различным формам одного и того же слова). Это говорит о том, что разделение на уровни — морфемы, слова, фразы — не вполне четкое.

Большой суммарный выигрыш в длине описания был получен от введения правил порождения таких сочетаний слов, как

«по крайней мере» ($n = 111$)
«в самом деле» ($n = 100$)
«несмотря на то что» ($n = 72$)
«то же время» ($n = 70$)
«для того чтобы» ($n = 63$)
«во всяком случае» ($n = 30$)
«точно так же» ($n = 42$)
«с тех пор» ($n = 58$)
«в то время как» ($n = 49$)

Они действительно могут быть охарактеризованы как неделимые фразы. Вместе с тем среди них оказались такие сочетания слов, как, например:

«я не могу» ($n = 130$)
«я не понимаю» ($n = 40$)
«я не хочу» ($n = 37$)
«я могу» ($n = 45$)
«я понимаю» ($n = 47$)
«я хочу» ($n = 60$)
«я могу быть» ($n = 5$)
«как я понимаю» ($n = 3$)
«я хочу знать» ($n = 4$)

«что же я могу» ($n = 3$)
«я понимаю что» ($n = 3$)
«я хочу быть» ($n = 3$)

Это частично обусловлено жанром художественных произведений, вошедших в корпус (к примеру, если бы корпус был составлен из научных статей, то эти сочетания слов не были бы выделены). Тем не менее высокая частота встречаемости таких фраз не случайна. При изучении языка подобные шаблонные фразы обычно усваиваются до усвоения грамматических правил языка, позволяющих правильно конструировать любые предложения.

Приведем также некоторые наиболее длинные сочетания слов, введение для которых собственного нетерминального символа позволяет уменьшить суммарную длину описания:

«во что бы то ни стало» ($n = 13$)
«как бы то ни было» ($n = 9$)
«в то же время» ($n = 32$)
«но в то же время» ($n = 11$)
«как ни в чем не бывало» ($n = 6$)
«на одном и том же месте» ($n = 3$)
«до тех пор пока не» ($n = 8$)
«как раз в то время когда» ($n = 3$)
«в одно и то же время» ($n = 6$)

Но и среди наиболее длинных сочетаний слов встретились такие, как

«не спуская с него глаз» ($n = 5$)
«ты не можешь себе представить» ($n = 4$)
«я до сих пор не могу» ($n = 3$)
«я только хочу сказать что» ($n = 3$)

Как и сочетания слов «я не хочу», «я не могу» и т. д., эти сочетания слов, безусловно, являются стереотипными и составляют элемент синтагматической структуры речи, но видно и заметное различие между фразами «во что бы то ни стало» и «ты не можешь себе представить»: первой фразе, в отличие от второй фразы, соответствует один концепт (это проявляется, в частности, в том, что она переводится на иностранный язык как единое целое, а не по частям).

Таким образом, в результате восстановления грамматики (4.4) определяются идиоматические выражения, часто используемые фразы, словоформы с примыкающими к ним служебными словами. Это, видимо, свидетельствует об ограниченности вида выбранной грамматики. Во-первых, отсутствует учет морфологии слов. Во-вторых, в грамматике отсутствует возможность введения классов слов.

Часто, чтобы исследовать именно проблему обучения фразам, отодвигаются еще дальше от предыдущего уровня, в частности, рассматривают только слова, наделенные самостоятельным смыслом, не делают различий между формами одного слова и т. д. Это, однако, требует введения информации о языке, которая сама может быть предметом автоматического анализа.

4.4.3. Разделение морфов на классы на основе принципа МДО

В п. 3.3.6 мы рассмотрели возможность получения списка морфов через поиск устойчивых сочетаний букв. Как было отмечено, эта процедура соответствует восстановлению грамматики вида (4.4), не содержащей никаких нетерминальных символов, кроме начального. Для решения задачи автоматического обучения морфологии языка применим восстановление грамматики другого частного вида — автоматной грамматики. Алгоритм восстановления подобной грамматики был приведен в п. 4.3.3, а способ задания ин-

формационной целевой функции грамматики — в п. 4.3.5, поэтому сам алгоритм мы повторять не будем.

В качестве терминальных символов здесь выступают морфы. Нетерминальные символы соответствуют классам морфов. Посмотрим, насколько эти классы будут совпадать с классами, выделенными в морфологии.

Возьмем весьма небольшую обучающую выборку словоформ, разделенных на морфы:

у-сматр-ива-ет	у-сматр-ива-ют	при-сматр-ива-ют-ся	у-смотр-им
при-лет-а-ет	с-лет-а-ет-ся	с-лет-а-ют-ся	с-лет-у
с-лет-е	при-лет-е	при-лет-а-ют	у-лет-а-ют
у-лет-им	при-ют-им	при-ют-им-ся	у-ют-е
при-ют-е	при-ют-у		

при-смотр-им
при-лет-у
с-лет-им-ся
у-ют-у

Начальная грамматика будет включать правила:

$$S \to \text{у}\, A_1;\ A_1 \to \text{сматр}\, A_2;\ A_2 \to \text{ива}\, A_3;\ A_3 \to \text{ет};$$
$$S \to \text{у}\, A_4;\ A_4 \to \text{сматр}\, A_5;\ A_5 \to \text{ива}\, A_6;\ A_6 \to \text{ют}$$

и т. д.

Не будем объединять правила при формировании первоначальной грамматики, поэтому словоформы «у-сматр-ива-ет» и «у-сматр-ива-ют» будут давать два правила, в левой части которых стоит начальный символ: $S \to \text{у}A_1$ и $S \to \text{у}A_4$. Неправомерность объединения нетерминальных символов на этапе формирования первоначальной грамматики хорошо понятна на примере двух словоформ: «у-лет-а-ют» и «у-лет-им». Если для них сразу же производить объединение нетерминальных символов, то в результате получатся правила: $S \to \text{у}A$; $A \to \text{лет}B$; $B \to \text{а}C$; $B \to \text{им}$; $C \to \text{ют}$, т. е. суффиксальный морф «а» будет объединен в один класс с флексийным морфом «им».

В нашем эксперименте грамматический вывод представлял собой последовательность итераций, на каждой из которых перебирались все пары нетерминальных символов и выбирались такие два символа, объединение которых давало максимальное уменьшение длины описания (подобный перебор для больших корпусов, разумеется, будет неприемлемым, и потребуется вводить дополнительные эвристики).

Сначала производились объединения правил вида $A_i \rightarrow$ о для разных i, где о — некоторое окончание. Такие объединения не увеличивали длину описания текстового корпуса посредством грамматики, но сокращали длину описания самой грамматики за счет уменьшения числа правил. При этом в системе грамматических правил появлялись правила с идентичной правой частью, но разными левыми частями, например, $A_i \rightarrow$ лет B, где $B \rightarrow$ у. Далее такие нетерминалы A_i объединялись между собой, давая один класс.

Видно, что на этом этапе грамматического вывода каждый нетерминальный символ соответствует классу, состоящему лишь из одного терминального символа. Тогда возникает вопрос: что представляли собой исходные классы до объединения? И если при объединении нетерминальных символов происходит обобщение, то в чем именно оно заключается? Заметим, что исходное число нетерминальных символов соответствует не числу терминальных символов, а числу их вхождений в текстовый корпус. Иными словами, каждое вхождение некоторого терминального символа считается уникальным. Эта уникальность обеспечивается контекстом, в котором этот символ появляется. Другими словами, некоторый нетерминальный символ описывает некоторый терминальный символ в некотором контексте. При первоначальном объединении нетерминальных символов происходило абстрагирование от того, в каком контексте встречается тот или иной символ (морф). Отметим, что это абстрагирование является неполным — определенная зависимость от контекста остается. В противном случае такая процедура была бы лишена смысла и исходную грамматику следовало бы формировать так, чтобы число нетерминальных символов соответствовало числу терминальных символов (если не считать того, что первые символы в словах генерируются из начального символа S).

В результате первоначального абстрагирования от контекста такие слова, как «при-ют-у» и «при-лет-у», оказываются порождаемыми правилами вида: $S \rightarrow$ при A_1; $S \rightarrow$ при A_2; $A_1 \rightarrow$ ют B; $A_2 \rightarrow$ лет B; $B \rightarrow$ у. Теперь уже объединение таких нетерминальных символов, как A_1 и A_2, соответствует не просто объединению в класс одного и того же терминального символа, но встречающегося в разных контекстах, а объединению различных терминальных символов. В результате все корневые морфы объединяются в один класс, после чего происходит постепенное объединение остальных морфов.

Приведем конечную грамматику, полученную в результате грамматического вывода на основе нашей обучающей выборки:

$S \rightarrow$ у A | при A | с A
$A \rightarrow$ сматр B | лет D | смотр D | ют D | лет B
$B \rightarrow$ ива C | аC | ся
$C \rightarrow$ ет | ют | ют B | ет B
$D \rightarrow$ им | им B | е | у

Полученные классы в достаточной степени соответствуют классам морфов, известным из морфологии. Одно небольшое отклонение заключается в том, что флексийные морфы разделены на две группы: «ют», «ет» и «им», «е», «у». Это деление полностью обусловлено малым текстовым корпусом: «ют» и «ет» в нем встречаются только после суффиксов «а» и «ива», в то время как остальные окончания встречаются только сразу после корней. Менее обоснованным выглядит объединение суффиксальных морфов «ива» и «а» с постфиксальным морфом «ся». Вероятно, причиной является то, что алгоритм редукции грамматики является «жадным» алгоритмом (в связи с чем дает неоптимальное решение).

Отметим, что в исходной грамматике приставки «у», «при», «с» сразу же были объединены в один класс, так как порождались из начального символа S. Если бы в корпусе присутствовали словоформы без приставок, то входящие в них корневые морфы оказались бы также в этом классе. В связи с этим при формировании исходной грамматики следует вводить несколько начальных символов и из каждого символа будет порождаться собственное слово.

Отметим также, что окончание «ют» и корень «ют» оказались в разных классах. При разборе некоторого слова в рамках данной грамматики встреченное «ют» будет распознано правильно. Таким образом, начальное разделение символов на классы по контексту позволяет автоматически решить проблему их омонимии. Тезис о неоднозначности естественного языка является общепринятым. Однако эта неоднозначность по большей части возникает из-за рассмотрения фрагмента текста вне всего контекста. Так, морф «ют» имеет неоднозначное толкование, но только до тех пор, пока не указано слово, в которое он входит. Слово «коса» имеет несколько значений, но эта неоднозначность разрешается, когда это слово используется в конкретном предложе-

нии. В предложении «Спрос рождает предложение» выбор между двумя кандидатами на роль подлежащего, скорее всего, мог бы быть разрешен, если бы это предложение использовалось в определенном контексте. Однако уже при выделении классов слов возникают определенные проблемы с подходом, используемым нами: слов очень много, а редуцировать грамматику, в которой под каждое вхождение в текстовый корпус каждого слова выделен собственный нетерминальный символ, крайне ресурсоемко. Применение же общих процедур грамматического вывода для выделения структуры целого текста (в терминах некоторой формальной грамматики), состоящего из многих предложений, практически неосуществимо. К сожалению, вопросы моделирования контекста выходят далеко за рамки данной книги.

Задача о выделении классов морфов тесно связана с задачей выделения классов слов, поскольку многие морфы являются характерными для определенных грамматических категорий, но для простоты изложения мы рассмотрим эти задачи раздельно.

4.4.4. Построение классов слов на основе принципа МДО

Одна из задач, которая не решалась системой CELL (см. п. 3.4), — это задача построения общих понятий. Такие общие понятия необходимы для того, чтобы перейти от конкретного предметного мышления к более высоким формам абстрактного мышления. Кластеризацию частных понятий естественно осуществлять по соответствующим им семантическим признакам. Однако это потребовало бы предварительного решения проблемы построения системы понятий, основанных на семантике, что является слишком трудоемким для решения ряда практических задач компьютерной лингвистики. Вместо этого ограничиваются использованием только лингвистической информации о появлении каждого из слов рядом с другими словами (в их контексте) в текстовом корпусе.

Задание отношения эквивалентности на множестве слов на основании совпадения контекстов, в которых эти слова допустимы, является общепринятым в математической лингвистике. Вводимые через подобное отношение эквивалентности дистрибутивные классы (или семейства) слов являются основой для формального определения различных

грамматических категорий [387]. Понятие допустимости слова в данном контексте подразумевает только грамматическую правильность построения соответствующего предложения, без учета его осмысленности. Если же определение классов слов осуществляется на практике по текстовому корпусу, то вовлеченные в этот процесс предложения будут корректными и в семантическом смысле, а значит, полученные классы слов будут захватывать как синтаксические, так и семантические категории (см., например, [388]).

Рассмотрим задачу построения классов слов на основе текстового корпуса как задачу грамматического вывода. Отдельные слова здесь — терминальные символы некоторой формальной грамматики. Классам слов, очевидно, должны соответствовать нетерминальные символы. Контекст может моделироваться по-разному в зависимости от выбранного вида правил вывода. Рассмотрим восстановление регулярных грамматик с правилами вида $A \to aB$ и $A \to b$. Правило $A \to aB$ может быть проинтерпретировано таким образом: слово «a» относится к классу слов A и встречается слева от некоторого слова из класса B.

Общий алгоритм восстановления подобных грамматик на основе принципа МДО был описан в пп. 4.3.3 и 4.3.5. Затронем некоторые особенности использования этого алгоритма при выявлении классов слов.

Во-первых, в исходно формируемой грамматике нетерминальные символы могут быть выделены под каждый терминальный символ (слово) или под каждое вхождение терминальных символов в текстовый корпус. В первом случае все возможные значения некоторого слова, которые могут быть различены по контексту, сливаются в один класс. Однако при построении классов слов проблемой омонимии часто пренебрегают. Как отмечалось выше, это позволяет существенно уменьшить число нетерминальных символов в начальной грамматике (или начальное число классов слов), что приводит к существенному повышению вычислительной эффективности. Используя полученные таким образом классы слов, можно разделять омонимы для каждого слова по отдельности.

Во-вторых, нужно решить, следует ли использовать в качестве терминальных символов слова или нужно различать их разные формы. При использовании словоформ, очевидно, будет наблюдаться тенденция к формированию классов на основе падежей (в связи с появлением определенных

форм слов рядом с соответствующими предлогами), а также на основе числа и рода (например, рядом со словоформой «цветом» может появиться «красным» или «зеленым», а рядом со словоформой «цвету» — «красному» или «зеленому», поэтому словоформы «красным» и «красному» не будут объединены в один класс, а «красному» и «зеленому» — будут). Как мы видели в п. 4.4.2, предлоги со словами, находящимися в соответствующих падежах, образуют устойчивые сочетания, которые выделяются в грамматиках вида (4.4). Именно поэтому при использовании слов в качестве терминальных символов грамматики необходимо не только каждую словоформу в текстовом корпусе привести к исходной форме, но также исключить из рассмотрения предлоги и частицы. Объединение словоформ в парадигмы — задача морфологического анализа. Выделение цепочек слов тоже является отдельной задачей (она рассматривалась выше). Неясности, возникающие с постановкой задачи выделения классов слов, связаны с вопросом, где проводить границу между этой задачей и другими задачами восстановления структуры языка. Разделение на подзадачи необходимо для создания практически применимых алгоритмов. В то же время между этими подзадачами остаются взаимосвязи, которые нельзя игнорировать.

Для эксперимента мы воспользовались следующим приемом: при сравнении слов длиннее пяти символов игнорировались различия в последних трех символах, а также были исключены из рассмотрения слова в три и менее символа. Это грубый, но простой способ в русском языке перейти от словоформ к словам.

В качестве текстового корпуса мы использовали текст п. 4.3 данной книги. При формировании первоначальной грамматики на основе каждого слова a формировался один нетерминальный символ A и набор правил подстановки $A \to aB_i$, где символу B_i соответствует слово b_i, встретившееся справа от слова a. Далее проводилась редукция этой грамматики под управлением принципа МДО.

В результате получились классы, включающие разделение как по грамматическим, так и по семантическим признакам. Из-за использования малого по объему текстового корпуса, что необходимо при применении достаточно общих процедур грамматического вывода, многие выделенные классы содержали и посторонние элементы. К примеру, были выделены такие классы:

A — иметься; использоваться; оказываться; являться.

B — следующий; неоднозначный; некоторый; любой.

C — малость; возрастание; убывание; перечисление.

D — напротив; во-первых; во-вторых; в-третьих; известно; полагая.

Видна не только связь между словами в этих классах по принадлежности к близким грамматическим категориям, но и определенная связь по смыслу. Если в классе *B* три из четырех слов — местоименные прилагательные (имеющие специфичную функцию указательных слов), то приведенный ниже класс содержит прилагательные с более абстрактными значениями:

E — сенсорный; универсальный; конкретный; синонимичный.

Приведем еще несколько выделенных классов:

F — поиск; редукция; выбирать; терять; ожидать; пригодиться.

G — термин; правило; ошибка; неправильный; новый; допустимый.

H — предпочтительный; затруднительный; неукорачивающий; NP-полный; конструирование; манипуляция.

I — исправленный; нестрогий; независимый; собственный; кодировать; трактовать.

J — формализм; трудность; сторона; приложение; целевой; ненадежный.

Далеко не все выделенные классы объединяют слова, имеющие четкое сходство по грамматическим категориям или смыслу, однако определенная взаимосвязь между ними прослеживается. К примеру, в классе *G* присутствуют существительные и прилагательные, так или иначе относящиеся к описанию данных посредством моделей (связь между словами «правило», «ошибка», «неправильный» достаточно очевидна). В классе *F* присутствуют слова «поиск» и «редукция» (напомним, что в п. 4.3 редукция грамматики была средством поиска при грамматическом выводе), которые в достаточной степени ассоциируются со словами «выбирать» и «терять».

Недостаточное разделение слов по грамматическим категориям вызвано использованием в данном тексте предложений с достаточно сложной структурой, а также с нестрогостью правил построения предложений при использовании небольшой обучающей выборки. Разделение слов на грамматические категории может также вестись путем

анализа их морфологии. Корректное использование двух видов информации — синтаксической и морфологической — возможно на основе принципа МДО. К примеру, именно этот принцип позволил авторам статьи [389] объединить информацию о соответствии суффиксов и грамматических категорий и о появлении суффиксов в словах с определенной основой.

Вместо задачи построения классов слов часто ставится задача построения тезауруса — иерархически сгруппированных классов слов. В представлении в форме тезауруса фиксируется степень близости различных слов внутри классов, что является более информативным, чем представление множества слов в виде планарных классов. В работе [390] развивается подход к восстановлению грамматик на основе принципа МДО на примере задачи построения тезауруса. Этот подход весьма близок к описанному нами (за тем исключением, что в нем используется понятие частичной грамматики). Тестирование проводилось на хорошо подготовленном корпусе с текстами на английском языке. Авторы указывают на то, что полученные в результате классы подвержены совместному влиянию как семантических, так и грамматических категорий. При этом грамматические категории превалируют (смысловое объединение слов различных грамматических категорий отсутствует, что может быть следствием более жесткого, чем в русском языке, синтаксиса). Внутренняя структура отдельных классов оказывается не слишком показательной, однако определенную дополнительную информацию несет. Приведем один пример из работы [390] (рис. 4.2).

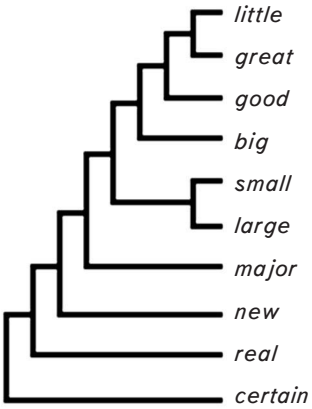

Рис. 4.2. Элемент автоматически построенного тезауруса: один из найденных классов слов, иерархически разбитый на подклассы

Лингвистическая информация о грамматических категориях является более доступной, чем информация о семантических категориях слов. В связи с этим на практике бывает полезным устранить влияние грамматики на процесс построения тезауруса, воспользовавшись априорной информацией о ней. К примеру, в работе [376] (см. также [391]) параллельно строятся классы для существительных и глаголов: существительные объединяются в классы на основе частот их появления рядом с глаголами (или классами глаголов), а для глаголов для той же цели в качестве опорных используются классы существительных (при этом формальные грамматики не используются). Обсуждение этого и других подобных подходов выходит за рамки данной книги. Отметим лишь несколько моментов. Во-первых, иерархическая структура классов оказывается здесь более релевантной: в классы имеют тенденцию группироваться слова, связанные по смыслу. Во-вторых, принцип МДО здесь также оказывается полезным. В частности, в работе [376] предлагается подход на основе принципа МДО с демонстрацией его преимущества по сравнению с методами максимального правдоподобия.

Такая модификация задачи построения тезауруса может быть предпочтительней с практической точки зрения, но для системы машинного обучения она подразумевает наличие априорного умения распознавания общих грамматических классов — подлежащих и сказуемых.

4.4.5. Проблема выделения подзадач при восстановлении грамматик

Мы рассмотрели некоторые приложения принципа минимальной длины описания и формальных грамматик к задачам анализа естественного языка. Формальные грамматики в рамках единого представления позволяют описывать разные структурные особенности языка: морфы и их классы, слова, их классы и устойчивые словосочетания и т. д. Однако автоматическое восстановление формальной грамматики, которая бы захватывала все эти структурные особенности, является крайне ресурсоемким, тем более на больших текстовых корпусах. Как правило, на практике задачи по обнаружению морфов, формированию их классов, построению тезауруса, выделению как отдельных фраз, так и

структуры целых предложений решаются раздельно. Существование возможности разделения (хотя и неполного) этих задач представляет несомненный интерес. В данном случае это разделение делалось человеком на основании своего опыта. Перечисленные частные задачи получаются из общей задачи восстановления грамматики путем введения ограничений на вид правил подстановки и выбора надлежащего алфавита терминальных символов. Формальные грамматики дают методологическое основание для синтеза из решения частных задач решения общей задачи в результате снятия этих ограничений. Использование в этом случае принципа минимальной длины описания позволяет выразить целевые функции всех задач в единых терминах. Однако нерешенным остается фундаментальный вопрос: как такое разделение и последующий синтез выполнять автоматически?

В частности, если система машинного обучения начинает свою работу с представления, в котором в качестве терминальных символов выступают буквы или фонемы, то как провести четкую границу между уровнями букв, морфем, слов, словосочетаний? Иными словами, как, к примеру, из исходного текста как совокупности букв вывести понятие (элемент информационного представления) слова, если оно не заложено в систему априори? Отметим еще раз, что в рамках такого общего представления, как формальные грамматики, потенциально допустимы объекты со структурой, состоящей из смеси любых терминальных и нетерминальных символов. Понятие слова в некоторой формальной грамматике можно считать выделенным только тогда, когда в ней образован нетерминальный символ, порождающий все прочие символы, соответствующие конкретным словам. Однако при практическом обучении по текстовым корпусам на формирование такого нетерминального символа надеяться не приходится: выигрыш от введения одного класса типа «слово» может быть слишком незначительным по сравнению с использованием нескольких столь же высокоуровневых классов. Более того, при последовательном слиянии нетерминальных символов слишком велика вероятность на более ранних этапах принять локальное решение об объединении в один класс объектов разных уровней.

Вернемся к случаю построения структурных описаний изображений. Сжатие, достигавшееся на нижнем (пиксельном) уровне, было существенно больше, чем сжатие от опи-

сания контуров посредством структурных элементов. Выигрышем от введения последних можно было бы фактически пренебречь, если бы целью было собственно сжатие изображения. Однако именно структурные элементы несли наиболее важную информацию о содержании изображения. Похожая ситуация имеет место и при анализе текстов на естественном языке. Таким образом, эмпирически наблюдаемая ценность высокоуровневой информации не находит численного отражения в классическом определении длины описания и в других альтернативных критериях.

Можно было бы утверждать, что понятие слова гораздо легче может быть выведено при использовании данных от различных сенсорных модальностей. В частности, поскольку визуально наблюдаемым объектам сопоставляются отдельные слова, то эти слова могут быть выделены таким же образом, как в системе CELL. В то же время можно заметить, что в этой системе понятие слова (вернее, понятие о сингулярных терминах, осуществляющих референцию на объекты реального мира) было введено априори (и только оно, так как система не могла выделить понятие слога или словосочетания).

При независимом решении таких задач, как выделение классов слов или морфем, формирование словосочетаний, в качестве терминальных символов используются нетерминальные символы грамматик, полученные при решении более низкоуровневых задач (например, при формировании алфавита морфем). Хотя те же языковые объекты и зависимости между ними могут быть описаны в рамках одной формальной грамматики общего вида, это представление будет отличаться от представления в виде иерархии грамматик частного вида. Отличие между этими представлениями заключается не столько в выразительной силе представлений, сколько в вычислительной сложности задач, решаемых с их помощью. К примеру, в случае использования системы грамматик при работе со словами как неделимыми объектами можно абстрагироваться от структуры слов и игнорировать все правила подстановки, относящиеся к грамматикам других уровней. В случае же использования одной грамматики общего вида правила подстановки в ней не организованы в какую-либо структуру.

Разговор о представлениях информации мы начали с того, что представления различаются не только тем, можно ли в них описать тот или иной набор данных, но и эффективностью этого описания. В качестве примера приводилось

описание точек экспоненциальной кривой посредством системы полиномов (см. п. 1.2.4); также указывалось (см. п. 2.7) на то, что при выборе модели может быть полезно учитывать не только критерий длины описания, но и число операций, необходимых для построения каждой из альтернативных моделей. Здесь мы видим, что это относится не только к отдельным моделям, но и к целым представлениям. При выборе среди различных представлений должна учитываться не только общая длина описания совокупности массивов данных, но и общая вычислительная сложность построения моделей по этим массивам. При использовании критерия, учитывающего вычислительную сложность, иерархические представления информации получат неоспоримое преимущество перед эквивалентными им неиерархическими представлениями.

Стратегии поиска имеют большое значение и для самого процесса построения представления при грамматическом выводе. Автором был использован метод последовательного улучшения грамматики в результате ее локального изменения: выполнения операций слияния или конструирования для пар нетерминальных символов. Как уже отмечалось, сходная стратегия поиска может использоваться при решении задач группирования и сегментации, могут также использоваться генетические алгоритмы, имитация отжига и другие стохастические методы оптимизации. Стратегии поиска могут варьироваться от «жадных» алгоритмов до полного перебора. Эти две крайности имеют много общего с двумя крайними способами описания данных — ad hoc- и «беспорядочной» моделями. «Жадный» алгоритм является одним из наиболее простых (в смысле вычислительной сложности, а не длины описания) алгоритмов поиска, но он может давать весьма неточное решение (но может давать и точное решение, если сложность этого алгоритма поиска соответствует сложности задачи оптимизации!). Полный перебор дает точное решение оптимизационной задачи, но сам обладает большой вычислительной сложностью. Очевидно, оптимальный алгоритм поиска должен быть компромиссом между сложностью поиска и его точностью. Также очевидно, что положение этого оптимума зависит от исходных данных. Так, полный перебор может оказаться лучшей альтернативой при решении оптимизационной задачи на малом объеме исходных данных, но при увеличении последнего он окажется неприемлемым.

Отметим, что в информационное представление принято включать не только структуру пространства моделей, но и сами алгоритмы интерпретации наблюдательных данных, т. е. алгоритмы построения описаний данных [235, с. 36]. К сожалению, этот элемент представления редко служит предметом рассмотрения при решении задач индуктивного вывода. В связи с этим проблема автоматического выбора между алгоритмами поиска разной вычислительной сложности исследована мало, хотя для построения системы машинного обучения общего назначения решение этой проблемы необходимо. Некоторые подсказки можно получить из методов эвристического программирования и автоматического построения эвристик поиска.

Возвращаясь к проблеме оптимизации процедуры грамматического вывода, нельзя не назвать еще одну возможность ее осуществления, связанную с инкрементным обучением. На данный момент наиболее популярным является использование больших текстовых корпусов, которые дают репрезентативную выборку для подсчета частот N-грамм. Однако вместо этого можно начинать с малых текстовых корпусов с ограниченным лексиконом и упрощенным строением предложений и постепенно усложнять тексты. Грамматический вывод, производимый по новым текстам, будет базироваться на выделенных на предыдущих этапах обучения лингвистических единицах и грамматических правилах, что существенно упростит процедуру вывода и сделает ее более надежной. К сожалению, формальные грамматики и методы их восстановления, допускающие инкрементное обучение, исследованы мало, а сам этот подход требует специально составленных обучающих выборок (здесь могут помочь книги для детей разного возраста).

Мы рассмотрели применение формальных грамматик в вычислительной лингвистике. Однако этим сфера их применения не ограничивается. Как отмечалось выше, формальные грамматики могут использоваться для распознавания образов, автоматического порождения и понимания программ и во многих других приложениях. Сделав акцент на задачи анализа естественного языка, мы показали возможные пути решения некоторых проблем (таких, как формирование общих понятий или восстановление синтагматической структуры языка), упоминавшихся в гл. 3. Использование только лингвистической информации существенно ограничивает описанные здесь методы. К сожалению, ис-

следования по использованию семантической информации в процессе восстановления структуры языка практически отсутствуют.

Помимо формальных грамматик существуют и другие символьные представления информации — наборы правил, деревья решений, графы решений и связанные леса решений, которые имеют между собой много общего. Эти представления хотя и имеют определенное отношение к формальным грамматикам, но используются при решении задач, слабо связанных с вычислительной лингвистикой.

4.5. НАБОРЫ ПРАВИЛ, ДЕРЕВЬЯ И ГРАФЫ РЕШЕНИЙ

4.5.1. Построение наборов порождающих правил

В основе формальных грамматик лежат порождающие правила, или продукции, вида $\alpha \rightarrow \beta$, которые имеют следующий смысл: «если в цепочке символов встретилась подстрока α, то заменить ее цепочкой β». Правила сходного вида используются для описания динамических систем, начиная с конечных автоматов и заканчивая машинами Тьюринга. Различного рода условные операторы в языках программирования имеют ту же природу. Все эти продукции можно представить общей схемой: «если верно <условие>, то выполнить <действие>», т. е. порождающее правило говорит, какое действие следует выполнить в данной ситуации.

Строку α правила $\alpha \rightarrow \beta$ обычно называют левой частью, условием, или антецедентом, а строку β — правой частью, действием, или консеквентом. Термины «антецедент» и «консеквент» заимствованы из логики высказываний и говорят о том, что порождающие правила могут использоваться для представления причинно-следственных связей вида «если..., то...». Однако и в этом случае консеквент часто описывает некоторое действие, например, «если сверкнула молния, то скоро прогремит гром».

Принято считать [209, с. 106], что порождающие правила являются хорошей моделью принятия решений человеком. Следует отметить непосредственное сходство порождающего правила и рефлекторной дуги, а также то, что классическая схема S—R (стимул—реакция) не объясняет ряда экспериментальных данных по реакции биологических си-

стем на стимулы и требует расширения, примером чего может служить схема Т—О—Т—Е (проба—операция—проба—результат), см., например, [207, с. 102–116]. Отсутствие подобного расширения в продукционных системах говорит об их ограниченности в качестве модели выбора действия биологическими системами. Тем не менее наборы порождающих правил широко и довольно успешно применяются в экспертных системах в качестве элемента подсистемы представления знаний (продукционной системы), описывающего взаимосвязь между наблюдательными данными и действиями, предпринимаемыми при их получении.

Как описание исходной ситуации, так и возможные действия в порождающих правилах, обычно задаются в символьной, дискретной форме. В целях синтеза оптимальных систем управления, напротив, рассматриваются преимущественно непрерывные воздействия. Так, система стабилизации курса корабля или система управления шагающим устройством должна непрерывно осуществлять управление, описываемое системами дифференциальных уравнений, а не дискретным набором инструкций <условие>—<действие>. К сожалению, рассмотрение этого нижнего уровня (непрерывного управления) выходит за рамки данной книги.

Связь <условие>—<действие> (или <причина>—<следствие>) может устанавливаться на основе опыта. Иными словами, построение набора порождающих правил — задача индуктивного вывода. Продукционные системы могут конструироваться вручную при участии людей-экспертов. Такой подход часто применяется при разработке экспертных систем (например, в целях медицинской диагностики). Более интересным для нас представляется использование методов машинного обучения для автоматического построения наборов порождающих правил по примерам. Это является и более удобным на практике. Например, вместо того, чтобы просить врачей-экспертов в явном виде формулировать правила <симптомы>—<болезнь>, можно использовать данные по ранее поставленным диагнозам для автоматического формирования соответствующих правил. Поставим задачу автоматического построения набора правил по примерам более формально.

Пусть есть пространство признаков $X = X_1 \times X_2 \times \ldots \times X_N$, где каждое из множеств X_i содержит конечное число элементов. В задаче построения набора правил (а также дерева или графа решений, о которых речь пойдет ниже) при-

знаки часто называются *атрибутами* (для обозначения переменных, множество значений которых конечно, также можно встретить название «категориальные переменные»). Пусть также дано конечное множество классов $A = \{a_1, a_2, ..., a_d\}$, где d — число классов. По обучающей выборке $D = \big((x_1, c_1), (x_2, c_2), ..., (x_M, c_M)\big)$, где $x_i \in X$, $c_i \in A$, требуется построить классификационный алгоритм, который бы для нового объекта по его атрибутам предсказывал номер класса, к которому он относится. Выходной класс может соответствовать как некоторому действию, так и отдельному концепту. В последнем случае обычно говорят о задаче *изучения понятий* или *обучения концептам* (concept learning).

Как видно, мы приходим к формулировке классической задачи распознавания образов (случай обучения с учителем), с которой уже встречались в гл. 2. Различие, однако, заключается в том, что в рассмотренных нами дискриминантных методах распознавания признаки считались вещественными величинами, в то время как в данном случае атрибуты являются дискретными. Таким образом, наборы порождающих правил (как и деревья решений) могут использоваться как средство решения задачи распознавания с учителем в дискретном пространстве признаков.

Весьма интересен частный случай, при котором все величины являются бинарными. Такое ограничение ведет к логическим методам распознавания (или к задаче восстановления пропозициональных формул). Отличие логических методов распознавания в том, что антецеденты (левые части правил) содержат не просто перечень значений атрибутов, при которых эти правила могут быть применены, но представляют собой логические выражения, в которых используются операторы «И», «НЕ», «ИЛИ» и т. д., например, $X_1 \,\&\, X_2 \vee \neg X_1 \,\&\, \neg X_2 \Rightarrow a$. Мы, однако, такой способ описания условий рассматривать не будем.

Как уже отмечалось в п. 2.1.3, системы распознавания с дискретными и непрерывными пространствами признаком могут быть объединены в многоуровневую систему распознавания. На это еще раз следует обратить внимание в связи с проблемой построения концептуальной системы (см. п. 3.4.1). Действительно, в гл. 3 книги мы рассмотрели методы распознавания сенсорных сигналов и формирования на их основе простейших концептов. Далее эти концепты могут использоваться как атрибуты в дискретной системе распознавания, строящей более абстрактные кон-

цепты. С возможностью построения классов слов мы уже сталкивались при обсуждении применений формальных грамматик. Формальные грамматики предоставляют более широкие возможности задания исходных описаний объектов и более широкое пространство моделей, поскольку допускают произвольные структурные взаимосвязи между атрибутами. Здесь же атрибуты оказываются в определенном смысле независимыми: изучаемые концепты инвариантны по отношению к порядку атрибутов. Задачу построения набора правил можно свести к задаче восстановления грамматики, однако для многих задач наборы правил являются более естественным представлением.

Сформулируем некоторые особенности представления концептов в виде набора правил. Для этого рассмотрим пример. Пусть имеются следующие атрибуты:

цвет: $X_1 = \{$желтый $= 0$, синий $= 1$, зеленый $= 2$, красный $= 3\}$;

форма: $X_2 = \{0 =$ круглый, $1 =$ квадратный, $2 =$ вытянутый$\}$;

размер: $X_3 = \{0 =$ маленький, $1 =$ большой$\}$;

вес: $X_4 = \{0 =$ легкий, $1 =$ тяжелый$\}$;

степень съедобности: $X_5 = \{0 =$ съедобный, $1 =$ несъедобный$\}$;

следующие классы:

$a_1 =$ яблоко; $a_2 =$ мяч; $a_3 =$ гиря; $a_4 =$ банан;

также имеются следующие правила:

1) $X_2 = 0$ & $X_5 = 0 \to a_1$;
2) $X_2 = 0$ & $X_4 = 1 \to a_3$; (4.5)
3) $X_2 = 0 \to a_2$;
4) $X_1 = 0$ & $X_2 = 2$ & $X_5 = 0 \to a_4$.

Отметим следующие особенности. Во-первых, число атрибутов N может быть достаточно большим. В связи с этим явное описание допустимых значений всех атрибутов в левой части правила не используется. Вместо этого в условии указываются лишь те атрибуты, значение которых релевантно данному концепту. От остальных атрибутов происходит абстрагирование. Например, 1-е правило ($X_2 = 0$ & &$X_5 = 1 \to a_1$) говорит о том, что яблоко — это нечто круглое и съедобное независимо от цвета, веса и размера (для данного набора классов).

Во-вторых, в наборе правил важна последовательность применения правил, так как для одних и тех же значений атрибутов может найтись несколько подходящих правил с разными правыми частями. Например, 3-е правило ($X_2 = 0 \to a_2$)

классифицирует все круглые объекты как мяч, но, несмотря на это, если оно применяется после 1-го и 2-го правил, то результат его применения будет корректным.

В-третьих, в наборе правил может не найтись ни одного правила, условие применения которого удовлетворяет данным значениям атрибутов. Из-за этого возникает новый вид ошибки, не встречавшийся в дискриминантных методах распознавания. Например, набор атрибутов ($X_1 = 2$ & $X_2 = 2$ & $X_5 = 0$) не подпадет ни под одно правило. Этот вид ошибки называется *пропуском*.

Вернемся к постановке задачи построения набора правил. Исходные данные были определены. Пространство гипотез также несложно установить: оно состоит из множества отображений $\{\varphi | \varphi : X \to A\}$. Для определения задачи построения набора правил как задачи индуктивного вывода не хватает задания целевой функции.

Часто в качестве целевой функции используется размер набора правил. При этом выбирается исходный набор правил, который просто повторяет данные примеры: $\{x_i \to c_i\}_{i=1}^{M}$. Его необходимо максимально упростить, сохранив при этом корректную классификацию обучающих примеров. Этот классический подход основывается на том эмпирической факте, что минимальный набор правил, совместимый с примерами, является наиболее полезным [74, с. 394].

Смысл этой полезности вполне очевиден в рамках принципа минимальной длины описания. Наименьший набор правил, совместимый с обучающей выборкой, обладает наибольшей предсказательной силой (с наибольшей вероятностью предсказывает истинные классы для примеров, не вошедших в выборку). Упрощение правил соответствует их обобщению. Поясним это на очень простом примере. Пусть в выборке даны следующие примеры:

круглый, легкий, зеленый → мяч;

круглый, легкий, синий → мяч;

круглый, легкий, красный → мяч.

Упрощение этих правил может дать правило «круглый, легкий → мяч», совместимое со всеми примерами. Упрощенное правило не только заметно короче трех исходных правил, но также позволяет корректно классифицировать такой пример, как «круглый, легкий, желтый», который до этого ни под одно правило не подпадал.

На практике интерес представляет случай зашумленных данных, т. е. таких выборок, примеры в которых могут со-

держать ошибки. Обучение правилам — типичное обучение с учителем, которое может происходить по примерам, полученным от разных людей (например, экспертов-медиков, ставящих разные диагнозы при похожих симптомах; людей, выполняющих различные действия в одинаковых ситуациях). При этом возникает необходимость обобщения. Таким образом, выводимый набор правил должен не просто минимизироваться по размеру, но в этом процессе должно также учитываться количество примеров-исключений, не соответствующих правилам.

Вполне очевидно, что здесь может быть применен принцип МДО. Исходя из этого принципа, наилучшим (при имеющихся данных) является тот набор правил, при котором минимизируется сумма

$$DL = DL_{rules} + DL_{exceptions}, \qquad (4.6)$$

где DL_{rules} — длина описания набора правил; $DL_{exceptions}$ — длина описания исключений.

Для более аккуратного анализа задачи вернемся к рассуждениям с отправителем и получателем сообщения (что чаще соотносится с принципом минимальной длины сообщения). При обучении с учителем (см. п. 2.2), как правило, полагается, что в сообщении отправитель передает информацию только о правых частях обучающих примеров, в то время как левые части получателю известны априори. Формула вида (4.6) как раз и соответствует передаче сообщения, несущего информацию о правых частях примеров обучающей выборки.

Рассмотрим слагаемые более подробно для случая, когда число классов равно двум (примеры, относящиеся к разным классам, при этом часто называются положительными и отрицательными).

Сначала рассмотрим кодирование ошибок (исключений) для определения слагаемого $DL_{exceptions}$. Ошибки могут быть двух типов. Во-первых, применение правил может приводить к неверной классификации некоторых примеров. Чтобы получатель мог восстановить правильные значения отсутствующих правых частей, ему должен быть передан список исключений (в сообщении просто необходимо указать, какие именно примеры классифицированы неправильно, так как число классов равно двум). Чтобы закодировать индексы M_{FP} неверно классифицированных примеров из всех M примеров, необходимо примерно $\log_2 C_M^{M_{FP}}$ бит информации.

Поясним это. Всего существует $C_M^{M_{FP}}$ сочетаний из M примеров по M_{FP} примеров. Если пронумеровать все возможные сочетания в лексикографическом порядке, то для указания номера конкретного сочетания, включающего M_{FP} примеров, потребуется $\log_2 C_M^{M_{FP}}$ бит информации. Для однозначного выбора требуется также указать количество примеров M_{FP}, для чего необходимо примерно $\log_2 M_{FP}$ бит (при использовании кода с саморазграничением для записи натуральных чисел требуется $\log_2 M_{FP} + \log_2 \log_2 M_{FP} + \ldots$ бит, см. п. 1.5.4). Последним слагаемым обычно пренебрегают, так как оно существенно меньше первого (за исключением случая $M_{FP} \approx M$).

Как уже отмечалось, некоторые примеры могут не подпадать ни под какое из правил. Такие примеры — это второй тип ошибок, обозначаемых как пропуски и требующих отдельного кодирования. Является ли некоторый пример пропуском, получатель может легко определить, имея соответствующий набор правил. Поэтому отправитель должен лишь закодировать информацию о том, какие из пропусков должны быть классифицированы как положительные, а какие — как отрицательные (для случая двух классов $d = 2$). Пусть M_{NS} — число пропусков — примеров, не покрытых набором правил, и пусть M_{FN} — число пропусков, которые должны быть классифицированы как положительные. Тогда для их указания требуется порядка $\log_2 C_{M_{NS}}^{M_{FN}}$ бит информации. Таким образом, мы приходим к формуле, предложенной Квинланом [392]:

$$DL = DL_{rules} + \log_2 C_M^{M_{FP}} + \log_2 C_{M_{NS}}^{M_{FN}}. \qquad (4.7)$$

Длина описания правил DL_{rules} обычно берется просто пропорциональной суммарной длине правил.

Исследователи [393, 394] отмечают следующие недостатки формулы (4.7):

1) формула для вычисления длины описания исключений симметрична, т. е. набор правил, неправильно классифицирующий все примеры, будет иметь такую же длину описания исключений, равную нулю, что и набор правил, правильно классифицирующий все примеры;

2) если число положительных примеров существенно больше (существенно меньше) числа отрицательных примеров, то выведенный набор правил будет иметь тенденцию

обобщать в недостаточной (избыточной) степени изучаемый концепт, особенно в присутствии шума или при обучающей выборке малого объема;

3) при обучении по большой выборке примеров в присутствии шума все еще остается тенденция следовать этому шуму (создавать на основе исключений дополнительные правила).

Как отмечается в работе [394] (и мы поддерживаем это мнение), первый недостаток таковым не является. Действительно, какое-то понятие может быть выучено как отрицание другого понятия. Например, понятие «нечто, не являющееся яблоком» сложно выучить через описание всех атрибутов, которые могут отсутствовать у яблока. Также следует отметить, что симметрия несколько нарушается, если при выводе формулы (4.7) быть более аккуратными и не опускать слагаемые $\log_2 M_{FP}$ и $\log_2 M_{NP}$, которыми обычно можно пренебречь, но именно для этого крайнего случая они имеют видимый эффект. В результате описание отрицания некоторого концепта оказывается несколько длиннее, чем описание самого концепта.

По утверждению автора работы [394], следующие два недостатка связываются с тем, что формула (4.7) является приближенной оценкой истинной длины описания. На наш взгляд, корректнее было бы сказать, что это сравнительно точная оценка длины описания, но сделанная в рамках представления информации, не вполне адекватного (в смысле задаваемого им априорного распределения вероятностей) тем задачам, в которых оно применяется.

В работах [394, 395] предлагается более изощренная (хотя все еще простая с вычислительной точки зрения) схема кодирования. Воспользуемся основными идеями указанных работ. Обратим внимание, что в формуле (4.7) кодирование ошибок, сделанных при применении всех правил, осуществляется совместно, так как отправителем передается номер сочетания из M примеров по M_{FP} неверно классифицированных примеров. Раздельно кодируя ошибки, допущенные при применении разных правил, мы в общем случае уменьшим длину описания. Это подтверждается на практике, а также очевидно из общих соображений: некоторые правила могут не иметь исключений, в то время как другие правила могут иметь много исключений; разумно описывать исключения для таких правил независимо. Для примеров, не покрытых набором правил, можно ввести до-

полнительное правило с пустой левой частью (условием) и негативной правой частью (она может быть взята и положительной — это не имеет значения). Такое правило должно находиться в самом конце набора правил и применяться после проверки условий других правил. Это обеспечивает более единообразную схему кодирования как правил, так и ошибок. Тогда общая длина описания будет

$$DL = \sum_i [L(rule_i) + L(err_i)]. \qquad (4.8)$$

Длина описания ошибок для каждого из правил может вычисляться следующим образом. Каждое правило выделяет некоторое подмножество примеров, которые под это правило подпадают. Пусть отправитель передает получателю закодированные правые части этих примеров. Для этого необходимо $L(err_i) = m_i H(e_i)$ бит информации, где m_i — число примеров, подпадающих под i-е правило; $H(e_i)$ — энтропия, рассчитанная на основе гистограммы правых частей примеров для данного правила.

Например, если под правило с условием $X_2 = 0$ & $X_4 = 1$ подпадают два примера, относящихся к классу a_2, и шесть примеров, относящихся к классу a_3, то $L(err_i) = -2\log_2 2/8 - -6\log_2 6/8 \approx 6,5$ бит. Если все примеры, подпадающие под данное правило, имеют одну и ту же правую часть, то, очевидно, $L(err_i) = 0$. Теперь становится более ясным, почему раздельное описание ошибок для разных правил предпочтительнее.

Обратим внимание на следующее: при таком представлении не имеет значения, что именно стоит в правой части правила, так как кодируются любые значения правых частей (это становится наиболее явным, когда правые части могут принимать не два, а более значений). Тогда $L(err_i)$ принимает смысл не длины закодированных ошибок, а длины закодированных правых частей, а правила оказываются стохастическими, т. е. допускающими разные правые части с разными вероятностями. Подобные правила исследуются отдельно [396], в чем есть смысл: неопределенность действия в какой-то ситуации может быть явно выражена в наборе правил. Таким образом, длины кодов для правых частей каждого из правил могут использоваться не только для оценки критерия качества набора правил, но и для последующей классификации, сообщая об апостериорном распределении вероятностей по классам для каждого нового примера.

В работе [394] также приводится более строгая схема кодирования самих правил. Длину описания каждого правила предлагается считать следующим образом. Условие применения правила — это набор тестов, осуществляемых над атрибутами примера (например, правило «$X_2 = 0$ & $X_4 = 1 \rightarrow a_3$» содержит два теста). Пусть условие i-го правила содержит L_i тестов. И пусть всего возможно N_{pt} различных тестов. Тогда для описания условия i-го правила необходимо примерно $\log_2 C_{N_{pt}}^{L_i}$ бит информации (с учетом того, что порядок проверки тестов значения не имеет и описываться не должен).

Следует отметить, что одно слагаемое здесь также было пропущено — это длина таблиц перекодировок правых частей для подмножеств примеров каждого из правил. Действительно, если $L(rule_i)$ — только длина кодирования условия применения правила, а $L(err_i)$ — только оценка длины кодов Хаффмана для возможных правых частей правила, то у получателя не будет связи между закодированными по Хаффману правыми частями и их исходными значениями. Иными словами, необходимо передавать таблицу перекодировки, которая, по сути, является частью правила (ее длина не зависит от числа примеров) и описывает возможные правые части правил с указанием их вероятностей. Мы уже приводили оценки для длины описания таблицы перекодировки, их несложно адаптировать к текущему случаю. Здесь просто обозначим это слагаемое через $L(t_i)$.

Чтобы получатель смог корректно декодировать набор правил, ему необходимо знать, где заканчивается сообщение. Иными словами, необходимо знать общее число правил. Для этого следует добавить еще одно небольшое слагаемое, которое грубо можно оценить как $\log_2 N_{rules}$, где N_{rules} — общее число правил в наборе. Таким образом, окончательная формула будет иметь вид

$$DL = \log_2 N_{rules} + \sum_i \left(\log_2 C_{Lpt}^{L_i} + L(t_i) + m_i H(e_i) \right). \quad (4.9)$$

Зачастую некоторые слагаемые опускают (сознательно или нет), что приводит к менее точной оценке длины описания и некоторому снижению обобщающей способности метода построения набора правил. Возможно, наш анализ тоже был недостаточно строгим. Существует также возможность различного рода модификаций схемы кодирования [именно так мы перешли от формулы (4.7) к формуле (4.9)],

что приведет к смещению распределения априорных вероятностей.

Путем изменения схемы кодирования для получения более компактных описаний набора правил можно прийти к другим представлениям информации, содержательно отличающимся от наборов правил. Обратим внимание на то, что в наборе правил многие условия применения разных правил могут содержать общие элементы. Возникает соблазн сжать описание правил таким образом, чтобы одинаковые блоки тестов не дублировались. Очевидно, такое сжатие набора правил приведет к тому, что похожие правила будут предпочтительнее, чем такое же количество разных правил. Иными словами, произойдет смещение распределения априорных вероятностей, что скажется на обобщающей способности набора (для многих задач — положительным образом). Таким представлением, выступающим в роли альтернативы наборам правил, являются деревья решений.

Здесь мы не будем описывать алгоритмы оптимизации информационной целевой функции для набора правил. Вопрос построения конкретных алгоритмов отложим до обсуждения методов восстановления деревьев решений, для которых, на наш взгляд, существуют более элегантные алгоритмы поиска.

4.5.2. Информационный критерий качества дерева решений

В задаче построения дерева решений, как и в задаче построения набора порождающих правил, дано конечное множество атрибутов $\mathrm{X} = X_1 \times X_2 \times \ldots \times X_N$, где N — число атрибутов, и конечное множество классов $\mathrm{A} = \{a_1, a_2, \ldots, a_d\}$, где d — число классов. Как и раньше, по обучающей выборке $D = \big((x_1, c_1), (x_2, c_2), \ldots, (x_M, c_M)\big)$, где $x_i \in \mathrm{X}$; $c_i \in \mathrm{A}$, требуется восстановить отображение (алгоритм), действующее из X в A. Задача построения деревьев решения отличается способом представления этого алгоритма. В данном случае классификационный алгоритм задается в виде дерева, в каждом узле которого осуществляется проверка одного из атрибутов и в зависимости от значения этого атрибута происходит переход к соответствующему дочернему узлу (рис. 4.3). В листьях дерева решений располагаются номера классов, которые и выступают в качестве результа-

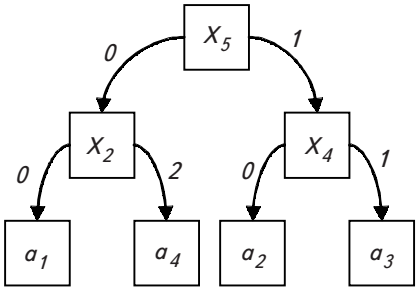

Рис. 4.3. Пример дерева решений, заменяющего систему правил (4.5). Здесь X_5 — степень съедобности (0 — съедобный, 1 — несъедобный); X_2 — форма (0 — круглый, 2 — вытянутый); X_4 — вес (0 — легкий, 1 — тяжелый); классы: a_1 = яблоко; a_2 = мяч; a_3 = гиря; a_4 = банан

та классификации. Узлы такого дерева также называются узлами принятия решений, а проверка значения атрибута в узле — тестовой процедурой.

Деревья решений были предложены Квинланом [397, 398] в качестве представления информации в рамках машинного обучения для решения задачи изучения понятий. Тем не менее деревья решений, как и наборы правил, пригодны также для описания связи <условие>—<действие>. Эти два представления имеют одинаковую выразительную силу, хотя компактность описания одних и тех же понятий у них может быть весьма различной (ср. рис. 4.3 и набор правил 4.5). Помимо этого основное отличие деревьев решений от наборов правил заключается в том, что в первых проверка атрибутов осуществляется последовательно, в то время как во вторых работа осуществляется сразу со всей совокупностью атрибутов.

Это приводит к некоторым отличиям в их использовании. В частности, деревья решений оказываются несколько более удобными для случая, в котором не все атрибуты доступны с самого начала и имеют разную сложность получения (в смысле материальных ресурсов). В качестве примера можно привести задачу геологической разведки (например, нефтяных месторождений). В этой задаче существуют разные способы обнаружения месторождений, которые имеют разную стоимость и разную эффективность. Оказывается выгоднее сначала проверить более дешевые в получении, хотя и менее информативные признаки (например, связанные с сейсмическими характеристиками местности). Более надежные способы проверки, такие, как бурение скважин, существенно дороже, и их следует использовать лишь после того, как вероятность наличия нефти повысилась в результате проверки других признаков. Похожим образом осуществляется диагностика и в других областях. Например, при по-

иске неисправности автомобиля в первую очередь осуществляются наиболее доступные тесты (например, проверка наличия бензина), даже если заранее известно, что эти тесты с высокой вероятностью не помогут выявить неисправность.

Если каждый атрибут обладает некоторой стоимостью, определяемой связанным с ним действием, то должно строиться такое дерево, которое минимизирует математическое ожидание затрат. В п. 1.2.2 уже отмечалось, что принципиальные различия между предсказанием и принятием решений отсутствуют: необходимость принятия решения лишь добавляет один уровень вывода, который базируется на результатах предсказания затрат.

Если информация о затратах отсутствует, т. е. атрибуты считаются в этом смысле равноправными, то минимизация затрат будет соответствовать минимизации размера дерева решений, его длины описания. Интересно, что минимальное дерево решений обеспечивает минимальные вычислительные затраты на классификацию новых примеров, т. е. строится не только самый короткий, но и самый быстрый классификационный алгоритм (к сожалению, для более широких классов алгоритмов минимизация длины описания не совпадает с минимизацией вычислительных затрат). Следует также упомянуть о связи поиска наиболее быстрых алгоритмов с принципом наименьшего действия, широко известным в физике. В частности, существует мнение, согласно которому имеет смысл интерпретировать физические процессы как оптимальные вычисления [102].

Последовательная проверка атрибутов также полезна в задачах планирования, в которых некоторые атрибуты появляются лишь после принятия решения о выполнении действия, связанного с проверкой предыдущего атрибута. К примеру, упрощенная инструкция по переходу дороги в виде набора правил может формулироваться так:

«Если на переходе есть работающий светофор, то подождать зеленый свет».

«Если горит зеленый свет, то переходить дорогу».

«Если нет работающего светофора, то посмотреть, есть ли машины».

«Если машин нет, то переходить дорогу».

Очевидно, что в виде линейной системы правил такая информация представима хуже, чем в виде дерева (рис. 4.4), так как в последнем порядок, в котором измеряются значения атрибутов, задается более явно.

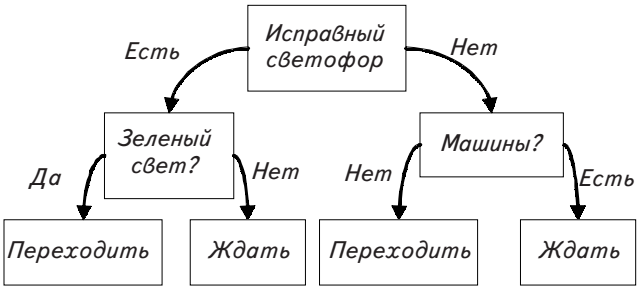

Рис. 4.4. Дерево решений, задающее порядок проверки атрибутов для упрощенного алгоритма перехода дороги

Итак, существуют задачи, в которых деревья решений предпочтительнее наборов правил, поэтому деревья решений заслуживают отдельного изучения. Рассмотрим задачу восстановления деревьев решений, в которой с атрибутами не связана стоимость получения их значения и значения всех атрибутов известны заранее. Такая ситуация более характерна для задачи обучения концептам, чем для задачи планирования.

Дерево решений представляет собой модель, в которой должны отражаться закономерности в данных наблюдений D. Как и при использовании набора правил, здесь может быть применен принцип МДО. Сначала перепишем уравнение (4.7)

$$DL = DL_{tree} + DL_{exceptions}, \qquad (4.10)$$

где DL_{tree} — длина описания соответствующего дерева решений; $DL_{exceptions}$ — как и ранее, длина описания исключений.

Если вернуться к аналогии с отправителем и получателем сообщения, то видно, что здесь, как и раньше, полагается, что левые части примеров обучающей выборки получателю известны априори. Отправитель должен передать модель (дерево решений), связывающую значения атрибутов с выходными классами, а также отклонения примеров обучающей выборки от этой модели (исключения).

В простом случае производится поиск только среди деревьев, которые точно описывают исходные данные (корректно классифицируют все имеющиеся примеры). Тогда задача построения дерева решений сводится к поиску наипростейшего дерева, согласованного с обучающей выборкой, а сообщение должно включать лишь закодированное дере-

во решений. Именно такой частный случай был сначала рассмотрен Квинланом [397, 398].

Интересно то, что, хотя сам Квинлан (см., например, [393, 399]), а также многие другие исследователи (см., например, [400–402]) апеллируют к принципу МДО, в обзорных книгах по искусственному интеллекту об этом принципе часто забывается. При этом указывается, что в основу минимизации размера дерева решений положен эвристический принцип бритвы Оккама и утверждается, что, «хотя эта идея основывается на интуитивных рассуждениях, ее можно проверить на практике» [74, с. 394]. В связи с этим еще раз отметим, что строгое теоретическое обоснование такого подхода давно существует.

Рассмотрим сначала вопрос о кодировании дерева решений. Воспользуемся классической схемой кодирования, предложенной Квинланом [399]. Кодирование будем начинать с корня дерева и далее записывать информацию о каждом узле в порядке обхода дерева (например, обход будет в ширину). Для каждого узла записывается следующая информация:

- является ли узел конечным (0, 1);
- если узел не конечный, то записывается атрибут, проверяемый в данном узле;
- если узел конечный, то записывается номер выходного класса.

На рис. 4.5 представлен пример дерева решений для трех атрибутов X_1, X_2 и X_3, первые два из которых являются бинарными, а третий может принимать три значения. Число классов равно трем: $\mathrm{A} = \{a_1, a_2, a_3\}$. Покажем по шагам, как будет формироваться описание этого дерева в виде строки:

корень: $1X_3$

первый уровень: $1X_2 0 a_2 1X_1$

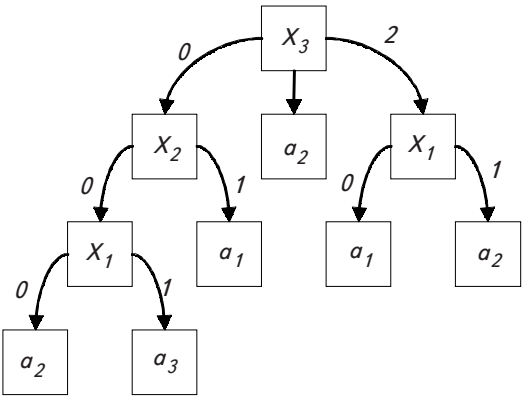

Рис. 4.5. Дерево решений, включающее проверку трех атрибутов X_1, X_2, X_3 и приводящее к выбору одного из трех классов: a_1, a_2, a_3

второй уровень: $1X_1 0a_1 0a_1 0a_3$

третий уровень: $0a_2 0a_3$

полное описание: $1X_3 1X_2 0a_2 1X_1 1X_1 0a_1 0a_1 0a_3 0a_2 0a_3$

Это описание сжимаемо. Первый символ в описании узла указывает, является ли узел листом. Пусть n_0 — количество листьев (в которые записаны результирующие классы); n_1 — число узлов, в которых осуществляется выбор. В нашем примере $n_0 = 6$ и $n_1 = 4$. Тогда требуется $-\log_2\left[n_0/(n_0+n_1)\right]$ бит для указания листа и $-\log_2\left[n_1/(n_0+n_1)\right]$ бит для указания неконечного узла. Для всех узлов на указание того, являются ли они листьями или нет, потребуется

$$-n_0 \log_2\left[n_0/(n_0+n_1)\right] - n_1 \log_2\left[n_1/(n_0+n_1)\right] \text{ бит.}$$

Общую суммарную длину этого элемента описания узлов можно грубо оценить как $n_0 + n_1$ бит (эта оценка точна, когда в дереве одинаковое число узлов и листьев: $n_0 = n_1$). Иногда этой оценкой и ограничиваются.

Поскольку листья могут содержать только номера классов, то для их описания требуется $n_0 H(\mathrm{A})$ (без учета длины таблицы перекодировки), где энтропия $H(\mathrm{A}) = -\sum_{a \in \mathrm{A}} P(a) \log_2 P(a)$. Здесь $P(a)$ — вероятность того, что в листе содержится указание на выбор класса a. В нашем примере $P(a) = 1/3$ для каждого из классов. Для упрощенной схемы кодирования в предположении о равновероятном появлении классов получаем $H(\mathrm{A}) = \log_2 d$, где d — общее число классов.

Также неконечные узлы могут содержать только указание на проверяемый атрибут. Для описания всех таких узлов требуется $-n_1 \sum_{i=1}^{N} P_i \log_2 P_i$, где P_i — частота проверки i-го атрибута в дереве решений. В нашем примере $P_1 = 2/4$, $P_2 = P_3 = 1/4$. Для упрощенной схемы кодирования в предположении $(\forall i, j) P_i = P_j$ это слагаемое будет равно $\log_2 N$.

Итак, длина описания дерева решений равна:

$$DL_{tree} = n_0 \left[-\log_2 \frac{n_0}{n_0+n_1} + H(\mathrm{A}) \right] + \\ + n_1 \left[-\log_2 \frac{n_1}{n_0+n_1} - \sum_{i=1}^{N} P_i \log_2 P_i \right]. \qquad (4.11)$$

В случае упрощенной схемы кодирования результирующая формула будет иметь вид

$$DL_{tree} = (n_0 + n_1) + n_0 \log_2 d + n_1 \log_2 N. \qquad (4.12)$$

В нашем примере длина описания дерева решений, вычисленная по формуле (4.11), примерно равна 25 бит, а вычисленная по приближенной формуле (4.12) — 26 бит. Формула (4.12) используется чаще благодаря своей простоте, однако в более сложных случаях она может давать заметное отклонение от более точной формулы (4.11), что может привести к выбору менее предпочтительного дерева решений.

Длина описания дерева решений не содержит явной зависимости от размера обучающей выборки. Если дерево решений правильно предсказывает классы для всех обучающих примеров, то выражение (4.11) является целевой функцией, которую нужно минимизировать выбором подходящего дерева решений.

Данное представление информации можно расширить, разрешив отправителю передавать такие деревья решений, которые позволяют корректно классифицировать не все обучающие примеры. Тогда помимо самого дерева решений отправитель должен передавать и информацию об ошибках классификации — только тогда получатель сможет правильно восстановить недостающую у него информацию о номерах классов.

Мы можем воспользоваться схемой кодирования исключений, уже описанной для наборов правил в п. 4.5.1. Для случая двух классов имеем

$$DL_{exceptions} = \log_2 C_M^{M_{FP}} + \log_2 M_{FP}. \qquad (4.13)$$

Напомним, что M — общее число примеров обучающей выборки; M_{FP} — число неверно классифицированных примеров. Слагаемое, связанное с кодированием пропусков, для деревьев решений обычно отсутствует, поскольку классификационный алгоритм на основе деревьев решений всегда дает на выходе какой-либо класс. Как мы увидим позже, возможность пропусков несложно реализовать и для деревьев решений, путем ввода дополнительного класса. Это бывает целесообразно для алгоритмов восстановления деревьев решений, так как обучающая выборка может не покрывать всех значений атрибутов.

Если для набора правил исключения могли кодироваться отдельно для каждого правила, то для деревьев решений может применяться кодирование исключений отдельно для каждого листа [400]. Для этого все примеры обучающей выборки разделяются на подвыборки в зависимости от того, какая из ветвей дерева решений используется при классификации конкретного примера (или в каком из листьев заканчивается анализ). В нашем примере (см. рис. 4.5) было бы шесть таких подвыборок. Далее кодирование ошибок осуществляется раздельно для каждой подвыборки, так что оценка длины описания ошибок будет включать сумму из выражений вида (4.13). Это может быть целесообразно в том случае, если размеры обучающей выборки достаточно большие по сравнению с числом листьев в дереве решений.

Не представляет трудности расширить эту схему кодирования и на случай многих классов [взяв соответствующее слагаемое из уравнения (4.9)]:

$$DL_{exceptions} = \sum_i \left(L(t_i) + m_i H(e_i) \right), \qquad (4.14)$$

где суммирование теперь осуществляется не по правилам из набора, а по листьям дерева решений. При таком способе кодирования в листьях дерева решений содержится не отдельный «истинный» класс, а перечень всех классов с указанием той вероятности, с которой соответствующий класс должен быть выбран, если классификационный алгоритм останавливается в данном листе.

Как и для всех остальных задач, рассмотренных в данной книге, в задаче построения дерева решений есть два крайних случая.

1. Чрезмерно упрощенное дерево решений («беспорядочная» гипотеза) имеет только корень, который допускает все классы с вероятностями $P(c = a_i)$. Здесь $P(c = a_i)$ — вероятность того, что некоторый пример обучающей выборки относится к классу a_i. Таким деревом любому примеру приписывается класс с той вероятностью, с которой этот класс встречается в обучающей выборке. Тогда суммарная длина описания примерно будет $m_0 H(e_0) = -M \sum_{i=1}^{d} P(c = a_i) \times$ $\times \log_2 P(c = a_i)$.

2. Чрезмерно усложненное дерево решений (гипотеза ad hoc) имеет число листьев, равное числу примеров обучаю-

щей выборки $n_0 = M$. Все примеры таким деревом классифицируются правильно (если только нет примеров с одинаковыми значениями атрибутов, но разными значениями классов). Для такого дерева порядок, в котором проверяются атрибуты, не имеет значения, и его длина описания может быть оценена одним слагаемым из выражения (4.11): $n_0 H(\text{A})$, где $n_0 = M$; $H(\text{A}) = -\sum_{i=1}^{d} P(c = a_i) \log_2 P(c = a_i)$, т. е. это дерево дает столь же высокую длину описания, что и максимально простое дерево решений.

Оптимальное решение, как и в других задачах, будет находиться между этими двумя крайностями. Здесь мы представили общие идеи вывода информационного критерия качества деревьев решений. Далее рассмотрим один из алгоритмов построения дерева решений путем оптимизации этого критерия качества.

4.5.3. «Жадные» алгоритмы построения деревьев решений

Деревья решений, по сравнению со всеми ранее рассмотренными нами задачами, предоставляют возможность наиболее непосредственного применения генетических алгоритмов и эволюционного программирования. Для применения генетических алгоритмов достаточно лишь немного модифицировать приведенную выше схему кодирования дерева решений в форме символьной строки, а для применения методов эволюционного программирования вполне удобно исходное представление в форме дерева решений. Эти методы стохастической оптимизации используются для построения деревьев решений, в том числе и совместно с принципом МДО [400]. Несмотря на сравнительную успешность применения генетических алгоритмов, воздержимся от их описания. Это не связано с какими-либо недостатками самих методов, но вызвано лишь нашим нежеланием говорить об этих методах слишком кратко, а подробное их обсуждение увело бы слишком далеко от основной темы. В связи с этим ограничимся классическими алгоритмами поиска, уже использованными в книге.

Легко заметить, что полный перебор деревьев нереализуем из-за своей вычислительной сложности: дерево, вклю-

чающее проверку всех атрибутов, содержит $|\mathrm{X}| = \prod_{i=1}^{N} |X_i|$ листьев, в каждом из которых может быть любое из d значений класса. Всего таких деревьев $d^{|\mathrm{X}|}$. Общее же число деревьев еще больше. Уже для сравнительно небольшого числа атрибутов полный перебор всех деревьев невозможен.

Однако существуют и весьма простые с вычислительной точки зрения алгоритмы построения деревьев решений, дающие удовлетворительный результат. Кратко опишем основную идею алгоритма ID3 (для более детального знакомства читатель может обратиться к книгам [74, с. 392–399; 209, с. 464–470] или к исходным работам Квинлана [397, 398]), затем покажем, что этот алгоритм подпадает под общую схему минимизации длины описания посредством жадных алгоритмов, и приведем другую возможную модификацию этой схемы.

В алгоритме ID3 дерево решений строится рекурсивно, начиная с корня. Сначала выбирается некоторый атрибут. На основе значений атрибута множество обучающих примеров разделяется на непересекающиеся подмножества, в каждом из которых значение этого атрибута постоянно. Выбранный атрибут помещается в текущий узел дерева решений, где проверяется его значение, и в зависимости от этого значения происходит перемещение по одной из исходящих ветвей в следующий узел. Для каждого узла следующего уровня производится та же процедура, но уже для соответствующего подмножества примеров обучающей выборки. Проиллюстрируем это на примере.

Пусть имеется три атрибута $X_1 = \{0, 1\}$, $X_2 = \{0, 1\}$, $X_3 = \{0, 1, 2\}$, три класса a_1, a_2, a_3, и обучающая выборка, состоящая из $M = 11$ примеров:

$x_1 = (0,0,0)$; $c_1 = a_2$; $x_2 = (1,0,0)$; $c_2 = a_3$; $x_3 = (0,1,0)$; $c_3 = a_1$;
$x_4 = (1,1,0)$; $c_4 = a_1$; $x_5 = (1,1,1)$; $c_5 = a_2$; $x_6 = (1,0,1)$; $c_6 = a_2$;
$x_7 = (0,1,1)$; $c_7 = a_2$; $x_8 = (0,0,2)$; $c_8 = a_1$; $x_9 = (0,1,2)$; $c_9 = a_1$;
$x_{10} = (1,0,2)$; $c_{10} = a_3$; $x_{11} = (1,1,2)$; $c_{11} = a_3$.

На рис. 4.6 представлена схема построения дерева решений, при котором атрибуты тестируются в порядке возрастания их номеров, начиная с атрибута X_1. Для получения результирующего дерева на рис. 4.6 достаточно лишь ис-

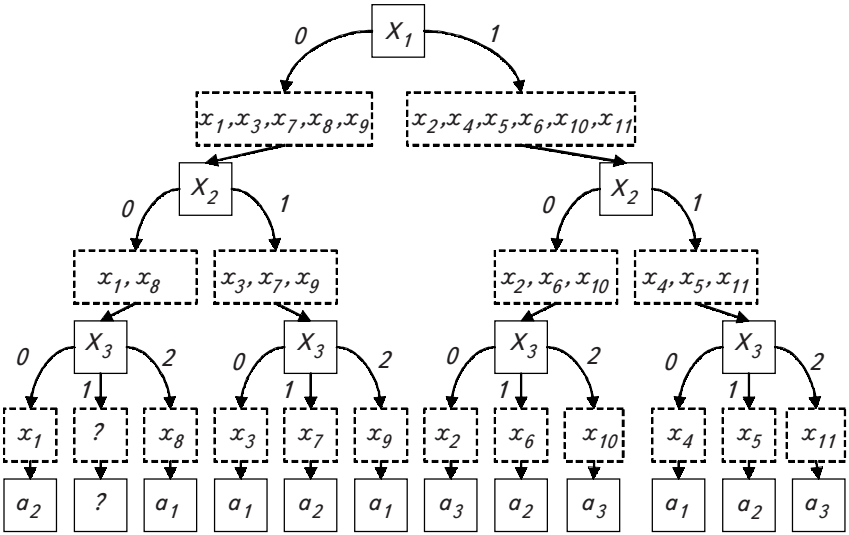

Рис. 4.6. Схема рекурсивного построения дерева решений с указанием подмножеств, на которые разделяется обучающая выборка в результате проверки атрибутов (порядок проверки атрибутов не изменяется)

ключить информацию о подмножествах, на которые разбиваются примеры обучающей выборки. Отметим, что подобное дерево решений может содержать узлы, которым не соответствует ни один пример обучающей выборки, а значит, отсутствует информация о том, какие классы должны содержаться в листьях, являющихся потомками этого узла. Это приводит к необходимости введения пропусков как некоторого дополнительного класса «?».

Как видно, такое рекурсивное построение дерева решений — это просто вполне очевидный способ формирования дерева ad hoc. Однако в алгоритме ID3 есть принципиальный прием, которым мы пока не воспользовались, — это выбор порядка проверки атрибутов по степени их информативности. Для рассмотрения этого приема нужно сформулировать понятие информативности признака.

Исходно присутствует неопределенность в номере класса, к которому относится данный объект. При выполнении классификации эта неопределенность устраняется за счет информации, содержащейся в атрибутах. Если для выполнения классификации требуется, скажем, один атрибут, принимающий с равной вероятностью одно из двух значений, то для выполнения классификации нужен один бит инфор-

мации. Если же требуется три таких атрибута, то неопределенность в номере класса будет равна восьми битам.

Информативность одного атрибута можно определить как разницу в количестве информации, требуемой для выполнения классификации до и после использования этого атрибута. Здесь полезно рассмотреть «беспорядочную» гипотезу, суть которой заключается в том, что она допускает на выходе любой класс с некоторой вероятностью. Пусть на основе обучающей выборки построено распределение вероятностей по классам $P(a)$. Тогда энтропия этого распределения, умноженная на размер обучающей выборки, будет определять длину описания «беспорядочной» гипотезы и будет ограничивать сверху количество информации, необходимой для завершения классификации (при описании алгоритма ID3 эта величина обычно называется количеством информации, необходимым для завершения построения дерева решений): $MH(\mathrm{A}) = -M\sum_{i=1}^{d} P(a_i)\log_2 P(a_i)$.

После использования какого-то атрибута выборка разделяется на подмножества, в каждом из которых имеется собственное распределение конечных классов. Наиболее предпочтителен тот атрибут, после проверки которого происходит наиболее четкое разделение гистограммы классов в подмножествах. Для индифферентного (по отношению к изучаемому концепту) атрибута гистограммы классов в подмножествах будут совпадать с гистограммой классов исходного множества, их энтропии будут равны и для завершения классификации примеров обучающей выборки потребуется столько же информации, сколько требовалось до проверки атрибута, т. е. информативность этого атрибута будет равна нулю. Приведем конкретные математические выражения.

Пусть в зависимости от значения атрибута выборка из M элементов разделяется на K выборок, причем в j-й выборке M_j элементов, дающих распределение вероятностей по классам $P_j(a)$. Тогда информативность атрибута будет:

$$I = MH(\mathrm{A}) - \sum_{j=1}^{K} M_j H_j(\mathrm{A}) =$$

$$= -M\sum_{i=1}^{d} P(a_i)\log_2 P(a_i) + \sum_{j=1}^{K} M_j \sum_{i=1}^{d} P_j(a_i)\log_2 P_j(a_i). \quad (4.15)$$

Так, в нашем примере исходное распределение вероятностей классов

$$P(a_1) = 4/11;\ P(a_2) = 4/11;\ P(a_3) = 3/11 \text{ и } M = 11,$$

т. е. исходная неопределенность номера классов для всех примеров составляет примерно 17,3 бит.

Атрибуты производят разделение на подмножества со следующими распределениями:

для X_1 —

$$P_1(a_1) = 3/5;\ P_1(a_2) = 2/5;\ P_1(a_3) = 0/5\ \ (M_1 = 5);$$

$$P_2(a_1) = 1/6;\ P_2(a_2) = 2/6;\ P_2(a_3) = 3/6\ \ (M_2 = 6);$$

$$M_1H_1(\text{A}) + M_2H_2(\text{A}) = 4,85 + 8,75 = 13,6 \text{ бит};$$

для X_2 —

$$P_1(a_1) = 1/5;\ P_1(a_2) = 2/5;\ P_1(a_3) = 2/5\ \ (M_1 = 5);$$

$$P_2(a_1) = 3/6;\ P_2(a_2) = 2/6;\ P_2(a_3) = 1/6\ \ (M_2 = 6);$$

$$M_1H_1(\text{A}) + M_2H_2(\text{A}) = 7,6 + 8,75 = 16,35 \text{ бит};$$

для X_3 —

$$P_1(a_1) = 2/4;\ P_1(a_2) = 1/4;\ P_1(a_3) = 1/4\ \ (M_1 = 4);$$

$$P_2(a_1) = 0/3;\ P_2(a_2) = 3/3;\ P_2(a_3) = 0/3\ \ (M_2 = 3);$$

$$P_3(a_1) = 2/4;\ P_3(a_2) = 0/4;\ P_3(a_3) = 2/4\ \ (M_3 = 4);$$

$$M_1H_1(\text{A}) + M_2H_2(\text{A}) + M_3H_3(\text{A}) = 6 + 0 + 4 = 10 \text{ бит}.$$

Итак, наибольшей информативностью обладает третий атрибут. Таким образом, он должен быть помещен в корень дерева. Выбор следующего атрибута должен осуществляться заново, причем отдельно для каждой ветви. Причина, по которой необходимо пересчитывать информативность атрибутов, вполне очевидна: проверенный атрибут может обладать разным количеством взаимной информации с другими атрибутами, т. е. смотреть следует не просто на количество информации о классах, которую несут атрибуты, а на количество новой информации.

На рис. 4.7 представлен первый шаг построения дерева решений, начиная с наиболее информативного атрибута X_3. Этот атрибут разбивает исходную выборку на три подмножества. Видно, что все примеры x_5, x_6, x_7 относятся к клас-

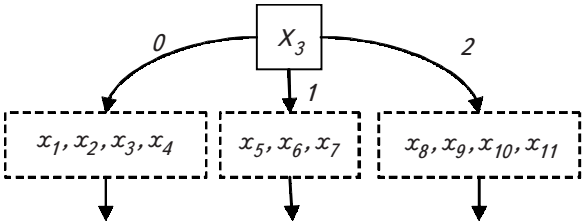

Рис. 4.7. Первый шаг выращивания дерева решений: помещение в корень проверки атрибута X_3, который разделяет обучающую выборку на три подмножества

су a_2, т. е. дальнейшую проверку атрибутов в этой ветви вести не нужно. Нетрудно определить, что для левой ветви далее должен тестироваться атрибут X_2, а для правой — X_1. В частности, для правой ветви атрибут X_1 разбивает множество примеров на два подмножества с распределениями вероятностей по классам $P_1(a_1) = 1$ и $P_2(a_2) = 1$, т. е. этот атрибут содержит информацию, достаточную для окончательной классификации примеров. Таким образом, на одном уровне дерева в разных его ветвях могут проверяться разные атрибуты. Если завершить построение дерева решений этим алгоритмом, то получим дерево, представленное на рис. 4.5. Это дерево гораздо компактнее дерева, представленного на рис. 4.6, при построении которого порядок проверки атрибутов не выбирался. При этом, очевидно, произошло и обобщение. В частности, деревом, представленным на рис. 4.5, пример (0, 0,1) классифицируется как a_2, а деревом на рис. 4.6 он считается пропуском. При малом размере выборки (или большом числе атрибутов) или зашумленных данных это обобщение было бы еще более принципиальным.

Подобный выбор порядка атрибутов можно рассматривать как итеративное улучшение (в смысле длины описания) «беспорядочного» стохастического дерева решений посредством «жадного» алгоритма. Действительно, стохастическое дерево решений с единственным узлом для нашего примера будет содержать один узел с уже приведенным распределением вероятностей $P(a)$ (рис. 4.8).

После замены этого узла узлом, в котором выполняется проверка атрибута X_3, получаем дерево решений, содержащее на три узла больше (что обычно обозначается как *выращивание* дерева). Вместо того чтобы считать объект, пред-

$a_1(P = 4/11); a_2(P = 4/11); a_3(P = 3/11)$

Рис. 4.8. «Беспорядочное» дерево решений, состоящее из единственного корневого узла, являющегося также и листом и содержащего в себе все конечные классы с указанными вероятностями

ставленный на рис. 4.7, некоторой промежуточной конструкцией, его вполне можно трактовать как вполне корректное дерево решений (рис. 4.9).

Тогда информативность признака, выражаемая формулой (4.15), представляет собой не что иное, как изменение длины описания (4.10) в результате добавления в дерево решений проверки дополнительного атрибута. Алгоритм ID3 является «жадным», поскольку на каждой его итерации дерево выращивается так, чтобы уменьшение длины описания было максимальным.

Отсюда видно, что формула (4.15) не совсем точна, так как в ней учитывается только длина описания исключений [см. формулу (4.14)], при этом не учитывается изменение длины описания самого дерева решений. Помещение в текущий лист нового атрибута вызывает изменение размера дерева на K узлов. Для разных атрибутов это значение может быть разным. Несложно представить такую ситуацию, когда значение K столь велико, что разбивает обучающую выборку на подмножества, в каждом из которых содержится лишь по одному примеру. По формуле (4.15) такой атрибут будет идеальным. В действительности же он ведет к построению гипотезы ad hoc и не несет информации. Если воспользоваться подсчетом изменения полной длины описания, введенной в п. 4.5.2, то окажется, что такой атрибут ничуть не лучше атрибута, принимающего лишь одно

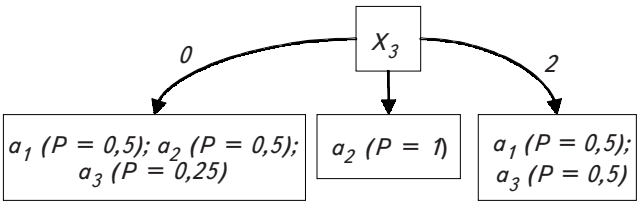

Рис. 4.9. Пример стохастического дерева решений, получившегося после первого шага процедуры выращивания

значение: $K = 1$. В процессе выращивания дерева решений такие атрибуты никогда не должны выбираться при наличии других атрибутов. Таким образом, мы видим, что эвристически введенная информативность атрибутов не является вполне адекватной задаче, в то время как обращение к принципу МДО предохраняет от ошибок.

Итак, алгоритм ID3 вполне подпадает под общую схему итеративной минимизации длины описания. В наиболее общем случае в рамках этой схемы выполняются следующие шаги:

- конструируется начальное дерево решений («беспорядочное» или ad hoc);
- среди множества допустимых операций по преобразованию дерева решений выбирается та, которая позволяет в наибольшей степени уменьшить суммарную длину описания (4.10);
- выбор операции повторяется до тех пор, пока есть хоть одна операция, приводящая к уменьшению длины описания.

В алгоритме ID3, как уже отмечалось, итеративное улучшение начинается с «беспорядочного» дерева, а число операций преобразования весьма ограничено (есть лишь операция по замене листа узлом с проверкой одного из атрибутов).

Мы на примере показали, что выбор порядка проверки атрибутов в алгоритме ID3 ведет к уменьшению суммарной длины описания. Однако это делается посредством «жадного» алгоритма поиска, которым на некотором шаге может быть принято неоптимальное (в глобальном плане) решение. Принятые решения при дальнейшем выращивании дерева не пересматриваются. Реализовать же такой пересмотр можно, организовав перебор на глубину, большую единицы.

Алгоритм построения дерева также может начинаться не с «беспорядочного», а с ad hoc-дерева (например, подобного представленному на рис. 4.6). Это дерево классифицирует все примеры обучающей выборки правильно, а все примеры, не вошедшие в обучающую выборку, как пропуски. Далее осуществляется последовательное упрощение дерева на основе каких-либо операций по преобразованию дерева. При этом, как и раньше, на каждом шаге выбирается та операция, которая приводит к максимальному уменьшению длины описания. Обычно рассматривается только один тип операций: удаление какого-то узла дерева со всеми его по-

томками. Как правило, этот тип операций называется *отсечением* (pruning). Порядок проверки атрибутов при этом не меняется, что делает отсечение довольно слабым средством по преобразованию деревьев решений, которое не может применяться самостоятельно, а лишь улучшает дерево, построенное некоторым другим способом. Например, в работе [401] отсечение дерева (выполняемое на основе принципа МДО) осуществляется после его выращивания. Более сложные операции по преобразованию деревьев (перестановка родительского и дочернего узлов, исключение некоторого узла без удаления его потомков и т. д.) исследованы меньше. В частности, указанные операции имеют ясный смысл только тогда, когда во всех дочерних узлах модифицируемого узла тестируется один и тот же атрибут.

Итак, отсечение и выращивание деревьев решений укладываются в общую схему итеративной оптимизации длины описания. Операции по добавлению узлов и их удалению могут использоваться совместно, а также могут быть дополнены некоторыми другими операциями преобразования деревьев. Сам же оптимизационный алгоритм может быть абстрагирован от того, что именно подвержено оптимизации, коль скоро определены целевая функция и операции преобразования. Это полностью согласуется с методами эвристического программирования, поэтому для построения деревьев решений могут использоваться не только «жадный» алгоритм и нерассмотренные здесь генетические алгоритмы, но и другие методы поиска.

4.5.4. Ограничения представления информации в форме деревьев решений

У деревьев решений есть ряд недостатков. В частности, нередко рассматриваются деревья решений, содержащие в листьях указание на единственный класс. Если в обучающей выборке есть примеры с идентичными атрибутами, но разными значениями классов (например, в случае зашумленных данных), то это, в принципе, не может быть выражено подобным деревом решений. Как указывалось ранее, подобная проблема легко обходится при использовании стохастических деревьев решений, которые (как и стохастические грамматики или наборы правил) вполне естественным образом возникают в рамках информационного подхода.

Общеизвестны и более глубокие недостатки деревьев решений как представления концептов, которые не разрешаются путем привлечения стохастичности. Эти недостатки связаны с тем, что структура деревьев решений не позволяет выразить некоторые регулярности данных, определяемые изучаемым концептом. Наиболее существенными являются проблемы репликации и фрагментации.

Проблема *репликации* связана с тем, что для представления некоторых концептов оказывается необходимым в точности дублировать часть поддеревьев. Проблема *фрагментации* заключается в том, что атрибуты, имеющие большое количество значений, разбивают обучающую выборку на большое число подмножеств малого размера.

Рассмотрим пример дерева, представленного на рис. 4.10. Поддеревья T_1 и T_2 могут быть очень большого размера. Если переставить местами тестирование атрибутов X_1 и X_2, то абсолютно ничего не изменится. Попытка поставить поддеревья T_1 и T_2 выше проверки атрибутов X_1 и X_2 (возможность этого зависит от структуры самих поддеревьев) может привести дерево к виду, где перед каждым из многочисленных листьев поддеревьев T_1 и T_2 будет осуществляться однотипное тестирование атрибутов X_1 и X_2. Иными словами, такое дерево может быть практически несжимаемым. В связи с этим длина описания не зависит от того, сколько из поддеревьев в точности совпадает. Приведенное дерево решений, в котором две пары поддеревьев совпадают, будет таким же по длине описания, что и дерево, в котором все четыре поддерева различны. Очевидно, первый случай существенно проще. Чем это плохо?

Давайте вспомним рекурсивный алгоритм построения дерева решений ID3. В этом алгоритме каждое поддерево строилось на основе тех примеров обучающей выборки, которые прошли тесты, задаваемые атрибутами, помещенными в родительские узлы. В связи с этим каждое из двух совпадающих поддеревьев T_1, как и каждое из двух поддере-

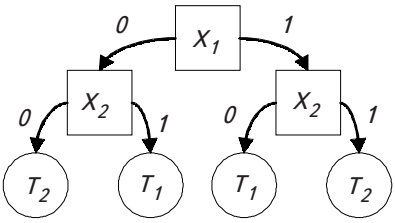

Рис. 4.10. Дерево решений, содержащее дублирующиеся поддеревья T_1 и T_2, которые могут быть очень большого размера

вьев T_2, строятся на основе только части примеров выборки, имеющих соответствующие значения атрибутов X_1 и X_2. Значит, для их построения используется меньшее количество информации, чем это могло быть, и они в большей степени будут подвержены влиянию шума. А в случае выборок малых размеров количества информации может оказаться недостаточно для восстановления каких-либо из этих поддеревьев. Заметим, что эта проблема репликации не является особенностью алгоритма построения дерева решений сверху вниз (такого, как ID3), но является общей проблемой для деревьев решений.

Проблема фрагментации имеет сходный смысл, что и проблема репликации. Чтобы это стало очевидным, представим, что атрибуту «вес», имевшему два значения — «маленький» и «большой», придается множество абсолютных значений, каждое из которых отличается, скажем, на один грамм. Тогда этот атрибут разделит обучающую выборку на подмножества, в каждом из которых окажется преимущественно по одному элементу. Как уже отмечалось, это ничем не отличается от гипотезы ad hoc, которая не выполняет никакого обобщения. При использовании выбора порядка признаков на основе принципа МДО будет наблюдаться тенденция к выбору атрибутов с большим количеством значений в последнюю очередь. Но если все атрибуты будут обладать таким свойством, то на их основе будет, в принципе, невозможно построить хорошее дерево решений. Несмотря на утрированный характер этого примера, такие ситуации вполне могут встретиться на практике, хотя и не в столь выраженном виде.

Проблемы репликации и фрагментации связаны и могут проявляться совместно (рис. 4.11). Обе проблемы вызывают уменьшение числа обучающих примеров в нижних узлах дерева решений. Как уже отмечалось, эти примеры несут информацию, необходимую для построения дерева решений. Отсутствие репрезентативных выборок примеров приводит

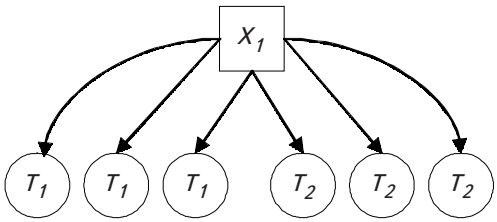

Рис. 4.11. Дерево решений, содержащее проверку атрибута большой арности, разные значения которого содержательно не отличаются, что приводит к многократному повторению поддеревьев

к формированию статистически недостоверных тестов в нижних узлах дерева. В конечном итоге, это отрицательным образом сказывается на качестве обобщения (или на требуемом объеме выборки), т. е. на точности предсказания классов новых примеров, не вошедших в выборку.

Подобные проблемы могут решаться изменением атрибутов (введением новых или уменьшением количества значений у имеющихся атрибутов). В нашем примере новый атрибут X' такой, что $X' = 0$, если $X_1 = 0 \,\&\, X_2 = 0 \vee X_1 = 1 \,\&\, X_2 = 1$, и $X' = 1$ в противном случае, даст весьма простое дерево решений. Заметим, что задача построения новых атрибутов граничит с задачей концептуальной кластеризации, которой мы, к сожалению, не касались.

В более ранних работах [403, 404] указанные проблемы (в основном проблема репликации) решались напрямую путем предварительного выращивания дерева решений и последующей идентификации дублирующихся поддеревьев. В частности, в работе [404] для вывода новых атрибутов обрабатывается край дерева (каждая пара узлов, расположенных над листьями). Новые атрибуты добавляются в список атрибутов, и дерево полностью выращивается заново. В работе [405] также проводилась попытка поиска дублирующихся поддеревьев. Подобные подходы, однако, наталкиваются на сложности идентификации совпадающих поддеревьев, особенно в случае, когда один и тот же концепт может быть выражен разными способами (что, в частности, характерно для булевых концептов, особенно когда используется неканоническое их представление).

Позднее были предложены расширения представления в виде деревьев решений. Эти представления в той или иной степени преодолевают проблемы репликации и фрагментации.

4.5.5. Представления, расширяющие деревья решений

Граф решений — это ориентированный корневой ациклический граф, в котором каждый нетерминальный узел (т. е. узел, у которого есть выходящие дуги) помечен номером атрибута, проверяемого в этом узле. Дуги, выходящие из данного узла, ставятся в соответствие со значениями атрибута, проверяемого в этом узле. Терминальные узлы (подобные листьям деревьев решений) содержат номера классов (или распределение вероятностей по номерам классов в случае

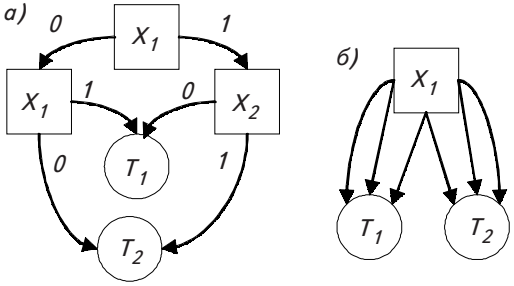

Рис. 4.12. Графы решений, соответствующие деревьям решений, представленным на рис. 4.10 (*а*) и 4.11 (*б*)

стохастических графов решений). Как видно, графы решений похожи на деревья решений, за тем исключением, что в них допускаются сходящиеся пути (в дереве решений каждый узел обладает не более чем одной входящей дугой). Как видно из рис. 4.12, введение сходящихся путей позволяет упростить деревья, представленные на рис. 4.10 и 4.11, т. е. позволяет смягчить проблемы репликации и фрагментации. Представление в виде графов решений позволяет получать более компактные (по сравнению с представлением в виде деревьев решений) описания наблюдательных данных во многих предметных областях, следовательно, оно позволяет строить более точные предсказания относительно примеров, не вошедших в обучающую выборку. Этим объясняется интерес исследователей к данному представлению [406–410].

Для восстановления графов решений были предложены алгоритмы, в которых, как и в случае поиска атрибутов, производится обнаружение дублирующихся поддеревьев в уже выращенном дереве решений, а также алгоритмы, в которых выращивается сам граф. Следует отметить, что и в том и в другом случае алгоритм выращивания или редукции графа часто осуществляется под управлением принципа МДО (см. [406, 409, 411] и [407, 408, 410] соответственно). Вывод формулы длины описания для графа решений не многим сложнее вывода формулы для графа решений. Различие между ними заключается лишь в том, что в случае графа решений для каждой дуги необходимо в явном виде указывать узел, в который эта дуга входит. Это означает, что дерево решений, представленное как граф решений без изменения внутренней структуры, будет обладать большей длиной описания. Если изучаемые концепты сводятся именно к таким деревьям, то их описание посредством графов решений будет менее предпочтительным. Это еще раз

подчеркивает, что не существует универсального представления, которое было бы лучшим для любой задачи, и выбор представления — это эмпирическая проблема, которая должна решаться на основе некоторого количества обучающих выборок.

К сожалению, восстановление графов решений оказывается задачей более сложной, чем восстановление деревьев решений, поскольку само это представление является менее жестким. В связи с этим часто вводят некоторые ограничения на вид графов решений. Существуют такие разновидности графов решений, как бинарные диаграммы решений [412], упорядоченные и редуцированные графы решений [407, 408] или графы решений с четким делением на уровни [409, 410].

В бинарных диаграммах из каждого нетерминального узла исходит две дуги (сам узел имеет смысл «если, то... иначе...»), т. е. ограничение накладывается на атрибуты исходных объектов, а не на их представление. Редуцированные графы — это графы, в которых: 1) не существует таких двух узлов, все исходящие дуги которых ведут в совпадающие узлы; 2) не существует такого узла, все исходящие дуги которого ведут в один и тот же узел. Эти ограничения разумны с точки зрения принципа МДО: их введение не сужает выразительной силы представления.

В ориентированных графах существует общий порядок атрибутов такой, что движение по любому пути в этом графе не нарушает данного порядка. Похожее ограничение имеется и в графах решений со строгим делением на уровни, во всех узлах одного и того же уровня проверяется один и тот же атрибут. Это ограничение достаточно сильное. В частности, при обсуждении алгоритма ID3 мы обратили внимание на полезность установления порядка проверки атрибутов для каждого поддерева в отдельности. Таким образом, не каждое дерево решений будет эффективно представляться ориентированным графом решений. В то же время эти графы частично решают проблемы фрагментации и репликации, характерные для деревьев решений, а также существуют вычислительно эффективные алгоритмы восстановления ориентированных графов решений (см., например, [406, 408, 410]).

Выращивание и упрощение графа решений подпадает под общую схему итеративной минимизации информационной целевой функции. Однако для графов решений на-

бор операций по их преобразованию должен быть расширен. В частности, при выращивании графа решений оказывается необходимой операция по слиянию двух терминальных узлов, которая рассматривается наряду с операцией по замене терминального узла нетерминальным узлом, содержащим проверку некоторого атрибута [406].

Как отмечается в работе [413], графы решений преодолевают проблему репликации не полностью, но лишь для таких дублирующихся поддеревьев, все пути в которых заканчиваются в листьях. Иными словами, графы решений не способны описывать структуру, повторяющуюся внутри дерева решений (рис. 4.13). Другим естественным расширением деревьев решений является представление информации в виде связанного леса решений.

В рамках этого представления концепты описываются не одним, а несколькими деревьями решений, ссылающимися друг на друга. Каждое такое дерево отвечает некоторому субконцепту (за исключением дерева, отмеченного как корневое). Изображенные на рис. 4.10 и 4.11 деревья можно рассматривать как корневые деревья, которые ссылаются на деревья T_1 и T_2, относящиеся к тому же лесу решений. Деревья T_1 и T_2 должны иметь тот же вид, что и обычные деревья решений, за тем исключением, что у них имеются соответствующие метки (T_1 и T_2).

Этого, однако, недостаточно, чтобы описывать повторение внутренней структуры в деревьях решений. В связи с этим в работе [413] вводятся два типа ссылок: ссылки по значению и ссылки по атрибуту. Ссылки по значению имеют смысл инструкции перехода: по этой ссылке просто осуществляется переход на соответствующее дерево леса, которое и управляет дальнейшей процедурой классификации.

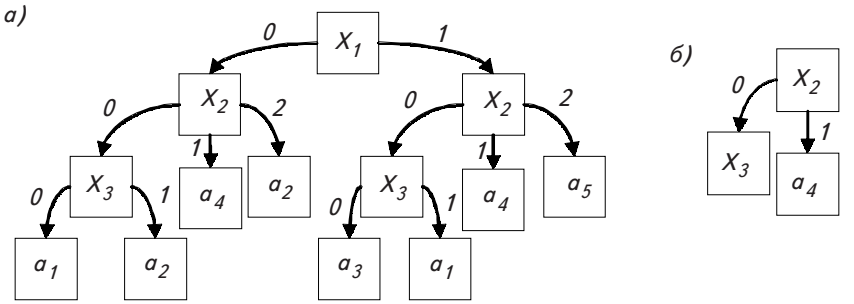

Рис. 4.13. Дерево решений (*а*) с повторяющейся внутренней структурой (*б*)

Ссылки по атрибуту аналогичны вызову функций: после достижения листа такого дерева происходит возврат в точку вызова с передачей метки листа, который был выбран на основе текущих значений атрибутов. Узел, в котором содержится ссылка по атрибуту, должен иметь столько выходных дуг, сколько у вызываемого дерева листьев, не содержащих номеров конечных классов. Дальнейшее продвижение осуществляется по дуге, соответствующей выбранному листу. Так, дерево, представленное на рис. 4.13, *а*, может быть описано с помощью леса решений, включающего ссылку по атрибуту (рис. 4.14).

Восстановление связанного леса решений может осуществляться путем его выращивания. Эта процедура сходна с выращиванием графа решений, но вместо объединения двух листьев здесь в эти два листа должна помещаться ссылка на новое дерево. В работе [413] предлагается алгоритм «дровосека», в котором операции по выращиванию дерева решений имеют более сложный вид: в новое дерево леса выделяются не совпадающие листья, а совпадающие двойные узлы, каждый из которых состоит из родительского узла и его потомка. При этом допускается, что двойные узлы являются внутренними. Это позволяет формировать ссылки не только по значению, но и по атрибуту.

В работе [413] указывается связь между связанным лесом решений и порождающими грамматиками, состоящими из многих правил подстановки. Здесь роль порождающих правил играют отдельные деревья. Аналогия будет более полной, если деревьям решений будет разрешено иметь циклические ссылки (такие леса решений, однако, малоинтересны в задачах, для решения которых они обычно используются). Лес решений можно также трактовать и как формальное описание алгоритма. Если продолжать дальше аналогию

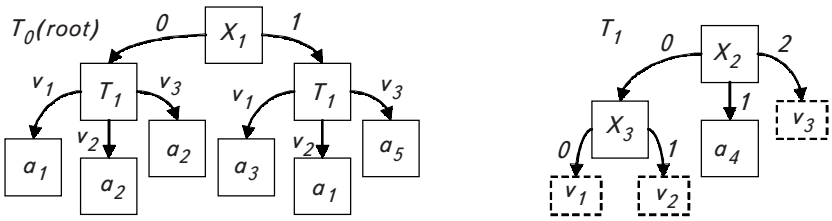

Рис. 4.14. Связанный лес решений, состоящий из двух деревьев — T_0 и T_1, решающий проблему репликации для внутренних частей дерева, представленного на рис. 4.13, *а*

с языком программирования, то может быть предложено расширение способов взаимных вызовов между деревьями. В частности, вызов может содержать параметры, задающие подстановку атрибутов. Например, если в исходном дереве встречается проверка выражений $X_1 \& X_2 \vee \neg X_1 \& \neg X_2$ и $X_3 \& X_4 \vee \neg X_3 \& \neg X_4$, то структура этих выражений может быть описана в качестве одного дерева T, которому будут передаваться конкретные атрибуты для проверки: $T(X_1, X_2)$ и $T(X_3, X_4)$.

Другая возможность заключается в том, чтобы совместить идею графов решений и идею леса решений. Действительно, нет никаких препятствий (разве что заключающихся в сложности алгоритмов поиска) для того, чтобы в лесе решений вместо деревьев использовались бы графы. Достаточно близким по смыслу является привлечение при конструировании графов решений составных атрибутов. Такой прием используется, например, в работе [408] при инициализации графов решений. Все эти возможности, однако, на данный момент исследованы мало.

4.5.6. Обсуждение символьных представлений

Мы рассмотрели несколько типов символьных представлений, таких как наборы правил, деревья решений, графы решений и связанные леса решений. Все эти представления имеют сходные приложения, но задают несколько отличающиеся априорные распределения и обладают разной сложностью вывода. Помимо отличий с точки зрения индуктивного вывода, указанные представления обладают определенными особенностями и с точки зрения их практического использования. Такие вопросы, как, скажем, разрешение конфликтов в продукционных системах, имеют отдаленное отношение к теме данной книги, однако при выборе представления в целях решения некоторой задачи их необходимо учитывать наряду с вопросами выразительной силы и сложности вывода.

Несмотря на то что описанные представления информации являются более узкими, чем неограниченные формальные грамматики, они широко применяются при решении таких задач, как обучение концептам или планирование. Это говорит о том, что такого рода представления адекватно описывают соответствующие предметные области и при

построении системы машинного обучения общего назначения должны быть включены в ее инструментарий.

Концепцию формальных грамматик можно трактовать как метапредставление, в рамках которого переход от одного представления к другому может осуществляться инкрементно путем разрешения или запрещения правил определенного вида. Так, не представляет трудности предложить общую схему перехода между грамматиками различных типов. Рассмотренные представления также имеют много общего, в частности, исторически графы и связанные леса решений были введены как расширения деревьев решений. Более того, после введения графов решений были предложены некоторые их упрощения (такие, как бинарные диаграммы решений и редуцированные ориентированные графы решений). Таким образом, интересной проблемой является разработка метапредставления, которое бы позволило формально описать переход между различными частными представлениями, имеющими отношение к деревьям и графам решений.

В результате сведения задачи восстановления символьных представлений к задаче индуктивного вывода становится ясно, что это задачи распознавания образов в дискретном пространстве признаков (атрибутов). Хотя любая задача распознавания образов формулируется одинаково — как восстановление отображения из одного множества в другое, — дискриминантные методы распознавания существенно отличаются от методов восстановления символьных представлений.

Указанное различие не является непреодолимым. Действительно, при увеличении арности атрибутов возникает проблема фрагментации обучающей выборки, которая разрешается в графах решений так, как показано на рис. 4.12, т. е. разные значения одного и того же атрибута трактуются как эквивалентные. Когда в каком-то дискриминантном методе проводится разделяющая поверхность, происходит примерно то же самое: значения признаков по каждую сторону поверхности принимаются эквивалентными в том плане, что они разделяют обучающую выборку на соответствующие подмножества. Разница же здесь заключается в том, что в дискриминантных методах такое объединение значений одного признака описывается единственной точкой на числовой оси, так как имеет место предположение непрерывности. В дискретном случае для каждого значения атрибута необходимо указывать, к какому из дочерних узлов ведет соответствующая дуга. К при-

меру, на рис. 4.12, *б* дуги, выходящие из узла X_1, могут в произвольном порядке вести то к узлу T_1, то к узлу T_2. Если арность атрибута возрастает очень сильно, то указание узла, в который входит каждая дуга, становится очень дорогим с точки зрения длины описания. Однако несложно изменить представление таким образом, чтобы для каждого дочернего узла было достаточно указывать лишь крайние дуги, входящие в этот узел, а все промежуточные дуги также считались бы входящими в него. Иными словами, длина описания оказывается пропорциональной числу дочерних узлов, а не арности атрибута. Естественно, это является дополнительным ограничением на структуру графа решений, которое далеко не всегда будет адекватным.

Другое видимое различие между дискриминантными и дискретными методами распознавания заключается в том, что в дискриминантных методах разделяющие поверхности могут иметь любой вид, в то время как в дискретных методах они проходят перпендикулярно к осям и имеют вид $X_i = \text{const}$ (если атрибуты имеют нечисловые значения, то по оси можно отложить порядковый номер соответствующего значения). В действительности, это различие тоже не является непреодолимым. К примеру, составные атрибуты (или некорневые деревья в связанном лесе) могут иметь вид, представленный на рис. 4.15. Значение такого атрибута будет не чем иным, как суммой значений двух других атрибутов. Естественно, такое ad hoc-описание процесса суммирования является крайне неэффективным, что говорит об ограниченности представлений в форме деревьев решений или наборов правил: с их помощью можно описать любую операцию над атрибутами, но все эти описания будут одинаково неэффективными. Однако мы видим, что дискретные и непрерывные методы распознавания — лишь два крайних случая, показавших свою полезность на практике. Это вовсе не значит,

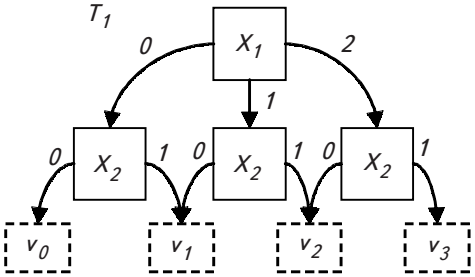

Рис. 4.15. Некорневой граф решений, описывающий операцию сложения атрибутов X_1 и X_2

что от них нужно отказаться в пользу некоторого общего метода. Тем не менее поиск представлений, расположенных между этими двумя крайностями, может быть полезным. Многоуровневые системы распознавания являются привлекательной, но не единственной возможностью.

4.6. ЗАКЛЮЧЕНИЕ

Исследование игровых задач, проблем автоматического перевода, доказательства теорем, планирования и принятия решений привело к формулированию гипотезы символьной физической системы, важное место в которой занимают символьные представления информации, такие как формальные грамматики, а также представления, родственные наборам правил: деревья и графы решений и связанные леса решений. Задача автоматического восстановления этих представлений имеет большое значение для области искусственного интеллекта.

Отличительной особенностью символьных представлений является то, что они строятся на основе наблюдательных данных, в которых каждый символ не обладает никакими свойствами, кроме собственной уникальности. Для любых двух символов мы можем сказать лишь то, являются ли они одинаковыми или различными. Численные данные, напротив, обладают богатым набором свойств, которые направляют поиск регулярностей в этих данных. И в связи с этим основной трудностью при восстановлении символьных представлений является не большой объем исходных данных, что имеет место в случае интерпретации необработанной сенсорной информации, а большая степень априорной неопределенности возможных связей между отдельными элементами данных, что приводит к необходимости привлечения перебора вариантов в процессе построения символьных представлений. Автоматическое построение адекватного по вычислительной сложности алгоритма поиска является чрезвычайно важной проблемой.

Помимо алгоритмов поиска при восстановлении символьных представлений большое значение имеет корректная постановка задачи анализа данных, которую необходимо решить. В частности, необходимо строго задать целевую функцию, что и является основным предметом данной книги. Мы показали, что для всех задач, связанных с восстановле-

нием символьных представлений, могут быть привлечены теоретико-информационный подход и принцип минимальной длины описания. Это позволит избежать ошибок, неизбежных при формулировании задач на уровне здравого смысла, а также рельефно выделить ограничения, связанные с тем или иным методом, что, в конечном счете, помогает улучшить решения, получаемые эвристически, что было показано выше.

В гл. 3 были приведены иерархические представления для задач интерпретации сенсорной информации. Было показано, что возможность использования единой целевой функции на разных уровнях представления является большим преимуществом информационного подхода. Это преимущество становится еще более явным, если рассматривать иерархическое представление сенсорной информации, которое будет дополнено верхними символьными уровнями, являющимися, например, деревьями или графами решений.

Одна из проблем, связанных с применением символьных представлений, заключается в том, что исходную информацию для их построения должен готовить человек. Использование символьных представлений в качестве верхних уровней непрерывных представлений позволяет ослабить эту проблему и, с другой стороны, может помочь разрешить ряд проблем в области анализа изображений и речи. Все это ведет к построению систем машинного обучения, приближающихся к концептуальным системам. Грамматический вывод и обучение концептам являются естественным продолжением интерпретации сенсорной информации и решают ряд проблем, лишь сформулированных в гл. 3. Принцип минимальной длины описания является важным средством для задания целевой функции, однако для второго компонента индуктивного вывода — оптимизационного алгоритма, осуществляющего поиск оптимальной модели, на данный момент не существует законченной теории. Создание такой теории является открытой научной проблемой, которая должна быть решена в будущем.

Мы упомянули далеко не обо всех существующих применениях принципа МДО (например, ничего не было сказано об использовании этого принципа при анализе структуры генома или предсказании экспрессии генов [414, 415]). Более того, существует множество задач, в которых принцип МДО еще только ждет своего применения, так что здесь у исследователей есть широкое поле для деятельности.

ЛИТЕРАТУРА

1. **Кайберг Г.** Вероятность и индуктивная логика. — М.: Прогресс, 1978. — 374 с.

2. **Baxter R. A.** Minimum Message Length Inference: Theory and Applications: PhD thesis, Department of Computer Science, Monash University, Clayton, Australia. 1996. — 246 p.

3. **Рахитов А.** Послесловие: философия, индукция и вероятность // Кайберг Г. Вероятность и индуктивная логика. — М.: Прогресс, 1978.

4. **Гаек П., Гавранек Т.** Автоматическое образование гипотез: Математические основы общей теории. — М.: Наука, 1984. — 280 с.

5. **Russell S., Norvig P.** Artificial Intelligence: A Modern Approach (Second Edition): Prentice Hall, New Jersey, 2003. — 1132 p.

6. **MacKay D. J. C.** Bayesian Methods for Adaptive Models: PhD thesis, Dept. of Computation and Neural Systems — California Institute of Technology, Pasadena, California. 1992. — 98 p.

7. **Solomonoff R.** A formal theory of inductive inference, part 1 and part 2 // Information and Control. — 1964. — Vol. 7. — P. 1–22, 224–254.

8. **Wallace C. S., Boulton D. M.** An information measure for classification // Comput. J. — 1968. — Vol. 11. — P. 185–195.

9. **Wallace C. S., Freeman P. R.** Estimation and inference by compact coding // J. Royal Stat. — Soc. — 1987. — Series B. — Vol. 49. — No 3. — P. 240–251. Discussion: ibid. — P. 252–265.

10. **Baxter R. A., Oliver J.** MDL and MML: Similarities and Differences (Introduction to Minimum Encoding Inference — Part III) // Technical report 207, Department of Computer Science. — Monash University, Clayton, Australia, 1994.

11. **Rissanen J. J.** Modeling by the shortest data description // Automatica-J. IFAC. — 1978. — Vol. 14. — P. 465–471.

12. **Rissanen J. J.** Stochastic Complexity and Statistical Inquiry: World Scientific Publishers, 1989.

13. **Li M., Vitányi P. M. B.** Inductive reasoning and Kolmogorov complexity // Proc. 4th IEEE Structure in Complexity Theory Conf. — 1989. — P. 165–185.

14. **Vitányi P., Li M.** Ideal MDL and its relation to Bayesianism // Proc. ISIS: Information, Statistics and Induction in Science. — 1996. — P. 282–291.

15. **Vovk V., Gammerman A.** Complexity Approximation Principle // The Computer Journal. — 1999. — Vol. 42. — No 4. — P. 318–322.

16. **Domingos P.** Occam's two razors: the sharp and the blunt // Proc. 4th Int. Conf. Knowledge Discovery and Data Mining: AAAI Press. — 1998. — P. 37–43.

17. **Solomonoff R. J.** The Discovery of Algorithmic Probability // J. of Computer and System Sciences. — 1997. — Vol. 55. — No 1. — P. 73–88.

18. **Zemel R. S.** A minimum description length framework for unsupervised learning: PhD thesis, Dept. of Computer Science — University of Toronto. — Toronto, Canada, 1993.

19. **Vitányi P. M. B., Li M.** Minimum description length induction, Bayesianism and Kolmogorov complexity // IEEE Trans. on Information Theory. — 2000. — Vol. 46. — No 2. — P. 446–464.

20. **Schmidhuber J. H.** Low-Complexity Art // Technical Report FKI-197-94 (revised). Fakultät für Informatik, Technische Universität München, 1994.

21. **Birkhoff G. D.** A mathematical approach to aesthetics // Scietia. — 1931. — Vol. 50. — P. 133–146.

22. **Birkhoff G. D.** A mathematical theory of aesthetics // Rice Institute Pamphelt. — 1932. — Vol. 19. — P. 189–342.

23. **Birkhoff G. D.** Aesthetic measure: Harvard University Press, Cambridge, MA, 1933.

24. **Miller A.I., Engler G.** Insights of Genius: Imagery, and Creativity in Science and Art // Brit. J. Aesthetics. — 2001. — Vol. 41. — P. 337–339.

25. **Engler G.** Aesthetics in Science and Art // British Journal of Aesthetics. — 1990. — Vol. 30. — N 1. — P. 24–33.

26. **Koshelev M.** Towards The Use of Aesthetics in Decision Making: Kolmogorov Complexity Formalizes Birkhoff's Idea // Bulletin of the European Association for Theoretical Computer Science (EATCS). — 1998. — Vol. 66. — P. 166–170.

27. **Kreinovich V., Longpre L., Ferson S., Ginzburg L.** Why Is Selecting the Simplest Hypothesis (Consistent with Data) a Good Idea? A Simple Explanation // Bulletin of the European Association for Theoretical Computer Science (EATCS). — 2002. — Vol. 77. — P. 191–194.

28. **Patrick J. D., Wallce C. S.** Stone circle geometries: An information theory approach // In: D. C. Heggie, editor. Archaeoastronomy in the Old World: Cambridge Univ. Press, 1982. — P. 231–264.

29. **Schmidt M.** Time-bounded Kolmogorov complexity may help in search for extra terrestrial intelligence (SETI) // Bulletin of the European Association for Theoretical Computer Science. — 1999. — Vol. 67. — P. 176–180.

30. **Bennett C. H.** The thermodynamics of computation — a review // Int. J. Theoret. Physics. — 1982. — Vol. 21. — No 12. — P. 905–940.

31. **Zurek W. H.** Algorithmic randomness and physical entropy // Physical Review. — 1989. — Series A40. — No 8. — P. 4731–4751.

32. **Complexity**, entropy and the physics of information / W. H. Zurek, editor. Addison-Wesley. 1990.

33. **Solomonoff R.** Does Algorithmic Probability Solve the Problem of Induction? // Oxbridge Research, P. O. B. 391887, Cambridge, Mass. 02139. 1997.

34. **Myung I. J., Pitt M. A., Zhang S., Balasubramanian V.** The use of MDL to select among computational models of cognitions // Advances in Neural Information Processing Systems. — 2001. — Vol. 13. — P. 38–44.

35. **Barlow H. B.** Possible principles underlying the transformations of sensory messages // In W. A. Rosenblith, editor. Sensory Communication: MIT Press, 1961. — P. 217–234.

36. **Barlow H. B.** Single units and sensation: A neuron doctrine for perceptual psychology? // Perception. — 1972. — Vol. 1. — P. 371–394.

37. **Barlow H. B., Kaushal T. P., Mitchison G. J.** Finding minimum entropy codes // Neural Computing. — 1989. — Vol. 1. — P. 412–423.

38. **Barlow H. B.** What is the computational goal of the neocortex? // In C. Koch, J. L. Davis, eds. Large-scale neuronal theories of the brain: MIT Press, Cambidge, MA, 1994. — P. 1–22.

39. **Field D. J.** Relations between the statistics of natural images and response properties of cortical cells // J. Opt. Soc. Am. — 1987. — Vol. 4. — N 12. — P. 2379–2394.

40. **Field D. J.** What is the goal of sensory coding? // Neural Computation. — 1994. — Vol. 6. — P. 559–601.

41. **Atick J. J.** Entropy minimization: A design principle for sensory perception? // International Journal of Neural Systems. — 1992. — Vol. 3. — P. 81–90.

42. **Atick J. J., Redlich A. N.** What Does the Retina Know about Natural Scenes? // Neural Computation. — 1992. — Vol. 4. — P. 196–210.

43. **Deco G., Obradovic D.** Linear redundancy reduction learning // Neural Networks. — 1995. — Vol. 8. — No 5. — P. 751–755.

44. **Schmidhuber J., Eldracher M., Foltin B.** Semilinear predictability minimization produces well-known feature detectors // Neural Computation. — 1996. — Vol. 8. — No 4. — P. 773–786.

45. **Fass D., Feldman J.** Categorization Under Complexity: A Unified MDL Account of Human Learning of Regular and Irregular Categories // Neural Information Processing Systems (NIPS 2002). — 2002. — P. 25–34.

46. **Feldman J.** Minimization of Boolean complexity in human concept learning // Nature. — 2000. — Vol. 407. — P. 630–632.

47. **Pothos E. M., Chater N.** Categorization by simplicity: A minimum description length approach to unsupervised clustering // In: U. Hahn and M. Ramscar, eds. Similarity and Categorization: Oxford University Press, Oxford, 2001. — Chap. 4. — P. 51.

48. **Myung I. J.** Maximum entropy interpretation of decision bound and context models of categorization // J. of Mathematical Psychology. — 1994. — Vol. 38. — P. 335–365.

49. **Chater N.** Reconciling simplicity and likelihood principles in perceptual organization // Psychological Review. — 1996. — Vol. 103. — N 3. — P. 566–581.

50. **Spiegelman S.** An in vitro analysis of a replicating molecule // American Scientist. — 1967. — Vol. 55. — N 3. — P. 221–264.

51. **Li M., Vitányi P. M. B.** Philosophical issues in Kolmogorov complexity (invited lecture) // In W. Kuich, editor, Proc. on Automata, Languages and Programming (ICALP'92). — 1992. — Vol. 623. — P. 1–15.

52. **Хант Э.** Искусственный интеллект. — М.: Мир, 1978. — 558 с.

53. **Goodman N.** The Structure of Appearance. — Harvard University Press, Cambridge, 1951.

54. **Колесник В. Д., Полтырев Г. Ш.** Курс теории информации. — М.: Наука, 1982. — 416 с.

55. **Shannon C. E.** A Mathematical Theory of Communications // Bell Syst. Tech. J. — 1948. — Vol. 27. — N 3. — P. 379–423 (Part I); No 4. — P. 623–656 (Part II). [Рус. пер.: Шеннон К. Математическая теория связи // Работы по теории информации и кибернетике. — М., 1963. — С. 243–332.]

56. **Fisher R. A.** Theory of statistical estimation // Proc. Cambridge Phil. Society. — 1925. — Vol. 22. — P. 700–725.

57. **Hartley R. V. L.** Transmission of information // The Bell System Technical J. — 1928. — Vol. 7. — No 3. — P. 535–563. [Рус. пер.: Хартли Р. В. Л. Передача информации // Теория информации и ее приложения. — М.: Физматгиз, 1959. — С. 5–35.]

58. **Fano R.** Transmission of Information: A statistical theory of communication. — N. Y.: Wiley, 1961. [Рус. пер.: Фано Р. Передача информации. Статистическая теория связи. — М.: Мир, 1965.]

59. **Стратонович Р. Л.** Теория информации. — М.: Сов. радио, 1975. — 424 с.

60. **Kullback S.** Information theory and statistics. — N.Y.: Dover publications, Inc., 1958. [Рус. пер.: Кульбак С. Теория информации и статистика. — М.: Наука, 1967.]

61. **Jaynes E.** Information theory and statistical mechanics // Phys. Rev. — 1957. — Vol. 106. — P. 620–630.

62. **Huffmen D. A.** A Method for the Construction of Minimum Redundancy Codes // Proc. IRE. — 1952. — Vol. 40. — P. 1098–1101. [Рус. пер.: Хаффмен Д. А. Метод построения кодов с минимальной избыточностью // Кибернетический сб. — М., 1961. — Вып. 3. — С. 79–87.]

63. **Viola P. A.** Alignment by Maximization of Mutual Information: PhD thesis, Massachusetts Institute of Technology (MIT). — Cambridge, Massachusetts, 1995.

64. **Principe J. C., Xu D., Fisher III J. W.** Information Theoretic Learning // In: Simon Haykin, editor. Unsupervised Adaptive Filtering. — N. Y.: Wiley, 2000. — Vol. 1. — P. 265–319.

65. **Колмогоров А. Н.** Комбинаторные основания теории информации и исчисления вероятностей // УМН. — 1983. — Т. 38. — Вып. 4. — С. 27–36.

66. **Gödel K.** Über formal unentscheidbare Sätze der Principia Mathematica und verwandter Systeme // Monatshefte für Mathematik und Physik. — 1931. — Vol. 38. — P. 173–98. [Gödel K. On formally undecidable propositions of Principia Mathematica and related systems // In M. David, editor. The Undecidable: Raven Press. — New York. — 1965. — P. 5–38.]

67. **Post E.** Finite combinatory processes — formulation 1 // J. Symbolic Logic. — 1936. — Vol. 1. — P. 103–105. [Рус. пер.: Пост Э. Конечные комбинаторные процессы, формулировка 1 // Успенский В. А. Машина Поста. — М.: Наука, 1979. — С. 89–95.]

68. **Turing A. M.** On computable numbers with an application to the Entscheidungsproblem // Proc. London Math. — Soc. 1936. — Vol. 42. — P. 230–265; 1937. — Vol. 43. — P. 544–546.

69. **Марков А. А.** Теория алгорифмов // Тр. Матем. ин-та АН СССР им. В. А. Стеклова, XLII. — М.: Изд.-во АН СССР, 1954. — Т. 42.

70. **Church A.** The Calculi of Lambda-Conversion: Princeton University Press, 1941.

71. **Gödel K.** On Undecidable Propositions of Formal Mathematical Systems (Princeton lecture notes, 1934) // In M. Davis (ed.). The Undecidable: Raven Press. — New York, 1965. — P. 39–71.

72. **Post E.** Formal reduction of the general combinatorial decision problem // Amer. J. Math. — 1943. — Vol. 65. — P. 197–215.

73. **Мендельсон Э.** Введение в математическую логику: Пер. с англ./ Под ред. С. И. А д я н а; 3-е изд. — М.: Наука, 1984. — 320 с.

74. **Люгер Д. Ф.** Искусственный интеллект: стратегии и методы решения сложных проблем: 4-е изд. / Пер. с англ. — М.: Вильямс, 2003. — 865 с.

75. **Стин Э.** Квантовые вычисления. — Ижевск: НИЦ «Регулярная и хаотическая динамика», 2000. — 111 с.

76. **Верещагин А. К., Шень А.** Лекции по математической логике и теории алгоритмов. Ч. 3. Вычислимые функции. — М.: МЦНМО, 1999. — 176 с.

77. **Колмогоров А. Н.** Три подхода к определению понятия «количество информации» // Проблемы передачи информации. — 1965. — Т. 1. — № 1. — С. 3–11.

78. **Kolmogorov A. N.** Logical basis for information theory and probability theory // IEEE Trans. Inform. Theory. — 1968. — Vol. IT-14. — P. 662–664. [Колмогоров А. Н. К логическим основам теории информации и теории вероятностей // Проблемы передачи информации. — 1969. — Т. 5. — № 3. — С. 3–7.]

79. **Chaitin G. J.** On the length of programs for computing finite binary sequences // J. of the Association for Computing Machinery. — 1966. — Vol. 13. — No 4. — P. 547–569.

80. **Chaitin G. J.** On the length of programs for computing finite binary sequences: statistical considerations // J. of the Association for Computing Machinery. — 1969. — Vol. 16. — P. 145–159.

81. **Li M., Vitányi P.** An Introduction to Kolmogorov Complexity and Its Applications. 2nd ed. — N. Y.: Springer-Verlag, 1997.

82. **Звонкин А. К., Левин Л. А.** Сложность объектов и обоснование понятий информации и случайности с помощью теории алгоритмов // УМН. — 1970. — Т. 25. — Вып. 6. — С. 85–127.

83. **Grünwald P. D.** Model selection based on minimum description length // J. of Mathematical Psychology. Special Issue on Model Selection. — 2000. — Vol. 44. — No 1. — P. 133–152.

84. **Martin-Löf P.** The Definition of Random Sequences // Information and Control. — 1966. — Vol. 9. — No 6. — P. 602–619.

85. **Schnorr C. P.** A survey of the theory of random sequences // In R. E. Butts and J. Hintikka, eds. Basic Problems in Methodology and Linguistics: D. Reidel, 1977. — P. 193–210.

86. **Kolmogorov A. N., Uspensky V. A.** Algorithms and randomness // SIAM J. Theory Probab. Appl. — 1987. — Vol. 32. — P. 389–412.

87. **Vitányi P. M. B., Li M.** Minimum description length induction, Bayesianism and Kolmogorov complexity // Manuscript, CWI. — Amsterdam,1996.

88. **Davies P. C. W.** Why is the Physical World so comprehensible? // In: Ed. W.H. Zurek. Complexity, Entropy and the Physics of Information, SFI Studies in the Sciences of Complexity: Addison-Wesley. — 1990. — Vol. VIII. — P. 61–70.

89. **Levin L. A.** Laws of information conservation (non-growth) and aspects of the foundation of probability theory // Problems Inform. Transmission. — 1974. — Vol. 10. — N 3. — P. 206–210.

90. **Willis D. G.** Computational Complexity and Probability Constructions // J. of the Assoc. of Comp. Math. — 1970. — P. 241–259.

91. **Solomonoff R. J.** Complexity-Based Induction Systems: Comparisons and Convergence Theorems // IEEE Trans. on Information Theory. — 1978. — Vol. IT-24. — No 4. — P. 422–432.

92. **Левин Л. А.** Универсальные задачи перебора // Проблемы передачи информации. — 1973. — Т. 9. — № 3. — С. 115–116.

93. **Solomonoff R.** The Application of Algorithmic Probability to Problems in Artificial Intelligence // In: L. N. Kanal and J. F. Lemmer (Eds.). Uncertainty in Artificial Intelligence. Elsevier Science Publishers, 1986. — P. 473–491.

94. **Solomonoff R.** A System for Incremental Learning Based on Algorithmic Probability // Proc. 6th Israeli Conference on Artificial Intelligence. Computer Vision and Pattern Recognition. — 1989. — P. 515–527.

95. **Solomonoff R.** Progress in incremental machine learning // Technical Report IDSIA-16-03, IDSIA, 2003.

96. **Hansen M. H., Yu B.** Model selection and the principle of minimum description length // American Statistical Association J. — 2001. — Vol. 96. — P. 746–774.

97. **Hutter M.** Towards a universal theory of artificial intelligence based on algorithmic probability and sequential decisions // Proc. 12th European Conference on Machine Learning (ECML-2001). — 2001. — P. 226–238.

98. **Hutter M.** The fastest and shortest algorithm for all well-defined problems // Int. J. of Foundations of Computer Science. — 2002. — Vol. 13. — No 3. — P. 431–443.

99. **Hutter M.** A gentle introduction to the universal algorithmic agent AIXI // Technical Report IDSIA-01-03, 2003.

100. **Schmidhuber J.** On learning how to learn learning strategies // Technical Report FKI-198-94, Fakultät für Informatik. — Technische Universität München.1994.

101. **Schmidhuber J., Zhao J., Wiering M.** Shifting inductive bias with success story algorithm, adaptive Levin search, and incremental self-improvement // Machine Learning. — 1997. — Vol. 28. — No 1. — P. 105–130.

102. **Schmidhuber J.** The Speed Prior: A New Simplicity Measure Yielding Near-Optimal Computable Predictions // Computational Learning Theory (COLT 2002). — 2002. — P. 216–228.

103. **Solomonoff R.** Two Kinds of Probabilistic Induction // The Computer Journal. — 1999. — Vol. 42. — No 4. — P. 256–259.

104. **Rissanen J.** Hypothesis Selection and Testing by the MDL Principle // The Computer Journal. — 1999. — Vol. 42. — No 4. — P. 260–269.

105. **Wallace C. S., Dowe D. L.** Minimum Message Length and Kolmogorov Complexity // The Computer Journal. — 1999. — Vol. 42. — No 4. — P. 270–283.

106. **Oliver J. J., Hand D.** Introduction to Minimum Encoding Inference // Technical report 205, Department of Computer Science, Monash University, Clayton, Australia, 1994.

107. **Oliver J.J., Baxter R.A.** MML and Bayesianism: Similarities and Differences (Introduction to Minimum Encoding Inference — Part II) // Technical report 206, Department of Computer Science, Monash University, Clayton, Australia, 1994.

108. **Lanterman A. D.** Schwarz, Wallace, and Rissanen: Intertwining themes in theories of model order estimation // International Statistical Review. — 2001. — Vol. 69. — No 2. — P. 185–212.

109. **Solomonoff R.** The Universal Distribution and Machine Learning // The Computer Journal. — 2003. — Vol. 46. — P. 598–601.

110. **Zemel R. S., Hinton G. E.** Developing population codes by minimizing description length // Neural Computation. — 1995. — Vol. 7. — P. 549–564.

111. **Lappalainen H.** Using an MDL-based cost function with neural networks // Proc. IJCNN'98. — 1998. — P. 2384–2389.

112. **Leonardis A., Bischof H.** An efficient MDL-Based construction of RBF networks // Neural Networks. — 1998. — Vol. 11. — N 5. — P. 963–973.

113. **Zhang B.-T., Ohm P., Mühlenbein H.** Evolutionary induction of sparse neural trees // Evolutionary Computation. — 1997. — Vol. 5. — No 2. — P. 213–236.

114. **Sporring, J.** Pruning with minimum description length // Proc. 5th Scandinavian Conference on Artificial Intelligence (SCAI'95). — 1995. — P. 157–168.

115. **Tirri H., Myllymäki P.** MDL Learning of Probabilistic Neural Networks for Discrete Problem Domains // Proc. IEEE World Congress on Computational Intelligence. — 1994. — P. 1493–1497.

116. **Bouckaert R. R.** Probabilistic network construction using the minimum description length principle // In Lectures Notes in Computer Science 747, Symbolic and Quantitative Approaches to Reasoning and Uncertainty (ECSQARU'93). — 1993. — P. 41–48.

117. **Lam W., Bacchus F.** Learning Bayesian Belief Networks: An Approach Based on the MDL Principle // Computational Intelligence. — 1994. — Vol. 10. — P. 269–293.

118. **Suzuki J.** Learning Bayesian Belief Networks Based on the MDL Principle: An Efficient Algorithm Using the Branch and Bound Technique // IEICE Trans. on Information and Systems. — 1999. — P. 356–367.

119. **Горелик А. Л., Скрепкин В. А.** Методы распознавания. — М.: Высш. шк., 1977. — 222 с.

120. **Ту Дж., Гонсалес Р.** Принципы распознавания образов. — М.: Мир, 1978. — 412 с.

121. **Фу К.** Структурные методы в распознавании образов. — М.: Мир, 1977. — 320 с.

122. **Pavlidis T.** Structural pattern recognition: primitives and juxtaposition relations // In: M. S. Watanabe, ed. Frontiers of Pattern Recognition. — N. Y.: Academic Press, 1972. — P. 421–451.

123. **Narasimhan R.** A Linguistic Approach to Pattern Recognition // Rep. 121, Digital Computer Lab., Univ. of Illinois, Urbana, 1962. [Рус. пер.: Нарасимхан Р. Лингвистический подход к распознаванию образов // Автоматический анализ сложных изображений. — М.: Мир, 1969.]

124. **Miller W. F., Shaw A. C.** Linguistic methods in picture processing — A survey // Proc. AFIPS Fall Joint Comput. Conf. 1968. — P. 279–290.

125. **Завалишин Н. В., Мучник И. Б.** Лингвистический (структурный) подход к проблеме распознавания образов // Автоматика и телемеханика. — 1969. — № 8. — P. 86–118.

126. **Shaw A. C.** The formal description and parsing of pictures // Rep. SLAC-84, UC-32, Stanford Linear Accelerator Center, Stanford Univ. — Stanford, California, 1968.

127. **Дасарахти Б. В., Шила Б. В.** Построение сложной системы распознавания образов: Теория и методика // ТИИЭР. — 1979. — Т. 67. — № 5. — С. 5–12.

128. **Патрик Э.** Основы теории распознавания образов: Пер. с англ. / Под ред. Б. Р. Левина. — М.: Сов. радио, 1980. — 408 с.

129. **Воронцов К. В.** Локальные базисы в алгебраическом подходе к проблеме распознавания: Тезисы диссертации. — М.: Вычислительный центр РАН, 1999. — 121 с.

130. **Журавлёв Ю. И.** Об алгебраических методах в задачах распознавания и классификации // Распознавание, классификация, прогноз. Вып. 1. — М.: Наука, 1988. — С. 9–16.

131. **Журавлёв Ю. И., Гуревич И. Б.** Распознавание образов и распознавание изображений // Распознавание, классификация, прогноз. Вып. 2. — М.: Наука, 1989. — С. 5–72.

132. **Tyron R. C.** Cluster Analysis: Ann Arbor, MI, Edwards Brothers, 1939.

133. **Лурия А. Р.** Основные проблемы нейролингвистики. — М.: Изд-во МГУ, 1975. — 253 с.

134. **Looney C. G.** Pattern recognition using neural networks: Theory and algorithms for engineers and scientists. — Oxford University Press, 1997. — 458 p.

135. **Вапник В. Н., Червоненкис А. Я.** Теория распознавания образов. — М.: Наука, 1974.

136. **Boser B. E., Guyon I. M., Vapnik V. N.** A training algorithm for optimal margin classifiers // Proc. 5th Annual ACM Workshop on Computational Learning Theory. — ACM Press, 1992. — P. 144–152.

137. **Cortes C., Vapnik V. N.** Support vector networks // Machine Learning. — 1995. — Vol. 20. — N 3. — P. 273–297.

138. **Ben-Hur A., Horn D., Siegelmann H. T., Vapnik V.** Support vector clustering // J. of Machine Learning Research. — 2001. — Vol. 2. — P. 125–137.

139. **Platt J. C.** Fast training of support vector machines using sequential minimum optimization // In: B. Scholkopf, C. Burges, A. Smola, eds. Advances in Kernel Methods-Support Vector Learning. — MIT Press, Cambridge, 1998. — P. 185–208.

140. **Keerthi S. S., Gilbert E. G.** Convergence of a generalized SMO algorithm for SVM classifier design // Machine Learning. — 2002. — Vol. 46. — P. 351–360.

141. **Burges C. J. C.** A Tutorial on Support Vector Machines for Pattern Recognition // Data Mining and Knowledge Discovery. — 1998. — Vol. 2. — No. 2. — P. 121–167.

142. **Hsu C.-W., Lin C.-J.** A comparison of methods for multi-class support vector machines // IEEE Trans. on Neural Networks. — 2002. — Vol. 13. — N 2. — P. 415–425.

143. **Vapnik V. N.** Statistical Learning Theory. — N. Y.: Wiley, 1998.

144. **Fix E., Hodges J. L.** Discriminatory Analysis; Nonparametric Discrimination: Consistency Properties // USAF School of Aviation Medicine Project Number 21-49-003, Rep. No 4: Randolph Field. — Texas, 1951.

145. **Vapnik V. N.** An Overview of Statistical Learning Theory // IEEE Trans. on Neural Networks. — 1999. — Vol. 10. — No 5. — P. 988–999.

146. **Whittle P.** On the Smoothing of the Probability Density Functions // J. Royal Statistical Soc. — 1958. — Ser. B. — Vol. 20. — P. 334–343.

147. **Parzen E.** On Estimation of a Probability Density Function and Mode // Ann. Math. Statistics. — 1962. — Vol. 33. — No 3. — P. 1065–1076.

148. **Watson G. S., Leadbetter M. R.** On the Estimation of the Probability Density // Ann. Math. Statistics. — 1963. — Vol. 34. — No 2. — P. 480–491.

149. **Cacoullus T.** Estimation of a Multivariate Density // TR-40, Dept. of Statistics, Univ. of Minnesota. — Minneapolis, 1964.

150. **Sato M., Kudo M., Toyama J., Shimbo M.** Construction of a nonlinear discrimination function based on the MDL criterion // 1st International Workshop on Statistical Techniques in Pattern Recognition. — 1997. — P. 141–146.

151. **U. von Luxburg, Bousquet O., Schölkopf B.** A compression approach to support vector model selection // J. of Machine Learning Research. — 2004. — Vol. 5. — P. 293–323.

152. **Tenmoto H., Kudo M., Shimbo M.** MDL-Based selection of the number of components in mixture models for pattern classification // In A. Amin, D. Dori, P. Pudil and H. Freeman, eds. Advances in Pattern Recognition, number 1451 in Lecture Notes in Computer Science. — Springer, 1998. — P. 831–836.

153. **Kudo M., Shimbo M.** Selection of Classifiers Based on the MDL Principle Using the VC Dimension // Proc. ICPR'96. — 1996. — P. 886–890.

154. **Jain A.** et al. Information-theoretic bounds on target recognition performance based on degraded image data // IEEE Trans. on Pattern Analysis and Machine Intelligence. — 2002. — Vol. 24. — N 9. — P. 1153–1166.

155. **MacQueen J. J.** Some Methods for Classification Analysis of Multivariate Observations // Proc. 5th Berkley Symp. Math. Statistics and Probability. — 1967. — Vol. 1. — N 281. — P. 281–297.

156. **Sebesteyen G., Edie J.** An algorithm for non-parametric pattern recognition // IEEE Trans. on Electronic Computers. — 1966. — Vol. EC-15. — P. 908–915.

157. **Ball G. H., Hall D. J.** ISODATA, a novel method of data analysis and pattern classification // Stanford Research Institute Technical Report (NTIS AD699616). — Stanford, CA, 1965.

158. **Ball G. H., Hall D. J.** ISODATA: An iterative method of multivariate data analysis and pattern classification // Proc. IEEE Int. Communications Conference. — Philadelphia, 1966.

159. **Ball G. H., Hall D. J.** A clustering technique for summarizing multivariate data // Behavioral Science. — 1967. — Vol. 12. — No 2. — P. 153–155.

160. **Day N. E.** Estimating the components of mixture of normal distributions // Biometrika. — 1969. — Vol. 56. — P. 463–474.

161. **Wolfe J. H.** Pattern clustering by multivariate mixture analysis // Multivariate Behavioral Research. — 1970. — Vol. 5. — P. 329–350.

162. **McLachlan G., Krishnan T.** The EM Algorithm and Extensions: John Wiley & Sons, 1996.

163. **Bezdek J. C.** Pattern Recognition with Fuzzy Objective Function Algorithms. — N. Y.: Plenum, 1981.

164. **Akaike H.** Information theory and an extension of the maximum likelihood principle // 2nd Int. Symposium on Information Theory. — 1973. — P. 267–281.

165. **Schwarz G.** Estimating Dimension of a Model // Ann. Stat. — 1978. — Vol. 6. — P. 461–464.

166. **Milligan G. W., Cooper M. C.** An examination of procedures for determining the number of clusters in a data set // Psychometrika. — 1985. — Vol. 50. — No 1. — P. 159–179.

167. **Bischof H., Leonardis A., Sleb A.** MDL principle for robust vector quantization // Pattern Analysis and Applications. — 1999. — Vol. 2. — P. 59–72.

168. **Computer** and Information Sciences / II, J. T. Tou, ed. — N. Y.: Academic Press, 1967.

169. **Almeida L. B.** MISEP — Linear and Nonlinear ICA Based on Mutual Information // J. of Machine Learning Research. — 2003. — Vol. 4. — P. 1297–1318.

170. **Ziehe A.** et al. Blind Separation of Post-nonlinear Mixtures using Linearizing Transformations and Temporal Decorrelation // J. of Machine Learning Research. — 2003. — Vol. 4. — P. 1319–1338.

171. **Karhunen J., Pajunen P., Oja E.** The nonlinear PCA criterion in blind source separation: Relations with other approaches // Neurocomputing. — 1998. — Vol. 22. — P. 5–20.

172. **Hyväninen A.** Survey on Independent Component Analysis // Neural Computing Surveys. — 1999. — Vol. 2. — P. 94–128.

173. **Minka T. P.** Automatic Choice of Dimensionality for PCA // Advances in Neural Information Processing Systems. — 2000. — Vol. 13. — P. 598–604.

174. **Jolliffe L. T.** Princi pal Component Analysis. — N. Y.: Springer-Verlag, 1986.

175. **Harman H. H.** Modern Factor Analysis, 2nd edition: University of Chicago Press, 1967.

176. **Huber P. J.** Projection pursuit // The Annals of Statistics. — 1985. — Vol. 13. — N 2. — P. 435–475.

177. **Jutten C.** Calcul neuromimétique et traitement du signal, analyse en composantes independents: PhD thesis. — INPG: Univ. Grenoble, 1987.

178. **Jutten C., Herault J.** Blind separation of sources, part I: An adaptive algorithm based on neuromimetic architecture // Signal Processing. — 1991. — Vol. 24. — P. 1–10.

179. **Bell A. J., Sejnowski T. J.** An information-maximization approach to blind separation and blind deconvolution // Neural Computation. — 1995. — Vol. 7. — P. 1129–1159.

180. **Hyväninen A., Oja E.** A fast fixed-point algorithm for independent component analysis // Neural Computation. — 1997. — Vol. 9. — No 7. — P. 1483–1492.

181. **Cichocki A.** et al. Modified Herault-Jutten algorithms for blind separation of sources // Digital Signal Processing. — 1997. — Vol. 7. — P. 80–93.

182. **Hyväninen A., Pajunen P.** Nonlinear independent component analysis: Existence and uniqueness results // Neural Networks. — 1999. — Vol. 12. — No 3. — P. 429–439.

183. **Oja E.** Nonlinear PCA criterion and maximum likelihood in independent component analysis // Proc. Int. Workshop on Independent Component Analysis and Signal Separation (ICA'99). — 1999. — P. 143–148.

184. **Hyväninen A., Karhunen J., Oja E.** Independent component analysis. — J. Wiley, 2001.

185. **Bach F. R., Jordan M. I.** Beyond Independent Components: Trees and Clusters // J. of Machine Learning Research. — 2003. — Vol. 4. — P. 1205–1233.

186. **Learned-Miller E. G., Fisher III J. W.** ICA Using Spacings Estimates of Entropy // J. of Machine Learning Research. — 2003. — Vol. 4. — P. 1271–1295.

187. **Pajunen P.** Blind source separation using algorithmic information theory // Neurocomputing. — 1998. — Vol. 22. — P. 35–48.

188. **Lee T.-W.** et al. A unifying information-theoretic framework for independent component analysis // Int. J. on Mathematical and Computer Modeling. — 2000. — No 39. — P. 1–21.

189. **Chen P. C., Pavlidis T.** Segmentation by texture using correlation // IEEE Pat. and Anal. Mach. Intell. — 1983. — Vol. 5. — No 1. — P. 64–69.

190. **Cross G., Jain A.** Markov random field texture models // IEEE Trans. PAMI. — 1983. — Vol. 5. — P. 25–39.

191. **Turner M. R.** Texture discrimination by Gabor functions // Biol. Cybern. — 1986. — Vol. 55. — P. 71–82.

192. **Portilla J., Simoncelli E. P.** A Parametric Texture Model based on Joint Statistics of Complex Wavelet Coefficients // Int. J. of Computer Vision. — 2000. — Vol. 40. — No 1. — P. 49–71.

193. **Haralick R. M., Shapiro L. G.** A Survey: Image Segmentation Techniques // J. of Computer Vision, Graphics and Image Processing. — 1985. — Vol. 29. — No 1. — P. 100–132.

194. **Sund R.** Minimum Description Length based model selection in linear regression // Lectures on Statistical Modeling Theory, fall 2001 course (J. Rissanen). — Department of Computer Science, University of Helsinki, Finland, 2001.

195. **Lee T. C. M.** Tree-based wavelet regression for correlated data using the minimum description length principle // Australian & New Zealand Journal of Statistics. — 2002. — Vol. 44. — No 1. — P. 23–39.

196. **Qian G.** Computing minimum description length for robust linear regression model selection // Pacific Symposium on Biocomputing. — 1999. — Vol. 4. — P. 214–325.

197. **Cohen I., Raz Sh., Malah D.** Translation-invariant denoising using the minimum description length criterion // Signal Processing. — 1999. — Vol. 75. — P. 201–223.

198. **Saito N.** Simultaneous noise suppression and signal compression using a library of orthonormal bases and the minimum length criterion // In: E. Foufoula-Georgiou and P. Kumar (eds.). Wavelets in Geophysics: Academic Press, 1994. — P. 299–324.

199. **Rissanen J.** MDL denoising // IEEE Trans. Inform. Theory. — 2000. — Vol. 46. — No 7. — P. 2537–2543.

200. **Hansen M.H., Yu B.** Wavelet thresholding via MDL: Simultaneous denoising and compression // IEEE Trans. Information Theory. — 2000. — Vol. 45. — P. 1778–1788.

201. **Baxter R. A., Oliver J. J.** The kindest cut: minimum message length segmentation // In S. Arikawa and A. Sharma, eds. Lecture Notes in Artificial Intelligence 1160, Algorithmic Learning Theory, ALT-96, 1996. — P. 83–90.

202. **Fitzgibbon L. J., Allison L., Dowe D. L.** Minimum message length grouping of ordered data // In Arimura, H., Jain, S., eds. Proc. 11th Int. Conf. on Algorithmic Learning Theory (ALT2000). — 2000. — P. 56–70.

203. **Козлов Ю. М.** Адаптация и обучение в робототехнике. — М.: Наука, 1990. — 248 с.

204. **Jain R. C., Binford T. O.** Ignorance, myopia and naivete in computer vision systems // CVGIP: Image Understanding. — 1991. — Vol. 53. — No 1. — P. 112–117.

205. **Tarr M. J., Black M. J.** A computational and evolutionary perspective on the role of representation in vision // CVGIP: Image Understanding. — 1994. — Vol. 60. — No 1. — P. 65–73.

206. **Roy D. K.** Learning Words from Sights and Sounds: A Computational Model: PhD thesis, Massachusetts Institute of Technology, 1999. — 176 p.

207. **Прибрам К.** Языки мозга. — М.: Прогресс, 1975. — 464 с.

208. **Shotwell P.** Reflections on language and philosophy in regard to cognitive psychology, artificial intelligence and educational studies of chess and go, 2002.

209. **Джексон П.** Введение в экспертные системы: Пер. с англ. — М.: Вильямс, 2001. — 624 с.

210. **Nacken P.** Image Analysis Methods Based on Hierarchies of Graphs and Multi-Scale Mathematical Morphology: PhD-thesis, University of Amsterdam, 1994. — 176 p.

211. **Aloimonos J.** Purposive and qualitative active vision // Proc. 10th International Conference on Pattern Recognition. — 1990. — Vol. 1. — P. 346–360.

212. **Jain R. C., Binford T. O.** Revolutions and experimental computer vision // CVGIP: Image Understand. 1991. — Vol. 53. — No 1. — P. 127–128.

213. **Фурман Я. А.** и др. Введение в контурный анализ и его приложения к обработке изображений и сигналов. — М.: Физматлит, 2002. — 592 с.

214. **Rares A., Reinders M. J. T., Hendriks E. A.** Image Interpretation Systems // Technical Report (MCCWS 2.1.1.3.C), MCCWS project, Information and Communication Theory Group, TU Delft, 1999.

215. **Pinz A.** Interpretation and fusion — recognition versus reconstruction // In: Pinz A. and Burger W., editors. Vision Milestones, OGAI lecture series, 1995. — P. 9–21.

216. **Искусственный** интеллект. В 3 кн. Кн. 1. Системы общения и экспертные системы: Справочник /Под ред. Э. В. Попова. — М.: Радио и связь, 1990. — 464 с.

217. **Chan T.F., Shen J., Vese L.** Variational PDE models in image processing // Notices of Amer. Math. Soc. — 2003. — Vol. 50. — P. 14–26.

218. **Ерош И. Л., Игнатьев М. Б., Москалёв Э. С.** Адаптивные робототехнические системы. — Л.: ЛИАП, 1985. — 144 с.

219. **Lillholm M., Nielsen M., Griffin L. D.** Feature-Base Image Analysis // Int. J. of Computer Vision. — 2003. — Vol. 52. — N 2/3. — P. 73–95.

220. **Тихонов А. Н., Арсенин В. Я.** Методы решения некорректных задач. — М.: Наука, 1986.

221. **Ruderman D. L., Bialek W.** Statistics of natural images: Scaling in the woods // Physical Review Letters. — 1994. — Vol. 73. — N 6. — P. 100–105.

222. **Розенфельд А., Дейвис Л. С.** Сегментация и модели изображения // ТИИЭР. — 1979. — Т. 67. — № 5. — С. 71–81.

223. **Cooper D.** Maximum Likelihood Estimation of Markov Process Blob Boundaries in Noisy Images // IEEE Trans. Pattern Analysis and Machine Intelligence. — 1979. — Vol. 1. — P. 372–384.

224. **Zhu S. C., Wu Y. N., Mumford D. B.** Filters, Random Fields and Maximum Entropy (FRAME): Towards A Unified Theory for Texture Modeling // Int. J. Computer Vision. — 1998. — Vol. 27. — No 2. — P. 1–20.

225. **Tu Z. W., Zhu S. C.** Image Segmentation by Data Driven Markov Chain Monte Carlo // IEEE Trans. PAMI. — 2002. — Vol. 24. — No 5. — P. 657–673.

226. **Прэтт У.** Цифровая обработка изображений. — М.: Мир, 1982. — Кн. 2. — 480 с.

227. **Geman D.** et al. Boundary detection by constrained optimization // IEEE Trans. Pattern Analysis and Machine Intelligence (PAMI). — 1990. — Vol. 12. — P. 609–628.

228. **Zhu S. C., Wu Y., Mumford D.** Minimax Entropy Principle and its Application to Texture Modeling // Neural Computation. — 1997. — No 9. — P. 1627–1660.

229. **Geman S., Geman D.** Stochastic relaxation, Gibbs distributions and Bayesian restoration of images // IEEE Trans. PAMI. — 1984. — Vol. 6. — P. 721–741.

230. **Mumford D., Shah J.** Optimal approximation by piecewise smooth functions // Comm. Pure and Appl. Math. — 1989. — Vol. 42. — P. 577–685.

231. **Mumford D., Gidas B.** Stochastic models for generic images // Quarterly of Applied Mathematics. — 2001. — Vol. 59. — P. 85–11.

232. **Kersten D.** Predictability and Redundancy of Natural Images // J. Optical Soc. Am. A. — 1987. — Vol. 4. — No 12. — P. 2395–2400.

233. **Koloydenko A.** Modeling Natural Microimage Statistics: PhD thesis, Dept. of Math and Statistics, Univ. of Massachusetts, Amherst, 2000.

234. **Huang J. G., Mumford D. B.** Statistics of natural images and models // Proc. of IEEE Int. Conf. on Computer Vision and Pattern Recognition, 1999. — P. 541–547.

235. **Марр Д.** Зрение. Информационный подход к изучению представления и обработки зрительных образов: Пер. с англ. — М.: Радио и связь, 1987. — 400 с.

236. **Абду И. Э., Прэтт У. К.** Количественный расчет детекторов контуров, основанных на подчеркивании перепадов яркости с последующим пороговым ограничением // ТИИЭР. — 1979. — Т. 67. — № 5. — С. 59–70.

237. **Павлидис Т.** Иерархические методы в структурном распознавании образов // ТИИЭР. — 1979. — Т. 67. — № 5. — С. 39–49.

238. **Робертс Л.** Автоматическое восприятие трехмерных сцен // Интегральные роботы. — М.: Мир, 1973. — С. 162–208.

239. **Prewitt J. M. S.** Object enhancement and extraction // In Lipkin B. S. and Rosenfeld A., eds. Picture processing and Psychopictorics. — Academic Press, New York, 1970. — P. 75–149.

240. **Дуда Р., Харт П.** Распознавание образов и анализ сцен. — М.: Мир, 1976. — 511 с.

241. **Пространственное** зрение / В. М. Бондарко, М. В. Данилова, Н. Н. Красильников и др. — СПб.: Наука, 1999. — 218 с.

242. **Canny J. F.** A computational approach to edge detection // IEEE Transactions on pattern analysis and Machine Intelligence. — 1986. — Vol. 8. — No 6. — P. 679–698.

243. **Deriche R.** Optimal edge detection using recursive filtering // Proc. 1st Int. Conf. Computer Vision. — 1987. — P. 501–505.

244. **Lindeberg T.** Edge detection with automatic scale selection // IEEE Computer Vision and Pattern Recognition. — 1996. — P. 465–470.

245. **Chiang P.-C., Binford T. O.** Generic, Model-Based Edge Estimation in the Image Surface // Proc. Image Understanding Workshop. — 1997. — Vol. 2. — P. 1237–1246.

246. **Park R.-H., Yoon K. S., Choi W. Y.** Eight-point discrete Hartley transform as an edge operator and its interpretation in the frequency domain // Pattern Recognition Letters. — 1998. — Vol. 19. — P. 569–574.

247. **Chanda B., Kundu M. K., Padmaja Y. V.** A multi-scale morphologic edge detector // Pattern Recognition. — 1998. — Vol. 31. — No 10. — P. 1469–1478.

248. **Olson C. F., Huttenlocher D.** Automated target recognition by matching oriented edge pixels // IEEE Trans. on Image processing. — 1997. — Vol. 6. — No 1. — P. 103–113.

249. **Olson C. F.** A probabilistic formulation for Hausdorff matching // Proc. IEEE Conf. on Computer Vision and Pattern Recognition. — 1998. — P. 150–156.

250. **Grimson W. E. L., Marr D.** A computer implementation of a theory of human stereo vision // Proc. ARPA Image Understanding Workshop, L. S. Baumann, ed. SRI, 1979. — P. 41–45.

251. **Lei B. J., Hendriks E. A., Reinders M. J. T.** On Feature Extraction from Images // Technical Report, Deliverable 2.1.1.2.A+B, MCCWS project, 1999.

252. **Moravec H. P.** Towards automatic visual obstacle avoidance // Proc. Int. Joint Conf. on Artificial Intelligence. — 1977. — P. 584.

253. **Moravec H. P.** Visual mapping by a robot rover // Proc. 6th Int. Joint Conf. on Artificial Intelligence. — 1979. — P. 598–600.

254. **Rohr K.** Modelling and identification of characteristic intensity variations // Image and Vision Computing. — 1992. — Vol. 10. — No 2. — P. 66–76.

255. **Smith S.M., Brady J.M.** SUSAN — a new approach to low level image processing // Int. J. Computer Vision. — 1997. — Vol. 23. — No 1. — P. 45–78.

256. **Parida L., Geiger D., Hummel R.** Junctions: Detection, Classification, and Reconstruction // IEEE Trans. on Pattern Analysis and Machine Intelligence. — 1998. — Vol. 20. — No 7. — P. 687–698.

257. **Baker S.** Design and Evaluation of Feature Detectors: PhD-thesis. Columbia University, 1998.

258. **Lagunovsky D., Ablameyko S.** Straight-line-primitive extraction in grey-scale object recognition // Pattern Recog. Letters. — 1999. — Vol. 20. — P. 1005–1014.

259. **Cappellini V., Fini S., Harrigan E., Mecocci A.** Circular shape detection in remote sensing multispectral images // In Arcelli C., Cordella L.P., Sanniti di Baja G. (Eds.) Visual Form Analysis and Recognition. — Plenum Press, New York, 1992. — P. 119–126.

260. **Kanatani K., Ohta N.** Automatic detection of circular objects by ellipsegrowing // Proc. SSII2002. — 2002. — P. 355–360.

261. **Gander W., Golub G.H., Strebel R.** Fitting of circles and ellipses — least squares solution // BIT. — 1994. — Vol. 34. — P. 556–577.

262. **Fitzgibbon A., Pilu M., Fisher R.** Direct least-square fitting of Ellipses // IEEE Trans. PAMI. — 1999. — Vol. 21. — No 5. — P. 476–480.

263. **Gull N., Zapata E. L.** Lower order circle and ellipse Hough Transform // Pattern Recognition. — 1997. — Vol. 30. — No 10. — P. 1792–1744.

264. **McLaughlin R. A.** Randomized Hough Transform: Improved ellipse detection with comparison // Pattern Recognition Letters. — 1998. — Vol. 19. — P. 299–305.

265. **Shaw A. C.** A formal picture description scheme as a basis for picture processing system // Information and Control. — 1969. — Vol. 14. — P. 9–52.

266. **Ma B., Hero A.** Image Registration with Minimum Spanning Tree Algorithm // ICIP'00. — 2000. — Vol. I. — P. 481–484.

267. **Ma B.** Parametric and Nonparametric Approaches for Multisensor Data Fusion. — PhD thesis, University of Michigan, 2001. — 196 p.

268. **Nevatia R., Lin C., Huertas A.** A System for Building Detection from Aerial Images // In: Automatic Extraction of Man-Made Objects from Aerial and Space Images. — Birkhaser Verlag, Basel, 1997. — P. 77–86.

269. **Noronha S., Nevatia R.** Detection and Modeling of Buildings from Multiple Aerial Images // IEEE Trans. PAMI. — 2001. — Vol. 23. — No 5. — P. 501–518.

270. **Iqbal Q., Aggarwal J. K.** Lower-level and High-level Approaches to Content-based Image Retrieval // Proc. IEEE South West Symposium on Image Analysis and Interpretation. — 2000. — P. 197–201.

271. **Linying S., Sharp B., Chibelushi C.** Knowledge-Based Image Understanding: A Rule-Based Production System for X-Ray Segmentation // Int. Conf. on Enterprise Information Systems (ICEIS), 2002. — P. 530–533.

272. **Минский М.** Фреймы для представления знаний: Пер. с англ. — М.: Энергия, 1979. — 151 с.

273. **Lowe D.** Object Recognition from Local Scale-Invariant Features // Proc. Int. Conf. on Computer Vision, 1999. — P. 1150–1157.

274. **Kreutz M., Völpel B., Janßen H.** Scale-invariant Image Recognition based on Higher-order Autocorrelation Features // Pattern Recognition. — 1996. — Vol. 29. — N 1. — P. 19–26.

275. **Чернявский А. Ф., Афанасьев Г. К., Михаилов В. П.** Выявление дефектов интегральных схем методом оптической пространственной фильтрации // Дефектоскопия. — 1974. — № 5. — С. 41–49.

276. **Котлецов Б. Н.** Микроизображения: Оптические методы получения и контроля. — Л.: Машиностроение, 1985. — 240 с.

277. **Choo Y. C.** On the use of the polycurve codes for structural pattern recognition // Proc IEEE Int. Conf. on Systems, Man and Cybernetics. — 1987. — Vol. 2. — P. 739.

278. **Olson C. F.** Improving the generalized Hough transform through imperfect grouping // Image and Vision Computing. — 1998. — Vol. 16. — P. 627–634.

279. **Liedtke C.-E., Grau O., Growe S.** Use of explicit knowledge for the reconstruction of 3-D object geometry // Int. Conf. on Computer Analysis of Images and Patterns, 1995. — P. 580–587.

280. **Liedtke C.-E., Buckner J., Grau O.** et al. AIDA: A system for the knowledge based interpretation of remote sensing data // 3^{d} Airborne Remote Sensing Conference and Exhibition. — 1997. — Vol. 2. — P. 313–320.

281. **Growe S.** Knowledge based interpretation of multisensor and multi-temporal remote sensing images // IAPRS. — 1999. — Vol. 32. — Part 7–4–3 W6. — P. 130–138.

282. **Tönjes R., Growe S., Bückner J., Liedtke C.-E.** Knowledge Based Interpretation of Remote Sensing Images Using Semantic Nets // Photo-grammetric Engineering and Remote Sensing. — 1999. — Vol. 65. — N 7. — P. 811–821.

283. **Draper B.A.** et al. The Schema System // Int. J. of Computer Vision. — 1989. — No. 2. — P. 209–250.

284. **Crevier D., Lepage R.** Knowledge-Based Image Understanding Systems: A Survey // Comp. Vision and Image Underst. — 1997. — Vol. 67. — N 2. — P. 161–185.

285. **Boissier O., Demazeau Y.** MAVI: A Multi-Agent system for Visual Integration // Proc. IEEE Conf. on Multisensor Fusion and Integration for Intelligent Systems. — 1994. — P. 731–738.

286. **Veenman C. J., Reinders M. J. T.** A Multi-Agent Framework for a Hybrid Facial Action Tracker // Proc. 4th annual conference of the Advanced School for Computing and Imaging. — 1998. — P. 127–132.

287. **Veenman C. J., Reinders M. J. T.** Agents Attacking A Face // Face to Face Symposium, 1999.

288. **Gorniak P., Roy D.** Grounded semantic composition for visual scenes // J. of Artificial Intelligence Research. — 2004. — Vol. 21. — P. 429–470.

289. **Roy D.** Learning visually grounded words and syntax for a scene description task // Computer speech and language. — 2002. — Vol. 16. — No 3. — P. 353–385.

290. **Roy D., Mukherjee N.** Towards situated speech understanding: visual context priming of language models // Computer Speech and Language. — 2005. — Vol. 19. — No 2. — P. 227–248.

291. **Ковалевский В. А.** Локальные и глобальные решения в распознавании изображений // ТИИЭР. — 1979. — Т. 67. — № 5. — С. 50–58.

292. **Thevenaz P.** et al. A pyramid approach to subpixel registration based on intensity // IEEE Trans. on Image Processing. — 1998. — Vol. 7. — N 1. — P. 27–41.

293. **Александров В. В., Бутузов В. В., Горский Н. Д.** и др. Пирамидально-рекурсивный процессор обработки изображений и видеоданных // Обработка изображений и дистанционные исследования. — Новосибирск: ВЦ СО АН СССР, 1987. — С. 23–24.

294. **Анисимов В. А.** Исследование и разработка методов идентификации объектов на изображениях на основе пирамидально-рекурсивных структур // Тез. дисс. — СПб.: СПИИА РАН, 1992. — 110 с.

295. **Mallat S.** A theory for multiresolution signal decomposition: The wavelet representation // IEEE Trans. PAMI. — 1989. — Vol. 11. — No 7. — P. 674–693.

296. **Wang Yu-Ping.** Image representations using multiscale differential operators // IEEE Trans. Image Processing. — 1999. — Vol. 8. — No 12. — P. 1757–1771.

297. **Lutsiv V., Malyshev I., Potapov A.** Hierarchical structural matching algorithms for registration of aerospace images // Proc. SPIE. — 2003. — Vol. 5238. — P. 164–175.

298. **Geisler W. S., Perry J. S.** Variable-Resolution Displays for Visual Communication and Simulation (SID 99) // Society for Information Display. — 1999. — Vol. 30. — P. 420–423.

299. **Grossberg S.** Adaptive pattern classification and universal recoding (I, II). Parallel development and coding of neural feature detectors // Biol. Cybernet. — 1976. — Vol. 23. — P. 121–134, 187–202.

300. **Carpenter G. A., Grossberg S.** ART 2: Self-organization of stable category recognition codes for analog input patterns // Applied Optics. — 1987. — Vol. 26. — P. 4919–4930.

301. **Carpenter G., Grossberg S.** Adaptive resonance theory (ART) // In Arbib M., ed. Handbook of Brain Theory and Neural Networks: MIT Press, 1995. — P. 79–82.

302. **Wunsch II D.C.** et al. An Optoelectronic Implementation of the Adaptive Resonance Neural Networks // IEEE Trans. on Neural Networks. — 1993. — Vol. 4. — No 4. — P. 673–684.

303. **Kuo R. J.** Integration of adaptive resonance theory II neural network and genetic K-Means algorithm for data mining // Journal of the Chinese Institute of Industrial Engineers. — 2002. — Vol. 19. — No 4. — P. 64–70.

304. **Zhu S.-C., Yuille A.** Region competition: unifying snakes, region growing and bayes/MDL for multiband image segmentation // IEEE Trans. on Pattern Analysis and Machine Intelligence. — 1996. — Vol. 18. — P. 884–900.

305. **Lee T. C. M.** A Minimum Description Length Based Image Segmentation Procedure, and Its Comparison with a Cross-Validation Based

Segmentation Procedure // J. of the American Statistical Association. — 2000. — Vol. 95. — P. 259–270.

306. **Lindeberg T., Li M.-X.** Segmentation and classification of edges using minimum description length approximation and complementary junction cues // Computer Vision and Image Understanding. — 1997. — Vol. 67. — No 1. — P. 88–98.

307. **Cazorla M.A.** et al. Bayesian models for finding and grouping junctions // Energy Minimization Methods in Computer Vision and Pattern Recognition. — 1999. — P. 70–82.

308. **Li M.** Minimum description length based 2-D shape description // In IEEE 4th Int. Conf. on Computer Vision. — 1992. — P. 512–517.

309. **Davies R.H.** et al. An information theoretic approach to statistical shape modeling // Proc. 12th British Machine Vision Conference. — 2001. — P. 3–12.

310. **Lanterman A.** Minimum Description Length understanding of infrared scenes // Proc. SPIE. — 1998. — Vol. 3371. — P. 375–386.

311. **Li M., Gao Q., Vitányi P. M. B.** Recognizing on-line handwritten characters using MDL // Proc. IEEE Information Theory Workshop. — 1993. — P. 24–25.

312. **Gao Q., Li M., Vitányi P. M. B.** Applying MDL to learning best model granularity // Artificial Intelligence. — 2000. — Vol. 121. — P. 1–29.

313. **Yvan G.** et al. Self-Consistency and MDL: A Paradigm for evaluating point-correspondence algorithms, and its application to detecting changes in surface elevation // Int. J. of Computer Vision. — 2003. — Vol. 51. — No 1. — P. 63–83.

314. **Maybank S. J., Sturm P. F.** MDL, Collineations and the Funda-mental Matrix // Proc. 10th British Machine Vision Conference, 1999. — P. 53–62.

315. **Ayer S., Sawhney H.** Layered representation of motion video using robust maximum-likelihood estimation of mixture models and MDL encoding // ICCV. — 1995. — P. 777–784.

316. **Mansouri A.-R., Konrad J.** Motion segmentation with level sets // Proc. IEEE Int. Conf. Image Processing. — 1999. — Vol. II. — P. 126–130.

317. **Mansouri A.-R., Konrad J.** Minimum description length region tracking with level sets // Proc. SPIE Image and Video Communications and Process. — 2000. — Vol. 3974. — P. 515–525.

318. **Maybank S. J., Sturm P. F.** Minimum description length and the inference of scene structure from images // IEE Colloquium on Applied Statistical Pattern Recognition. — 1999. — P. 9–16.

319. **Feldman J.** Perceptual grouping by selection of a logically minimal model // Int. J. of Computer Vision. — 2003. — Vol. 55. — N 1. — P. 5–25.

320. **Pilu M., Fisher R. B.** Part segmentation from 2D edge images by the MDL criterion // Image and Vision Computing. — 1997. — Vol. 15. — N 8. — P. 563–573.

321. **Brown L. G.** A survey of Image Registration Techniques // ACM Computing surveys. — 1992. — Vol. 24. — P. 325–376.

322. **Baumberg A.** Reliable feature matching across widely separated views // Conf. on Computer Vision and Pattern Recognition. — 2000. — P. 774–781.

323. **Efrat A., Gotsman C.** Subpixel Image Registration Using Circular Fiducials // Int. J. of Comp. Geom. and Appl. — 1994. — Vol. 4. — N 4. — P. 403–422.

324. **Pinz A., Prantl M., Ganster H.** A Robust Affine Matching Algorithm Using an Exponentially Decreasing Distance Function // J. of Universal Computer Science. — 1995. — Vol. 1. — No 8. — P. 614–631.

325. **Gabrani M., Tretiak O. J.** Surface-based matching using elastic transformations // Pattern Recognition. — 1999. — Vol. 32. — P. 87–97.

326. **Lan Z.-D., Mohr R., Remagnino P.** Robust matching by partial correlation // Proc. of 6th British Machine Vision Conference. — 1995. — P. 651–660.

327. **Rohr K.** Image registration based on thin-plate splines and local estimation of anisotropic landmark uncertainties // Medical Image Computing and Computer-Assisted Intervention. — 1998. — Vol. 1496. — P. 1174–1183.

328. **Thevenaz P., Ruttimann U. E., Unser M.** Iterative multi-scale registration without landmarks // Proc. IEEE Int. Conf. Image Processing. — 1995. — Vol. 3. — P. 228–231.

329. **Christensen G. E.** Consistent Linear-Elastic Transformations for Image Matching // Proc. Information Processing in Medical Imaging. — 1999. — P. 224–237.

330. **Growe S., Tonjes R.** A Knowledge Based Approach to Automatic Image Registration // Proc. Int. Conf. on Image Processing. — 1997. — Vol. 3. — P. 228–231.

331. **Potapov A. S.** Image matching with the use of the minimum description length approach // Proc. SPIE. — 2004. — Vol. 5426. — P. 164–175.

332. **Потапов А. С.** Иерархические структурные методы автоматического анализа аэрокосмических изображений // Тез. дисс. СПб.: ФГУП ВНЦ «ГОИ им. С. И. Вавилова», 2003. — 158 с.

333. **Potapov A. S., Lutsiv V. R.** Information-theoretic approach to image description and interpretation // Proceedings of SPIE. — 2003. — Vol. 5400. — P. 277–283.

334. **Wells W.M., Viola P., Atsumi H.** et al. Multi-modal volume registration by maximization of mutual information // Medical Image Analysis. — 1996. — Vol. 1. — No 1. — P. 35–51.

335. **Long C. J., Datta S.** Wavelet Based Feature Extraction for Phoneme Recognition // Proc. ICSLP 96. — 1996. — P. 264–267.

336. **Kadambe S., Srinivasan P.** Application of Adaptive Wavelets for Speech Coding // Proc. of the IEEE-SP Int. Symp. on Time-Frequency and Time-Scale analysis. — 1994. — P. 632–635.

337. **Coifman R. R., Wicherhauser M. V.** Entropy-based algorithms for best basis selection // IEEE Trans. on Information Theory. — 1992. — Vol. 32. — P. 712–718.

338. **Thomas I.** et al. Lexical Access for Speech Understanding using Minimum Message Length Encoding // Proc. 13th Annual Conference on Uncertainty in Artificial Intelligence (UAI-97). — 1997. — P. 464–471.

339. **Thomas I.** et al. Lexical Access using Minimum Message Length Encoding // 4th Pacific Rim International Conference on Artificial Intelligence (PRICAI'96). — 1996. — P. 229–240.

340. **Thomas I.** et al. A minimum message length evaluation metric for lexical access in speech understanding // In H.Y. Lee and H. Motoda, eds. Proc. 5th Pacific Rim Int. Conf. on Artificial Intelligence (PRICAI'98). — 1998. — P. 49–54.

341. **Sankoff D., Kruskal J. B.** Time warps, string edits and macromolecules: the theory and practice of sequence comparison. — London: Addison Wesley, 1983.

342. **Жолковский А. К., Мельчук И. А.** О семантическом синтезе // Проблемы кибернетики. — 1967. — Вып. 19.

343. **Hua Y.** Unsupervised word induction using MDL criterion // Proc. Int. Symposium on Chinese Spoken Language Processing (ISCSLP). — 2000. — P. 275–279.

344. **Hübener K., Carson-Berndsen J.** Phoneme recognition using acoustic events // Proc. 3d Int. Conf. on Spoken Language Processing. — 1994. — Vol. 4. — P. 1919–1922.

345. **Siivola V.** et al. Unlimited vocabulary speech recognition based on morphs discovered in an unsupervised manner // EUROSPEECH-2003. — 2003. — P. 2293–2296.

346. **Brent M.R., Murthy S.K., Lunsberg A.** Discovering morphemic suffixes: A case study in MDL induction // Proc. 5th Int. Workshop on AI and Statistics, 1995.

347. **Чуковский К. И.** От двух до пяти. — М.: Сов. писатель, 1960. — 375 с.

348. **Rissanen J., Ristad E. S.** Language acquisition in the MDL framework // In: E. S. Ristad, ed. Language Computations. 1992. Series in Discrete Mathematics and Theoretical Computer Science. — Vol. 17. — P. 149–166.

349. **Cartwright T. A., Brent M. R.** Segmenting speech without a lexicon: The roles of phonotactics and speech source // Proc. 1st Meeting of the ACL Special Interest Group in Computational Phonology. — 1994. — P. 83–90.

350. **Thomas I., Zukerman I., Raskutti Bh.** Extracting phoneme pronunciation information from corpora // In D. M. W. Powers, ed. New Methods in Language Processing and Computational Natural Language Learning, ACL, 1998. — P. 175–183.

351. **Kit C. Y., Wilks Y.** Unsupervised learning of word boundary with description length gain // In: M. Osborne & E. T. K. Sang, eds. CoNLL-99. — 1999. — P. 1–6.

352. **Павилёнис Р. И.** Проблема смысла. — М.: Мысль, 1983. — 286 с.

353. **Roy D., Hsiao K.-Y., Mavridis N.** Mental imagery for a conversational robot // IEEE Trans. on Systems, Man, and Cybernetics. — 2004. — Vol. 34. — N 3. — P. 1374–1383.

354. **Roy D., Gorniak P., Mukherjee N., Juster J.** A trainable spoken language understanding system for visual object selection // Proc. Int. Conf. of Spoken Language Processing, 2002.

355. **Brooks R. A., Stain L. A.** Building brains for bodies // Autonomous Robots. — 1994. — N 1. — P. 7–25.

356. **Иванов В. В.** Чет и нечет: Асимметрия мозга и знаковых систем. — М.: Сов. радио, 1978. — 184 с.

357. **Выготский Л. С., Лурия А. Р.** Этюды по истории поведения: Обезьяна. Примитив. Ребенок. — М.: Педагогика-Пресс, 1993. — 224 с.

358. **Gorniak P., Roy D.** Augmenting user interfaces with adaptive speech commands. Proc. 5th Int. Conf. on Multimodal Interface. — 2003. — P. 176–179.

359. **Roy D.** A trainable visually-grounded spoken language generation system // Proc. Int. Conf. of Spoken Language Processing. — 2002.

360. **Hsiao K.-Y., Mavridis N., Roy D.** Coupling perception and simulation: Steps towards conversational robotics // Proc. IEEE/RSJ Int. Conf. on Intelligent Robots and Systems, 2003.

361. **Roy D., Hsiao K.-Y., Mavridis N.** Conversational robots: building blocks for grounding word meanings // Proc. HLT-NAACL03 Workshop on Learning Word Meaning from Non-Linguistic Data. — 2003. — P. 70–77.

362. **Gorniak P., Roy D.** A visually grounded natural language interface for reference to spatial scenes // Proc. Int. Conf. for Multimodal Interfaces. — 2003. — P. 219–226.

363. **Roy D.** Learning visually grounded words and syntax of natural spoken language. Evolution of Communication. — 2001. — Vol. 4. — No 1. — P. 33–56.

364. **Лурия А. Р.** Нейропсихология памяти. (Нарушения памяти при локальных повреждениях мозга). — М.: Педагогика, 1974. — 311 с.

365. **Warren R. M.** Perception restoration of missing speech sounds // Science. — 1970. — Vol. 167. — P. 393–395.

366. **Горский Н., Анисимов В., Горская Л.** Распознавание рукописного текста: От теории к практике. — СПб.: Политехника, 1997. — 126 с.

367. **Турчин В. Ф.** Феномен науки: Кибернетический подход к эволюции. Изд. 2-е. — М.: ЭТС, 2000. — 368 с.

368. **Turchin V.** The concept of supercompiler // ACM Trans. on Programming Languages and Systems. — 1986. — Vol. 8. — P. 292–325.

369. **Newell A.** Physical symbol system // In: Norman D. A., ed. Perspectives on Cognitive Science: Norwood, NJ, Ablex. Chapter 4, 1981.

370. **Вейль Г.** Математическое мышление: Пер. с англ. и нем. / Под ред. Б. В. Бирюкова и А. Н. Паршина. — М.: Наука, 1989. — 400 с.

371. **Lenat D. B.** EURISKO: A program that learns new heuristics // Artificial Intelligence. — 1983. — Vol. 21. — N 1, 2. — P. 61–98.

372. **Chomsky N.** Three models for the description of language // IRE Trans. on Information Theory. — 1956. — Vol. IT-2. — N 3. — P. 113–124. [Рус. пер.: Хомский Н. Три модели для описания языка // Кибернетический сборник. — 1961. — Вып. 2. — С. 237–266.]

373. **Chomsky N.** Syntactic structures: Gravenhage, Mouton. 1957. [Рус. пер.: Хомский Н. Синтаксические структуры // Новое в лингвистике. — 1962. — Вып. 2. — С. 412–527.]

374. **Chomsky N.** Formal properties of grammars // In: Handbook of Mathematical Psychology. — N. Y.: Wiley, 1963. — V. II. — P. 323–418. [Рус. пер.: Хомский Н. Формальные свойства грамматик // Кибернетический сборник. — 1966. — Вып. 2. — С. 121–230.]

375. **Алфёрова З. В.** Теория алгоритмов. — М.: Статистика, 1973. — 164 с.

376. **Li H., Abe N.** Clustering words with the MDL principle // Proc. 16th Conf. on Computational Linguistics. — 1996. — Vol. 1. — P. 4–9.

377. **Gold E. M.** Language identification in the limit // Information and Control. — 1967. — Vol. 10. — P. 447–474.

378. **Feldman J. A.** Some decidability results on grammatical inference and complexity // Stanford Artificial Intelligence Proj. Memo. AI-93, Stanford Univ., Stanford, California, 1969.

379. **Keller B., Lutz R.** Evolving stochastic context-free grammars from examples using a minimum description length principle // Workshop on Automata Induction Grammatical Inference and Language Acquisition, ICML-97, 1997.

380. **Lankhorst M. M.** A genetic algorithm for induction of nondeterministic pushdown automata // Technical Report CS-R 9502, University of Groningen, The Netherlands, 1995.

381. **Korkmaz E. E.** Controlled genetic programming search for solving deceptive problems: PhD thesis. The Middle East Technical University. The Department of Computer Engineering, 2003.

382. **Belz A., Eskikaya B.** A genetic algorithm for finite state automata induction with an application to phonotactics // In: B. Keller, ed. Proc. ESSLLI-98 Workshop on Automated Acquisition of Syntax and Parsing, August. — 1998. — P. 9–17.

383. **Kit C.** Current progress in learning phrase structure with the minimum description length principle // Technical Report CS-96-09, Department of Computer Science, University of Sheffield, 1996.

384. **Kit C.** A goodness measure for phrase learning via compression with the MDL principle // The ESSLLI-98 Student Session. — 1998. — Chapter 13. — P. 175–187.

385. **Osborne M.** DCG induction using MDL and parsed corpora // Proc. 1st Workshop on Learning Language in Logic, 1999. — P. 63–71.

386. **Osborne M.** MDL-based DCG induction for NP identification // Proc. CoNLL99. — 1999. — P. 61–68.

387. **Маркус С.** Теоретико-множественные модели языков. — М.: Наука, 1970. — 332 с.

388. **Scheler G.** Constructing Semantic Representations Using the MDL Principle // Technical Report TR-97025, Institüt für Informatik, München, 1997.

389. **Brent M. R., Murthy S. K., Lundberg A.** Discovering morphemic suffixes: a case study in minimum description length induction // Proc. 5th Int. Workshop on Artificial Intelligence and Statistics. — 1995. — P. 482–490.

390. **Grünwald P.** A minimum description length approach to grammar inference // In: G. Scheler, S. Wernter and E. Riloff, eds. Connectionist, Statistical and Symbolic Approaches to Learning for Natural Language: Berlin, Springer-Verlag. — 1994. — Vol. 1004 of Lecture Notes in AI. — P. 203–216.

391. **Li H., Abe N.** Generalizing Case Frames using a Thesaurus and the MDL Principle // Computational Linguistics. — 1998. — Vol. 24. — No 2. — P. 217–244.

392. **Quinlan J. R.** C4.5: Programs for Machine Learning: San Mateo, CA, Morgan Kaufmann, 1993.

393. **Quinlan J. R.** The Minimum Description Length Principle and Categorial Theories // In: Cohen W.W. and Hirsh H., eds. Proc. 11th Int. Conf. on Machine Learning (ICML94). — 1994. — P. 233–241.

394. **Pfahringer B.** A new MDL measure for robust rule induction // Proc. 8th European Conf. on Machine Learning. — 1995. — Vol. 912 of LNAI. — P. 331–334.

395. **Pfahringer B.** Practical uses of the Minimum Description Length Principle in Inductive Learning: PhD thesis. Technische Universitat Wien, 1995.

396. **Cleary J. J. G., Legg S., Witten I. H.** An MDL estimate of the Significance of Rules // Proc. Information, Statistics and Induction in Science Conference. — 1996. — P. 43–53.

397. **Quinlan J. R.** Learning efficient classification procedures and their application to chess end games // In: R. S. Michalski, J. G. Carbonell, T. M. Mitchell, eds. Machine Learning: Springer-Verlag, 1983.

398. **Quinlan J. R.** Induction of Decision Tress // Machine Learning. — 1986. — Vol. 1. — N 1. — P. 81–106.

399. **Quinlan J. R.** Inferring Decision Trees using the Minimum Description Length Principles // Information and Computation. — 1989. — Vol. 80. — P. 227–248.

400. **Iba H., de Garis H., Sato T.** Genetic programming using a minimum description length principle // In: E. Kenneth, Jr. Kinnear, eds. Advances in Genetic Programming: MIT Press, 1994. — Chapter 12. — P. 265–284.

401. **Mehta M., Rissanen J., Agrawal R.** MDL-Based Decision Tree Pruning // Proc. of the First International Conference on Knowledge Discovery and Data Mining (KDD'95). — 1995. — P. 216–221.

402. **Ferri-Ramírez C., Hernández-Orallo J., Ramírez-Quintana M. J.** Learning MDL-guided Decision Trees for Constructor-Based Languages // Proc. Work-in-Progress Track at the 11th Int. Conf. on Inductive Logic Programming. — 2001. — P. 39–50.

403. **Matheus C. J., Rendell L. A.** Constructive induction of decision trees // Proc. IJCAI-89. — 1989. — P. 645–650.

404. **Pagallo G., Haussler D.** Boolean feature discovery in empirical learning // Machine Learning. — 1990. — Vol. 5. — P. 71–99.

405. **Mahoney J. J., Mooney R. J.** Initializing ID5R with a domain theory: some negative results // Technical Report 91-154, CS Dept., University of Texas, Austin, TX, 1991.

406. **Oliver J. J.** Decision Graphs — An Extension of Decision Trees // Technical Report 92/173, Dept. of Computer Science, Monash University, 1993.

407. **Oliveira A. L., Sangiovanni-Vincentelli A.** Inferring Reduced Ordered Decision Graphs of Minimal Description Length // Proc. 12th Int. Conf. on Machine Learning. — 1995. — P. 421–429.

408. **Oliveira A. R., Sangiovanni-Vincentelli A.** Using the minimum description length principle to infer reduced ordered decision graphs // Machine Learning Journal. — 1996. — Vol. 25. — P. 23–50.

409. **Kohavi R.** Bottom-up Induction of Oblivious Read-Once Decision Graphs: Strength and Limitations // 12th National Conf. on Artificial Intelligence. — 1994. — P. 613–618.

410. **Kohavi R., Li C.-H.** Oblivious Decision Trees, Graphs and Top-Down Pruning // Proc. 14th Int. Joint Conf. on Artificial Intelligence. — 1995. — P. 1071–1077.

411. **Oliver J., Wallace C. S.** Inferring decision graphs // Technical Report 91/170. — Dept. of Computer Science, Monash University, 1992.

412. **Oliveira A. L.** et al. Exact minimization of binary decision diagrams using implicit techniques // IEEE Trans. on Computers. — 1998. — Vol. 47. — N 11. — P. 1282–1296.

413. **Uther W. T. B., Veloso M. M.** The Lumberjack algorithm for learning linked decision forests // In: R. Mizoguchi, J. K. Slaney, eds. 6th Pacific Rim Int. Conf. on Artificial Intelligence (PRICAI 2000). — 2000. — Vol. 1886 of Lecture Notes in Computer Science. — P. 156–166.

414. **Tabus I., Astola J.** On the use of MDL principle in gene expression prediction // J. Appl. Signal Proces. — 2001. — N 4. — P. 297–303.

415. **Koivisto M.** et al. An MDL method for finding haplotype blocks and for estimating the strength of haplotype block boundaries // Pacific Symposium on Biocomputing (PSB'03), World Scientific. — 2003. — P. 502–513.

ISBN 5-7325-0881-3

9 785732 508819

НАУЧНОЕ ИЗДАНИЕ

Потапов Алексей Сергеевич

РАСПОЗНАВАНИЕ ОБРАЗОВ И МАШИННОЕ ВОСПРИЯТИЕ: ОБЩИЙ ПОДХОД НА ОСНОВЕ ПРИНЦИПА МИНИМАЛЬНОЙ ДЛИНЫ ОПИСАНИЯ

Заведующая редакцией *Е. В. Шарова*
Редактор *Л. М. Манучарян*
Переплет *М. Л. Черненко*
Технический редактор *Т. М. Жилич*
Корректоры *Т. Н. Гринчук, Н. Б. Старостина*
Компьютерная верстка *Т. М. Каргапольцевой*

Сдано в набор 11.09.2006. Подписано в печать 21.12.2006.
Формат издания 60×90 $^{1}/_{16}$. Бумага офсетная. Гарнитура SchoolBook.
Печать офсетная. Усл. печ. л. 34,5. Уч.-изд. л. 31,9.
Тираж 1000 экз. Заказ .

ОАО «Издательство “Политехника”».
191023, Санкт-Петербург, Инженерная ул., д. 6.
Отпечатано с готовых диапозитивов
в ГУП РК «Республиканская типография им. П. Ф. Анохина»
185005, г. Петрозаводск, ул. «Правды», 4.