Предисловие

Методы анализа данных (АД), по-видимому, прошли главный пик своего развития. Прояснились особенности, отличающие область АД от других направлений прикладной математики, достаточно четко обозначился круг задач и накоплен большой опыт их решения. Ясны и направления дальнейшего развития методов АД. Утихли споры с внешними оппонентами о том, наука это или не наука, математика или не математика. Более корректными стали и дискуссии между разработчиками методов АД. И на этом спокойном фоне стали уже появляться статьи и диссертации, заново предлагающие идеи, которые обсуждались 20–25 лет назад.

Все это создает объективные предпосылки для попытки написать книгу обзорно-учебного характера, ориентированную на тех, кому приходится сейчас и предстоит в обозримом будущем пользоваться методами АД в практике своей работы. Данная книга и является результатом такой попытки. В ней отражен опыт работы автора и его коллег в области распознавания образов и анализа данных в течение последних 30-ти лет. Хотелось бы, чтобы эта книга содержала описание современных методов АД и идей, лежащих в их основе, и была понятна прикладнику нематематику, т. е. инженеру, агроному, экономисту, врачу и т. д.

Автор считает приятным долгом поблагодарить своих друзей и сотрудников за многолетнюю совместную работу над основной частью описываемых в книге методов и, прежде всего, В. Н. Ёлкину и Г. С. Лбова.

Большое значение для автора имели научные дискуссии о проблемах анализа данных и распознавания образов с коллегами из нашего и других коллективов — С. А. Айвазяном, А. А. Боровковым, В. Н. Вапником, Г. Я. Волошиным, Ю. А. Ворониным, В. Д. Гусевым, Э. Дидэ, Э. В. Евреиновым, Ю. И. Журавлевым, Л. Кеналом, В. А. Ковалевским, Ю. Г. Косаревым, Л. Д. Мешалкиным, Д. Мики, Р. Михальским, Ж.-К. Симоном, В. В. Ольшевским, Л. А. Растригиным, Ш. Раудисом, К. Ф. Самохваловым, Ф. П. Тарасенко, Э. И. Цветковым и многими другими.

Автор тепло благодарит также свою постоянную вдохновительницу и помощницу В. М. Загоруйко.

Часть I
Введение в анализ данных

Глава 1
Основные понятия

В этой книге описываются методы обработки информации, представленной в различной форме — в виде «данных», «знаний», «структур» и т. д. В основе анализа всех этих видов информации лежат две процедуры: процедура обнаружения закономерностей, содержащихся в представленной информации, и процедура использования обнаруженных закономерностей для предсказания значения одной части информации по известным значениям другой ее части. Но прежде чем переходить к описанию этих процедур, нужно пояснить смысл употребляемых в книге терминов, в частности таких распространенных, как данные, знания, гипотеза, закономерность и т. п.

§ 1. Чем отличаются «данные» от «знаний»?

Исходная информация, которую нужно обрабатывать, чаще всего имеет вид числовых таблиц (матриц), состоящих из m строк и n столбцов. Строки $a_1, a_2, \ldots, a_i, \ldots, a_m$ отражают информацию об изучаемых объектах или явлениях, а столбцы $x_1, x_2, \ldots, x_j, \ldots, x_n$ отражают свойства (признаки, характеристики) этих

объектов или явлений. Природа объектов может быть любой — это могут быть физические тела, живые организмы, сигналы, отдельные социальные процессы, заводы, виды спорта, месторождения и т. д. Понятно, что набор признаков, описывающих эти объекты, будет в каждом случае своим и должен отражать их наиболее важные свойства.

На пересечении i-й строки и j-го столбца указывается значение (b_{ij}) j-го признака у i-го объекта. Такой факт (например, что i-й дом имеет высоту 12 м) считаем атомарной частью данных о конкретном i-м объекте. Полные данные об i-м объекте содержатся в совокупности всех элементов i-й строки. Информация же о всех заданных свойствах всех изучаемых объектов, записанная в таблице «объект-свойство», и называется *таблицей данных*. Таким образом, данные представляют собой совокупность отдельных конкретных фактов.

Пусть в таблице данных представлены описания большого количества жилых домов, а нас интересуют только три свойства этих домов: из какого материала они построены, в какой цвет покрашены их стены и какой они высоты. После изучения таблицы данных мы можем обнаружить некоторые закономерности. Например, выясняется, что все панельные дома, окрашенные в серый цвет, имеют высоту от 15 до 25 м, панельные зеленые дома — от 8 до 16 м, а кирпичные, вне зависимости от цвета стен, имеют высоту меньше 10 м. Обозначим признак «вид строительного материала» через x_1. Этот признак принимает два понятных значения: $x_1 =$ п (панель) или $x_1 =$ к (кирпич). Признак «цвет стен», обозначаемый через x_2, принимает значения: $x_2 =$ серый, $x_2 =$ зеленый или $x_2 =$ любой. Признак «высота» x_3 может принимать любое числовое значение от нуля до 30 м . Тогда обнаруженные закономерности можно сжато записать в виде таких логических высказываний:

если $(x_1 =$ п$)$ и $(x_2 =$ серый$)$, то $(x_3 = 15 - 25)$;
если $(x_1 =$ п$)$ и $(x_2 =$ зеленый$)$, то $(x_3 = 8 - 16)$;
если $(x_1 =$ к$)$ и $(x_2 =$ любой$)$, то $(x_3 < 10)$.

Эти высказывания не содержат информации в виде конкретных характеристик каждого отдельного дома, но зато отражают наши знания о некоторых обобщенных характеристиках всех домов, описанных в таблице данных.

Так выглядит переход от данных к знаниям. Знания представляют собой краткое обобщенное описание основного содержания информации, представленной в данных. Знания могут быть пред-

ставлены в различной форме. В дальнейшем мы будем пользоваться приведенной выше формой в виде логических правил типа «если ... то ...».

§ 2. Что такое анализ данных?

Среди задач прикладной математики поясним место того направления, которое с подачи французских математиков получило название *анализа данных* [14, 51].

Классическое направление прикладной математики связано с методами вычислений одних характеристик изучаемого объекта или явления по известным значениям других его характеристик. При этом модель объекта считается известной, а зависимости между характеристиками описываются аналитическим выражением в виде уравнения или системы уравнений или неравенств. Проблемы, возникающие при решении таких задач, связаны, например, с большими объемами вычислений, с защитой от погрешностей, накапливающихся в компьютере из-за округления чисел.

Позже появились задачи анализа объектов, математическая модель которых известна с точностью до параметров. Известен набор характеристик, влияющих на целевую характеристику, известен также общий вид зависимости между характеристиками, но коэффициенты, показатели степени и другие параметры модели неизвестны, и, чтобы их определить, используются протоколы наблюдений, отражающие значения одних характеристик при разных значениях других. Делается серия предположений о значениях неизвестных параметров модели и эти предположения проверяются на протоколах. В результате выбираются такие значения параметров, при которых модель с заданной точностью позволяет по одним (входным) характеристикам определять другие (выходные или целевые) характеристики. Такого рода задачи называются задачами *идентификации моделей*.

Наконец, с появлением кибернетики стали формулироваться задачи анализа «черного ящика»: исследователю известен набор характеристик, среди которых имеются характеристики, влияющие на целевое свойство объекта, но какие из них являются определяющими (информативными) и какой математической моделью описываются закономерности их влияния на целевую характеристику, не известно. Нужно выбрать информативные характеристики и построить модель, позволяющую вычислять значения целевой характеристики по значениям других характеристик.

Единственным источником информации для решения такой задачи служит таблица экспериментальных данных типа «стимул-реакция» с описанием входных и выходных характеристик наблюдаемого объекта или множества объектов. Как мы видели раньше, такие таблицы данных называют таблицами «объект-свойство». Теперь выбор модели и ее параметров делается путем проверки разных эмпирических гипотез на материале таблицы данных. Возникающий при этом круг задач и составляет направление, именуемое *задачами анализа данных*.

Возвращаясь к началу, можно отметить, что вычислительная математика обычно не имеет дела с этапом выдвижения гипотез о том, какие характеристики должны включаться в модель объекта и какой должна быть эта модель. Риск сделать ошибку в выборе модели и ее параметров остается вне поля внимания, а аккуратные вычисления по имеющейся модели создают впечатление высокого качества решения проблемы в целом.

Задачи идентификации моделей требуют от математика ответственности за правильный выбор параметров модели. Наличие этого рискованного шага в процессе решения задачи лишает результат ореола строгой математической чистоты.

На результатах решения задач анализа данных лежит явный след большого числа эвристических или экспертных предположений — и о выборе характеристик объекта, и о классе моделей, и о параметрах выбранной модели. Эти предположения представляются на языке математических формул, но истоки их появления лежат вне математики, так что основная часть процесса решения задач анализа данных связана с проникновением в природу изучаемого явления и характерна скорее для естественно-научных областей.

Ситуация усугубляется еще и тем, что реальные данные обладают такими особенностями, которые затрудняют применение строгих математических методов. Достаточно отметить, что таблицы данных часто бывают представлены малыми выборками в пространствах большой размерности при отсутствии информации о характере и степени зависимости одних характеристик от других, разнотипности измерительных шкал, наличии шумов и пробелов. В этих условиях методы решения задач анализа данных вынужденно основываются как на корректных математических процедурах, так и на чисто эвристических приемах. Не удивительно, что получаемые решения воспринимаются настороженно, а многие методы решения выглядят недостаточно строго

обоснованными.

Это обстоятельство объективно отражает тот факт, что на любом этапе развития прикладной математики возникают реальные задачи, для решения которых хорошо обоснованные математические методы еще не готовы. Вместе с тем важность задач не позволяет отложить их решение и вынуждает принимать рискованные эмпирические гипотезы и использовать нестрогие эвристические приемы. Если получаемые при этом результаты (предсказания, прогнозы) подтверждаются фактами, то настороженность в восприятии использованной модели сменяется уверенностью в ее адекватности изучаемому явлению, а внимание математика переносится на аналитическое исследование модели и вычислительные трудности, связанные с ее использованием. А доброжелательные и стимулирующие замечания типа «голая эвристика», «мутный поток литературы» применяются строгими критиками уже к попыткам решения других нетрадиционных проблем.

§ 3. Принятие решений по прецедентам и моделям

В последнее время обозначилось еще большее отдаление методов анализа данных от традиционных методов вычислительной математики. Упоминаемые выше модели изучаемых систем строятся не всегда. Все чаще стали использоваться такие алгоритмы анализа данных, которые опираются не на общие модели «черного ящика», а на конкретные факты его поведения, зафиксированные в протоколах «вход-выход», или на «прецеденты». При этом используется простая, но фундаментальная гипотеза о монотонности пространства решений, которую можно выразить так: «Похожие входные ситуации приводят к похожим выходным реакциям системы». Для каждой новой ситуации достаточно найти в протоколе одну или несколько самых близких, похожих на нее ситуаций и принимать решение, опираясь на эти прецеденты. Кстати, задолго до возникновения анализа данных это правило принятия решений по прецедентам было положено в основу древнегреческой медицины, а также британского судопроизводства.

При таком подходе мы не пытаемся познать систему так глубоко, чтобы уметь предсказывать ее реакцию на любые возможные внешние воздействия. Мы знаем лишь одно ее фундаментальное свойство: монотонность поведения в окрестностях име-

ющихся прецедентов. И этого обычно оказывается достаточно для получения практически приемлемых решений в каждом конкретном случае. Данный факт приводит некоторых авторов к заключению о возможности и даже целесообразности отказа от построения модели изучаемой системы вообще.

Однако необходимо отличать друг от друга модели разного уровня в иерархии древа познания. По мере углубления понимания изучаемой системы мы имеем дело с моделями, одни из которых отвечают на вопрос «Что происходит?», другие — на вопрос «Как это происходит?», а третьи — «Почему именно так, а не иначе?» [91]. В анализе данных чаще всего под моделью понимают аналитическое описание наблюдаемых экспериментальных значений в виде некоторого закона распределения. Такие феноменологические модели отражают то, что происходит, но ничего не говорят ни о механизме, ни о причинах происходящего. Ситуация принципиально меняется, если удается построить модель, объясняющую (имитирующую) механизм функционирования системы. Еще лучше, если модель объясняет метамеханизм, т. е. причины именно такого функционирования, а не другого. Такие модели могут помочь избежать принятия ошибочных решений.

Проиллюстрируем это на одном живом примере. Лет 20 назад в узких кругах заинтересованных специалистов появилась сенсационная информация о том, что удалось построить распознающее устройство, безошибочно отличающее атомные подводные лодки от дизельных по некоторым особенностям излучаемых ими гидроакустических сигналов. Протоколы реальных наблюдений были тщательно обработаны, были построены распределения сигналов в пространстве наблюдаемых характеристик и оптимальные для этих распределений решающие правила. В присутствии очень авторитетной комиссии надежность распознавания на контрольной выборке оказалась равной 100 %! Однако такой большой успех вызвал сомнения не только у наблюдателей, но и у самих разработчиков системы. Они начали искать физическое объяснение информативным особенностям сигналов и в итоге обнаружили, что сигналы от дизельных лодок писались на одном магнитофоне, а от атомных — в другой экспедиции и на другом магнитофоне. И тонвал одного из магнитофонов имел биения, которые вносили искажения в сигнал. Эти искажения и улавливались распознающим устройством.

Если иметь в виду только «Что?»-модели, то методы принятия решений с опорой на отдельные прецеденты или на их обоб-

щенное (модельное) описание имеют приблизительно равные методологические права. Более того, при построении модели, как и при всяком обобщении, теряются некоторые особенности поведения системы в каждой конкретной точке пространства решений. При опоре же на прецедент, как показывает опыт, удается учесть эти локальные особенности, что часто позволяет получать более точные решения. Наличие в протоколах наблюдений систематических погрешностей, искажение реальной картины случайной непредставительной выборкой и другие несовершенства имеющихся данных могут приводить к недоразумениям и никакими моделями первого уровня исправить эту ситуацию нельзя. Всегда, если есть возможность, нужно пытаться проникнуть в суть изучаемого явления и найти разумное объяснение получаемым результатам. Т. е. нужно выдвигать гипотезы и строить модели, отвечающие на вопросы «Как?» и «Почему?».

§ 4. Что такое анализ знаний?

В задачах АД мы имеем дело с различными методами обработки таблиц данных. В последнее время появились методы, с помощью которых можно обрабатывать и наборы логических высказываний (знаний), представленных в форме конъюнкций «если ... то ...». При этом на знаниях ставятся и решаются задачи, аналогичные тем, что ставятся и решаются на данных: обнаружение закономерностей в массиве знаний (т. е. знаний о знаниях) и использование этих закономерностей (метазнаний) для предсказания одних частей знаний по известным значениям их других частей.

Методы анализа знаний (АЗ) имеют много общего с методами АД. В частности, как и в АД, здесь можно опираться как на эвристические предположения о моделях закономерностей и параметрах этих моделей, так и на отдельные знания («прецеденты») из тех, что имеются в базе знаний. По этой причине значительная часть успеха зависит от того, насколько удачно выбраны предположения о моделях или прецеденты, насколько представителен анализируемый материал и т. д.

§ 5. Что такое закономерность?

Выше мы свободно применяли широко известное слово закономерность, не давая строгого определения этого понятия. Нам

придется и в будущем изложении часто употреблять это слово и потому попытаемся объяснить, что именно мы будем им обозначать. В основе понятия закономерность лежит понятие эмпирическая гипотеза, которое мы попытаемся сформулировать более строго [90]. Под гипотезой h мы подразумеваем набор из четырех элементов $h = \langle W, O, V, T \rangle$. Здесь

W — множество тех объектов, относительно которых высказывается данная гипотеза. Оно может быть конечным («для данного набора сортов пшеницы ...») или бесконечным («для всех материальных тел ...»);

O — конечный набор средств наблюдения или измерения;

V — словарь или конечный набор символов для записи результатов наблюдений в протоколе pr;

T — тестовый алгоритм, анализирующий протоколы и выносящий одно из двух решений: $T(\mathrm{pr}) = 1$, если данный наблюдаемый протокол согласуется с гипотезой h, и $T(\mathrm{pr}) = 0$, если наблюдаемый протокол не согласуется с гипотезой, т. е. опровергает ее.

Таким образом, когда мы выдвигаем какую-нибудь гипотезу, мы должны четко сказать, о каких объектах (W) мы говорим, какие свойства этих объектов нас интересуют, чем и как мы их будем измерять (O), какими символами (V) — цифрами, буквами и т. д. — будем записывать результаты наблюдений и как мы будем проверять гипотезу на «прочность», т. е. какими протоколами наблюдений (pr) гипотеза будет подтверждаться ($T = 1$) и появлением каких протоколов (с точки зрения гипотезы недопустимых) она может быть опровергнута ($T = 0$). Если эти элементы строго не оговорены, то по поводу любого результата эксперимента можно сказать, что он «не опровергает моей гипотезы, потому что я имел в виду не совсем то, что вы подумали».

К разным гипотезам у нас возникает разное отношение в зависимости от их содержания и формы, т. е. от того, что они утверждают и как они сформулированы. На какие же свойства гипотез, связанные с их содержанием и формой, мы обычно обращаем внимание?

Несколько гипотез могут говорить об одних и тех же объектах или явлениях (W) на одном и том же языке приборов (O) и протоколов (V), но отличаться строгостью своих тестовых алгоритмов (T), что проявляется в разном количестве мыслимых протоколов (N'), которые способны опровергнуть каждую из этих гипотез. Чем больше величина N', тем больше для данной гипотезы

риск быть опровергнутой при всяком новом эксперименте. Если общее число всех возможных различных (неизоморфных протоколов обозначить через N, то отношение $Q = N'/N$ называется *потенциальной опровержимостью* и является одной из наиболее важных характеристик практической полезности эмпирической гипотезы.

Сравним для примера три гипотезы:

1. «В пассивных электрических цепях могут встретиться любые сочетания значений тока I, сопротивления R и напряжения V».

2. «В пассивных электрических цепях при постоянном сопротивлении R сила тока I прямо пропорциональна напряжению V».

3. «В пассивных электрических цепях всегда выполняется соотношение $V = IR$».

Первую гипотезу, которая похожа на высказывание типа «В этом мире все возможно», опровергнуть ничем нельзя, и она не имеет никакой практической ценности. Вторая гипотеза могла бы быть опровергнута значительным числом мыслимых протоколов, и в некоторых практических задачах знание замеченных ею свойств электрических цепей может оказаться полезным. Третья же гипотеза (закон Ома) крайне рискованна, она могла бы быть опровергнутой бесконечным числом мыслимых ситуаций. Однако таких ситуаций пока никому обнаружить не удалось, и мы с большой пользой для приложений можем, опираясь на нее, предсказывать одну из электрических величин по значениям двух других величин.

Потенциальная опровержимость позволяет отличать содержательные научные гипотезы или теории от бессодержательных, псевдонаучных [96].

Очевидна также важность и другой характеристики эмпирических гипотез — степени их подтвержденности (P). Чем большее число проведенных в прошлом разных экспериментов не опровергали, а подтверждали гипотезу, тем с бóльшим доверием мы относимся к такой гипотезе. Доверие к гипотезе еще больше возрастает, если нам удается найти объяснение фактам, про которые она говорит. Степень объясненности (R), т. е. система ответов на вопросы «Как это происходит?», «Почему именно так, а не иначе?», является еще одной характеристикой, по которой одна гипотеза отличается от другой.

Помимо характеристик, касающихся содержания гипотез, важную роль играет и форма, в которой представлена гипотеза.

В методологической литературе [119] широко известен принцип простоты (S). Средневековый философ Оккам говорил: «Сущности не должны быть умножены сверх необходимости». Это означает, что при одном и том же эмпирическом содержании следует предпочитать наиболее простую гипотезу или теорию. Часто обращают внимание и на такую тонкую характеристику гипотез, как изящество или красота ее формулировки (B). В истории науки имеется много удивительных примеров плодотворного использования этих характеристик гипотез. Так, в книге [152], описывающей историю открытия структуры молекул ДНК, приводится пример использования рассуждения такого рода: «Теоретически все верно, но этого не может быть в природе: уж очень это некрасиво».

Из вышесказанного видно, что при знакомстве с гипотезой мы обращаем внимание не только на эмпирическое содержание, но и на ее внешние характеристики. Т. е. мы имеем дело всегда с объектом $Z = \langle h, Q, P, R, S, B \rangle$. Этот объект Z мы и называем *закономерностью*. Следует отметить, что попытки найти способы количественной оценки указанных выше свойств эмпирических гипотез пока не увенчались особыми успехами. Сейчас нельзя сказать, как одну закономерность можно предпочесть другой, если они отличаются в противоположных направлениях по двум или большему числу своих характеристик.

Можно различать разные стадии развития закономерностей. В самом начале своего пути гипотезы могут ограничивать малую часть мысленно возможных ситуаций, быть мало подтвержденными и слабо объясненными, сложно и неряшливо сформулированными. Это — гипотезы-претенденты, которые служат сырьем для дальнейшего исследования. Некоторые из них, выдержав все испытания, достигают своего совершенства: они предельно рискованны, но многочисленные эксперименты не опровергали, а лишь подтверждали их, природа описываемых ими явлений нашла глубокое всестороннее объяснение, они сформулированы просто и изящно. Такие гипотезы мы называем *законами природы*, наука ими больше не занимается, они передаются практикам для уверенного и полезного применения.

В промежутке между этими крайностями находятся закономерности, с которыми и имеет дело наука, пытаясь усиливать их: сделать гипотезы более строгими и проверенными, найти их содержанию более глубокие объяснения, сформулировать утверждения более просто и красиво. Строго говоря, наука редко имеет

дело с обнаружением закономерностей, т. е. с актом выдвижения самых первых сырых гипотез. Как человек выделяет какую-то часть мира, догадывается из бесконечного числа свойств измерять некоторый конечный их набор и затем формулирует свои исходные предположения о том, что может и чего не может быть — наука пока не знает. Наука сейчас только начинает подступаться к изучению этих фундаментальных творческих процессов.

В данной книге мы имеем дело с методами усиления только одной характеристики гипотез — потенциальной опровержимости Q. Если наблюдать за тем, как работают исследователи, то можно прийти к выводу, что существует некоторый загадочный алгоритм Q-усиления гипотез (F), на вход которого подается исходная гипотеза h_0 и «обучающий» протокол наблюдений pr_0, а на его выходе получается более сильная гипотеза h_1: $F(h_0, \mathrm{pr}_0) = h_1$. При этом h_1 отличается от h_0 только своим более строгим и потому более рискованным тестовым алгоритмом: существуют такие протоколы pr, которые гипотеза h_0 считала допустимыми, и для них $T_0(\mathrm{pr}) = 1$, а новая гипотеза h_1 считает их недопустимыми и выдает результат $T_1(\mathrm{pr}) = 0$. Требования к такого рода алгоритмам усиления F и вариант одного из алгоритмов, удовлетворяющих этим требованиям, можно найти в работах [90, 143].

Глава 2

Классификация задач анализа данных

§ 1. Теория измерений

Специалист, записывающий протокол своих наблюдений в виде таблицы данных «объект-свойство», свободен в выборе языка ведения протокола. Чаще всего в таблицы записываются цифры, однако встречаются и буквы, рисунки, тексты и т. д. Если протокол предназначен только для «ручного» использования его самим автором, то важно лишь, чтобы он помнил, какие свойства он измерял и как он их закодировал — цифрами или знаками. Если же протокол будет использоваться и другими людьми или компьютерными программами, то требуется обеспечить однозначное понимание смысла протокола любым его пользователем. Под воздействием этого очевидного требования выработалась некоторая дисциплина ведения протоколов. Из потенциально неограниченного круга способов отображения свойств наблюдаемого мира в протоколы распространение получило всего несколько, которые стали «общепринятыми», хорошо изученными и потому однозначно всеми понимаемыми. Их изучением и описанием занимается теория измерений [146]. Здесь мы представим основные сведения из этой теории, которые нужны для описания методов анализа данных.

1.1. Типы измерительных шкал. В процессе измерения участвуют два объекта: измерительный прибор и измеряемый

объект. В результате их взаимодействия прибор приходит в некоторое состояние, которое в зависимости от вида прибора и измерительной процедуры фиксируется тем или иным способом: положением стрелки на физической приборной шкале, цветом лакмусовой бумажки, цифрами на электронном табло, положительным или отрицательным ответом на вопрос социолога и проч. Затем это состояние прибора отображается в протоколе различными символами — цифрами, буквами, словами.

Теория измерений оперирует понятием «эмпирическая система с отношениями» (E), которая включает в себя множество измеряемых объектов (A) и набор интересующих исследователя отношений между этими объектами (R): $E = \{A, R\}$. Например, множество A — это множество физических тел, а набор R — отношения между ними по весу, твердости, размерам и т. п. Для записи результатов наблюдений используется символьная система с отношениями (N), состоящая из множества символов (M), например множества всех действительных чисел, и конечного набора отношений (P) на этих символах: $N = \{M, P\}$.

Отношения P выбираются так, чтобы ими было удобно отображать наблюдаемые эмпирические отношения R. Если тело t тяжелее тела q, т. е. если имеет место отношение R $(t > q)$, то цифровая запись веса тел $t = 5$ и $q = 3$ позволяет наглядно увидеть это эмпирическое событие в записи P $(5 > 3)$. Договоренность использовать именно такое отображение системы E на систему N означает выбор некоторого определенного правила отображения g. Тройка элементов $\langle E, N, g\rangle$ называется *шкалой* (не следует путать с физической приборной шкалой).

Но мы можем договориться и о некотором другом способе отображения w и тогда будем иметь дело с другой шкалой $\langle E, N, w\rangle$. Например, g рекомендует записывать вес тел в килограммах, а w — в граммах или тоннах. Цифровая запись в протоколах будет при этом разная, но эмпирическое содержание протоколов будет одинаковым. Это означает, что мы выбрали не любые способы отображений (g, w и т. д.), а только те, которые связаны между собой взаимно однозначными преобразованиями. Т. е. имеется такое преобразование f, с помощью которого по записи в языке g можно точно определить, какой будет запись в языке w (и наоборот): $g = f(w)$ и $w = f'(g)$. Преобразование f объединяет указанные выше по-разному выглядящие шкалы в определенную группу, которая называется *типом шкалы*. Зафиксировав допустимое преобразование f, мы тем самым фиксируем конкретный

тип шкалы.

В практике научных исследований получили распространение шкалы всего нескольких типов. Приведем описание шкал основных типов.

1. Абсолютная шкала. Допустимое преобразование для шкал данного типа представляет собой тождество, т. е. если на одном языке в протоколе записано y, а на другом языке x, то между ними должно выполняться простое соотношение: $y = x$. Этот тип шкалы удобен для записи количества элементов в некотором конечном множестве. Если, пересчитав количество яблок, один запишет в протоколе 6, а другой запишет VI, то нам достаточно знать, что 6 и VI означают одно и то же, т. е. что между этими записями существует тождественное отношение: 6 = VI.

2. Шкала отношений. Между разными протоколами, фиксирующими один и тот же эмпирический факт на разных языках, при этом типе шкалы должно выполняться соотношение: $y = ax$, где a — любое положительное число. Один и тот же эмпирический смысл имеют протоколы: 16 кг, 16000 г, 0,016 т, 1 пуд, 40 фунтов. От любой записи можно перейти к любой другой, подобрав соответствующий множитель a. Этот тип шкалы удобен для измерения весов, длин и т. д. Если нам не известно, в каких именно единицах записаны веса тел в разных протоколах, то мы можем полагаться только на отношение весов двух тел. Например, тело с весом 10 единиц в два раза тяжелее тела с весом 5 единиц вне зависимости от того, что было взято за единицу — тонна или грамм. Инвариантность отношений отражена в названии шкалы данного типа. Если же в протоколе указана единица веса, то такой протокол отражает свойства тел в абсолютной шкале.

3. Шкала интервалов. Здесь между протоколами y и x допустимы линейные преобразования: $y = ax + b$, где a — любое положительное число, а b может быть как положительным, так и отрицательным. Это значит, что в разных протоколах может использоваться разный масштаб единиц (a) и разные начала отсчета (b). Примером шкал этого типа могут быть шкалы для измерения температуры. Если в протоколе указаны градусы, но не говорится в какой шкале (Цельсия, Кельвина и т. д.), то во избежание недоразумений при описании закономерностей можно использовать только отношения интервалов, так как при любых

значениях a и b сохраняется равенство

$$(y_1 - y_2)/(y_3 - y_4) = \{(ax_1 + b) - (ax_2 + b)\}/\{(ax_3 + b) - (ax_4 + b)\}.$$

Если записи в протоколе сопровождаются информацией о том, какие именно градусы имеются в виду (например, 18 °C), то мы имеем дело с протоколом в абсолютной шкале.

4. Шкала порядка. Допустимыми преобразованиями для данного типа шкалы являются все монотонные преобразования, т. е. такие, которые не нарушают порядок следования значений измеряемых величин. Такие протоколы появляются, например, в результате сравнения тел по твердости. Записи «1; 2; 3» и «5,3; 12,5; 109,2» содержат одинаковую информацию о том, что первое тело самое твердое, второе менее твердое, а третье — самое мягкое. И никакой информации о том, во сколько раз одно тверже другого, на сколько единиц оно тверже, в этих записях нет и полагаться на конкретные значения чисел, на их отношения или разности нельзя.

Разновидностью шкалы порядка является шкала рангов, где используются только числа, идущие подряд от 1 вверх по возрастанию. Если среди m измеряемых объектов одинаковых нет, то ранговое место каждого объекта в протоколе будет указано одним из целых чисел от 1 до m. При одинаковом значении измеряемого свойства у k объектов, занимающих порядковые места с t-го по $(t+k)$-е, их ранги будут обозначены одинаковым числом, равным их «среднему» рангу x, где $x = (1/k)\sum_{i=1}^{k}(i + t - 1)$. Такая разновидность шкалы порядка называется *нормированной шкалой рангов*.

К типу шкал порядка относится и широко используемая шкала баллов. При этом используются целые числа в ограниченном диапазоне их значений: от 1 до 5 в системе образования, от 0 до 6 или 10 в спорте и т. д. В любом из этих случаев протокол содержит информацию только о трех эмпирических отношениях: $<$, $>$ и $=$.

5. Шкала наименований. Здесь фиксируется только два отношения: «равно» и «не равно». Следовательно, допустимы любые преобразования, лишь бы в протоколе одинаковые объекты были поименованы одинаковыми символами (числами, буквами, словами), а разные объекты имели разные имена. Так фиксируются в протоколах такие характеристики, как собственные имена людей, их национальность, названия населенных пунктов и т. п.

1.2. Сравнительная информативность шкал. Представляет интерес вопрос об относительной информативности измерительных шкал разного типа. С позиций порядковой шкалы ответ на этот вопрос ясен: информативность шкал убывает в том порядке, как они приведены выше. Действительно, пусть абсолютная шкала указывает, что множество A содержит 30 элементов, а множество B — 10 элементов. На языке шкалы отношений этот факт будет отражен в виде записи «количество элементов в A в три раза больше, чем в B». Протокол в шкале порядка будет говорить о том, что в A элементов больше, чем в B. А на языке шкалы наименований можно записать лишь то, что в A и B содержится разное количество элементов. Так что информации, содержащейся в абсолютной шкале, достаточно, чтобы сделать ее однозначное отображение на более слабую шкалу. Обратное отображение будет неоднозначным: из того факта, что A не равно B, нельзя узнать, какое из этих множеств больше, на сколько или во сколько раз больше и, тем более, сколько элементов содержится в A и B.

Шкалы первых трех типов содержат более богатую информацию, их показания можно подвергать определенным математическим преобразованиям, и потому их часто называют сильными, количественными или арифметическими. Шкалы порядка и наименований уступают им по информативности и отражают качественные свойства — их обычно называют слабыми и качественными. Однако рекомендовать пользоваться только сильными шкалами нельзя. Приборы для измерения сильных свойств более дорогие, для измерения многих свойств в сильных шкалах (особенно в гуманитарных областях) таких приборов еще нет.

Было бы интересно узнать, как много информации мы теряем, переходя от некоторой сильной шкалы к более слабой. Представление об этом можно получить следующим путем [89]. Будем считать, что измерительный прибор может принимать одно из m состояний. Пусть этим прибором измеряется некоторое свойство у n объектов. Если $n = 1$, то независимо от типа шкалы количество возможных разных (неизоморфных) протоколов равно m. Если объектов больше одного, то количество неизоморфных протоколов будет для разных типов шкал различным. Так, например, протоколы «2; 6» и «3; 9» в абсолютной шкале будут неизоморфными, а в шкале отношений, порядка и наименований одинаковыми (изоморфными). Отсюда появляется возможность сравнивать информативность шкал разного типа путем опреде-

ления количества возможных неизоморфных протоколов, которые можно получить на языке этих шкал при заданных значениях m и n.

По этой методике было проведено сравнение шкал трех типов: абсолютной, порядковой и наименований. Сравнение делалось в шкале отношений: количества неизоморфных протоколов для шкалы порядка (s_o) и шкалы наименований (s_n) соотносились с количеством разных протоколов для абсолютной шкалы (s_a). Выяснилось, что для фиксированного значения числа градаций m с ростом количества измеряемых объектов n различия в информативности разных шкал уменьшаются. Однако отношение s_n/s_a меняется слабо и остается малым. Отношение же s_o/s_a растет быстро и при $n > 5m$ достигает величины порядка $0,9$. Т. е. информативность шкалы порядка при экспериментах с большим числом объектов приближается к информативности абсолютной шкалы. Так что в ряде случаев с помощью более простых приборов или процедур можно получить почти столько же информации, сколько и с помощью сложных и дорогих.

Этот вывод подтверждается результатами эксперимента, проведенного с экспертами. 28 экспертов должны были оценить некоторое неформализованное свойство 10 объектов в шкале порядка. Каждый эксперт упорядочивал объекты по своему усмотрению и приписывал им целочисленные порядковые значения в диапазоне от 1 до 10. Затем им было предложено оценить свойство тех же объектов в шкале отношений (в процентах к самому лучшему). Эта задача оценивалась всеми экспертами как существенно более трудная. После завершения этой работы для каждого объекта были определены средние значения их порядковых мест и средние значения процентных оценок. Оказалось, что коэффициент линейной корреляции между этими средними оценками равен $0,93$! Отсюда можно сделать полезный вывод для практики группового экспертного оценивания: не нужно заставлять экспертов давать ответы в сильных шкалах. При количестве экспертов около 30 достаточно ограничиться оценками в шкале порядка и лишь для двух объектов, получивших самый высокий и самый низкий средний порядковый балл, сделать оценку в сильной шкале. Этих калибровочных величин будет достаточно для перехода от средних значений в шкале порядка к средним значениям в шкале отношений.

§ 2. Классификация задач анализа данных [72]

Анализ данных, представленных таблицей «объект-свойство-время» (ТОСВ) (трехвходовая таблица или куб данных на рис. 1), всегда включает в себя решение задач двух связанных между собой направлений:

а) обнаружение закономерных связей между элементами таблицы и

б) использование обнаруженных закономерностей для предсказания (прогнозирования) значений одних элементов таблицы по известным значениям других ее элементов.

Бывают случаи, когда требуется решать задачу только одного из этих направлений. Например, нужно выявить имеющиеся в таблице закономерности и зафиксировать их для использования в будущем (случай а) или сделать прогноз на базе ранее обнаруженных закономерностей (случай б).

Часто встречается ситуация и комбинированного характера, когда для прогнозирования конкретной характеристики данного объекта требуется сначала обнаружить специфичные закономерности, нужные именно для этого конкретного случая. Такой комбинированный случай (а, б) является более общим и классификацию задач анализа данных мы будем строить, имея в виду именно его.

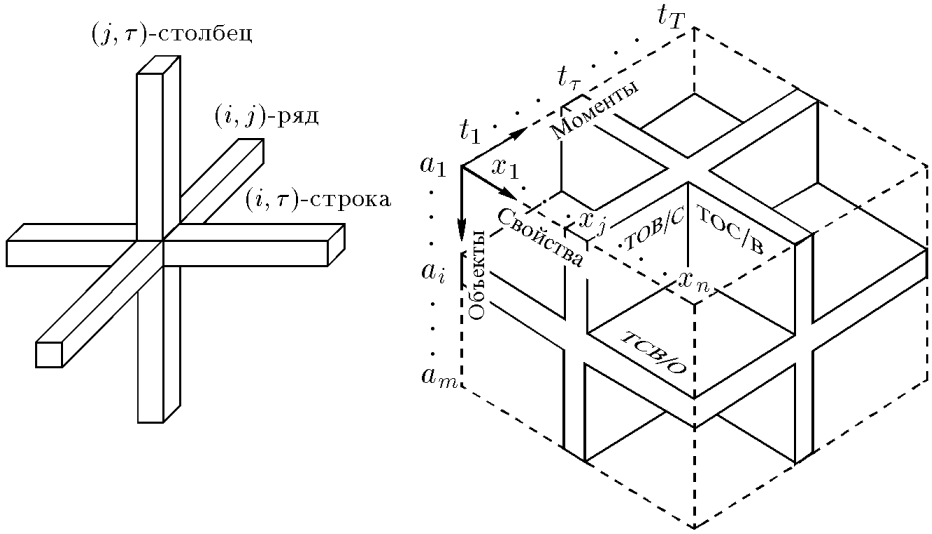


Рис. 1

I. Рассмотрим задачи предсказания элементов в двумерной таблице типа «объект-свойство» (двумерная таблица ТОС/В на рис. 1), строки в которой a_i $(i = 1, 2, \ldots, m)$ описывают m объектов, а столбцы x_j $(j = 1, 2, \ldots, n)$ соответствуют n свойствам (характеристикам) этих объектов. Отсутствие информации о времени (что отмечено знаком /В) означает, что все измерения проведены в один и тот же момент времени t_τ или что свойства изучаемых объектов от времени не зависят.

Предсказываемые элементы (b) в ТОС/В могут располагаться по-разному. В зависимости от этого выделим три *семейства задач*:

1) все элементы b_{i0} расположены в одном столбце;
2) все элементы b_{j0} расположены в одной строке;
3) элементы b_{ij0} принадлежат разным столбцам и строкам.

В каждом семействе выделяем *классы задач* в зависимости от того, какое количество (q) элементов нужно предсказывать. Первое семейство по этой характеристике имеет три класса задач:

1.1) предсказывается один элемент $(q = 1)$;
1.2) предсказывается сразу несколько элементов $(1 < q < m)$;
1.3) предсказываются сразу все элементы столбца $(q = m)$.

Аналогичным способом выделим классы задач во втором семействе:

2.1) $q = 1$;
2.2) $1 < q < n$;
2.3) $q = n$.

В третьем семействе выделяется два класса задач:

3.2) $1 < q < mn$;
3.3) $q = mn$.

В каждом из этих восьми классов различаем *типы задач* в соответствии со шкалами, в которых измеряются значения предсказываемых элементов. Выделим три группы шкал: наименований (Н), порядка (П) и количественные (К). Ситуацию, при которой предсказываются разнотипные элементы, обозначим символом (Р). Описанная классификация задач анализа данных приведена в табл. 1.

Приведем примеры некоторых распространенных типов задач предсказания.

Таблица 1

Классификация задач анализа данных

Семейство задач			Предсказание элементов столбца			Предсказание элементов строки			Предсказание элементов таблицы	
Классы задач			1.1 $q{=}1$	1.2 $1{<}q{<}m$	1.3 $q{=}m$	2.1 $q{=}1$	2.2 $1{<}q{<}n$	2.3 $q{=}n$	3.2 $1{<}q{<}mn$	3.3 $q{=}mn$
Типы задач	Шкалы	Н	*	*	*	*	*	*	*	*
		П	*	*	*	*	*	*	*	*
		К	*	*	*	*	*	*	*	*
		Р	–	–	–	–	–	–	*	*

Задача 1.1.Н состоит в предсказании одного элемента в столбце, измеренном в шкале наименований. В этом обычно состоит задача распознавания образов: указать имя образа (класса), которому принадлежит некоторый новый объект a (определить тип заболевания, предсказать наличие или отсутствие нефти и пр.). В задаче 1.1.П все объекты упорядочены по целевому свойству x_0 и требуется определить место нового объекта a в этом порядке (например, предсказать, что нефти в месторождении a больше, чем в a_i, но меньше, чем в a_{i+1}). В случае 1.1.К нужно указать количественную характеристику x_0 объекта a (например, предсказать запасы нефти в миллионах тонн).

Похожие по смыслу задачи составляют класс 1.2. Только здесь нужно принимать решение сразу о нескольких элементах: распознать q объектов (тип 1.2.Н), определить порядковые позиции группы объектов (тип 1.2.П) или оценить количественную характеристику x_0 сразу для q объектов (1.2.К).

Важную роль играют задачи класса 1.3. Разделить объекты по похожести их свойств, т. е. установить некоторую их классификацию, — значит сформировать в ТОС некоторый новый столбец x_0, измеряемый в шкале наименований (задача типа 1.3.Н). Часто ее называют задачей автоматической классификации или

таксономии. При экспертном оценивании m объектов с участием n экспертов требуется определить итоговую оценку либо в шкале порядка (тогда это задача 1.3.П), либо в более сильной шкале, например в процентах (задача типа 1.3.К).

С задачами второго семейства сталкиваются, когда нужно, например, оценивать информативность свойств, представленных в таблице. Если имеющиеся свойства предварительно разделены на информативные и неинформативные классы, то при необходимости определения, к какому из этих классов следует отнести некоторое новое свойство, мы имеем дело с задачей типа 2.1.Н. Если требуется указать порядковое место нового свойства в предварительно упорядоченном наборе свойств, то решается задача 2.1.П. А если требуется оценить информативность свойства в битах, то имеет место задача 2.1.К. Для группы свойств в этом классе формулируются задачи 2.2.Н, 2.2.П и 2.2.К. Очевидна интерпретация и задач оценки всей совокупности свойств сразу (задач типа 2.3.Н, 2.3.П и 2.3.К).

Представим себе таблицу с пробелами в разных столбцах и строках. Для предсказания значений пропущенных элементов приходится решать задачи разных типов из класса 3.2, в том числе и задачу предсказания разнотипных элементов 3.2.Р. Наконец, класс 3.3 охватывает задачи генерации таблиц с заданными свойствами: тестовых таблиц для проверки программ распознавания образов, таблиц случайных чисел и т. п. В зависимости от требуемого типа шкал имеют место задачи типа 3.3.Н, 3.3.П, 3.3.К или 3.3.Р.

II. Рассмотрим теперь таблицу, которая отображает значения некоторого свойства x_j для всех m объектов в каждый из t_τ, $\tau = 1, 2, \ldots, T$, моментов времени (двумерная таблица ТОВ/С на рис. 1). Такого рода таблица описывает, например, урожай зерновых во всех m хозяйствах в разные годы за последние T лет. Все исходные данные в этой таблице измерены в шкале одного и того же типа. Что касается выделенного (целевого) столбца t_0, то тип его шкалы может быть любым — Н, П или К. В зависимости от этого для предсказания элементов целевого столбца могут потребоваться методы решения задач семейства 1. Так, при необходимости сделать автоматическую классификацию объектов по похожести их характеристики x_j во времени решается задача типа 1.3.Н. Для указания порядковых позиций для некоторого набора новых объектов среди ранее упорядоченных объектов по динамике их характеристики x_j решается задача 1.2.П и т. д.

Если задана целевая строка a_0, то формулируемые на ней задачи принадлежат семейству 2. Например, группировка (таксономия) моментов времени по похожести значений характеристики x_j для всех m объектов приводит к формулировке задачи типа 2.3.Н. Если нужно нескольким выделенным моментам времени приписать некоторую количественную характеристику, то нужно будет решать задачу 2.2.К и т. д.

В случае, когда предсказываемые элементы разбросаны по разным столбцам и строкам, решаются задачи семейства 3.

Существенное отличие таблицы ТОВ/С от таблицы ТОС/В состоит в том, что столбцы в ТОВ/С связаны друг с другом отношением порядка следования. Эта дополнительная информация может оказаться полезной при решении некоторых задач, например задачи прогнозирования многомерных динамических рядов.

III. Теперь обратимся еще к одному сечению куба данных — к таблице ТСВ/О (см. рис. 1), строками которой считаем свойства некоторого выделенного объекта, а столбцами — моменты времени измерения этих свойств. Примером такой таблицы может служить протокол наблюдения за n симптомами одного пациента в течение T дней.

Здесь снова можно группировать строки (т. е. симптомы по похожести их динамики во времени) и столбцы (моменты времени с похожими «профилями» симптомов), для чего потребуются методы таксономии типа 1.3.Н или 2.3.Н. Для предсказания порядковых позиций группы новых свойств среди предварительно упорядоченных свойств решается задача типа 1.2.П. Если нужно определить, к какому из ранее выделенных типов (классов) моментов времени следует отнести некоторый новый момент времени, то нужно решать задачу распознавания образов, т. е. задачу типа 1.1.Н. Понятна на этой таблице интерпретация и других типов задач. Снова отметим, что порядок столбцов в таблице менять нельзя. На это ограничение можно опереться при решении некоторых задач.

IV. Наконец от рассмотрения задач на плоских сечениях куба данных перейдем к рассмотрению задач на кубе в целом. При решении описанных задач использовались закономерности, обнаруживаемые только на той или иной двумерной таблице. Хотелось бы иметь возможность работать с закономерностями, рассеянными в кубе и не представленными целиком ни в одном из его сечений.

В дальнейшем мы опишем методы решения ряда задач ана-

лиза данных для случая, когда исходная информация представлена именно такими трехвходовыми таблицами или кубами данных типа ТОСВ. Некоторые из этих методов предварительно делают из куба данных большую двумерную таблицу. Это можно сделать, «склеивая» плоские сечения (например, представляющие собой таблицы ТОС/В) в единое «полотно», в котором содержится m строк (объектов) и nT столбцов: n признаков, измеренных в разные T моментов времени. К такой таблице дальше применяются методы, разработанные для данных типа ТОС/В. Имеются и методы, не использующие предварительного преобразования кубов в таблицы.

V. Не все описанные типы задач одинаково хорошо изучены: некоторые имеют давнюю историю, широко известны, имеют хорошо отработанные алгоритмы и программы для их решения, которые применяются в разных прикладных областях. Другие известны меньше, но понятны и иногда используются. Есть и такие, которые пока ясно не формулировались и интерпретация которых вызывает затруднения.

В дальнейших разделах книги описываются в основном хорошо изученные типы задач, методы решения и примеры их приложений в различных содержательных областях. Основная часть описываемых методов реализована в программах пакета ОТЭКС [82].

Глава 3

Базовые гипотезы, лежащие в основе методов анализа данных

Как будет видно из дальнейшего, строгие математические методы, используемые в математической статистике, разработаны для случаев, когда о распределениях анализируемых генеральных совокупностей известно все, что только может потребоваться в процессе решения задачи: известны виды законов распределений и все их параметры, априорные вероятности появления образов, матрица потерь от ошибок и т. д.

К сожалению, при решении реальных задач анализа данных такие условия не встречаются. Так, в задаче распознавания обучающая выборка каждого из k образов представлена конечным числом m реализаций a_i $(i = 1, 2, \ldots, m)$, описанных n характеристиками x_j $(j = 1, 2, \ldots, n)$. Сведений о законах и параметрах распределения генеральных совокупностей (G) образов нет. В частности, ничего не известно о зависимости одних признаков от других. Не известна связь обучающей выборки с генеральной совокупностью, т. е. не известна степень представительности (R) выборки. Владелец обучающей выборки («заказчик») имеет туманные представления об априорной вероятности появления разных образов и о матрице стоимости ошибок распознавания. (Оставим пока в стороне те обычно сопутствующие факты, что выборка бывает очень небольшой, в данных есть ошибки и пробелы, признаки измерены в разных шкалах и среди них имеются неинформативные, шумящие признаки и пр.)

Совершенно очевидно, что для приведения ситуации к виду, при котором можно было бы применить тот или иной статисти-

ческий алгоритм, нужно к имеющейся объективной информации добавить ряд субъективно выбираемых предположений или гипотез. Этот этап привнесения эвристических гипотез, значение которого подчеркивалось в первой главе, имеет место во всех случаях решения реальных задач распознавания образов и потому деление алгоритмов на строгие статистические и нестрогие эвристические не имеет смысла.

Дополнительные гипотезы могут носить общий характер или касаться мелких частностей. Здесь будут описаны две базовых гипотезы — компактности и λ-компактности [74] — и показано их влияние на характер алгоритмов анализа данных.

§ 1. Гипотеза компактности

Одна из давно используемых эмпирических гипотез, известная в литературе по распознаванию образов под именем гипотезы компактности (обозначим ее через H), состоит в том, что реализации одного и того же образа обычно отражаются в признаковом пространстве в геометрически близкие точки, образуя «компактные» сгустки [6]. При всей кажущейся тривиальности и легкости опровержения указанная гипотеза лежит в основании большинства алгоритмов не только распознавания, но и всех других задач анализа данных.

Конечно, она подтверждается не всегда. Если, например, среди признаков имеется много случайных, неинформативных, то точки одного образа могут оказаться далекими друг от друга и рассеянными среди точек других образов. Но дополнительно предполагается, что в многомерном признаковом пространстве уже было найдено такое (информативное) подпространство, в котором точки одного класса действительно образуют явно выделяемые компактные сгустки. Назовем n признаков, входящих в информативное подмножество X, *описывающими*, а номинальный $(n+1)$-й признак z, указывающий имя образа, *целевым*. Обозначим множество объектов обучающей выборки через A, новый распознаваемый объект через q, а тот факт, что объекты множества A компактны (эквивалентны, похожи или близки друг другу) в пространстве n характеристик X — через C_A^X. Мера компактности может быть любой: она может характеризоваться средним расстоянием от центра тяжести до всех точек образа; средней длиной ребра полного графа или ребра кратчайшего незамкнутого пути, соединяющего точки одного образа; максималь-

ным расстоянием между двумя точками образа и т. д. Например, компактными (эквивалентными) считаем два объекта, если все признаки одного объекта равны соответствующим признакам другого. Или: объекты компактны, если евклидово расстояние между векторами их признаков не превышает величину r.

Фактически гипотеза H равнозначна предположению о наличии закономерной связи между признаками X и z, и с учетом вышесказанного ее тестовый алгоритм может быть представлен следующим выражением: if $\left(C_A^{X,z} \& C_{A,q}^X\right)$ then $C_{A,q}^z$. Т. е. если объекты множества A компактны в пространстве (X, z) и объекты множества (A, q) компактны в пространстве описывающих свойств X, то объекты A и q будут компактными и в пространстве целевого признака z. Часто эту гипотезу формулируют так: «Объекты, похожие по n описывающим свойствам X, похожи и по $(n+1)$-му целевому свойству z». Легко видеть, что в этой более краткой формулировке опущены весьма существенные дополнительные условия.

Заметим, что деление свойств на описывающие и целевые является условным. Мы можем целевой признак z включить в число описывающих, а в качестве целевого принять любой признак x_j из информативной системы X. Если при этом обучающие объекты множества A компактны в новом пространстве свойств $\{x_1, x_2, \ldots, x_{n-1}, z, x_j\}$ и множество (A, q) компактно в пространстве $\{x_1, x_2, \ldots, x_{n-1}, z\}$, то значение нового целевого признака x_j у объекта q будет эквивалентным его значению у объектов множества A. Целевыми могут быть не одна, а несколько характеристик. В частности, гипотеза H позволяет решать не только задачу анализа, когда по признакам X распознается образ z, но и обратную задачу — задачу синтеза, когда по имени образа z восстанавливаются наиболее правдоподобные значения характеристик X (например, путем приписывания объекту q с признаком z свойств «типичного» представителя образа z).

Указаний на то, какое число n признаков и какое число m объектов обучающей выборки A нужно иметь, чтобы гипотеза H гарантированно подтверждалась, здесь нет и быть не может. Информативность признаков и представительность выборки являются понятиями условными. Система признаков информативна, если при заданной обучающей выборке и заданном типе решающих правил удается построить правило, распознающее объекты контрольной выборки с заданной точностью. Обучающая вы-

борка представительна, если при заданном наборе признаков и заданном типе решающих правил удается то же самое.

Можно найти случаи, когда для успешного решения задачи достаточно иметь всего один признак и по одной обучающей реализации на образ. Пусть, например, образы A и B представляют теплокровных млекопитающих, а описывающий признак есть их вес. Если A и B — это слоны и мыши, то достаточно измерить вес одного любого представителя множества A и любого представителя множества B, чтобы построить безошибочное правило распознавания любого нового представителя этих образов.

Совсем другая ситуация возникает, если мы захотим распознавать этих же млекопитающих, но по признаку окраса их волосяного покрова. Если в конце концов окажется, что мыши темнее слонов, то для установления этого факта потребуется обучающая выборка гораздо большего объема. Можно отметить, что с ростом числа обучающих реализаций уверенность в правильности, неслучайности обнаруживаемой закономерности растет.

§ 2. Гипотеза λ-компактности

Гипотеза компактности оперирует абсолютными значениями расстояний между векторами в пространстве характеристик. Однако на некоторых примерах можно показать, что важную роль в задачах анализа данных играют не только сами расстояния, но и отношения между ними. Так, расстояние между точками 5 и 6 на рис. 2, *а* меньше, чем между 6 и 7, но, делая «вручную» таксономию этого множества точек на два таксона, эксперты обычно проводят границу по ребру 5–6. Глаз человека улавливает на этой границе нарушение однородности расстояний между соседними точками и придает этому факту большее значение, чем абсолютной величине расстояний.

Зрительный аппарат человека обладает уникальными способностями делать классификацию (таксономию) множества объектов, если они представлены точками на плоскости [68]. На рис. 3 представлены примеры множеств, таксономия которых для человека не составляет труда. Результаты получаемой при этом естественной для человека таксономии (два сгустка и фон) не могут быть получены или объяснены с позиций гипотезы компактности. Гипотеза же λ-компактности позволяет легко получать и просто объяснять такие результаты.

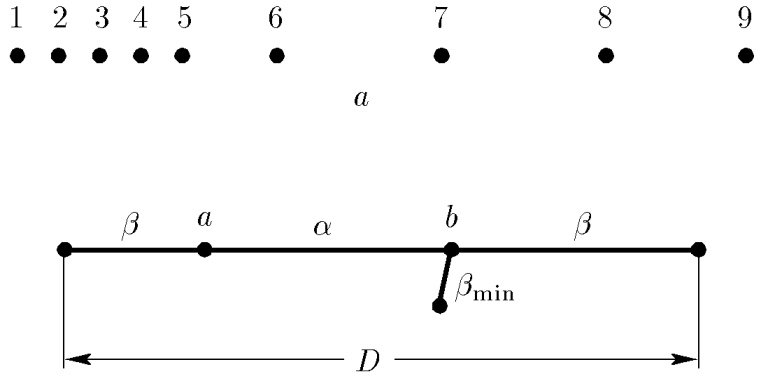


Рис. 2

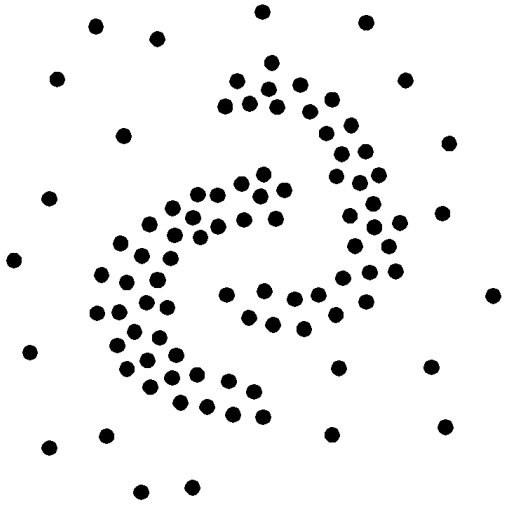


Рис. 3

Формулировка гипотезы λ-компактности опирается на понятие λ-расстояния, которое учитывает нормированное расстояние d между элементами множества и характеристику τ локальной плотности множества в окрестностях этих элементов.

Если определить расстояния между всеми парами точек мно-

жества A, то можно построить полный граф, соединяющий все точки со всеми, и найти самое длинное ребро — диаметр графа (D). Выделим две любые точки a и b и обозначим длину связывающего их ребра через $\alpha(ab)$. Будем считать нормированным расстоянием между этими точками величину $d = \alpha/D$ (см. рис. 2, *б*).

Теперь среди ребер, смежных ребру (ab), найдем самое короткое, длину которого обозначим через $\beta_{\min}$. Отношение длин этих смежных отрезков обозначим через $\tau^* = \alpha/\beta_{\min}$. Чтобы сделать эту величину нормированной в диапазоне от нуля до единицы, найдем в полном графе наибольшее значение $\tau_{\max}$. Величина $\tau = \tau^*/\tau_{\max}$ является нормированной характеристикой локальной неоднородности плотности множества в окрестностях точек a и b. Величину $\lambda = f(\tau, d)$ называем *λ-расстоянием* между точками a и b. Для определения степени влияния параметров τ и d на λ-расстояние были проведены эксперименты, в которых сравнивались результаты таксономии двумерного множества точек экспертами и программами, использующими разные виды функции $f(\tau, d)$. Выяснилось, что использование λ-расстояний вместо евклидовых позволяет получать более естественную с точки зрения экспертов таксономию, совпадающую с результатами, полученными экспертами. При этом оказалось, что параметр d играет более важную роль по сравнению с параметром τ. Наилучшее совпадение экспертных суждений с формальными получалось в том случае, если в качестве меры расстояния использовалась величина $\lambda = \tau^2 \times d$.

Находя λ-характеристики для всех отрезков, соединяющих точки множества A, мы делаем отображение этого множества из евклидова пространства в новое λ-пространство. В этом пространстве можно построить граф без петель, который связывает между собой все точки и имеет минимальную суммарную λ-длину своих ребер. Такой граф в евклидовом пространстве называется *кратчайшим незамкнутым путем* (КНП). По аналогии с этим, КНП в λ-пространстве обозначим через λ-КНП. Строятся такие графы следующим способом [135]. Сначала находятся две самые близкие точки, которые соединяются ребром. Затем соединяется ребром следующая пара самых близких точек. Для каждой следующей пары ближайших точек предварительно проверяется, нельзя ли пройти из одной из них в другую по ребрам уже построенного графа. Если можно, то они из дальнейшего рассмотрения исключаются, а если нет, то строится ребро графа между ними. Так продолжается до объединения в общий граф

всех m точек множества A. На рис. 4 представлен пример полного графа (ПГ) и графов КНП и λ-КНП для одного и того же множества A. Теперь λ-расстоянием между двумя любыми точками считаем сумму λ-характеристик тех ребер, по которым проходит путь между ними по λ-КНП.

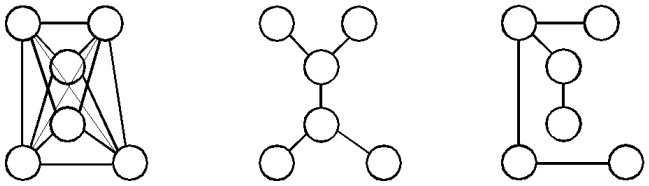


Рис. 4

Если геометрическая близость точек связывалась с понятием компактность, то близость по λ-расстояниям называем *λ-компактностью*. Исходя из этого по аналогии с гипотезой компактности гипотезу λ-компактности (λH) можно сформулировать следующим образом: реализации одного и того же образа обычно отражаются в признаковом λ-пространстве в близкие точки, образуя λ-компактные сгустки.

Если выделить n описывающих признаков X и целевой признак z, то в пространстве (X, z) λ-компактное множество объектов A характеризуется тем, что оно λ-компактно одновременно и по признакам X, и по признаку z. Это означает, что между признаками X и z есть закономерная связь или что система описывающих признаков X информативна для предсказания значений целевого признака z. Если к множеству A добавить новый объект q с известными значениями описывающих признаков X и неизвестным значением целевого признака z и при этом окажется, что множество (A, q) λ-компактно в пространстве X, то из гипотезы λ-компактности следует, что оно будет λ-компактно и в пространстве (X, z). Это дает возможность предсказывать значение целевого признака z для нового объекта q. Обозначим факт λ-компактности объектов A в пространстве X символом λC_A^X. Тогда тестовый алгоритм гипотезы λ-компактности можно представить следующим выражением: if $\left(\lambda C_A^{X,z} \& \lambda C_{A,q}^X\right)$ then $\lambda C_{A,q}^z$.

Как видно из дальнейшего, в задачах разделения множества A на таксоны простой формы, которые описываются непере-

секающимися выпуклыми оболочками, стремление к наибольшей компактности или λ-компактности приводит к одинаковым результатам. Но для более сложных случаев гипотеза λ-компактности обеспечивает получение результата, более естественного по сравнению с гипотезой компактности. Следовательно, гипотеза λ-компактности является более сильной эмпирической гипотезой, чем широко применяемая сейчас гипотеза компактности.

Однако за гипотезой компактности — многолетние традиции и большое количество алгоритмов и программ, построенных на ее основе. Кроме того, переход к λ-пространству сопряжен с дополнительными затратами машинных ресурсов, что в простых случаях себя не оправдывает. В связи с этим в дальнейшем будут описаны методы анализа, основанные на той и другой гипотезах.

Часть II
Методы анализа данных

Глава 4
Задачи таксономии

§ 1. Природа задач таксономии

Содержательную постановку задачи таксономии можно прочитать в работе [121], написанной еще во II в. до нашей эры. В «Письме ученому соседу» Демокрит пишет такие слова: «Если тебе, дорогой друг, нужно разобраться в сложном нагромождении фактов или вещей, ты сначала разложи их на небольшое число куч по похожести. Картина прояснится, и ты поймешь природу этих вещей».

Некоторое время назад новосибирские социологи изучали причины массовой миграции сельского населения в города. В города и села Алтайского края были направлены экспедиции, которые находили людей, недавно переехавших из села в город, а также тех сельских жителей, которые планируют такой переезд в ближайшем будущем. Каждому из них социологи задавали около ста вопросов, записывая в анкету ответы о возрасте, образовании, семейном положении и прочих характеристиках опрашиваемого.

Когда социологи вернулись из экспедиций и выложили на стол заполненные анкеты, они увидели впечатляющую картину. Перед ними была гора из семи тысяч анкет, каждая их которых содержала ответы на сто вопросов. Как подступиться к анализу этих данных? Переписывание данных в протокол «объект-свойство» делу не поможет: таблица размером 7000 строк на 100 столбцов так же не обозрима, как и эта гора анкет.

Пошли по единственно возможному пути: перенесли анкеты на машинные носители и применили к полученной таблице программы таксономии. В результате получили 7 основных таксонов, средние характеристики которых позволили дать им вполне понятную содержательную интерпретацию. Выделился, например, таксон, куда вошли люди женского пола в возрасте более 60 лет, дети которых живут в городе. Ясно, что представители этого таксона, который социологи назвали «бабушки», едут в город нянчить своих внуков. Были также понятные таксоны «демобилизованные солдаты», «невесты» и другие [86].

Группировка объектов (часто употребляют также термины автоматическая классификация, самообучение, кластеризация) по похожести их свойств упрощает решение многих практических задач анализа данных. Так, если объекты описаны свойствами, которые влияют на общую оценку их качества, то в одну группу (таксон) будут собраны объекты, обладающие приблизительно одинаковым качеством. И вместо того, чтобы хранить в памяти все объекты, достаточно сохранить описание типичного представителя каждого таксона (прецедента), перечислить номера объектов, входящих в данный таксон, и указать максимальное отклонение каждого свойства от его среднего значения для данного таксона. Этой информации обычно бывает достаточно для дальнейшего анализа изучаемого множества объектов.

Как же делается таксономия? Одно и тоже множество из m объектов можно разбить на k таксонов ($k < m$) по-разному. Если бы мы записали такое исходное состояние нашего знания в виде эмпирической гипотезы h_0, то ее тестовый алгоритм T_0 считал бы допустимой любую таксономию. Но мы знаем, что человек, делая группировку чего бы то ни было, руководствуется каким-то критерием (обозначим его F), который позволяет отличать хорошие группировки от плохих и выбирать наилучший вариант таксономии. Теперь наше знание позволяет сформулировать более сильную гипотезу, тестовый алгоритм которой считает допустимой только такую таксономию, которая удовлетворяет критерию F.

§ 2. Алгоритмы таксономии класса FOREL

Самый известный критерий F состоит в том, что в один таксон должны собираться объекты, похожие, близкие по своим характеристикам. Но термины похожесть, близость можно понимать по-разному, и в зависимости от того, какой их вариант мы выберем, получится тот или иной вариант таксономии. Остановимся вначале на разновидности меры «похожести» в виде «похожести на центр».

Оговоримся, что здесь мы рассматриваем случай таксономии объектов, признаки которых измерены в сильных шкалах, что позволяет оценивать похожесть через евклидово расстояние между точками в многомерном пространстве. Как поступать в случае с разнотипными признаками, мы обсудим позже.

Если координаты центра j-го таксона обозначить символом C_j, то сумма расстояний $\rho(C_j, a_i)$ между центром и всеми m_j точками a_i этого таксона $\rho_j = \sum \rho(C_j, a_i)$, где $i = 1 \div m_j$, а сумма таких внутренних расстояний для всех k таксонов $F = \sum \rho_j$, $j = 1 \div k$. Смысл критерия похожести на центр состоит в том, что нужно найти такое разбиение m объектов на k таксонов, чтобы приведенная выше величина F была минимальной. Выполнение этого условия можно достичь с помощью алгоритма FOREL [53, 82]. Опишем базовую версию и некоторые модификации этого алгоритма.

2.1. Алгоритм FOREL. Таксоны, получаемые этим алгоритмом, имеют сферическую форму. Количество таксонов зависит от радиуса сфер: чем меньше радиус, тем больше получается таксонов. Вначале признаки объектов нормируются так, чтобы значения всех признаков находились в диапазоне от нуля до единицы. Затем строится гиперсфера минимального радиуса R_0, которая охватывает все m точек. Если бы нам был нужен один таксон, то он был бы представлен именно этой начальной сферой. Но такое огрубление экспериментального материала нас обычно не устраивает, и мы пытаемся получить большее количество таксонов.

Для этого мы постепенно уменьшаем радиус сфер. Берем радиус $0,9R_0$ и помещаем центр сферы в любую из имеющихся точек. Находим точки, расстояние до которых меньше радиуса, и вычисляем координаты центра тяжести этих «внутренних» точек. Переносим центр сферы в этот центр тяжести и снова нахо-

дим внутренние точки. Сфера как бы плывет в сторону локального сгущения точек. Такая процедура определения внутренних точек и переноса центра сферы продолжается до тех пор, пока сфера не остановится, т. е. пока на очередном шаге мы не обнаружим, что состав внутренних точек, а следовательно и их центр тяжести, не меняется. Это значит, что сфера остановилась в области локального максимума плотности точек в признаковом пространстве.

Точки, оказавшиеся внутри остановившейся сферы, мы объявляем принадлежащими таксону номер 1 и исключаем их из дальнейшего рассмотрения. Для оставшихся точек описанная выше процедура повторяется до тех пор, пока все точки не окажутся включенными в таксоны. Доказана сходимость алгоритма за конечное число шагов, однако легко видеть, что решение может быть не единственным. Так, на рис. 5 видно, что результат таксономии зависит от того, с какой первой точки был начат процесс.

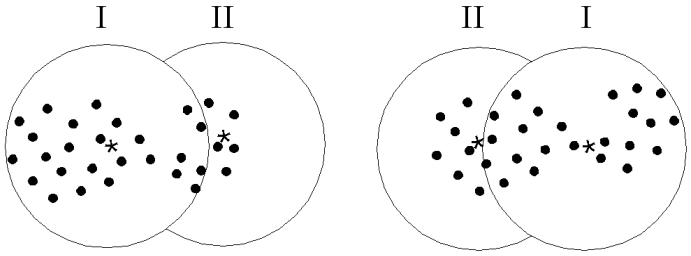


Рис. 5

Если начальную точку менять случайным образом, то может получиться несколько разных вариантов таксономии, и тогда нужно останавливаться на таком варианте, который соответствует минимальному значению величины F.

2.2. Алгоритм FOREL–2. Эта модификация описанного выше базового алгоритма предназначена для получения таксономии с точно заданным числом таксонов k. Здесь радиус сферы по мере надобности увеличивается или уменьшается на величину R, которая на каждой очередной итерации становится все меньше, например уменьшается вдвое. Этот метод последовательных приближений позволяет быстро подойти к заданному числу таксонов при минимально возможном радиусе сфер. Функционал качества

таксономии F в этом алгоритме выглядит следующим образом:

$$F = f(k_i)\sum_{j=1}^{k}\rho_j, \quad \text{где} \quad f(k_i) = \begin{cases} 1, & \text{если } k_i = k, \\ \infty, & \text{если } k_i \neq k. \end{cases}$$

Наилучшему варианту таксономии соответствует минимальное значение F.

2.3. Алгоритм SKAT. Если при многократном случайном выборе начальной точки получается большое число неодинаковых таксономий или если таксоны сильно отличаются друг от друга по количеству своих точек, то это может означать, что наш материал наряду с несколькими локальными сгустками точек содержит еще и одиночные точки или небольшие их скопления, случайно разбросанные в пространстве между сгустками. Создается ощущение того, что имеется несколько «самостоятельных» таксонов и ряд случайно образовавшихся, «несамостоятельных» таксонов, которые было бы целесообразно присоединить к ближайшим самостоятельным.

Каждый очередной таксон находился нами в условиях, когда точки, попавшие в предыдущие таксоны, исключались из рассмотрения. А что происходит, если таксоны формируются в присутствии всех m точек? Может случиться, что некоторые из более поздних таксонов включат в свой состав точки, ранее вошедшие в другие таксоны, и не останутся на месте, а станут скатываться в сторону соседнего сгустка точек и сольются с одним из своих предшественников. Такая ситуация изображена на рис. 6: таксон 2 начнет смещаться и сольется с таксоном 1. Другие же таксоны останутся на прежнем месте и с прежним составом своих внутренних точек. Будем считать таксоны 1, 3 и 4 устойчивыми, самостоятельными, а таксон 2 — неустойчивым, случайным. Случайные таксоны могут появляться из-за помех в данных или из-за неудачного выбора радиуса сфер.

Проверку на устойчивость таксономии можно было бы делать строгими статистическими методами. Однако они разработаны для случаев, когда речь идет о простых распределениях (обычно нормальных) в пространстве малой размерности. Анализ данных же часто имеет дело с относительно небольшим числом объектов (прецедентов) в пространстве большой размерности, и говорить о каком бы то ни было распределении не возможно. Поэтому приходится применять единственно возможные в такой ситуации и, как кажется, достаточно разумные эвристические приемы.

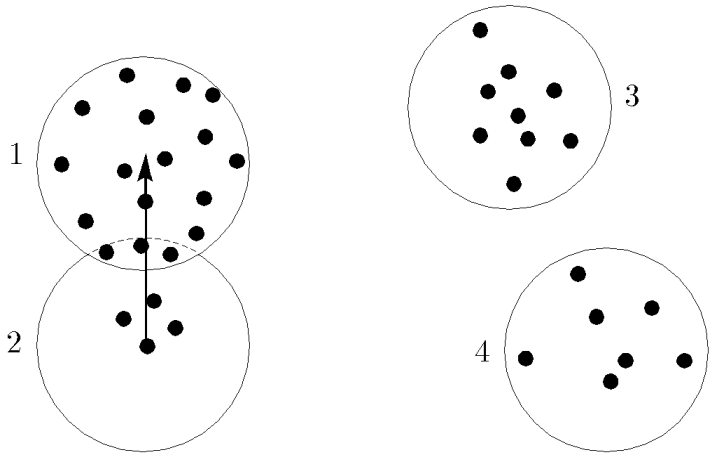


Рис. 6

Один из таких приемов реализован в алгоритме SKAT. На вход программы подается множество m объектов и результаты его таксономии S с помощью алгоритма FOREL при радиусе сферы, равном R. Процедуры таксономии повторяются с таким же радиусом сфер, но теперь в качестве начальных точек выбираются центры, полученные в таксономии S, и формирование каждого нового таксона делается с участием всех m точек. В результате обнаруживаются неустойчивые таксоны, которые скатываются к таксонам-предшественникам. Решение выдается в виде перечня устойчивых таксонов и указания тех неустойчивых, которые к ним тяготеют. Если мы хотим ограничиться только устойчивыми таксонами, тогда мы должны стремиться к такому варианту таксономии, при котором количество точек в устойчивых таксонах было бы максимальным, т. е. максимизировать функционал

$$F = \sum_{j=1}^{k} m_j f(j), \quad \text{где} \quad f(j) = \begin{cases} 1, & \text{если } j\text{-й таксон устойчив,} \\ 0, & \text{если } j\text{-й таксон неустойчив.} \end{cases}$$

2.4. Алгоритм KOLAPS. Взглянув на рис. 7, мы скорее всего отметим следующие его особенности: здесь выделяются три разных по диаметру сгустка точек на равномерном сером фоне. Хотелось бы, чтобы алгоритм таксономии мог выделить эти три сгустка, каждый со своим диаметром, отделив их от точек фона,

т. е. мог бы решать задачу выделения ярких созвездий на звездном небе. Для решения таких задач предназначен один из алгоритмов семейства FOREL — алгоритм KOLAPS*). Его можно разделить на два этапа.

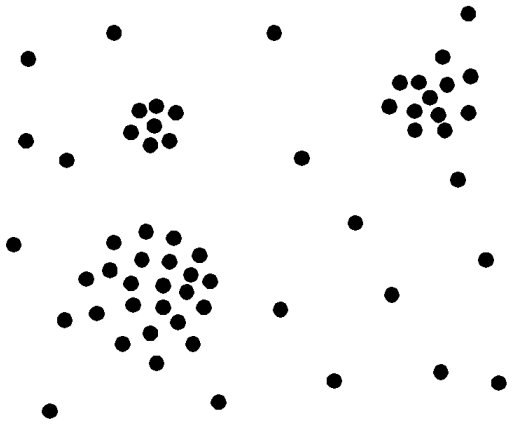


Рис. 7

На первом этапе ищутся потенциальные центры будущих таксонов, а на втором — делается проверка, действительно ли выбранная точка является центром устойчивого таксона.

В начале первого этапа сфера достаточно большого радиуса $R < R_0$ помещается в любую точку множества и смещается, как и в алгоритме FOREL, в центр локального сгустка точек. Количество m_j внутренних точек полученного таксона служит мерой локальной плотности точек в данном месте признакового пространства. Если m_j больше некоторого порога d, то центр такого мощного таксона заносится в список претендентов на роль центра таксона-созвездия, а попавшие в него внутренние точки из дальнейшего рассмотрения исключаются. Если m_j меньше d, то список претендентов не меняется, но внутренние точки этого таксона также гасятся. Затем центр сферы помещается в любую из оставшихся точек и процесс выделения следующих таксонов продолжается до исчерпания всех точек.

После этого восстанавливается все множество m точек и выделяется в списке претендентов таксон с наибольшим значением локальной плотности m_j. В центр этого таксона помещается

*) Алгоритм предложен Е. А. Чиркиным.

сфера, и ее радиус начинает сжиматься от величины R до величины $R_{\min}$. На каждом i-м шаге сжатия определяется число точек, оставшихся внутренними. Если начальный радиус был слишком большим для данного таксона (т. е. если он захватывал много разреженного пространства), то в начале процесса сжатия скорость убывания числа внутренних точек будет небольшой. По мере вхождения сферы в более плотную часть таксона количество теряемых точек начнет увеличиваться, что служит сигналом к остановке сжатия. В результате находится наиболее естественный для данного таксона радиус сферы и фиксируется число его внутренних точек m'_j. Та же процедура постепенного сжатия повторяется и для других таксонов из списка претендентов, упорядоченных по характеристике локальной плотности. В результате выбирается k таких таксонов, в состав которых после сжатия попадает наибольшее количество точек. Это условие равнозначно максимизации функционала $F = \sum\limits_{j=1}^{k} m'_i$.

2.5. Алгоритм BIGFOR. Иногда встречаются такие большие таблицы данных, которые не умещаются целиком в оперативную память и должны обрабатываться по частям. Для таксономии таких массивов предназначен итеративный алгоритм BIGFOR.

Вначале в оперативную память вводится массив из V объектов ($V \ll m$), который с помощью алгоритма FOREL–2 делится на k' таксонов. Описание таксона состоит из координат его центра и количества m'_j его внутренних точек («веса» центра). Это краткое описание запоминается в отведенном месте памяти. Затем в память вводится следующая порция данных, с которой проделывается та же процедура. После повторения этих шагов t раз, где $t = m/V$, получается массив из $q = tk'$ точек, представляющих собой центры таксонов, возникавших на каждом шаге. Группировка этих точек в k таксонов ($k < k'$) делается с помощью варианта алгоритма FOREL–2, который при вычислении координат центра тяжести внутренних точек учитывает вес этих точек. После нахождения центров k таксонов весь массив из m точек перераспределяется между ними. При поочередном просмотре каждая точка относится к тому таксону j, расстояние до центра которого оказывается минимальным.

Если число исходных объектов m слишком велико, процедура укрупнения таксонов может быть не двух-, а многоступенчатой,

скатывание таксонов во все более крупные «шарики» делается несколько раз. Критерий качества таксономии, получаемой алгоритмом BIGFOR, тот же, что и у алгоритма FOREL.

2.6. Иерархическая таксономия. Можно считать, что исходное множество m точек представляет собой результат таксономии с радиусом $R = 0$, при котором получается m таксонов, каждый из которых содержит по одной точке. Применим к ним алгоритм BIGFOR, исключив из него описанную выше процедуру перераспределения точек. На первом шаге алгоритма BIGFOR выбирается небольшой радиус $R > 0$, дающий $k_1 < m$ таксонов первого уровня. На втором шаге происходит слияние некоторых близких друг к другу мелких таксонов в более крупные, в результате чего появляются k_2 таксонов второго уровня ($k_2 < k_1$). Если эти шаги продолжать, то на некотором шаге p будет получена таксономия, объединяющая все точки в один-единственный таксон.

Отношения между таксонами разных уровней можно представить себе в виде иерархической структуры или дерева, состоящего из m объектов на нулевом уровне (уровне листьев) и k_i таксонов ($i = 1, \ldots, p$) на каждом из p уровней. Корневая (p-я) вершина этого дерева содержит $k_p = 1$ таксон со всеми m объектами. Из нее выходит $(k_p - 1)$ ребер, соединяющих корневую вершину с центрами таксонов $(p-1)$-го уровня. Такие связи между таксонами прослеживаются вплоть до m ребер, которые соединяют m точек нулевого уровня с k_1 центрами самых мелких таксонов первого уровня (см. рис. 8).

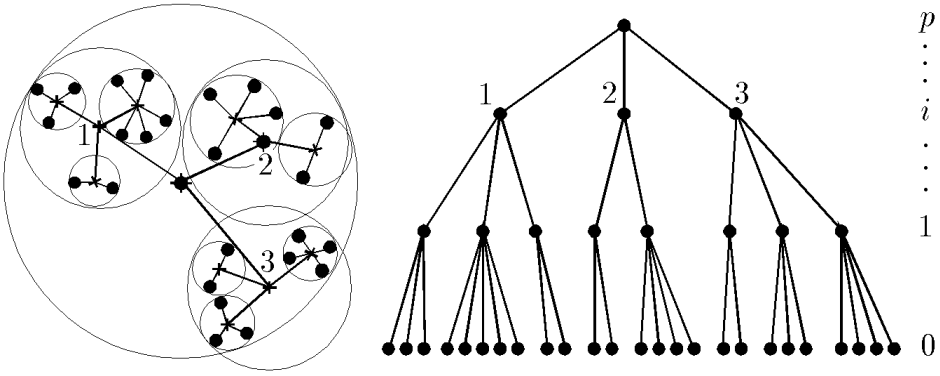


Рис. 8

Так выглядит иерархическая таксономия, полученная методом скатывания мелких таксонов во все более крупные, который называют *методом агломерации*. Такое «генеалогическое» дерево позволяет видеть связи разных объектов и их групп друг с другом, что иногда бывает полезно при содержательном анализе массива данных. Иерархическое дерево может быть получено и другим путем — путем дробления крупных таксонов на более мелкие. Процедура такой таксономии практически совпадает с алгоритмом FOREL. Вначале определяется радиус R наименьшей сферы, описывающей все m точек. Эта сфера есть таксон верхнего корневого (p-го) уровня. Затем радиус уменьшают и находят k_{p-1} таксон следующего $(p-1)$-го уровня. Затем для объектов каждого из полученных таксонов процедура таксономии повторяется при еще меньшем радиусе. Некоторые из этих таксонов распадутся на несколько более мелких таксонов $(p-2)$-го уровня. Такой процесс дробления таксонов продолжается до тех пор, пока на некотором шаге не окажется, что количество таксонов стало равным числу объектов m. В результате получится дерево такого же вида, что и при агломерации.

§ 3. Динамичная таксономия

Иногда возникает ситуация, когда таксономию нужно делать не на одновременно заданном множестве объектов, а на объектах, возникающих по одному или небольшими группами в ходе исследуемого процесса. При этом таксономия может меняться после появлении каждого нового объекта и окончательный вид она приобретает тогда, когда обработан последний из исследуемых объектов. Таким образом, по ходу процесса происходит свертка информации: вместо характеристик отдельных объектов нужно хранить лишь краткое описание полученных к данному моменту времени таксонов с указанием числа объектов, включенных в каждый таксон.

3.1. Алгоритм DINA. Для решения такой задачи в алгоритме DINA задается радиус сферы R и первая из появившихся точек объявляется центром первого таксона. При появлении второй точки происходит проверка, не попадает ли она в первый таксон. Для этого вычисляется расстояние от нее до центра первого таксона, и если это расстояние меньше R, то вторая точка включается в состав первого таксона, а центр этого таксона, как

в алгоритме FOREL, смещается в центр тяжести двух своих внутренних точек. Если же вторая точка отстоит от центра первого таксона на расстояние, большее R, то она объявляется центром нового, второго таксона. При появлении каждой очередной точки эти процедуры повторяются и часть этих точек попадает в имеющиеся таксоны, а другие становятся зародышами новых таксонов.

Можно следить за тем, чтобы количество объектов в таксонах было по возможности одинаковым. Если, например, обнаружится, что какой-нибудь таксон включил в себя слишком большое число точек по сравнению с другими таксонами, то его можно разделить на два таксона половинной мощности, расположив центры этих новых таксонов на диаметрально противоположных сторонах сферы родительского таксона.

Переход от описания исходных объектов к описанию их таксонов эквивалентен переходу от данных к знаниям. Если таксономия имеет иерархический характер, то она отображает структуру нашего знания об изучаемом процессе или явлении. Частные знания нижнего уровня объединяются на следующем уровне в знания более высокого уровня или метазнания. В работах В. П. Гладуна и его коллег [27, 38, 162] описываются алгоритмы динамического построения иерархии понятий («растущие пирамидальные сети») в процессе накопления новых фактов об изучаемом явлении. В этих алгоритмах новые таксоны могут возникать не только при поступлении нового оригинального объекта, но также и при появлении таксона с чрезмерно большим количеством объектов. Такое перегруженное содержанием знание как бы детализируется, разделяется на более мелкие составляющие понятия. Применение данного подхода показало свою высокую эффективность в системах искусственного интеллекта.

3.2. Алгоритм SETTIP. Алгоритм SETTIP [82] предназначен для анализа динамических рядов. Представим себе, что ведется наблюдение за n характеристиками процесса в разные последовательные моменты времени от $t = 1$ до $t = m$. Протокол наблюдений можно записать в виде таблицы «время-свойство» размером $n \times m$. Если m строк этой таблицы разделить на k таксонов и номер таксона, которому принадлежит процесс в данный момент времени, записать в качестве $(n + 1)$-го признака, то такое расширенное описание поведения процесса позволяет обнаруживать некоторые интересные динамические закономерности.

В частности, можно увидеть наличие неслучайных после-

довательностей из двух или большего числа номеров таксонов. Встречаются промежутки времени, когда номер таксона остается неизменным, что говорит о стационарности процесса на этом промежутке времени. Можно наблюдать периодически повторяющиеся во времени номера таксонов или связки этих номеров, что говорит об одинаковых стадиях протекания процесса в разные промежутки времени. Если номера таксонов считать символами из конечного алфавита, то каждый динамический процесс представляет собой некоторый текст на языке этих символов. Пользуясь мерами редакционного расстояния между текстами, можно находить меру близости, похожести двух текстов и использовать ее для таксономии текстов или в данном случае для выявления группы процессов с одинаковой динамикой протекания.

§ 4. Таксономия с суперцелью

В качестве синонима слова таксономия иногда употребляются термины «обучение без учителя», «самообучение». Этим отражается тот факт, что при решении задачи таксономии задается критерий качества (максимальные связи внутри таксона, минимальные связи между таксонами и т. д.), но не указывается, для какой конкретной цели эта группировка делается. Т. е. не указывается, что будет описываться в терминах полученных таксонов, что от чего нужно будет отличать по этим описаниям и т. д.

Но такая таксономия «на все случаи жизни» для каждого конкретного случая может оказаться неоптимальной. Например, если при распознавании устных слов мы пользуемся не отдельными фонемами, а их группами (звукотипами) и при этом объединим в один таксон очень похожие по своим характеристикам звуки «п», «т» и «к», то такая таксономия окажется неприемлемой, если в составе распознаваемого словаря имеются слова типа «кот», «ток», «кто» и т. д. В этом примере показано место таксономии в многоуровневой системе. Таксоны на уровне звукотипов должны строиться с учетом «суперцели»: помимо того, что таксоны должны объединять похожие элементы, количество таксонов еще должно быть минимальным и достаточным для принятия правильных решений на следующем более высоком уровне системы.

Этими условиями четко фиксируется цель таксономии, вследствие чего исключается возможность для использования терминов самообучение, обучение без учителя и т. п. Все, что требуется

от учителя — конечная цель таксономии, методы ее достижения и критерии для оценки получаемых результатов — здесь заданы однозначно. Таксономия с учетом суперцели может быть получена алгоритмом ROST.

4.1. Алгоритм ROST. Каждый объект того уровня, на котором делается таксономия, представлен описанием (точкой) в n-мерном пространстве своих свойств. В нашем примере это фонемы в пространстве спектральных характеристик. Алгоритм ROST действует вначале так же, как и алгоритм FOREL с малым радиусом гиперсферы. Затем радиус увеличивается и после каждого шага делается проверка на соответствие суперцели: не возникают ли ошибки из-за этого укрупнения звукотипов. Если нет, то процесс продолжается, а если да, то точки, попавшие в таксон, приводящий к ошибкам, подвергаются повторной таксономии с меньшим радиусом и из дальнейшего рассмотрения исключаются. Укрупнение звукотипов продолжается до тех пор, пока из процесса не будут исключены все точки.

§ 5. Таксономия в анизотропном пространстве [71]

До сих пор мы работали при вполне естественном предположении, что пространство признаков изотропно, так что расстояние $r(pq)$ между точками p и q не зависит от того, идем ли мы от точки p к точке q или от q к p. Однако если близость, похожесть точек измерять, например, затратами сил на преодоление пути между ними, то направление движения (например, идти в гору или под гору) часто бывает не безразлично.

Такая ситуация возникает в известной задаче выбора оптимального ряда типоразмеров изделий или машин [15]. Так, перевозка груза в 2,5 тонны на 5-тонном грузовике вызывает потери от полухолостых пробегов. Если же на эту машину погрузить 7,5 тонн, то она может выйти из строя и потери будут гораздо бóльшими, чем в предыдущем случае. Для того чтобы учесть этот эффект, при таксономии в анизотропном пространстве направление анализа фиксируется правилом, по которому расстояния измеряются всегда от центра таксона к его внутренним точкам. При этом относительная значимость направления вдоль координаты j отмечается весовыми коэффициентами v_j^+, если направление измерения совпадает с направлением увеличения j-й координаты, и v_j^- — в противоположном случае.

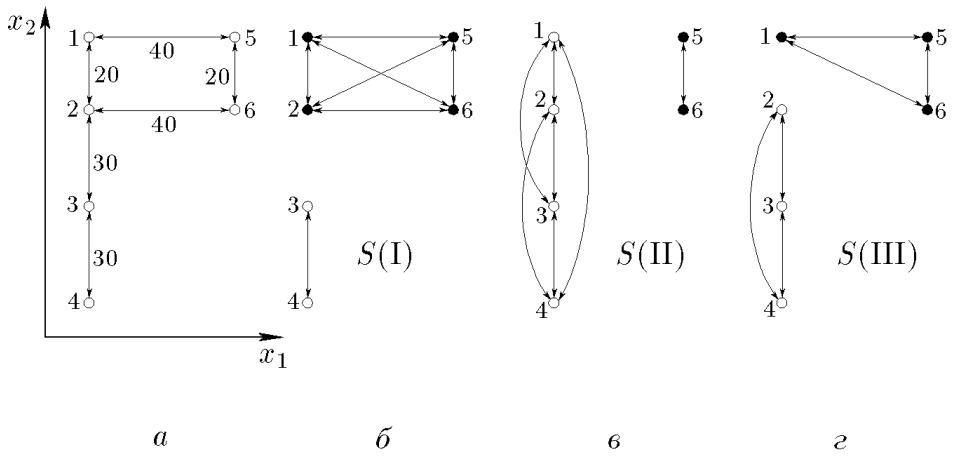


Рис. 9

Предположим, что мы хотели бы обеспечить возможность замены любого объекта таксона на любой другой объект этого же таксона. При этом потери от всех замен будут равны сумме потерь от каждой отдельной замены. Потери от взаимных замен объектов a_i и a_v равны сумме расстояний между ними в двух противоположных направлениях: $R(i,v) = r(a_i, a_v) + r(a_v, a_i)$. Тогда потери от всех парных замен внутри j-го таксона, состоящего из m точек, выражаются величиной $R_j = \sum R(i,v)$ для всех i и v от 1 до m. В результате качество таксономии может быть оценено через сумму потерь для всех k таксонов: $R = \sum R_j$, $j = 1 \div k$. Чем меньше R, тем лучше таксономия.

На рис. 9, *а* приведен пример множества точек в двумерном пространстве, подвергаемого таксономии с помощью алгоритма семейства FOREL. Если принять, что $v_j^+ = v_j^- = 1$, то пространство будет изотропным и таксономия S(I), приведенная на рис 9, *б*, представляется вполне естественной. Если же считать, что свойства пространства вдоль оси x_1 анизотропны и при этом $v_1^+ = 1$, $v_1^- = 5$, $v_2^+ = v_2^- = 1$, то получаем таксономию S(II), показанную на рис. 9, *в*. Если имеет место одновременная анизотропия по обеим координатам и такая, что $v_1^+ = 1$, $v_1^- = 5$, $v_2^+ = 3$, $v_2^- = 1$, то таксономия имеет вид S(III), показанный на рис. 9, *г*.

§ 6. Сравнение алгоритмов таксономии

Помимо описанных в этой главе существует большое количество других алгоритмов таксономии (см., например, [12, 49, 95, 166]). Естественно, возникает потребность в их сравнении и выборе алгоритма, в некотором смысле лучшего, чем другие. Алгоритмы можно сравнивать по требуемым машинным ресурсам (памяти и времени), по применимости к трудным случаям (большие массивы данных, разнотипные признаки, наличие в данных помех и пробелов и т. д.).

Однако главное, что интересует пользователя — качество получаемых решений. Чтобы сформулировать критерий качества, по которому можно было бы сравнивать алгоритмы таксономии, напомним, что таксономия обычно делается не только для компактной перекодировки множества m объектов в k таксонов. Эти таксоны или их типичные представители (прецеденты) используются в дальнейшем как для краткого описания имеющихся объектов, так и (что более важно) для распознавания новых объектов генеральной совокупности. Каждый новый объект относится к тому таксону (образу), присоединение к которому наилучшим образом удовлетворяет критерию качества таксономии.

Пусть множество m объектов представляет собой некоторую выборку из генеральной совокупности, состоящей из M объектов ($m < M$), и на множестве m тем или иным алгоритмом F сделана таксономия на k таксонов. Если теперь предъявлять по очереди все остальные $(M-m)$ объектов и присоединять их к соответствующим таксонам, то в итоге будет получен вариант S' таксономии генеральной совокупности M. Если бы мы тем же алгоритмом F сделали таксономию сразу всей совокупности M объектов, то получили бы вариант таксономии S. Будем называть таксономии S и S' *базовой* и *выборочной* соответственно.

Возникает вопрос: будет ли таксономия S' совпадать с таксономией S? Если да, то алгоритм F удачно угадал структуру множества M по случайной выборке m. Эта способность по малой выборке правильно угадывать структурные закономерности генеральной совокупности и есть, по-видимому, основная характеристика (Q) качества алгоритмов таксономии. Сравнивать различные алгоритмы таксономии по этому качеству можно с помощью программного испытательного полигона «Таксон» [84]. Прежде, чем описать его работу, введем некоторые обозначения.

Если объекты p и q принадлежат разным таксонам, то счи-

таем, что таксономическое расстояние $r(p,q)$ между ними равно единице, а если они из одного таксона, то $r(p,q)=0$. Представим себе квадратную бинарную матрицу размером $M \times M$, столбцы и строки которой соответствуют объектам генеральной совокупности, а на пересечении p-й строки с q-м столбцом стоит значение $r(p,q)$. Все диагональные элементы такой матрицы равны нулю, так как $r(p,p)=r(q,q)=0$, и матрица симметрична, так как $r(p,q)=r(q,p)$. Тогда различие $R(S,S')$ между двумя таксономиями S и S' можно определить, наложив одну на другую матрицы, соответствующие этим таксономиям, и суммируя число элементов с несовпадающими значениями $r(p,q)$. Если полученную сумму разделить на максимально возможное число несовпадений (M^2-M), то получим хеммингово расстояние между матрицами, нормированное в диапазоне от нуля до единицы:

$$R(S,S')=\sum_{p,q=1}^{M}\{r(p,q)-r'(p,q)\}/(M^2-M),$$

где $r(p,q)$ и $r'(p,q)$ — таксономические расстояния для объектов из таксономий S и S' соответственно. Чем меньше величина различия $R(S,S')$, тем лучше испытуемый алгоритм таксономии угадал структуру генеральной совокупности.

В полигоне «Таксон» можно менять объем генеральной совокупности M, объем случайной выборки m и размерность пространства признаков n. Множество m представляет собой либо данные некоторой реальной задачи, либо данные, порождаемые генератором с заданным законом распределения. В последнем случае можно задавать число таксонов k и расстояния между таксонами d. Испытуемому алгоритму F_i количество таксонов k либо задается, либо он должен сам выбрать наилучшее число таксонов с точки зрения своего критерия качества F. В процессе испытания объем выборки m меняется от малых долей M до полного объема M. При каждом значении m выборка формируется случайным образом, и эксперимент с m объектами повторяется t раз $(t \geq 0)$. Получаемые при этом величины $R(S,S')$ усредняются. По итогам экспериментов фиксируется доля M, при которой базовая и выборочная таксономии перестают отличаться друг от друга. Тот алгоритм считается лучшим, который достигает этого эффекта при минимальном значении m.

В ходе испытания алгоритмов таксономии выяснилось, что их способность угадывать структуру генеральной совокупности

практически не зависит от размерности признакового пространства. Можно сказать, что по этому свойству машина существенно превосходит человека, который успешно решает задачи таксономии лишь тогда, когда есть возможность непосредственно видеть разделяемое множество, т. е. не более чем в трехмерном пространстве. При большем числе признаков он переходит к примитивному способу деления по каждому признаку в отдельности (например, *от* и *до* по возрасту и образованию), разрезая многомерное пространство на гиперпараллелепипеды.

Из рассмотренных в данной книге алгоритмов таксономии при сравнении на полигоне лучшим, как и ожидалось, оказался алгоритм KRAB (см. главу 10), на втором месте был SKAT и на третьем FOREL. Однако следует учитывать, что алгоритм KRAB гораздо более трудоемкий, чем его конкуренты. Кроме того, он выдает результат в виде таксонов произвольной формы и сложности, что требует больших затрат памяти на их описание и затрудняет понимание результата человеком. В итоге приходится сложные таксоны описывать набором более простых форм, например набором гиперсфер или гиперкубов. Алгоритм же FOREL дает быстрые и простые решения. Так что в практическом использовании всем этим алгоритмам находится свое место.

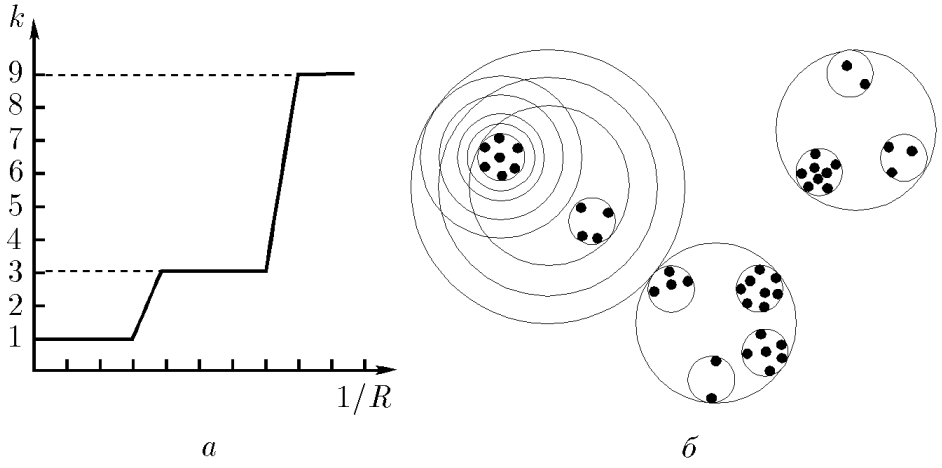
§ 7. Выбор числа таксонов

Иногда встречаются случаи, когда заказчик (владелец данных) точно знает, сколько таксонов он хотел бы получить. Чаще заказчик высказывает менее строгое пожелание: хотелось бы в диапазоне от k_1 до k_2. В этих случаях алгоритм должен автоматически выбрать наиболее «естественное» значение k в этом диапазоне. Обычно же заказчик на вопрос о числе таксонов отвечает: «Не знаю. Сколько получится. А что бы вы посоветовали?»

Совет зависит от назначения таксономии. Если таксоны служат для дальнейшего машинного использования, то можно выбирать большие значения k, сообразуясь только с имеющимся объемом памяти для их хранения. Если же таксонами будет пользоваться человек «вручную», то в диалоге с заказчиком обычно выясняется, что двух или трех таксонов не достаточно. Это слишком грубая классификация, приводящая к неприемлемым потерям информации об индивидуальных особенностях объектов, входящих в большие таксоны. Десять или больше таксонов тоже не

устраивают заказчика: такое большое число таксонов трудно запомнить и потому использовать их для объяснения структуры изучаемой системы неудобно. Договориться чаще всего удается на количестве таксонов от пяти до девяти. Это хорошо согласуется с наблюдением американского психолога Г. Миллера [125] о предпочтительности для человека оперировать количеством элементов, равным 7 ± 2, что объясняется ограниченностью объема оперативной памяти человека.

Описанные выше алгоритмы таксономии имеют некоторые средства для выбора наиболее предпочтительного числа таксонов в заданном диапазоне. В алгоритмах класса FOREL при постепенном уменьшении радиуса сферы R на графике зависимости числа таксонов от радиуса нередко можно наблюдать так называемый эффект «полочки» (см. рис. 10, *а*). Обнаруживается несколько соседних значений радиуса, при которых количество таксонов не меняется, а затем на следующем шаге начинает резко увеличиваться.



а *б*

Рис. 10

Природу этого явления можно пояснить с помощью рис. 10, *б*. На некотором шаге выделяется три таксона, и это число таксонов сохраняется вплоть до шага, на котором сфера перестает умещать в себя все точки этих таксонов. Затем число таксонов хаотически возрастает и снова стабилизируется на количестве таксонов, равном 9. Эти числа — 3 и 9 — хорошо соответствуют иерархической структуре анализируемого множества точек.

§ 8. Примеры решения практических задач

Описанные методы таксономии реализованы в виде блока программ «Таксономия» в пакете прикладных программ для обработки таблиц экспериментальных данных (ППП ОТЭКС), который был передан для использования более чем в 100 организаций бывшего СССР. За 30 лет со времени появления первых наших алгоритмов таксономии накоплен большой опыт их применения и у нас. Приведем некоторые примеры решения прикладных задач.

8.1. Задачи палеонтологии и геологической разведки. Алгоритм FOREL был разработан в 1967 году [53], и первой задачей для его опробования оказалась задача из области палеонтологии. Некоторые животные, жившие в разные геологические эпохи, имели твердый хитиновый покров, который хорошо отпечатался в древних породах. Палеонтологи находят такие отпечатки и изучают их, определяя вид, семейство и род бывших носителей этих панцирей. По таким следам прошлого определяется геологическое время возникновения того или иного слоя земной коры. Одними из таких ископаемых существ являются трилобиты (предки современных тараканов).

Нам была предъявлена таблица с описанием 30 характеристик 250 трилобитов. Среди признаков были размеры хитинового покрова, число бороздок на головной части и другие. Алгоритм FOREL при разных радиусах R сфер выдавал разное число таксонов. При некотором значении R было получено такое же число таксонов, которое было ранее установлено палеонтологами при ручной классификации этой коллекции трилобитов. Было построено таксономическое иерархическое дерево, и при этом к нашему общему удивлению состав таксонов в точности соответствовал составу ручных классов. Особенно порадовал палеонтологов тот факт, что один трилобит даже при очень больших радиусах R не хотел присоединяться ни к одному таксону. Оказалось, что он является представителем уникального вида из совсем другого семейства, и это удивительно и замечательно, что машина догадалась об этом! Такой успех произвел на наших коллег палеонтологов настолько сильное впечатление, что на одном из семинаров нам был задан вопрос: «А не может ли машина определить, каким латинским термином назван тот или иной вид трилобитов?». Мы были вынуждены сказать, что без подсказки — вряд ли.

Другой интересной задачей из геологической области была задача таксономии территорий северо-востока Чукотки [55]. Изучаемая местность была разделена геологами на 1992 ячейки в виде квадратов 10×10 км. Каждый квадрат был описан 45 двоичными признаками, отражающими наличие или отсутствие различных геологических свойств: шлиховые ореолы ртути, глубина залегания мезозоид 3–4 км, геосинклинальные прогибы и пр.

Таксономия этих участков делалась разновидностью алгоритма FOREL (FOREL–5), предназначенной для таксономии двоичных данных. Из разных вариантов таксономии заказчики выбрали вариант, содержавший 318 таксонов. Этот вариант привлек их внимание тем, что 46 из 318 таксонов включали в свой состав участки, которые были раньше хорошо изучены и на которых имелись золотоносные месторождения. И если планировать экспедиционные работы по поиску золота, то в первую очередь целесообразно обратить внимание на те неизученные участки, которые также оказались в составе этих «золотых» таксонов. Результаты экспедиций подтвердили высокую эффективность такого способа планирования геологоразведки. Таким же образом были разработаны рекомендации по поиску месторождений и ряда других минералов.

8.2. Задачи социологии и экономики [56, 86]. При разработке планов экспедиционных работ, связанных с изучением социальных проблем сельского населения Алтайского края, социологам нужно было выбрать для исследования k населенных пунктов, причем таких, которые представляли бы по возможности разные типы сел и деревень края. С этой целью было подготовлено описание всех сельских населенных пунктов края характеристиками типа численность населения, количество школ, клубов, характер водоснабжения, наличие электричества, дорог с твердым покрытием. С помощью алгоритма FOREL множество из нескольких сотен сел было разделено на заданное количество таксонов k, из которых были выбраны типичные представители каждого таксона. Таким путем гарантировалось, что выбранные k сел достаточно хорошо представляют всё разнообразие сел края, что не будут потеряны из виду какие-то типы сел и не будут тратиться средства на изучение сел-близнецов.

После завершения экспедиционных работ социологи привезли большой материал в виде анкет с ответами на разные вопросы. Для обработки этих данных также использовались алгоритмы

таксономии. В § 1 данной главы была описана задача таксономического анализа такого рода материалов, касающихся выявления причин миграции сельского населения в города. Еще одна задача, связанная с проблемами миграции населения, решалась на материалах с описанием всех областей, краев и автономных республик Российской Федерации. Были выделены таксоны, в состав которых входили административные единицы с положительным, нулевым или отрицательным балансом миграции населения. Анализ характеристик этих таксонов позволил понять относительную значимость отдельных факторов на процессы миграции. Так, было обнаружено, что уровень заработной платы по своему значению существенно уступает уровню обеспечения населения государственным жильем. И если решать, куда направлять денежные ресурсы, то в первую очередь следовало обращать внимание на жилищное строительство.

Широкое применение нашли методы таксономии в задачах анализа статистических данных экономического характера. Одна из них была связана с получением объективной оценки деятельности отдельных предприятий или административных образований (совхозов, шахт, заводов, сельских районов, областей). Применительно к совхозам Новосибирской области (их было около 1000) анализировались данные о посевных площадях, энерговооруженности, наличии шоссейных и железных дорог, числе работающих и т. д. По этим данным группировались совхозы с приблизительно одинаковыми объективными характеристиками. Затем сравнивались результаты деятельности совхозов из одного и того же таксона (сбор зерновых, надои молока). Выяснялось, например, что совхоз, которому ежегодно присуждались красные знамена, по своим объективным данным должен был бы работать гораздо эффективнее. А некоторый совхоз-середнячок дает гораздо больше продукции, чем все другие совхозы с аналогичными исходными ресурсами. Так что отмечать наградами следовало бы не того, кого обычно отмечали по валовым показателям.

Интересно, что областное начальство высоко оценило эту методику, но применять ее при подведении итогов года так и не стало. Объективные оценки лишали тех, кто принимает решение «казнить или миловать», возможности наказывать строптивых тружеников и поощрять верноподданных, пусть даже и не очень хорошо работающих. Сейчас эта методика в своем первоначальном виде потеряла актуальность. Но она может оказаться полезной для самооценки отдельных предприятий или компаний.

Если в результате анализа данных о себе и о конкурентах окажется, что по некоторым выходным характеристикам компания начинает проигрывать соревнование своим конкурентам, то такой ранний сигнал позволит своевременно принять необходимые корректирующие меры.

8.3. Задачи биологии. При выведении новых видов растений или пород животных селекционеры стремятся выбирать для скрещивания виды или породы, наиболее не похожие друг на друга, избегая скрещивания «близнецов». С этой целью описание свойств потенциальных родителей подвергается таксономии и для скрещивания отбираются особи, принадлежащие разным таксонам. Данная задача по своему характеру похожа на одну из вышеописанных задач социологии.

Биофизики изучают влияние различного рода воздействий на живые организмы разных видов. Первая серия экспериментов проводилась на большом числе видов. В результате протокол наблюдений представлял собой таблицу из более 20-ти видов, каждый из которых был описан двумя группами характеристик: 8 — характеристиками воздействия и 14 — характеристиками реакции организма. При каждом новом сочетании значений воздействующих факторов наблюдалось новое сочетание реакций организмов и фиксировался протокол в виде новой таблицы «объект-свойство».

Каждая такая таблица подвергалась таксономии, что позволяло автоматически выбирать наилучшее число таксонов k в заданном диапазоне $k_{\min} < k < k_{\max}$. При этом делалась таксономия отдельно по группам характеристик и в полном 22-мерном пространстве. В итоге сравнения таксономий разных таблиц было обнаружено, что имеются виды живых организмов, которые приблизительно одинаково реагируют на одинаковые внешние воздействия и попадают в один таксон. По одному типичному представителю таких устойчивых таксонов было отобрано для более детальных экспериментов, что позволило существенно ускорить исследования и сократить расходы на эксперименты.

8.4. Задачи океанологии. Данные в одной из задач океанологии представляли собой следующее. В определенной точке мирового океана делался эксперимент по измерению температуры и солености воды на 16 разных глубинах. В протокол записывались 2 координаты точки и еще 32 характеристики (16 температур и 16 соленостей). Всего таких точек в мировом океане было иссле-

довано около 20 000. Так что таксономию нужно было делать на таблице размером 20000×34, и она осуществлялась алгоритмом FOREL.

Авторы данных выбрали один из вариантов таксономии с числом таксонов, равным 15. Когда они покрасили на карте точки каждого таксона в один и тот же цвет, то обнаружились зоны с одинаковыми профилями температур и соленостей. В частности, были хорошо видны известные морские течения (Гольфстрим, Куросио), выявились и другие интересные для океанологов закономерности структуры мирового океана.

8.5. Задачи распознавания речевых сигналов («кодовая книга»). В системах распознавания речи часто используются спектральные характеристики, измеряемые на коротких участках сигнала, следующих друг за другом. Каждый участок отображается в n-мерном пространстве спектральных признаков точкой, а слово можно представить в виде траектории, помеченной этими точками. После накопления обучающего материала пространство признаков может содержать сотни тысяч точек, и естественно было бы хранить в памяти не все точки, а описывающие их таксоны.

Методами таксономии делается таксономия точек на k таксонов и вычисляются все парные расстояния между ними. Такая матрица парных расстояний называется *кодовой книгой*. Каждый участок произносимого слова попадает в окрестности центра того или иного таксона. Если фиксировать номера (коды) этих самых близких таксонов, то слово можно представить последовательностью таких кодов. После обучения в памяти машины появляются эталоны слов в виде кодовых последовательностей.

Для распознавания контрольного слова его кодовая последовательность сравнивается со всеми эталонными последовательностями и выбирается самый похожий эталон. При этом используется динамическое программирование, которое требует знания расстояний от всех кодов эталона до всех кодов распознаваемого слова. Наличие кодовой книги позволяет существенно упростить этот трудоемкий этап. Теперь достаточно лишь указать номера двух кодов и расстояние между ними будет извлечено из кодовой книги.

Первичное векторное описание большого числа участков речи требует чрезмерно больших затрат машинной памяти. В связи с этим для таксономии такого массива данных применяется ал-

горитм типа описанного выше алгоритма DINA. Для формирования кодовых последовательностей может оказаться полезным алгоритм SETTIP.

8.6. Другие области применения. В почвоведении алгоритмы таксономии применяются для классификации типов почв, что особенно важно при разработке кадастра почв для целей приватизации земли.

Анализ погодных таксонов, полученных на массиве трехлетних метеорологических наблюдений в зоне лесов Красноярского края, позволил обнаружить несколько таксонов, куда попали дни, когда в месте наблюдения возникали пожары. Были таксоны с 50 процентами пожарных дней и были дни (т. е. такие сочетания погодных условий), когда пожаров не наблюдалось. Эти данные в сочетании с прогнозом погоды позволяют планировать оптимальное распределение ресурсов противопожарных служб.

Анализ психологических характеристик студентов Пермского университета позволил выделить группы студентов с одинаковыми характеристиками. Такой материал может помочь в оптимальном формировании состава учебных групп, в выборе типичных методик коррекции психологических характеристик студентов и т. д.

§ 9. Некоторые дополнительные замечания о таксономии

Неискушенного пользователя обычно занимает вопрос, существует ли «объективная», «естественная» таксономия или она всегда «субъективна»? Ответ на этот вопрос состоит в том, что в каждой таксономии или классификации имеются элементы как субъективного, так и объективного. Это хорошо иллюстрирует пример из книги М. Бонгарда [20], приведенный на рис. 11. Здесь изображены шесть фигур, которые можно разделить по-разному и на разное число таксонов. Так, если обращать внимание на цвет, то выделится два таксона: светлые и заштрихованные фигуры. Если измерять число углов, то обнаружатся три таксона: фигуры с тремя, четырьмя и бесконечным числом углов. Если смотреть на площадь фигур, то можно выделить и два таксона (большие и малые), и три таксона (большие, средние и малые).

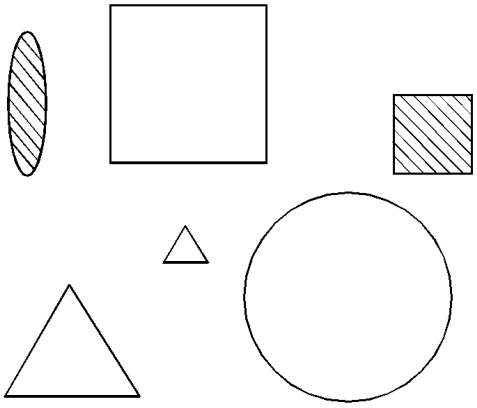


Рис. 11

Отсюда видно, что одной, «самой естественной», «абсолютно объективной», таксономии не существует. Все реальные объекты имеют бесконечное число свойств, и выделение некоторого конечного подмножества этих свойств — акт субъективный. Меры близости, критерии качества также выбираются субъективно. Если известна цель, для достижения которой делается таксономия (т. е. при наличии «суперцели»), то качество таксономии проверяется тем, хорошо ли она способствует достижению этой цели, удобна ли, экономична и т. д. Эта проверка носит объективный характер, но выбор суперцели опять-таки субъективен и для одной суперцели данная таксономия будет хорошей, а для другой — нет.

Иногда можно встретиться с суждением такого рода: «Алгоритм таксономии дал плохой результат: выделился один очень большой таксон, три поменьше и остальные точки рассыпались по единичным таксонам». Не всегда в таком результате повинен алгоритм таксономии. Встречаются данные, которые порождены одним однородным процессом, могут быть описаны нормальным законом распределения, и никакой алгоритм таксономии не разделит такую выборку на 5 или 7 «самостоятельных» таксонов. В таком случае в утешение можно сказать, что таксономия не только позволяет выявить структуру хорошо структурированного множества, но и показать, что некоторое множество гомогенно, оно не расслаивается на изолированные подмножества. Часто именно это и надо было узнать.

Бывают и такие ситуации: «Меня не устраивает такая таксономия. Один таксон получился хороший, в него попали дей-

ствительно объекты одной и той же природы. А в других все перемешано». Да, таксономия не исключает такого результата, причина которого может лежать и в плохом качестве алгоритма, но может отражать и неудачный выбор характеристик, описывающих объекты. Можно обнаружить, что характеристики неинформативны с точки зрения той суперцели, которую интуитивно ставит перед собой пользователь. Так что алгоритмы таксономии могут помочь разобраться в том, достаточно ли информативны имеющиеся признаки. Кстати, если пользователю известна частичная классификация, т. е. если он знает относительно некоторой части объектов, какие должны быть в одном таксоне, а какие обязательно в разных, то эту информацию можно использовать с пользой для дела, например в алгоритме ROST. При одних и тех же свойствах объектов результат таксономии может быть разным, если мы учитываем их относительные веса («важность»). При вычислении расстояния между объектами p и q вклад признака x_j должен быть пропорционален его весовому коэффициенту γ_j, так что евклидово расстояние $r(pq)$ в n-мерном пространстве определяется следующим образом:

$$r(pq) = \sum_{j=1}^{k} \sqrt{\gamma_j (x_{jp} - x_{jq})^2}.$$

Значение весов γ_j можно установить заранее, но иногда задача состоит именно в том, чтобы найти относительную важность различных характеристик. Если известна желательная таксономия, то, решая обратную задачу, можно подобрать такое сочетание весов γ_j, при котором получается именно эта таксономия.

Многолетний опыт применения алгоритмов таксономии показал, что таксономический анализ данных является мощным средством познания закономерностей изучаемых объектов или явлений.

Глава 5

Распознавание образов

Методы таксономии, описанные в предыдущей главе, позволяют создать начальную классификацию заданного множества m объектов. Эту классификацию $S = \langle s_1, s_2, \ldots, s_l, \ldots, s_k \rangle$ можно зафиксировать для будущего по-разному, в зависимости от ее назначения. Напомним, что по классификации задач анализа данных (гл. 2, § 2) задача типа 1.3.Н или задача таксономии заключается в предсказании всех элементов нового $(n+1)$-го (классификационного) столбца z в шкале наименований, в котором для каждого объекта a_i должен быть указыван номер его таксона (класса) s_l. Поэтому наиболее распространенный способ представления результата таксономии состоит в переформировании исходной таблицы данных путем собирания в отдельные слои всех m_l строк (объектов), входящих в один и тот же l-й таксон.

Для более краткого представления основного содержания такой таблицы можно записать, например, средние значения и дисперсию характеристик объектов каждого таксона. Можно сохранить по одному или несколько типичных представителей (прецедентов) из каждого таксона. Можно в пространстве характеристик описать границы, которыми таксоны отделяются друг от друга.

Любое из таких описаний представляет собой обобщенный образ каждого класса. Если после этого предъявляется новый объект q, не участвовавший в таксономии, и требуется отнести его к одному из k имеющихся классов, то нужно проанализировать характеристики объекта q и распознать образ того класса s_l, на который данный объект наиболее похож. Такая процедура по-

лучила в литературе по анализу данных название *распознавание образов* и соответствует задаче типа 1.1.Н, в которой требуется предсказать один элемент в столбце, измеренном в шкале наименований. Ее решению посвящено большое число работ (например, [26, 62, 63, 69, 94, 132, 140]). На вход алгоритма распознавания обычно подается таблица данных, которая содержит m объектов $(a_1, a_2, \ldots, a_i, \ldots, a_m)$, описанных характеристиками $(x_1, x_2, \ldots, x_j, \ldots, x_n, z)$. Характеристика z измерена в номинальной шкале и отражает результат предварительно проведенной классификации (таксономии). Эта таблица обычно носит название *обучающая выборка*.

Процесс распознавания включает в себя два основных этапа: этап обучения и этап принятия решения или контроля. На первом этапе алгоритм должен обнаружить закономерную связь между значениями описывающих характеристик $(x_1 \div x_n)$ и значением целевой характеристики z. Эта закономерность выражается в виде решающего правила, с помощью которого на этапе контроля по характеристикам любого объекта q можно принимать решение о его принадлежности к одному из k имеющихся образов.

§ 1. Алгоритмы построения решающих правил

В идеальном случае (если бы он кому-нибудь встретился в жизни) каждый образ был бы представлен не обучающей выборкой конечного объема, а полным аналитическим описанием распределения всех существующих в природе объектов этого образа (генеральной совокупностью). Для самых простых вариантов этого идеального случая, когда распределения подчиняются унимодальному закону (лучше всего, если нормальному) и все характеристики измерены в сильных шкалах, в литературе по математической статистике (например, в [5, 111]) описаны строгие и изящные методы построения решающих правил, гарантирующих минимум суммарных потерь R от ошибок распознавания. Практически все реальные задачи распознавания отличаются от такого идеального случая самым важным свойством: отсутствием знаний о генеральной совокупности изучаемых объектов.

Стратегии поведения распознавателей в таком неопределенном положении делятся на два направления. Идея первого направления состоит в стремлении максимально приблизить реальную ситуацию к идеальному случаю и затем спокойно пользоваться

строгими аналитическими методами построения решающих правил. Для этого делается предположение о том, что имеющаяся конечная выборка хорошо отражает свойства генеральной совокупности (гипотеза о представительности выборки) и что генеральная совокупность подчиняется одному или смеси из нескольких наиболее простых законов распределения (гипотеза о типе распределения). Принятие этих предположений эквивалентно принятию гипотезы унимодальной компактности H_u. После этого строится гипотетическая модель идеального случая. Умом эту модель можно понять, но все равно в нее, как и в любую другую эмпирическую гипотезу, можно только верить. В защиту этой веры можно привести тот регулярно повторяющийся факт, что с ростом объема обучающей выборки разница между величиной реальных ошибок распознавания и величиной ошибок, предсказываемых моделью, обычно уменьшается. Следовательно, при большом объеме обучающей выборки рассматриваемая гипотеза имеет достаточно высокую степень подтвержденности.

К тому же человек постоянно принимает решения без гарантий их безошибочности, и описанное выше поведение распознавателей носит вполне естественный характер. Переход к идеальной модели позволяет на последующих шагах процесса построения решающего правила не «изобретать велосипед», а использовать строгую и хорошо разработанную математическую технику. Вместе с тем следует подчеркнуть, что наличие этого рискованного эвристического перехода к модели не позволяет считать решение реальной задачи распознавания в целом безупречно математически корректным. Так что применительно к реальным задачам распознавания деление методов на «хорошие» — статистические и «плохие» — эвристические не имеет под собой оснований.

Второе направление не ставит перед собой цели дотянуться до высоких планок математической статистики. Объем выборки во многих реальных задачах бывает слишком малым и сравнимым с размерностью признакового пространства. Более того, иногда число объектов даже меньше числа признаков. Так, геологи в результате тщательного изучения кимберлитовых трубок могут свести все данные в таблицу, состоящую, например, из 40 объектов и 200 признаков, и предложить построить по ней решающее правило для различения алмазоносных трубок от пустых. Странной в этих условиях выглядела бы попытка строить по таким данным модель распределения и рассуждать о ее параметрах.

Единственно, что остается делать — опереться на гипотезу локальной компактности H_l. Если новая трубка по своим свойствам больше всего похожа на одну из известных алмазоносных трубок, то следует отнести ее к образу алмазоносных. Если же самая близкая трубка оказалась пустой, значит, и эта новая трубка не содержит алмазов.

В свете сказанного мы будем рассматривать задачу построения решающих правил для трех различных случаев: идеального, с опорой на модели и с опорой на прецеденты.

§ 2. Статистические решающие правила

Будем считать, что распределения генеральных совокупностей всех k образов известны и подчиняются нормальному закону с одинаковыми и единичными матрицами ковариаций [5, 21]. Поверхности равной плотности вероятностей в этом случае представляют собой гиперсферы одинакового для всех образов радиуса. Аналитически такие распределения описываются следующими уравнениями:

$$f(x) = ke^{-(x-\mu)\alpha(x-\mu)/2} \tag{1}$$

в одномерном случае и

$$f(x_1, x_2, \ldots, x_n) = ke^{-(X-M)'A(X-M)/2} \tag{2}$$

в многомерном случае.

Нормирующий коэффициент K выбирается таким образом, чтобы интеграл по всему n-мерному евклидову пространству переменных $x_1, \ldots, x_n$ был равен единице. Положительная постоянная α в выражении (1) заменяется в (2) положительно определенной (симметрической) матрицей

$$A = \begin{pmatrix} a_{11} & a_{12} & \ldots & a_{1n} \\ a_{21} & a_{22} & \ldots & a_{2n} \\ a_{31} & a_{32} & \ldots & a_{3n} \\ \ldots\ldots\ldots\ldots\ldots\ldots \\ a_{n1} & a_{n2} & \ldots & a_{nn} \end{pmatrix}.$$

Скалярная величина μ и скалярная переменная x заменены соответственно векторами

$$M = \begin{pmatrix} \mu_1 \\ \mu_2 \\ \dots \\ \mu_n \end{pmatrix}, \quad X = \begin{pmatrix} x_1 \\ x_2 \\ \dots \\ x_n \end{pmatrix}.$$

Выполнение условий, наложенных на K, приводит формулу (2) к виду

$$f(x_1, x_2, \dots, x_n) = (2\pi)^{-n/2} \sqrt{|A|}\, e^{-(X-M)' A (X-M)/2}. \qquad (3)$$

Анализ этого выражения показывает, что M — математическое ожидание многомерного вектора X, а A — матрица, обратная матрице ковариаций компонент этого вектора, т. е. $M = M[X]$, $A = V^{-1}$, где V — положительно определенная ковариационная матрица вектора X.

Учитывая, что $(X-M)'V^{-1}(X-M) = Q^{-1}$ — положительно определенная квадратичная форма и матрицы ковариаций всех k образов равны между собой, т. е. $V_1 = V_2 = \dots = V_k$, получаем, что отношение вероятности $P_i(X)$ присутствия в точке X образа i к вероятности $P_j(X)$ присутствия в X образа j (отношение правдоподобия) имеет вид

$$P_i(X)/P_j(x) = e^{0{,}5(Q_j^{-1} - Q_i^{-1})}.$$

Удобно пользоваться логарифмом отношения правдоподобия:

$$\ln \frac{P_i(X)}{P_j(X)} = \frac{1}{2}\left(Q_j^{-1} - Q_i^{-1}\right) = U_{ij}(X). \qquad (4)$$

В соответствии с критерием Байеса, если $U_{ij}(X) \geq 0$, то точка X принадлежит области образа i, а если $U_{ij}(X) < 0$, то X считается относящейся к образу j. Следовательно, оптимальная граница проходит по точкам, в которых $U_{ij}(X) = 0$. Уравнение (4) запишем в следующем виде:

$$\begin{aligned} U_{ij}(X) &= \frac{1}{2}\left(Q_j^{-1} - Q_i^{-1}\right) \\ &= X'V^{-1}(M_i - M_j) - \frac{1}{2}(M_i - M_j)'V^{-1}(M_i - M_j). \end{aligned} \qquad (5)$$

Если параметры X независимы, то матрицы ковариаций V и обратные им матрицы V^{-1} диагональны:

$$V = \begin{pmatrix} \sigma_{12} & 0 & \dots & 0 \\ 0 & \sigma_{22} & \dots & 0 \\ \dots & \dots & \dots & \dots \\ 0 & \dots & 0 & \sigma_{n_2} \end{pmatrix},$$

$$V^{-1} = \begin{pmatrix} 1/\sigma_{12} & 0 & \dots & 0 \\ 0 & 1/\sigma_{22} & \dots & 0 \\ \dots & \dots & \dots & \dots \\ 0 & \dots & 0 & 1/\sigma_{n_2} \end{pmatrix}.$$

При этом уравнение (5) принимает вид

$$U_{ij}(X) = \sum_{l=1}^{n} \left\{ [\mu_i(l) - \mu_j(l)]/\sigma_l^2 \right\} x_l - \frac{1}{2} \sum_{l=1}^{n} \left\{ [\mu_i(l)^2 - \mu_j(l)^2]/\sigma_l^2 \right\} = 0. \quad (6)$$

Введя очевидные обозначения, получим уравнение (6) в форме, обычной для представления гиперплоскостей:

$$U_{ij}(X) = \sum_{l=1}^{n} a_l(ij) x_l - a_0(ij) = 0. \quad (7)$$

Теперь, подставив в уравнение (7) координаты контрольной точки q, мы получим значение величины U_q и по ее знаку определим, по какую сторону от разделяющей границы находится точка X и, следовательно, какому из двух конкурирующих образов i или j она принадлежит.

Вид линейной решающей границы для двумерного случая показан на рис. 12, *а*. В многомерном пространстве разделяющая граница представляет собой гиперплоскость, которая перпендикулярна отрезку прямой, соединяющей математические ожидания, и делит этот отрезок пополам. Результат распознавания по описанному правилу будет точно таким же, если вычислить расстояния $r(X, l)$ от точки X до математических ожиданий l всех k

образов и выбрать тот образ, расстояние до которого окажется наименьшим. Этот метод, известный под названием корреляционный [101], не требует хранения в памяти машины аналитического описания распределений образов или уравнения разделяющих гиперплоскостей. Достаточно помнить только координаты математических ожиданий образов.

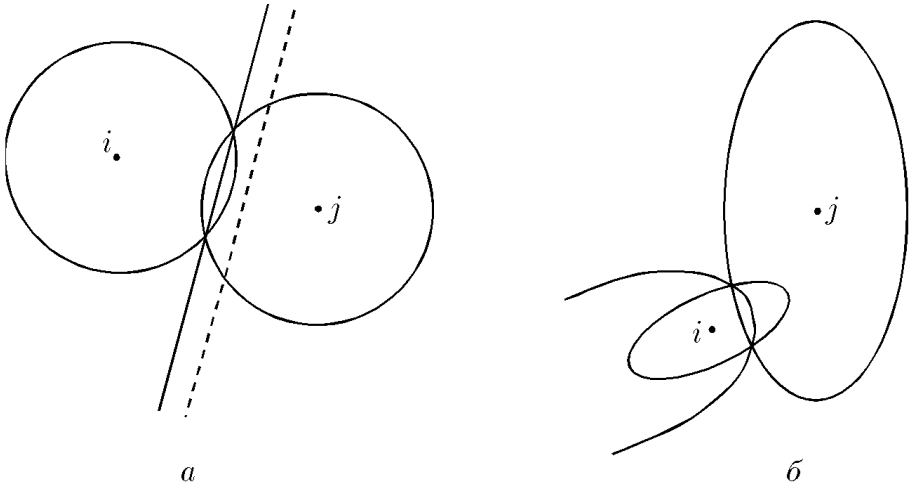


Рис. 12

Если принять, что матрицы ковариаций разных образов не одинаковы, то поверхности равной плотности вероятности будут иметь вид эллипсоидов разной ориентации и размеров (см. рис. 12, *б*). Решающее правило при этом использует разделяющую поверхность второго порядка и имеет следующий вид:

$$U_{ij} = \ln \frac{P_i(X)}{P_j(X)} = \ln \sqrt{|V_j|/|V_i|} + \frac{1}{2}\left(Q_j^{-1} - Q_i^{-1}\right). \qquad (8)$$

Считаем, что точка q с координатами X относится к i-му образу, если $U_{ij}(X) \le 0$.

Стратегия, при которой учитываются только апостериорные вероятности $P_i(X)$ и $P_j(X)$, называется стратегией *идеального наблюдателя* [158]. Реальный же наблюдатель в процессе принятия решений учитывает и другие факторы. В частности, если реализацию i-го образа по ошибке отнести к j-му образу, то это приведет к потерям $C(j/i)$. При ошибочном распознавании представителя j-го образа в качестве реализации i-го образа потери

равны $C(i/j)$. Если эти потери не одинаковы, например, если $C(j/i) > C(i/j)$, то выгоднее разделяющую границу сдвинуть в сторону центра j-го образа (пунктирная линия на рис. 12, a). При этом суммарные потери меньше, чем при использовании стратегии идеального наблюдателя.

В середине 50-х годов у одного из наших туристов, вернувшихся из Индии, были обнаружены признаки заболевания чумой. Были приняты экстренные меры, направленные против распространения этого заболевания: изолированы на несколько дней и тщательно обследованы не только туристы данной группы, но также все их родственники, друзья и сослуживцы, с которыми успели повидаться эти туристы после поездки. Затраты на это мероприятие были немалыми, но они были несравненно меньшими по сравнению с затратами, которые потребовались бы для ликвидации возможного очага эпидемии чумы. Если бациллоносителей считать представителями образа i, а здоровых людей — образа j, то стоимость «пропуска цели» $C(j/i)$ гораздо больше стоимости «ложной тревоги» $C(i/j)$. В этом случае лучше обследовать лишнюю сотню здоровых людей (т. е. отнести их по ошибке к образу i), чем оставить незамеченным одного реального бациллоносителя из образа i, отнеся его по ошибке к образу здоровых j.

При прочих равных условиях целесообразно обращать внимание на априорную вероятность появления образа $P_0(X)$. Если, например, реализации образа i встречаются чаще, чем реализации образа j, то для минимизации ошибок разделяющая поверхность должна быть сдвинута в сторону центра более редкого образа j. Учет этих дополнительных соображений приводит уравнение решающего правила к следующему виду: реализация X относится к образу i, если

$$U_{ij}(X) = \ln \frac{P_i(X) P_{ai} C(j/i)}{P_j(X) P_{aj} C(i/j)} \geq 0. \qquad (9)$$

Если это неравенство не выполняется, то контрольная реализация X считается принадлежащей образу j.

§ 3. Алгебраические методы построения решающих правил

В последнее время широкое распространение получили алгебраические методы построения алгоритмов распознавания и про-

гнозирования [42, 62, 63, 116, 138]. Суть алгебраического подхода коротко может быть описана так. Представим, что некоторая задача распознавания решается с помощью конечного набора C решающих функций: C_1, например, линейная решающая функция, C_2 — квадратичная, C_3 — правило k ближайших соседей. Если качество полученных этими функциями решений окажется неудовлетворительным, то можно расширить круг используемых функций и среди этого расширенного множества попытаться найти функцию, которая давала бы более высокий результат.

Рассматривается два типа расширений. Вначале некоторые параметры исходных функций из констант превращаются в переменные. Варьирование значениями этих переменных порождает широкий класс решающих функций того или иного типа: конечный или бесконечный набор различных гиперплоскостей, набор правил ближайшего соседа с разными значениями k и разной метрикой для вычисления расстояний между точками. Доказано, что почти всегда в этом параметрическом расширении можно найти решающую функцию, которая дает оптимальное решение данной задачи.

Если же встретился такой сложный случай, что оптимального решения получить не удается, тогда применяется другой (алгебраический) способ расширения разнообразия решающих правил. Рассмотрим множество операторов B над множеством простейших решающих правил C. С помощью алгебраических операторов B можно из набора простых правил C сконструировать любое более сложное правило A для решения задачи z: $A(z) = BC$. Доказано, что множество $M(A)$ алгебраически порожденных правил содержит оптимальное правило для решения любой задачи распознавания. Разработан также способ локализации подмножества правил, среди которых находится оптимальное правило. Однако и после этого количество оставшихся вариантов может оказаться большим. Для сокращения вычислительных трудностей применяются естественные эвристические приемы как на этапе отбора наиболее перспективных правил для включения в исходное множество C, так и на этапе конструирования классов их параметрического и алгебраического расширения.

Алгебраический подход успешно применяется при решении задач распознавания образов, в частности в распознавании и анализе изображений и в задачах прогнозирования многомерных динамических процессов. В русле этого подхода находятся, например, метод коллективов решающих правил (КРП) [138] и метод

комитетов [116].

Идея метода КРП состоит в следующем. Пусть в нашем распоряжении имеется обучающая выборка A в пространстве X и несколько решающих правил. Предполагается, что разные правила могут оказаться «хорошими» в одной части пространства X и «плохими» в другой. Каждый признак системы X имеет конечное число градаций, так что пространство X можно представить состоящим из конечного количества «клеточек» (гиперпараллелепипедов). Распознаваемый объект q поместим в произвольную клеточку пространства X и применим для его распознавания все решающие правила по очереди. Отметим те правила, которые приняли правильное решение. Затем переместим объект q в другую клеточку пространства X и повторим распознавание. Снова отметим правила, успешно работавшие в этой части пространства. Таким способом просмотрим все части пространства X и для каждого решающего правила укажем границы области или перечень клеточек, в которых оно оказалось наиболее компетентным. На этом этап обучения заканчивается.

На этапе распознавания контрольного объекта q сначала определяется правило, которое было наиболее компетентным для той части пространства, в которую попал данный объект. Затем по этому правилу определяется принадлежность объекта к одному из распознаваемых образов.

В методе комитетов в начале рассматривается широкий набор решающих правил, например параметрическое семейство из конечного числа гиперплоскостей. Каждая плоскость делит пространство X на две части, и при распознавании двух образов (i и j) можно указать вероятность присутствия представителей этих образов в одной (a) и другой (b) части пространства: P_{ia}, P_{ja} и P_{ib}, P_{jb}. Если в каждой из частей вероятности разных образов окажутся одинаковыми ($P_{ia} = P_{ja}$ и $P_{jb} = P_{ib}$) , то такая плоскость интереса не представляет. Более полезными будут плоскости, которые отделяют друг от друга области с преобладающим присутствием одного из двух образов, например $P_{ia} \gg P_{ja}$ и $P_{jb} \gg P_{ib}$. По этой информации можно выбрать подмножество (коллектив) из S наиболее «информативных» плоскостей.

Решение о принадлежности распознаваемого объекта q к тому или иному образу принимается коллективом правил путем голосования. Если объект q относительно плоскости s находится в области a, то эта плоскость голосует в пользу образа i с весом P_{sia}, а в пользу образа j — с весом P_{sja}. Можно просуммировать

голоса, поданные всеми S плоскостями за i-й образ, и получить оценку P_i. Аналогично получается сумма голосов P_j за j-й образ. Решение в пользу i-го образа принимается, если $P_i > P_j$. Можно пользоваться не суммами, а произведениями голосов.

Процедура построения коллективного решающего правила хорошо иллюстрирует важную роль методов распознавания в процессе познания. Исходная ситуация характеризовалась высокой степенью неопределенности, отсутствием какой бы то ни было модели изучаемого явления. Каждая отдельная гиперплоскость не позволяла надежно отличать один образ от другого, т. е. была «некорректной» распознающей моделью. Параметрический класс линейных решающих правил позволил сформировать из своего состава «корректную» распознающую модель. Как подчеркивает Ю. И. Журавлев [62], именно таким путем с помощью методов распознавания ситуации в неформализованных или слабо формализованных естественнонаучных областях оснащаются формализованными средствами познания. Создаваемые при этом модели позволяют ответить хотя бы на вопрос «Что происходит?». Если в обучающей выборке имеется соответствующая информация, то ее дальнейший анализ может привести к обнаружению закономерностей причинно-следственного характера и сформировать модель для ответа на вопрос «Как это происходит?» или даже на вопрос «Почему именно так, а не иначе?».

§ 4. Распознавание большого числа образов

В большинстве реальных задач распознавания выбор приходится делать не из двух, а из k альтернатив, где k может быть очень большим. Так, в некоторых системах автоматического распознавания устной речи список распознаваемых слов достигает 20 тысяч. Можно, конечно, порекомендовать провести по описанным выше правилам все парные сравнения (их будет 19 миллионов) и выбрать образ с наибольшим значением модуля величины $U_{ij}(X)$. Но вряд ли кому-нибудь захочется следовать такой рекомендации. Для распознавания большого числа образов разработан ряд ускоряющих процедур. Рассмотрим некоторые из них.

4.1. Метод отбора сильнейшего конкурента (МСК). Процесс начинается с проверки принадлежности точки X одному из двух образов, стоящих в начале списка распознаваемых образов. Для этого вычисляются величины $p_1 = P_1(X)P_{01}C(2/1)$ и

$p_2 = P_2(X)P_{02}C(1/2)$ и выбирается тот образ i, для которого величина p_i оказалась большей. С этим победителем тем же методом сравнивается третий по списку образ и из них двоих выбирается более сильный конкурент. Затем делается аналогичное сравнение с четвертым по списку и т. д. до завершения списка из k распознаваемых образов. В итоге этих $(k-1)$ сравнений выбирается образ, который выигрывает соревнование со всеми сильными конкурентами. Если считать априорные вероятности P_{0i} и цены ошибок $C(j/i)$ для всех образов одинаковыми, то задача сводится к вычислению k величин $P_i(X)$ и выбору наибольшей из них.

4.2. Метод попутного разделения (ПОРА) [66]. Если построить оптимальную разделяющую границу между образами i и j, то можно обнаружить, что эта граница попутно разделяет и другие некоторые пары образов с ошибкой R, не превышающей допустимого значения. Так, решающая граница (см. рис. 13), построенная для разделения образов 2 и 4, может использоваться также для разделения образов 1 и 3 от образов 5, 6 и 7.

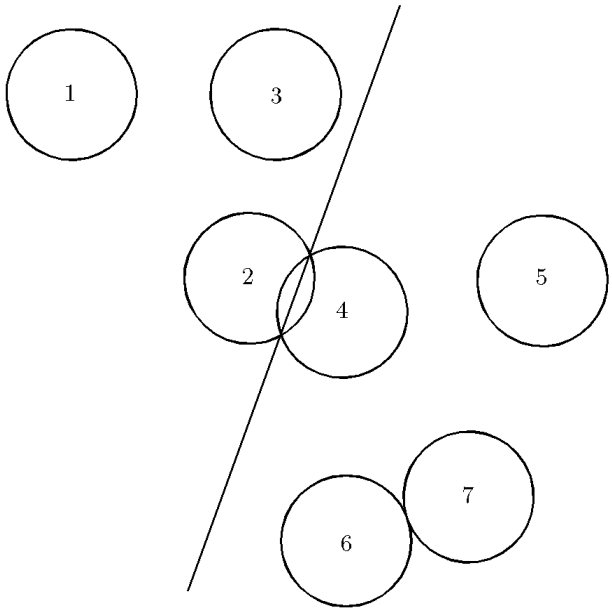


Рис. 13

Процесс нужно начинать с разделения двух наиболее трудно разделяемых образов. Отмечаются все пары образов, успешно разделенные построенной границей. Среди пар, оставшихся неразделенными, снова выбирается пара самых трудно разделяемых, для которой строится следующая разделяющая граница. Из списка неразделенных вычеркиваются пары, которые разделились этой границей. Такие шаги повторяются до полного исчерпания списка неразделенных пар. При удачном расположении образов число h разделяющих границ может оказаться значительно меньшим, чем число распознаваемых образов. Для каждой границы фиксируется список тех k_1 образов, которые находятся по одну сторону от нее, и тех k_2 образов, которые находятся по другую сторону. Будем характеризовать информативность границы величиной $I = 1-(|k_1-k_2|/k)$, которая принимает значения в диапазоне от нуля до единицы и равна единице, если граница делит множество образов на две равные части.

Процесс распознавания с помощью построенных границ состоит в следующем. Начиная с самой информативной границы, определяется, по какую сторону от нее находится контрольная точка X, и вычеркиваются из списка претендентов все образы, находящиеся с противоположной стороны от границы. С помощью следующей по информативности границы список оставшихся претендентов сокращается снова. Так продолжается до тех пор, пока в списке не останется один единственный образ, к которому и относится распознаваемая точка X.

Если образы в пространстве своих характеристик упакованы очень плотно, то даже при использовании всех оптимальных решающих границ ошибки распознавания будут большими, и может оказаться, что число h границ, нагруженных попутным разделением, сравнимо или больше числа образов k. В этом случае более экономичным по требуемой памяти и количеству операций будет упоминавшийся выше корреляционный метод, основанный на расстояниях от контрольной точки до математических ожиданий k образов.

Желательно предварительно оценить величину исходной надежности распознавания N, при которой можно ожидать, что число h не превышает числа k. Определение N было сделано путем машинного моделирования. В пространстве n переменных с помощью датчика случайных чисел задавались координаты математических ожиданий k образов. Дисперсия образов так согласовывалась с объемом пространства, чтобы потери при оптимальном

разделении образов не превышали $R = 1 - N$. Затем применялся метод попутного разделения и определялось число h границ, необходимых и достаточных для обеспечения того же результата.

Эксперимент проводился для $n = 2$ при числе образов $k = 4, 6, 8, 10, 15, 20, 30, 40$ и 100 для надежности оптимального распознавания $N = 0,8;\ 0,85;\ 0,9;\ 0,95;\ 0,999$. Для каждого такого сочетания условий проводилось по 10 опытов, отличающихся случайным расположением образов друг относительно друга. Оказалось, что число h меньше числа k в тех случаях, когда надежность оптимального распознавания находится в диапазоне от 0,9 до 1. От числа образов этот результат не зависит. Отсюда следует, что пользоваться методом попутного разделения вместо определения расстояний до математических ожиданий целесообразно тогда, когда надежность распознавания при оптимальных решающих границах равна или больше 90 %.

Можно ожидать, что в пространстве большей размерности метод попутного разделения даст еще более ощутимый выигрыш, так как с ростом размерности увеличивается число областей, на которые можно разделить пространство с помощью одного и того же числа границ.

4.3. Метод покоординатного вычеркивания (МПВ) [67]. Представим себе, что мы считаем допустимыми потери типа «пропуск цели» равными R_0. Такие потери могут возникать тогда, когда разделяющая граница проходит на расстоянии d от математического ожидания μ_i образа i. Если контрольная точка X удалена от μ_i на расстояние, большее d, то можно считать, что она не принадлежит образу i. Свой вклад в это расстояние вносит каждая координата пространства признаков и, если хотя бы по одной l-й координате расстояние $r(x_l, \mu_{il})$ для образа i равно или больше d, то i-й образ из списка конкурентов можно вычеркнуть. На этом соображении основан метод покоординатного вычеркивания, который состоит в следующем (см. рис. 14).

Рассматриваются проекции точки X и распределений всех образов на каждую координату в отдельности. По первой (l-й) координате определяются расстояния r между точкой x_l и математическими ожиданиями μ_{il} всех k образов. Те образы, для которых выполняется условие $r(x_l, \mu_{il)} \geq d$ (заштрихованная область), из списка претендентов на включение точки X в свой состав исключаются. Для оставшихся образов та же процедура повторяется с использованием проекции на вторую координату, и это про-

должается до тех пор, пока в списке претендентов не останется заданное число k^* образов. Для этих самых сильных претендентов вычисление расстояний и оценка ожидаемых потерь делается в исходном n-мерном пространстве с использованием оптимальных решающих правил.

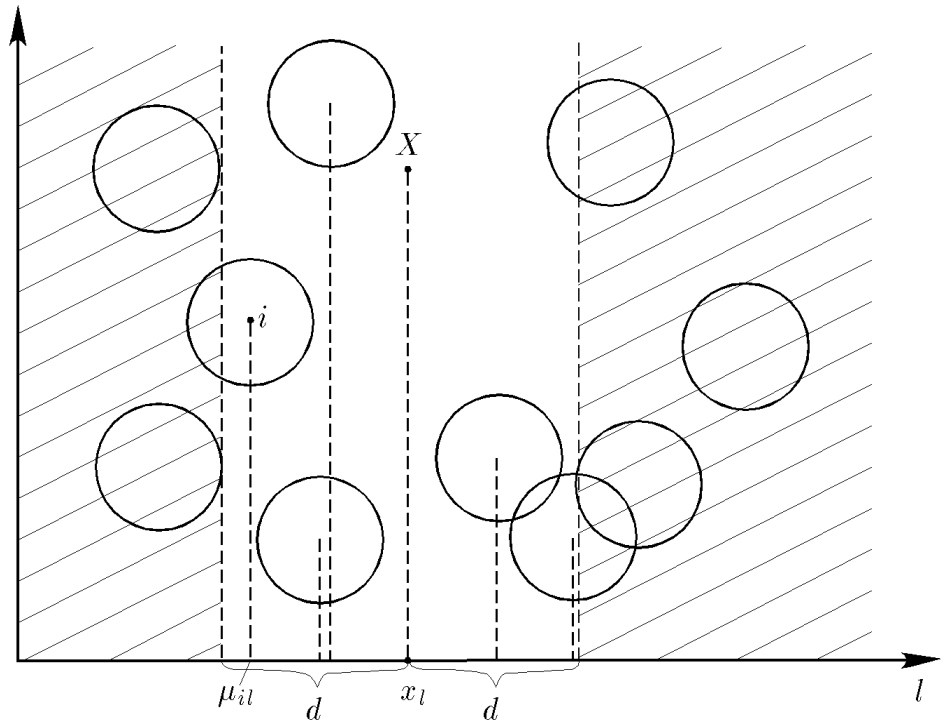


Рис. 14

Сравнение временных затрат на распознавание методом покоординатного вычеркивания (МПВ) и корреляционным методом (КМ) показывает, что с ростом размерности пространства эффективность МПВ быстро растет. Зависимости от числа образов не наблюдается, важно лишь, какая их доля вычеркивается на каждом шаге. Так, для случаев распознавания в пространствах размерности $n = 150$ и $n = 75$ достаточно, чтобы на каждом шаге вычеркивалось по 7 и 13 % образов соответственно. При этом время на принятие решения по МПВ меньше, чем по КМ. Реально эти времена могут отличаться на порядки.

§ 5. Оценка потерь

Выше мы часто говорили о потерях, возникающих от ошибок распознавания. Уточним это понятие.

Начнем с наиболее простого случая: количество образов $k = 2$, распределения нормальные, матрицы ковариаций, априорные вероятности и стоимости потерь для обоих образов одинаковы, т. е. $V_i = V_j$, $P_{0i} = P_{0j} = 0,5$, $C(j/i) = C(i/j)$. Оптимальной решающей границей для этого случая является гиперплоскость $U_{ij}(X)$, причем реализация X распознается в качестве представителя образа i, если $U_{ij}(X) \geq 0$ (т. е. если X попадает в область Y_i), и j-го образа, если $U_{ij}(X) < 0$ (т. е. если X лежит вне области Y_j).

Вероятность ошибочного отнесения реализации i-го образа к j-му принимает значение $R(j/i) = \int\limits_{Y_i} P_i(X)\,dX$, так что $R(j/i)$ при заданном законе распределения $P_i(X)$ в одномерном случае можно найти по таблице интегралов вероятности одномерного нормального распределения [29]. По этой же таблице находится и вероятность $R(i/j)$ ошибочного отнесения реализаций j-го образа к i-му, так что средние потери от ошибок распознавания этих двух образов выражаются величиной $R = [R(j/i) + R(i/j)]/2$.

Если пространство признаков двумерно, то достаточно повернуть координатные оси так, чтобы одна координата (x_1) стала параллельной линии, соединяющей математические ожидания образов (рис. 15). При этом проекции образов на ось x_2 полностью совпадают друг с другом и разделение образов возможно только с использованием проекции распределений и решающей границы на ось x_1. Таким способом задача оценки потерь сводится снова к одномерному случаю. Аналогично решается задача и при $n > 2$: ищется главная компонента и рассматривается проекция образов на нее.

Если параметры задачи P_0 и C для образов не одинаковы, то средние потери двух образов выражаются следующей величиной: $R = P_{0i}R(j/i)C(j/i) + P_{0j}R(i/j)C(i/j)$. Если число образов больше двух, то общие потери определяются выражением

$$R = \sum_{i=1}^{k} P_{0i} \sum_{\substack{j=1 \\ j \neq i}} R(j/i)C(j/i) = \sum_{i=1}^{k} P_{0i}R_i,$$

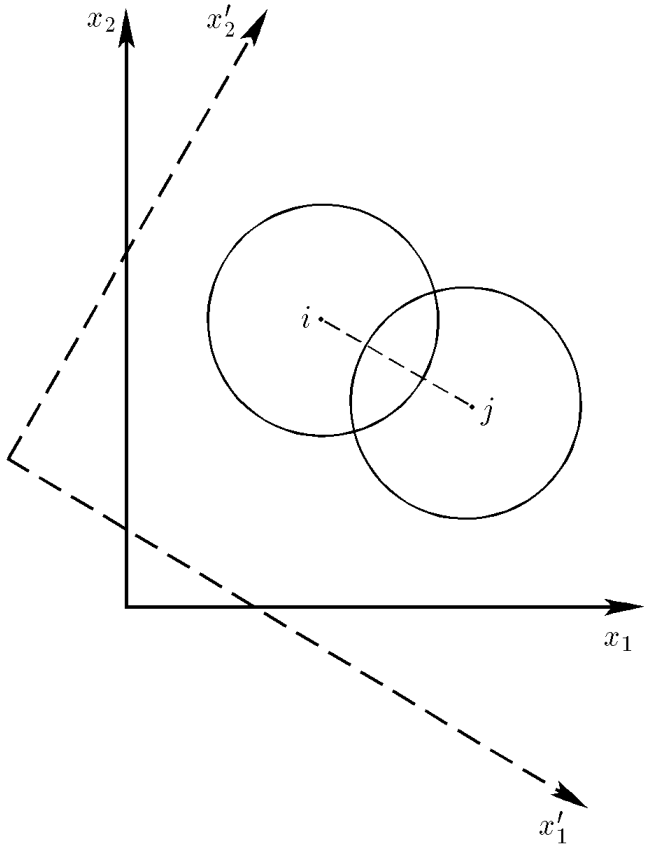


Рис. 15

где R_i — потери, связанные с ошибками распознавания i-го образа. При неравных матрицах ковариаций ($V(j/i) \neq V(i/j)$) аналитические выражения для оценки потерь имеют более сложный вид, и при необходимости с ними можно познакомиться по работам [5, 107].

§ 6. Гипотеза компактности в распознавании образов

Сформулированное в главе 3 условие компактности для решения задачи распознавания образов является необходимым, но не достаточным. Мало того, чтобы точки образа A были близкими

друг к другу, нужно еще, чтобы точки образа B не оказались к ним такими же близкими, т. е. нужно, чтобы сгустки точек разных образов не налагались друг на друга, что обозначаем следующим образом: $C^{-X}_{A,B}$. С учетом этого гипотезу компактности H для распознавания образов можно записать в следующем виде:

$$\text{if } \left(C^{-X}_{A,B} \& C^{X,z}_{A} \& C^{X}_{A,q}\right) \text{ then } C^{z}_{A,q}.$$

Если предполагать, что реализации одного и того же образа образуют один компактный сгусток, то его можно аппроксимировать унимодальным распределением. Этот случай соответствует *гипотезе унимодальной компактности* (H_u).

Ослабленный вариант обсуждаемой гипотезы (назовем его *гипотезой полимодальной компактности* (H_p)) утверждает, что точки одного образа могут образовывать не один, а несколько компактных сгустков. На этом основании можно представлять образ многосвязными областями или смесью нескольких простых распределений.

Полимодальную компактность можно в пределе представить в виде локальной компактности. *Гипотеза локальной компактности* (H_l) выражает осторожное утверждение о свойстве ближайшего соседства: «обо всем распределении судить не берусь, но в некоторой малой ε-окрестности каждой реализации обучающей выборки образа i может появиться только представитель этого же образа». На указанном основании построить общую модель распределения образа нельзя, но можно построить правило распознавания с опорой на все или на часть объектов обучающей выборки (т. е. с опорой на прецеденты).

Проекции компактных сгустков на координатные оси будут также компактными. Если, кроме того, эти проекции окажутся еще и не совпадающими друг с другом, то появляется возможность разделять образы не сразу в многомерном пространстве признаков, а поочередно, по каждому признаку в отдельности. На этом основании можно сформулировать гипотезы проективной унимодальной (H_{pu}), проективной полимодальной (H_{pp}) и проективной локальной (H_{pl}) компактности.

Из сказанного выше следует, что все алгоритмы распознавания, отправляющиеся от обучающей выборки, в своей основе отличаются друг от друга лишь вариантом принимаемой гипотезы компактности вне зависимости от того, формулируется ли она в явном виде или интуитивно подразумевается.

А дальше пути разных школ распознавания расходятся. Как указывалось выше, часть из них строит модели унимодальных распределений генеральных совокупностей распознаваемых образов и потом для построения решающих правил применяет аппарат математической статистики. Другие ориентируются на гипотезу полимодальной компактности, строят модели распределений в виде смеси простых распределений, после чего применяют те же статистические методы, но как бы для большего числа образов. Те, кто ориентируется на гипотезу локальной компактности, строят решающие правила, опирающиеся на прецеденты. Наконец, те, кто предполагает, что компактность в многомерном пространстве проявляется в компактности проекций сгустков на координатные оси, используют последовательные покоординатные процедуры. Рассмотрим особенности этих направлений.

§ 7. Построение решающих правил по конечной выборке

Если мы хотим воспользоваться техникой построения статистических решающих правил, то нам нужно построить модели распределений образов. А для этого нужно помимо гипотезы компактности принять еще несколько дополнительных предположений, которые давали бы ответы на следующие вопросы: какому закону подчиняются распределения образов, каков характер зависимости между признаками, какова априорная вероятность появления различных образов, как выглядит матрица стоимости ошибочного распознавания, какова степень представительности обучающей выборки? Так как обычно нет объективных способов ответить на эти вопросы, то в качестве ответов принимаются самые простые предположения. Чаще всего предполагается, что распределения подчиняются нормальному закону, что матрицы ковариаций диагональны (значит, признаки системы X не зависимы), что априорные вероятности и стоимости взаимных ошибок всех образов одинаковы и что обучающая выборка представительна. После добавления к фактическим данным этого букета эвристических подпорок можно приступать к вычислению по обучающей выборке параметров распределений и построению по ним «оптимальных» решающих правил.

Задаче подбора вида распределения и его параметров по конечной выборке посвящено большое количество работ [8, 17, 43, 48,

111]. Описаны программные системы, предназначенные для решения этой задачи при различных критериях согласия [3, 47, 112].

Самый простой, но часто применяемый прием состоит в следующем. Для каждого образа вычисляются координаты центра тяжести его точек, и эти координаты принимаются в качестве координат математического ожидания нормального распределения данного образа. Затем строится одно из решающих правил, описанных в § 2 данной главы. При опоре на гипотезу полимодальной компактности модель распределения аппроксимируется смесью простых (например, нормальных) распределений. Этот аппроксимационный подход развивается в работах [35, 118] и ряде других. Если для заданного критерия согласия наилучшая аппроксимация достигается с помощью одного распределения, то ситуация сводится к той, что рассмотрена выше. Если для образа i потребовалось использовать t_i распределений ($t_i > 1$), то общее число распределений, построенных в признаковом пространстве, окажется равным сумме этих чисел: $T = \sum t_i$, $i = 1, \ldots, k$. Все эти распределения рассматриваются в качестве описаний T отдельных образов, среди которых имеется k подмножеств из t_i образов с одинаковыми именами. При построении решающих правил следует применять методы, предназначенные для распознавания большого числа образов (см. § 3 данной главы). При использовании метода отбора сильнейшего конкурента необходимо следить за списком образов, оставшихся непроверенными. Если в этом списке остались образы с одним и тем же именем i, то дальнейшая проверка прекращается. Так, на рис. 16 показан случай, в котором проверка первых трех образов выявит в качестве сильнейшего претендента образ 3 с именем B. После 5-го шага на первое место выйдет образ 5 с именем A, и в списке оставшихся претендентов останутся только образы с тем же именем A, так что дальнейшие сравнения окончательного результата распознавания не изменят.

Метод попутного разделения (ПОРА) модифицируется для данной задачи так, чтобы из списка неразделенных образов с самого начала были удалены пары распределений с одинаковыми именами. В итоге фиксируются только границы, разделяющие образы с разными именами. Так, для разделения смесей образов с именами A и B, представленных на рис. 16, достаточно использовать две границы из 21 возможных. Разделяющая граница, составленная из нескольких линейных границ (в n-мерном пространстве гиперплоскостей), называется *кусочно-линейной*. Способам

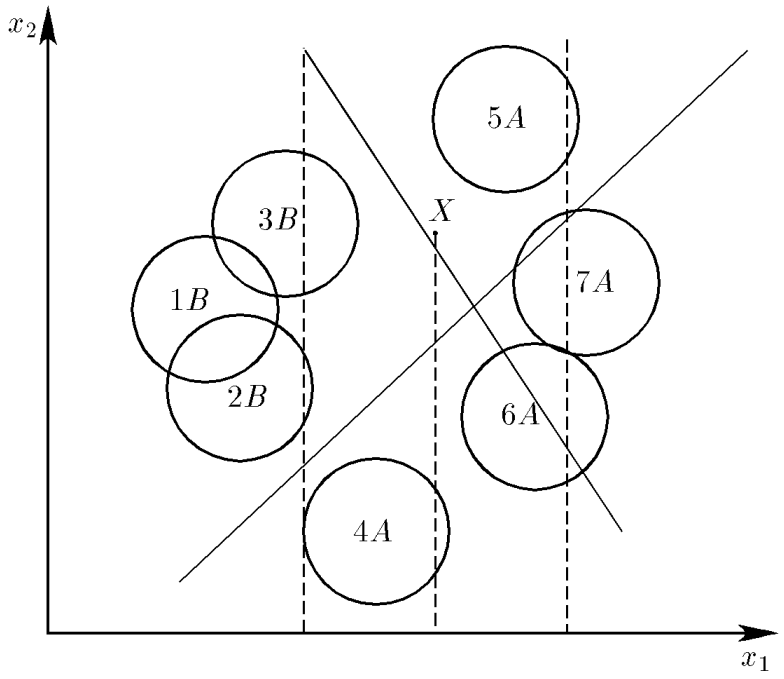


Рис. 16

построения аппроксимаций нелинейных поверхностей кусочно-линейными поверхностями посвящены работы [105, 141]. При использовании метода покоординатного вычеркивания (МПК) также нужно следить за составом невычеркнутых образов. Если на некотором шаге вычеркивания обнаружится, что все оставшиеся претенденты имеют одно и тоже имя, то процесс распознавания заканчивается. Так, на рис. 16 вычеркивание по признаку x_1 (пунктирные линии) оставляет невычеркнутыми только образы с именем A и точка X распознается в качестве реализации образа A.

§ 8. Решающие правила, опирающиеся на прецеденты

Напомним, что не все распознаватели верят в свою интуицию и прозорливость настолько, чтобы, глядя на бедную обучающую выборку, утверждать что-либо определенное о виде законов распределения, параметрах этих распределений, зависимости между признаками, представительности выборки и т. д. Но совсем без

добавочных предположений обойтись нельзя, и в качестве единственной эмпирической гипотезы они используют самый слабый вариант рассмотренной выше гипотезы компактности — локальную компактность H_l, из которой следует, что похожесть двух объектов по n признакам обычно сопровождается их похожестью и по $(n+1)$-му признаку. А отсюда следует, что рядом, в малой ε-окрестности от имеющихся реализаций i-го образа обучающей выборки, могут появляться только реализации того же i-го образа. Причем чем ближе контрольная реализация q находится к имеющейся реализации образа i, тем с большей вероятностью отнесение точки q к i-му образу будет правильным.

Исходя из этого можно предложить в качестве решающего правила следующую процедуру: оставить в памяти машины все реализации обучающей выборки и контрольную точку q относить к тому образу, чья реализация оказалась ближе всего к точке q. Это правило действительно используется в практике решения некоторых задач, и оно носит название *правила ближайшего соседа* [103]. Однако следует учитывать, что реальные измерения признаков нередко сопровождаются помехами и ошибками, так что свидетельству одного прецедента доверять опасно. Целесообразно учитывать свидетельства и других объектов обучающей выборки. С этой целью обращают внимание не на одну, а на несколько ближайших точек. Такие правила называются *правилами k ближайших соседей*. Если больше половины из k соседей принадлежат образу i, то и точка q относится к i-му образу.

Иногда в голосовании принимают участие все реализации обучающей выборки, но с разными весами, зависящими от их расстояний до распознаваемой точки q. Одним из первых алгоритмов такого рода был алгоритм потенциальных функций [4]. Его сущность иллюстрирует рис. 17. Реализации образа «крестики» как бы излучают потенциал, величина которого убывает с расстоянием r от точки q. Характер убывания может быть самым разным: $1/r$, $1/(r^2+a)$, e^{-ar^2} и т. д.

Точки образа «кружочки» излучают потенциал той же величины, но противоположного знака. В распознаваемой точке q вычисляется «наведенный потенциал» в виде суммы потенциалов от всех точек. Если сумма окажется положительной, то q относится к образу крестики, если отрицательной — к образу кружочки.

Если окажется, что какая-нибудь из точек обучающей выборки распознается по этому правилу с ошибкой, то картину по-

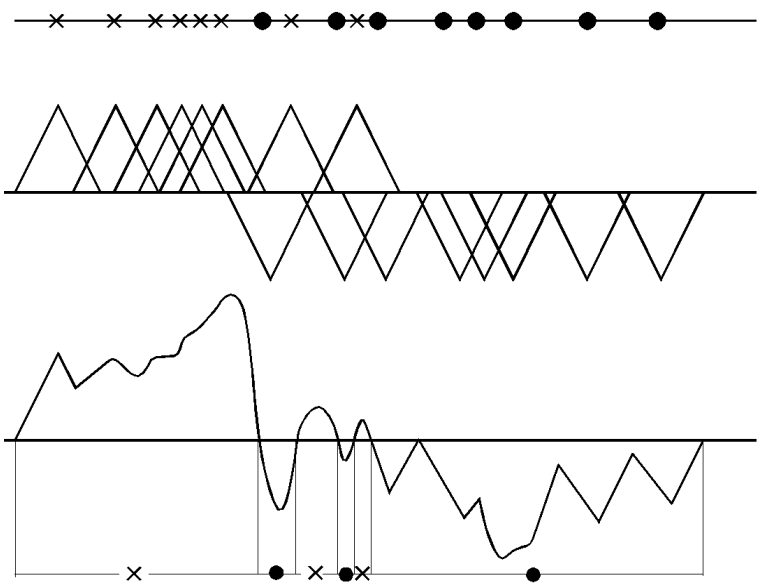


Рис. 17

тенциального поля можно скорректировать. Для этого потенциал в этой точке увеличивается в нужную сторону до тех пор, пока эта точка не станет распознаваться правильно. Авторами метода доказана его сходимость к оптимальному при увеличении объема обучающей выборки и конечность числа шагов обучения (коррекции поля) для случаев образов, хорошо разделяемых с помощью «не слишком вычурных» границ.

8.1. Минимизация набора прецедентов (алгоритм STOLP). Очевидный недостаток описанных здесь методов состоит в том, что в памяти машины нужно хранить сведения обо всех реализациях обучающей выборки. Естественно возникает вопрос: нельзя ли сохранять в качестве прецедентов не все, а лишь часть реализаций каждого образа? Например, оставить в качестве прецедентов только такие «точки опоры» образа i, которые обеспечивают выполнение следующего условия: расстояние от любой точки обучающей выборки i-го образа до ближайшего своего прецедента меньше расстояния до ближайшего прецедента чужого образа. Такой набор прецедентов обеспечит безошибочное распознавание всех реализаций обучающей выборки. Контрольные реализации распознаются по методу ближайшего соседа.

Сложность решения этой задачи состоит в том, что состав прецедентов i-го образа зависит от того, какие реализации других образов выбраны в качестве прецедентов [149]. Из этого становится очевидным комбинаторный характер задачи, оптимальное решение которой в общем случае требует полного перебора всех вариантов. Если количество образов k, число реализаций каждого (i-го) образа в обучающей выборке равно m_i, то число возможных вариантов оставления по одному прецеденту для каждого образа равно $\prod_{i=1}^{k} m_i$. Если же оставлять по t прецедентов на образ, то число вариантов возрастает до величины $\prod_{j=1}^{k} \left(C_{m_i}^{t}\right)$. Перебор вариантов можно сократить с помощью алгоритма STOLP. Рассмотрим его работу на примере с двумя образами (см. рис. 18.)

Сначала находятся самые «напряженные» пограничные точки. С этой целью для каждой точки определяются расстояния до ближайшей точки своего образа (r_{in}) и ближайшей точки чужого образа (r_{out}). Отношение $W = r_{\text{in}}/r_{\text{out}}$ характеризует величину риска для данной точки быть распознанной в качестве точки чужого образа. Среди точек каждого образа выбирается по одной точке с максимальным значением величины W. Эти точки заносятся в список прецедентов.

Затем делается пробное распознавание всех точек обучающей выборки с опорой на прецеденты и с использованием правила ближайшего соседа: точка относится к тому образу, расстояние до прецедента которого минимально. Среди точек, распознанных неправильно, выбирается точка с максимальным значением W и ею пополняется список прецедентов, после чего повторяется процедура пробного распознавания всех точек. Так продолжается до тех пор, пока все точки обучающей выборки не станут распознаваться без ошибок. В примере на рис. 18 в качестве первых прецедентов были выбраны точки 1 и 2. Затем список прецедентов пополнился точками 3, 4 и 5.

При количестве образов большем, чем два, целесообразно применять ускоряющие приемы, описанные в § 4. В частности, описанная процедура выбора прецедентов должна начинаться с пары самых близких образов. В качестве расстояния между образами i и j принимается расстояние $r(a_i, b_j)$ между двумя самыми близкими точками a_i и b_j, принадлежащими разным образам (расстояние Хаусдорфа). После решения задачи для двух первых образов выбирается следующая самая близкая пара образов. Если

среди них окажется образ i (или j) с выделенными на предыдущем этапе прецедентами, то эти прецеденты включается с самого начала в рассмотрение и в случае необходимости будут дополнены новыми.

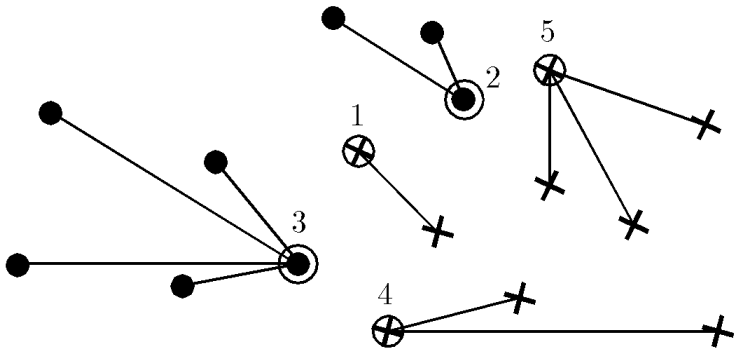


Рис. 18

К концу рассмотрения парных сочетаний всех образов каждый из них будет представлен в памяти набором прецедентов, достаточным для безошибочного распознавания всех своих объектов обучающей выборки. Распознавание новых объектов можно делать по правилу одного или нескольких ближайших соседей. Если число образов k и объем обучающей выборки m очень велики, то алгоритм STOLP может потребовать слишком большого машинного времени. Для его сокращения необходима разработка аналогов тех алгоритмов, которые были описаны в § 4 для случая распознавания большого числа образов с помощью статистических решающих правил (алгоритмы попутного разделения, покоординатного вычеркивания, отбора сильнейшего конкурента). Сложность разработки этих аналогов состоит в том, что здесь не удастся ограничиться сравнением с одним «эталоном» на образ, в качестве которого там использовался вектор математического ожидания нормального распределения. Теперь придется иметь дело с заметно бóльшим числом прецедентов. Отсутствует здесь и помогавшее в прошлом понятие разделяющей поверхности.

Можно предположить, что среди распознаваемых объектов встречаются не только реализации одного из заданных k образов, но и некоторого неизвестного $(k+1)$-го образа. Введем пороговую величину d_i и будем считать, что точки, удаленные от прецедентов i-го образа на расстояние, большее d_i, заведомо не принадле-

жит i-му образу. Если объект q не принадлежит ни одному из k образов, то будем относить его к $(k+1)$-му образу. В качестве порога d_i примем расстояние между двумя самыми далекими друг от друга точками i-го образа («диаметр» образа).

При таком пороговом правиле принятия решений появляется возможность сократить перебор в алгоритме STOLP. Если ближайшие точки двух разных образов i и j удалены друг от друга на расстояние, большее $(d_i + d_j)$, то роль прецедентов в противостоянии этих образов могут выполнить любые две реализации из обучающей выборки (по одной реализации из каждого образа). И никакая контрольная точка q не будет для этих образов спорной: она однозначно будет отнесена либо к одному из них, либо ни к одному. Если в образах i и j, оказавшихся в очередной самой близкой паре, уже имеются прецеденты, выделенные на предыдущих шагах алгоритма, и эти образы удалены на расстояние, большее $(d_i + d_j)$, то попытками дополнить список прецедентов этих образов можно не заниматься. А такие пары по мере перехода от самых близких пар к более далеким будут встречаться все чаще.

Количество прецедентов s, потребовавшихся для безошибочного распознавания всех остальных $(m - s)$ объектов обучающей выборки, может служить индикатором информативности выборочного пространства. В самом благоприятном случае необходимо оставить k прецедентов (по одному представителю на каждый образ). Для самого трудного случая в качестве прецедентов потребуется оставить все m объектов. Величина $J = (m - s)/(m - k)$ указывает на информативность признаков. Если $J = 0$, то безошибочное распознавание обучающей выборки обеспечено по определению, но рассчитывать на успешное распознавание контрольных объектов нельзя.

8.2. Метод «дробящихся эталонов» (алгоритм ДРЭТ). Так же, как и в предыдущих случаях, будем стремиться к безошибочному распознаванию обучающей выборки, но с использованием покрытий обучающей выборки каждого образа простыми фигурами, усложняющимися по мере необходимости [69].

Один из вариантов этого метода предусматривает использование в качестве покрывающих фигур набора гиперсфер. Для каждого из k образов строится сфера минимального радиуса, покрывающая все его обучающие реализации. Значения радиусов этих сфер и расстояний между их центрами позволяет определить

образы, сферы которых не пересекаются со сферами других образов. Такие сферы считаются эталонными (см. образы 1 и 8 на рис. 19), а их центры и радиусы запоминаются в качестве эталонов первого поколения.

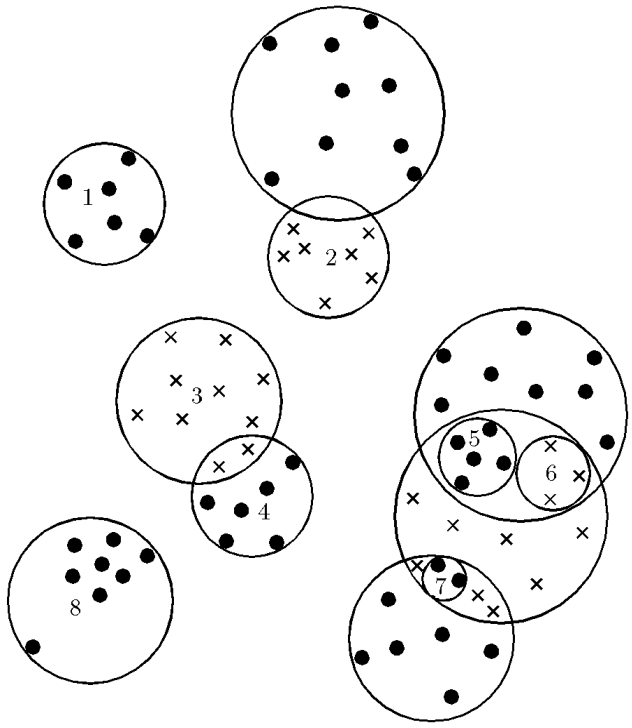


Рис. 19

Если два образа пересекаются, но в области пересечения не оказалось ни одной реализации обучающей выборки, такое пересечение считается фиктивным, центры и радиусы этих сфер также вносятся в список эталонов первого поколения. При этом область пересечения считается принадлежащей сфере с меньшим радиусом (сфере 2 на рис. 19). Если в зоне пересечения оказались точки только одного образа, то эта зона считается принадлежащей этому образу (сфере 3 на рис. 19). Точка считается относящейся к образу 4, если она попадает в сферу 4 и не попадает в сферу 3. Если же область пересечения содержит точки разных образов, то для этих точек строятся эталоны второго поколения (сферы 5, 6 и 7). Если и они пересекаются, то для точек их зоны

пересечения строятся эталоны третьего поколения. Процедура дробления эталонов продолжается до получения заданной надежности распознавания обучающей последовательности.

Дальнейшим упрощением алгоритма ДРЭТ служит его разновидность, отличающаяся только тем, что в качестве покрывающих фигур используются гиперпараллелепипеды. Если их стороны параллельны координатным осям, то задание их положения, проверка условия попадания точек внутрь параллелепипедов или в область их пересечения осуществляется с помощью очень простых процедур, что позволяет экономить приблизительно 30 % машинного времени по сравнению с предыдущим вариантом.

Напомним, что в соответствии с гипотезой H_l чем меньше расстояние от контрольной точки до реализации обучающей выборки i-го образа, тем выше вероятность того, что эта точка принадлежит образу i. И наоборот, вероятность появления реализаций чужого образа возрастает с удалением от обучающих точек i-го образа. Следовательно, чем выше плотность точек в i-й эталонной сфере, тем меньше вероятность $P(i/j)$ отнесения чужой реализации к этому образу.

Если постановка задачи распознавания предостерегает от ошибочного включения реализаций $(k+1)$-го образа в состав любого из k заданных образов (т. е. если цена ошибки $C(i/k+1)$ достаточно велика), то нужно избегать излишней «пустоты» в эталонных сферах (это имеет место, например, в сфере 8). С этой целью вычисляется отношение V числа точек m_i к радиусу сферы R_i. И если V меньше заданного порога V_0, такая сфера подвергается дополнительному анализу: делается попытка заменить одну разреженную сферу несколькими более плотными сферами меньшего диаметра. Это можно сделать с помощью алгоритма таксономии типа FOREL. Если средняя плотность новых сфер окажется выше плотности начальной сферы, то эти сферы вносятся в список эталонов вместо начальной. Следует предостеречь от чрезмерного увлечения процедурой повышения плотности сфер: каждая новая сфера — это дополнительные затраты машинной памяти на эталоны и времени на принятие решений. Кроме того, увеличение числа поколений эталонов фактически приводит к построению «слишком вычурных» границ между образами, что при ограниченной выборке статистически мало оправдано.

Опыт показал, что даже в очень сложных случаях для хорошего распознавания обучающей выборки бывает достаточно в среднем не более трех поколений эталонов. Так, при распознава-

нии 11 устных команд для безошибочного распознавания обучающей последовательности потребовалось от 2 до 5 сфер на образ (в среднем 2,8). При распознавании же по одному эталону на образ эти реализации распознавались с ошибкой 45 %, что говорит о высокой сложности данной задачи.

Если бы мы могли сопоставлять в единых ценах расходы машинных ресурсов на эталоны $C_э$ и стоимость потерь $C(i/k+1)$, то нам следует стремиться к минимизации общих потерь N:

$$\min N = \min \left\{ C_э + \sum_{i=1}^{k} C(i/k+1) P(i/k+1) \right\}.$$

При распознавании новых реализаций процесс проверки попадания в ту или иную эталонную сферу начинается со сфер наименьшего диаметра.

Решающие правила отличаются друг от друга так называемой емкостной характеристикой [25, 102]. Она оценивает отношение сложности правила к его разделяющей способности. Алгоритм ДРЭТ вырабатывает очень простые правила, которые вместе с тем могут отделить друг от друга образы со сложными границами, так что эти правила обладают очень хорошей емкостной характеристикой.

В заключение отметим, что здесь так же, как и в предыдущем алгоритме, имеется возможность по количеству эталонов судить об информативности признакового пространства. Если для безошибочного распознавания m обучающих объектов k образов ($m \gg k$) потребовалось s эталонов, то характеристика $J = (m-s)/(m-k)$ позволяет определить меру информативности признаков. Она меняется в диапазоне значений от единицы (один эталон на образ) до нуля (число эталонов равно числу обучающих объектов). В последнем случае мы имеем дело с очень плохо подготовленной задачей распознавания. Такого рода ситуация образно выражается известной метафорой «вода в губке», для которой гипотеза компактности не выполняется. Ясно, что о надежном распознавании контрольных реализаций здесь говорить не приходится.

8.3. Таксономические решающие функции (алгоритм ТРФ) [24, 54]. Обычно на этапе обучения строится решающее правило, параметры которого определяются и жестко фиксируются по информации, содержащейся в обучающей выборке. Какие

бы контрольные реализации не предъявлялись потом для распознавания, решающее правило не меняется. Но возможен и другой подход: решающее правило строить непосредственно в процессе распознавания, опираясь на информацию, содержащуюся как в обучающей, так и в контрольной выборке.

На рис. 20 объекты обучающей выборки двух образов обозначены кружочками и крестиками, а реализации контрольной выборки обозначены точками. Если аппроксимировать сгустки точек обучающей выборки унимодальными распределениями, то решающая граница представляет линию a. Такое решающее правило, построенное по непредставительной выборке, даст много ошибок при распознавании контрольной выборки.

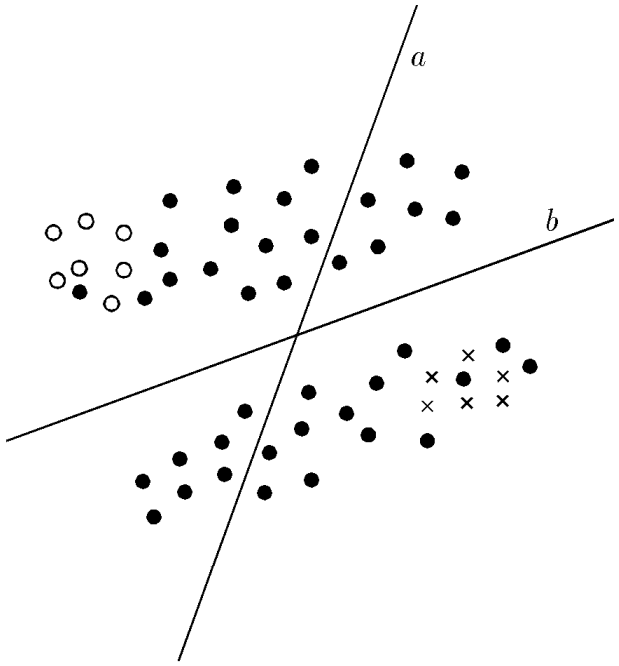


Рис. 20

Если же обучающую и контрольную выборку рассматривать совместно, то можно построить решающую границу в виде линии b, которая обучающую выборку распознает так же хорошо, как и линия a, но снижает количество ошибок на контрольной выборке. Такое решение получается, если применить критерии

таксономии: большие расстояния между таксонами и малые расстояния внутри таксонов. Это и положено в основу принятия решений методом таксономических решающих функций.

Смесь обучающих и контрольных объектов подвергается таксономии на k таксонов с помощью алгоритмов типа FOREL или KRAB. Если в некотором таксоне есть точки обучающей выборки только одного i-го образа, то все контрольные точки, попавшие в этот таксон, относятся также к образу i. Если в таксоне есть точки из k' разных образов, то такой таксон разбивается на k' более мелких таксонов. Эта процедура продолжается до тех пор, пока в каждом таксоне не окажутся обучающие точки только одного образа. Контрольные точки, попавшие в таксон, не содержащий обучающих точек, считаются принадлежащими новому, $(k+1)$-му образу. При желании их можно отнести к одному из k образов, присоединение к которому даст наибольшее значение критерия качества таксономии.

Описанный алгоритм позволяет использовать не только информацию из обучающей выборки, но и из контрольной. Этим обеспечивается более высокая устойчивость алгоритма ТРФ к такому часто встречающемуся неприятному явлению, как непредставительность обучающей выборки.

§ 9. Логические решающие правила

Выше указывалось, что один из вариантов гипотезы компактности состоит в предположении, что множество A, компактное в n-мерном пространстве признаков, обычно компактно и в его проекциях на координатные оси (гипотеза проективной компактности H_p). Применительно к распознаванию образов это очевидное предположение не является достаточным. Требуется еще, чтобы компактные и несовпадающие в n-мерном пространстве сгустки обучающих точек разных образов были бы компактными и несовпадающими и в одномерных проекциях (гипотеза проективной локальной компактности H_{pl}). Обычно это требование в чистом виде не выполняется. Проекции точек разных образов на координатные оси образуют перекрывающиеся области. Однако области перекрытия на разных координатах выглядят по-разному, и есть надежда, что комбинация несовпадающих перекрытий на нескольких осях позволит построить эффективное решающее правило. Пример, иллюстрирующий это, показан на рис. 21. Правило принятия решения о принадлежности

нового объекта q к образу 1, например, выглядит так: «Если $(x_{1q} < a)\&(x_{2q} > b)$, то $z_q = 1$».

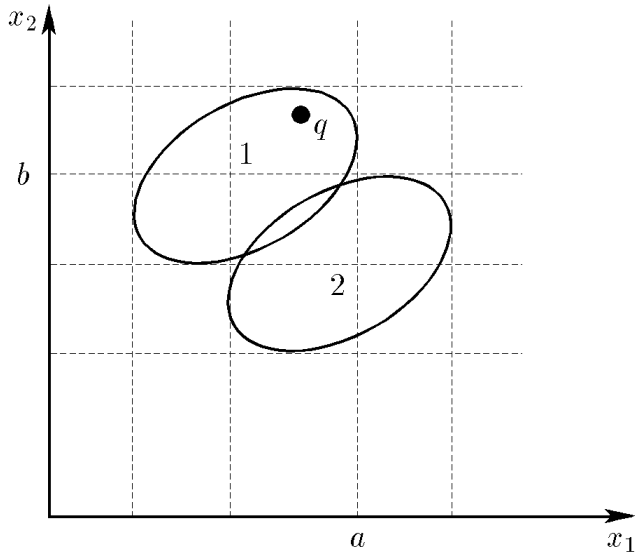


Рис. 21

Правила такого типа получили название *логических решающих правил* (ЛРП) и их исследованию посвящены многие работы, в частности [108, 120, 126]. Приведем описание двух алгоритмов построения ЛРП.

9.1. Алгоритм CORAL [108]. Выделим некоторое подмножество X_{jv} значений признака X_j. Для сильных признаков — это интервал значений, для шкал порядка — ряд соседних порядковых позиций, для шкал наименований — одно или несколько имен. Тот факт, что значение признака X_j у объекта a_i принадлежит подмножеству X_{jv}, обозначаем через $J(a_i, X_{jv})$. Тогда попадание объекта a в область v, образованную границами подмножеств X_{jv}, т. е. в гиперпараллелепипед $(X_{1v} \times X_{2v} \times \cdots \times X_{nv})$, запишем в форме логического высказывания:

$$S(a, X) = J(a, X_{1v})\&J(a, X_{2v})\& \ldots \& J(a, X_{n'v}).$$

В высказывание S могут входить не все n признаков, а любое их непустое подмножество из n' признаков, $n' \leq n$. Число n'

называется *длиной высказывания. Логической закономерностью* называем высказывание, удовлетворяющее двум условиям:

$$P_{ws} = m_{ws}/m_w \le \alpha \quad \text{и} \quad P_{ws^-} = m_{ws^-}/m_{w^-} \ge \beta.$$

Здесь w есть индекс объектов своего образа, w^- — индекс объектов всех чужих образов, m_w — общее число своих объектов, m_{w^-} — общее число чужих объектов, m_{ws} — число своих, удовлетворяющих высказыванию S, m_{ws^-} — число чужих объектов, удовлетворяющих этому же высказыванию S, а α и β — некоторые пороговые величины в диапазоне от нуля до единицы. Желательно, чтобы высказывание S, используемое для различения своих от чужих, отбирало побольше своих и поменьше чужих объектов, т. е. чтобы α было как можно большим, а β — как можно меньшим. Если условие (α, β) на обучающей выборке удовлетворяется, то высказывание S включается в список «покрывающего набора» закономерностей. Набор закономерностей называется *покрывающим для образа* w, если для любой его реализации выполняется хотя бы одна закономерность из этого набора. Желательно, чтобы число закономерностей в покрывающем наборе было минимальным.

Поиск закономерностей S начинается с больших значений α (например, с $\alpha = 1$) и малых значений β (например, $\beta = 0,02$). Просматриваются все возможные подмножества значений первого случайно выбранного признака и находится высказывание S, удовлетворяющее требованию (α, β). Если таковое не находится, то процесс поиска повторяется при более низком пороге α, устанавливаемом автоматически. Если снижение порога вплоть до величины $\alpha = 0,5$ не дает желаемого результата, тогда увеличивается порог β допустимой доли чужих среди своих. Если условие $(\alpha\beta)$ не выполняется и при $\alpha = \beta = 0,5$, то делается переход к рассмотрению второго признака, случайно выбранного из оставшихся. Если на каком то шаге условие $(\alpha\beta)$ выполняется, то те объекты своего образа w, которые удовлетворяют высказыванию S, из дальнейшего рассмотрения исключаются. Для оставшихся объектов образа w длина высказывания увеличивается на единицу. Описанный процесс продолжается до получения покрывающего набора закономерностей для всех объектов образа w. Аналогично строятся покрывающие наборы и для всех других распознаваемых образов.

Можно потребовать, чтобы алгоритм делал для каждого образа не по одному, а по несколько покрывающих наборов. Это

требование перекликается с высказыванием Р. Фейнмана [155] о том, что мы можем говорить, что понимаем явление, если в состоянии объяснить его несколькими разными способами. С этой целью после получения первого покрытия исключаем из рассмотрения первый признак, включенный в это покрытие, и процесс поиска закономерностей начинается с другого случайно выбираемого признака.

Распознавание контрольного объекта q с помощью покрывающих наборов закономерностей сводится к проверке того, каким высказываниям удовлетворяют его характеристики. Если такое высказывание одно или если их несколько и все они находятся в списке образа w, то объект q распознается в качестве реализации образа w. Если же объект q удовлетворяет закономерностям нескольких образов, то решение принимается в пользу того образа, которому принадлежит закономерность с наибольшим значением величины P_{ws}.

Анализ общего списка закономерностей может показать, что некоторые признаки из исходной системы X в них отсутствуют. Это означает, что они оказались неинформативными и в процессе принятия решений на них можно не обращать внимания. Для каждого i-го образа подмножество информативных признаков может оказаться разным. Это значит, что при проверке гипотезы о принадлежности объекта q к тому или иному образу нужно анализировать не все пространство признаков, а i-е его подпространство, что хорошо согласуется с интуитивными методами неформального распознавания. Действительно, одних людей мы узнаем по росту, других по походке, третьих по фигуре и цвету волос и т. д.

9.2. Алгоритм DW [120]. В основе этого алгоритма также лежит гипотеза проективной компактности. Рассмотрим его работу на примере распознавания двух образов. Элементарным высказыванием J в этом алгоритме называется выражение следующего вида:

а) $X(a) = x$ для признака X, измеренного в номинальной шкале;

б) $X(a) \leq x$ для признака в любой более сильной шкале.

В этих выражениях x — фиксированное (пороговое) значение. Будем называть высказывания $X(a) \neq x$ и $X(a) > x$ дополнительными к вышеприведенным и обозначать через J^-. Конъюнкцией длины l на элементарных высказываниях называем выра-

жение вида $S = J_1 \& J_2 \& \ldots \& J_l$. Набор конъюнкций представим в виде дихотомического дерева D, в котором каждой ветви соответствует одна конъюнкция. Конечные вершины дерева содержат имена объектов обучающей выборки, прошедших сюда от начальной вершины дерева. Если в некоторой конечной вершине имеются объекты разных образов, то считается, что эта вершина принадлежит тому образу, чьих объектов в ней больше.

Работа по проверке истинности или ложности выражений S по отношению к объекту q организуется с помощью таблицы $T(i,j)$. Вершине дерева с номером i (т. е. высказыванию J) сопоставляется i-я строка таблицы, в которой описываются характеристики j этой вершины:

$T(i,1)$ — номер строки, занимаемый данной вершиной;

$T(i,2)$ — номер строки, куда следует переходить, если J истинно;

$T(i,3)$ — номер строки, куда следует переходить, если J ложно;

$T(i,4)$ — номер признака, на котором построено высказывание J,

$T(i,5)$ — порог x, использованный при построении высказывания J;

$T(i,6)$ — количество объектов первого образа в i-й вершине;

$T(i,7)$ — количество объектов второго образа в i-й вершине.

Конечные вершины дерева будут помечены тем, что у них $T(i,4) = 0$. Движение по дереву с помощью таблицы $T(i,j)$ осуществляется следующим образом. Пусть мы находимся в вершине i. Если $T(i,4) \neq 0$, то проверяем, удовлетворяет ли объект q условиям высказывания J, отраженным в $T(i,4)$ и $T(i,5)$. Если да, то идем в вершину $T(i,2)$, если нет — в вершину $T(i,3)$. Если очередная вершина является конечной, то проверяется содержимое ее элементов $T(i,6)$ и $T(i,7)$. Если окажется, что $T(i,6) > T(i,7)$, то объект q будет распознан в качестве реализации первого образа, и наоборот. Если же $T(i,6) = T(i,7)$, то решение принимается в пользу того или иного образа с вероятностью 0,5.

При выборе вариантов пороговых значений x и их сочетаний возникает большой перебор, сокращение которого основано на методе наращивания «лучшего к лучшему» [2]. На первом шаге строится наилучшее в смысле некоторого критерия G элементарное высказывание $J(11)$ и его дополнение $J(12) = J(11^{-})$. С помощью этих высказываний обучающая выборка делится на две группы: группу объектов $U(11)$, удовлетворяющих $J(11)$, и

группу $U(12)$, удовлетворяющих $J(12)$. Если группа $U(11)$ содержит объекты разных образов, то для нее ищется высказывание $J(21)$ и его дополнение $J(22)$, наилучшие с точки зрения критерия G. Та же процедура для группы $U(12)$ дает наилучшее высказывание $J(23)$ и его дополнение $J(24)$. В результате обучающая выборка делится на четыре группы. Процесс такого деления можно представить в виде дерева D, показанного на рис. 22.

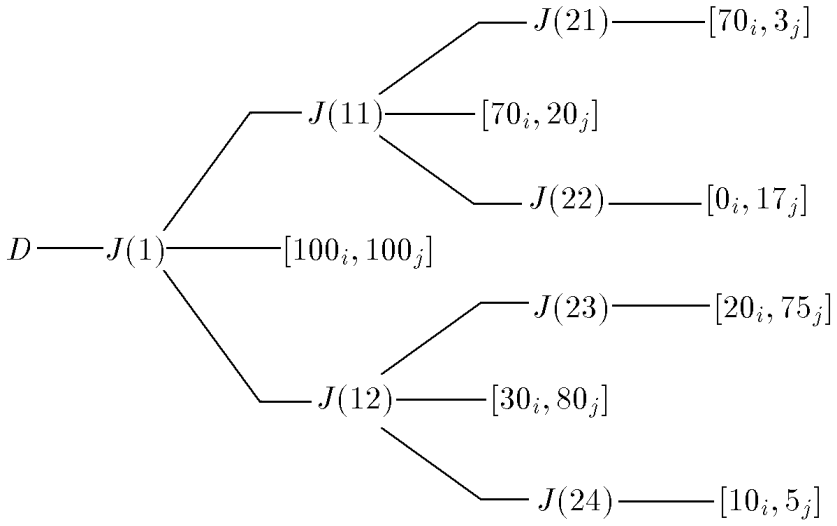


Рис. 22

Вопрос о том, нужно ли дальше продолжать процесс наращивания ветвей, решается проверкой содержимого каждой из полученных групп. Если в группе U обнаружится h_1 объектов первого образа и h_2 объектов второго образа и $\min(h_1, h_2) \geq fm$, где f — положительная константа, меньшая единицы, а m — общее число объектов обучающей выборки, то деление группы U на подгруппы должно быть продолжено. В противном случае группа U образует конечную вершину, и новый объект, попадающий в нее, относится к тому образу, чьих обучающих реализаций больше в этой вершине. Если принять $f = 0,03$, то в нашем примере при $m = 200$ дальнейшее деление требуется делать для вершины $J(23)$.

Логические решающие правила оказались очень эффективным средством решения задач распознавания. Они могут работать с разнотипными признаками. Не страшно, если какой-то

признак у нового объекта не известен: может оказаться, что по имеющимся признакам он удовлетворяет определенным закономерностям и хорошо распознается. Процесс построения ЛРП сложен, но процесс принятия решений по найденным правилам очень прост и может делаться даже вручную.

Большое значение имеет наглядная форма представления ЛРП в виде списка правил типа «если ... то ...». Как показал опыт, некоторые заказчики, получив решение в виде ЛРП, признаются, что оно само по себе им более интересно и полезно, чем возможность использовать его для автоматического распознавания новых объектов: им становятся видны и понятны простые закономерности, которые имеют место в изучаемой ими области.

Наконец, самое важное достоинство ЛРП состоит в том, что формулировка закономерностей в виде конъюнкций соответствует форме представления знаний в интеллектуальных системах. Следовательно, методы поиска ЛРП могут использоваться для автоматического извлечения знаний из данных.

§ 10. Представительность выборки

Простое определение представительности выборки могло бы выглядеть так: «Представительной считается такая обучающая выборка A, которая в заданном пространстве признаков и заданном классе решающих функций позволяет построить правило распознавания новых объектов (контрольной выборки Q) с ошибкой, не превышающей заданной величины».

К сожалению, такое определение логично, но не конструктивно. Обучение делается на имеющемся материале, а проверка качества обучения будет делаться на материале, которого в процессе обучения нет и о котором нет никакой предварительной информации. Так что проверить, хорошо ли она будет распознаваться, не возможно.

Как всегда, в условиях недостатка информации приходится привлекать дополнительные эмпирические гипотезы. В данном случае используется следующее предположение: закономерная связь описывающих признаков X с целевым признаком z на множестве обучающих (A) и контрольных (Q) объектов одна и та же. Эта гипотеза подтверждается, например, при следующих условиях: связь между X и z на генеральной совокупности объектов G подчиняется закономерности типа $C_G^{X,z}$, и выборки A и Q каждая в отдельности хорошо представляют эту генеральную сово-

купность. Тогда на них наблюдается та же закономерность $C_A^{X,z}$ и $C_Q^{X,z}$. Это позволит, обучившись на объектах A, хорошо распознавать и объекты Q.

Прямых доказательств того, выполняются ли эти условия в конкретной реальной задаче, получить нельзя. Можно говорить лишь о косвенных ответах на этот вопрос. Прежде всего, обращается внимание на объем m обучающей выборки A. Считается, что чем больше m, тем больше вероятность того, что закономерность $C_A^{X,z}$ совпадает с закономерностью $C_G^{X,z}$. В общем случае это так, но положиться только на число m не позволяет соображение «независимости» выборки, которое хорошо иллюстрирует следующая притча. Приходит геолог к распознавателю, показывает ему два камня и просит построить решающее правило для распознавания образов двух минералов, представленных этой выборкой. Распознаватель ему говорит: «Коллега, ваша обучающая выборка явно недостаточна». — «А сколько образцов вам нужно?» — «Ну, хотя бы по двадцать штук на каждый признак». — «Пожалуйста», — говорит геолог и через некоторое время приносит нужное количество образцов каждого минерала. «Вот теперь все в порядке! — говорит распознаватель. — А где вы их взяли?» — «Я просто разбил те камни на нужное число частей». — «Нет, так нельзя! — говорит распознаватель. — Нужно, чтобы каждый кусочек был взят независимо от других, был бы частью других более крупных образований». — «А как мне узнать, что кусочки минерала, которые я найду в экспедиции независимо один от другого, не будут на самом деле осколками одного и того же более крупного образования?» Ответа на этот вопрос нет ни у геолога, ни у распознавателя.

При одном и том же числе m представительность выборки как свойство, обеспечивающее хорошее качество обучения, будет разным в зависимости от того, какие части генеральной совокупности она представляет. Если распознается два образа, то не важно, как выглядит генеральная совокупность во всем пространстве признаков. Гораздо важнее, как она выглядит в районе границы между образами. И если есть возможность планировать эксперимент по набору обучающего материала, то нужно организовать его так, чтобы m экспериментов были проведены в критичной зоне пространства, где имеется наибольший риск получить ошибочное решение.

Из приведенных здесь рассуждений можно сделать вывод,

что назвать определенное число обучающих объектов, необходимых и достаточных для успешного обучения в конкретной задаче распознавания, нельзя. Глубокие теоретические исследования этой проблемы делались в предположении, что природа будет играть с распознавателем по самым коварным правилам — предъявлять в обучающей выборке максимально непредставительный материал [24]. Чтобы преодолеть это нежелание природы открыть истинные закономерности, требуются выборки чрезвычайно большого объема, что делает такие оценки неприемлемыми для практического использования.

При другом подходе оценивался требуемый объем выборки для случая, когда стратегия природы не так коварна и проявляется как в легких, так и в сложных ситуациях. При этом предположении исследуются параметры «средних» случаев и для них получаются оценки необходимых объемов обучающей выборки [109]. Они получаются меньшими, чем для самого плохого случая, но за это уменьшение выборки нужно расплачиваться риском столкнуться в реальной задаче со стратегией такой сложности, для которой эта выборка окажется недостаточно представительной.

В условиях такой неопределенности обычно используются различные эвристические приемы для косвенной оценки достаточности обучающей выборки. Если позволяют условия, то процесс обучения делится на два этапа — «предварительного обучения» и «дообучения». На первом этапе строится решающее правило с использованием имеющейся обучающей выборки. Затем система распознавания переводится в режим опытной эксплуатации. Предъявляются объекты контрольной выборки и ведется протокол результатов их распознавания. При появлении ошибки состав обучающей выборки дополняется реализацией, вызвавшей ошибку, и решающее правило корректируется. Так продолжается до тех пор, пока частота появления ошибок не снизится до приемлемого уровня.

В [4, 25] приводятся оценки необходимого объема дообучающей выборки как функции требуемого качества обучения. Считается, что решающая функция задана хорошо, если при случайном выборе контрольных объектов L подряд следующих реализаций не вносит в нее никаких изменений. Если P — вероятность ошибки распознавания после завершения обучения, то при любых положительных ε и δ вероятность $\Pr(P < \varepsilon)$ того, что P не превышает ε, всегда больше величины $(1 - \delta)$, если выполняется

следующее неравенство:

$$L_0 > \ln(\varepsilon\delta)/\ln(1-\varepsilon).$$

Здесь L_0 связано с доверительным числом L показов без исправления решающего правила соотношением $L = L_0 + \varphi$, где φ — число имевшихся ранее исправлений решающей функции.

Для проверки качества обучения часто применяется метод «скользящего экзамена», состоящий в следующем. Часть (t) объектов (иногда $t = 1$) изымается из обучающей выборки, проводится обучение на оставшихся объектах, а изъятые объекты предъявляются для распознавания. Фиксируется число ошибок. Затем эти «контрольные» объекты возвращаются в выборку A, а из нее изымаются другие t объектов, и процедура повторяется до тех пор, пока все m объектов не побывают в роли контрольных. Если сумма полученных ошибок не превышает заданной величины, то считается, что система обучена хорошо.

Глава 6

Выбор системы информативных признаков

§ 1. Постановка задачи

Информативность признаков — понятие относительное. Одна и та же система признаков может быть информативной для решения одной задачи распознавания и не информативной для другой. Так, кандидатов в сборную команду для участия в математической и спортивной Олимпиадах выбирают по разным системам признаков. Оценка информативности признаков зависит от того, что от чего нужно отличать, т. е. от списка распознаваемых образов $S = \langle s_1, s_2, \ldots, s_i, \ldots, s_k \rangle$. Зависит она и от типа решающих функций D. Так что указать типичные, часто используемые признаки не возможно. Для каждой задачи нужно находить свое информативное множество описывающих признаков X.

Первоначальный состав признаков (система X_g) задается неформализованным путем, на основе опыта и интуиции специалиста. Формальные методы применяются к обучающей выборке A для проверки этой исходной системы на достаточность и необходимость. Среди всех B возможных систем признаков достаточной считаем систему, которая при заданных S и D обеспечивает затраты N, не превышающие определенного порога N_0. Под затратами N здесь понимается стоимость измерения признаков (N_x) и стоимость потерь, вызываемых ошибками распознавания (N_r): $N = N_x + N_r$.

Необходимой является достаточная система минимальной сложности (стоимости). Так что фактически на обучающей выборке A решается переборная задача типа

$$\beta = \arg\min_{\beta \in B} N(X_\beta)/S, D, A, N_0.$$

Эта задача одновременной минимизации N_x и N_r впервые была сформулирована в [100]. Затраты на измерения зависят от того, сколько и каких признаков нужно измерять и какое число разрядов требуется для представления результатов измерений. По понятным причинам основное внимание уделяется уменьшению количества измеряемых признаков, т. е. поиску информативной подсистемы из n признаков (X_n) среди g признаков исходной системы (X_g).

§ 2. Критерии информативности признаков

Решающим критерием информативности признаков в задаче распознавания образов является, конечно, величина потерь от ошибок R. Даже если распределения генеральной совокупности известны, вычисление потерь R связано с очень большими затратами машинного времени. В связи с этим делаются попытки найти критерии, более просто вычисляемые и вместе с тем жестко, если не однозначно, коррелированные с оценкой потерь R.

Если распределение реализаций каждого образа подчиняется нормальному закону с диагональными матрицами ковариаций (при этом поверхности равной плотности представляют собой сферы одинакового радиуса), то мерой трудности распознавания D, обратно пропорциональной ожидаемым потерям, может служить среднее значение евклидова расстояния между математическими ожиданиями всех пар образов:

$$D = \left(1/C_k^2\right) \sum_{i,j=1}^{k} \rho(ij),$$

где $\rho(ij)$ — евклидово расстояние между математическими ожиданиями i-го и j-го образов.

В терминах теории информации мерой трудности распознавания служит энтропия H распределений плотности вероятности образов. Пусть распределения k образов спроектированы на

одну ось x, измеряемую с точностью до t градаций (см. рис. 23). Вероятность попадания реализаций i-го образа в j-ю градацию равна $P(j/i)$. Просуммировав для j-й градации вероятности всех k образов, мы получим величину $P_j = \sum_{i=1}^{k} P(j/i)$. Вклад i-го образа в эту сумму $r_i = P(j/i)/P_j$, так что энтропия j-й градации выражается следующим значением:

$$H_j = -(r_1 \log r_1 + r_2 \log r_2 + \cdots + r_i \log r_i + \cdots + r_k \log r_k).$$

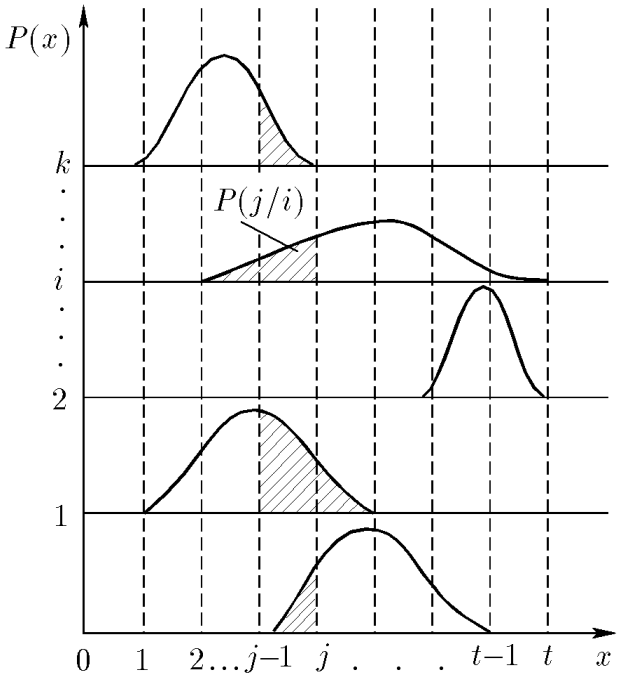


Рис. 23

Из принципа аддитивности энтропии следует, что общая неопределенность при распознавании образов по признаку x имеет вид $H_x = \sum_{j=1}^{t} H_j P_j$. Если исходная неопределенность H_0 ситуации равнялась $\log k$, то количество информации I_x, получаемой в результате измерения признака x, равно $H_0 - H_x$.

Теперь снова вспомним, что в реальных задачах законы распределений реализаций образов нам не известны. Объем обучающей выборки часто бывает небольшим, и делать оценки параметров моделей распределений, а по ним оценки информативности — очень рискованно. В этих условиях целесообразно использовать методы, которые не требуют построения моделей распределения и опираются на конкретные объекты, имеющиеся в обучающей выборке A. По этим прецедентам строится решающая функция (например, правило k ближайших соседей), распознается контрольная последовательность, и по количеству полученных ошибок выносится оценка информативности отдельного признака или их системы.

Возможны и другие способы оценки информативности. Гипотеза компактности дает нам основу для оценки информативности пространства признаков через проявление характеристик компактности. Из нее следует, что для хорошего распознавания образов желательно, чтобы расстояния между своими точками каждого образа были малыми, а расстояния до точек других образов по возможности большими. А если выпуклые оболочки разных образов налагаются друг на друга, то желательно, чтобы они как можно больше отличались по своим размерам. Компактность (плотность) W_i образа i, представленного в обучающей выборке m_i точками $1, 2, \ldots, t, \ldots, l, \ldots, m_i$, можно характеризовать средней длиной ребер соединяющего их полного графа:

$$W_i = \left(1/C^2_{m_i}\right) \sum_{t,l=1}^{m_i} r(t,l).$$

Аналогично, компактность W_j точек $1, 2, \ldots, s, \ldots, v, \ldots, m_j$, представляющих образ j, имеет вид

$$W_j = \left(1/C^2_{m_j}\right) \sum_{s,v=1}^{m_j} r(s,v).$$

Разнесенность образов в пространстве характеристик можно оценивать через среднее расстояние между всеми парами точек из разных образов:

$$W(i,j) = (1/m_i m_j) \sum r(t,s) \quad \text{для} \quad t = 1 \div m_i,\ s = 1 \div m_j.$$

На основании сказанного информативность пространства признаков тем больше, чем больше величина $J = W(i,j)/(W_i + W_j)$.

Оценку информативности признаков можно получить и непосредственно в процессе построения решающего правила в виде дерева дихотомических делений выборки по отдельным признакам. Представим себе, что мы имеем возможность разделить признак X только на две градации: $x \le l$ и $x > l$. Посмотрим состав реализаций, попавших в эти градации. Если в первой градации обнаружится m_{il} реализаций i-го образа и m_{vl} реализаций v-го образа, то неоднородность состава этой градации можно оценить величиной

$$R_1 = \sum_{\substack{i=1 \\ v=i+1}}^{k} m_{il} m_{vl}.$$

Аналогично можно найти неоднородность состава второй градации R_2. Величина $R_l = R_1 + R_2$ характеризует информативность признака x при пороге деления на две градации $x = l$. Меняя порог l, можно найти такое его положение, при котором R_l достигает минимального значения R'. Если исходную неопределенность оценивать через

$$R_0 = \sum_{\substack{i=1 \\ v=i+1}}^{k} m_i m_v,$$

то уменьшение неопределенности после извлечения информации из признака x, т. е. информативность признака x, можно оценить величиной $J_x = (R_0 - R')/R_0$. Если $R' = 0$, то информативность J_x признака будет максимальной и равной единице. Если R' не уменьшило исходной неопределенности, то $J_x = 0$ и признак x естественно считать неинформативным.

Если известно, что признаки не зависят друг от друга, то можно с помощью одного из описанных методов оценить информативность всех g признаков исходной системы и затем выбрать из них n самых информативных. Но в реальных таблицах данных зависимость между признаками наблюдается очень часто. А если признаки зависимы, то при выборе наиболее информативной подсистемы оценками их индивидуальной информативности руководствоваться нельзя.

В табл. 2 приведен пример обучающей выборки двух образов (i и j) в пространстве двух бинарных признаков x_1 и x_2. Проекции реализаций на каждую ось показывают, что оба признака по отдельности абсолютно неинформативны. Использование же этих признаков в системе позволяет найти простое правило для распознавания этих образов: признаки x_1 и x_2 у реализаций i-го образа имеют одинаковые значения, а у j-го образа — разные.

Таблица 2

Пример выборки с зависимыми признаками

	x_1	x_2
i	0	0
	1	1
j	0	1
	1	0

Зависимости могут носить и более сложный характер и проявляться на множестве из более чем двух признаков. Следовательно, для выбора n признаков из g нужно перебрать и испытать на информативность C_g^n их комбинаций. Однажды нам встретилась интересная и важная задача анализа сигналов, одновременно поступающих от 2500 датчиков. Нестационарные помехи искажали эти сигналы, и каждые 10 секунд нужно было отбирать 100 датчиков, наименее пораженных помехами. У нас хватило энтузиазма, чтобы определить, что число сочетаний из $2500 \times 100 \approx 10^{143}$, что на 100 порядков больше числа атомов в видимой части вселенной. А менее экзотические случаи, например, когда нужно выбрать 25 признаков из 50 (для чего нужно просмотреть 10^{15} комбинаций), встречаются регулярно. Поэтому естественно, что о нахождении оптимального решения таких задач речь идти не может. Разрабатываются эвристические алгоритмы направленного перебора, которые за приемлемое время давали бы решения, по возможности близкие к оптимальным.

Рассмотрим некоторые из таких алгоритмов.

§ 3. Метод последовательного сокращения (алгоритм Del) [123]

Пусть задача состоит в том, что из 50 признаков нужно выбрать наиболее информативную систему, состоящую из 25.

Оценим ошибку распознавания при использовании всех 50 признаков (α_0). Затем исключим из системы первый признак и найдем ошибку (α_{11}), которую дают оставшиеся 49 признаков. Поменяем ролями первый и второй признаки и найдем ошибку (α_{12}) в новом 49-мерном пространстве. Эту операцию поочередного исключения одного признака проведем 50 раз. Среди полученных величин $\alpha_{11}, \ldots, \alpha_{1j}, \ldots, \alpha_{50}$ найдем самую малую. Она укажет нам на признак, исключение которого из системы было наименее ощутимым. Исключим этот признак из системы и приступим к испытанию оставшихся 49 признаков. Их поочередное исключение из системы позволит найти самый неинформативный и снизить размерность пространства до 48. Эти процедуры повторяются $(g-n)$ раз, т. е. до тех пор, пока в системе не останется заданное число признаков n.

Количество проверяемых систем признаков при этом методе выражается следующим равенством:

$$L = \{g + (g-1) + (g-2) + \cdots + (n+1)\} = \sum_{j=0}^{g-n} (g-i),$$

что значительно меньше, чем C_g^n. В нашем примере $L = 900$, что на 12 порядков меньше объема полного перебора. В литературе [123] приводятся примеры достаточно хорошего решения задач этим методом.

§ 4. Метод последовательного добавления признаков (алгоритм Add) [2]

Этот алгоритм отличается от предыдущего лишь тем, что порядок проверки подсистем признаков начинается не с g-мерного пространства, а с одномерных пространств. Вначале все g признаков проверяются на информативность. Для этого делается распознавание контрольной последовательности по каждому из g признаков в отдельности и в информативную подсистему включается признак, давший наименьшее число ошибок. Затем к нему по очереди добавляются все $(g-1)$ признаков по одному. Получающиеся двумерные подпространства оцениваются по количеству

ошибок распознавания. Выбирается самая информативная пара признаков. К ней таким же путем подбирается наилучший третий признак из оставшихся $(g-2)$ и так продолжается до получения системы из n признаков.

Трудоемкость этого алгоритма приблизительно такая же, как и алгоритма Del, однако результаты, получаемые алгоритмом Add, обычно лучше, чем у Del. Объясняется этот факт влиянием малой представительности обучающей выборки: при одном и том же объеме выборки чем выше размерность признакового пространства, тем меньше обоснованность получаемых статистических выводов (в нашем случае — оценки информативности). Средняя размерность выборочного пространства в алгоритме Del равна $(g+n)/2$, а в алгоритме Add — $n/2$, так что риск ошибочного признания информативного признака неинформативным в Del выше, чем в Add.

Оба описанных алгоритма дают оптимальное решение на каждом шаге, но это не обеспечивает глобального оптимума. Причину такого явления можно проиллюстрировать примером из психологии малых коллективов: известно, что два самых дружных между собой человека не всегда входят в тройку самых дружных людей.

Для ослабления влияния ошибок на первых шагах алгоритма применяется релаксационный метод. В алгоритме Add набирается некоторое количество (n_1) информативных признаков и затем часть из них $(n_2 < n_1)$ исключается методом Del. После этого алгоритмом Add размерность информативных признаков наращивается на величину n_1 и становится равной $(2n_1 - n_2)$. В этот момент снова включается алгоритм Del, который исключает из системы n_2 «наименее ценных» признаков. Такое чередование алгоритмов Add и Del, которое получило название алгоритма AddDel, продолжается до достижения заданного количества признаков n.

Возможна и обратная стратегия: вначале работает алгоритм Del, после сокращения исходной системы на n_1 признаков включается алгоритм Add, который возвращает в систему n_2 ошибочно исключенных из нее признаков. Повторение этих процедур (алгоритм DelAdd) продолжается до получения системы из n наиболее информативных признаков. Наши эксперименты с этими алгоритмами показали, что алгоритм AddDel приводит к лучшим результатам, чем алгоритмы Add, Del и DelAdd. При этом n_2 бралось равным $n_1/3$.

§ 5. Метод случайного поиска с адаптацией (алгоритм СПА) [82, 108]

Разобьем отрезок (0–1) на g одинаковых участков и сопоставим каждый участок своему признаку: 1-й участок соответствует первому признаку, 2-й — второму и т. д. Каждый участок имеет ширину $1/g$. Запускается датчик случайных чисел с равномерным распределением в диапазоне (0–1). Если число попадает в j-й участок, то j-й признак включается в испытуемый набор признаков. После n шагов работы датчика выбранными оказываются n признаков. Качество этой случайно выбранной подсистемы оценивается по одному из критериев, например по числу получаемых ошибок распознавания α_1.

Описанная процедура случайного выбора n-мерных подсистем признаков и оценки их качества повторяется r раз. Рассмотрение полученного списка оценок $\alpha_1, \alpha_2, \ldots, \alpha_i, \ldots, \alpha_r$ позволяет выбрать наилучшую и наихудшую из подсистем. На этом основании делается процедура «поощрения» и «наказания»: участки, соответствующие признакам, попавшим в наилучшую подсистему, поощряются путем расширения их границ на величину h, а участки, соответствующие признакам из самой неинформативной подсистемы, наказываются тем, что их ширина уменьшается на величину h ($h < 1/g$). Суммарная длина всех участков попрежнему равна единице.

После этого случайным образом выбираются и испытываются r новых подсистем. Но теперь вероятность попадания признаков в эти подсистемы не одинакова: поощренные признаки, представленные более широкими отрезками, имеют больше шансов войти в очередную подсистему, чем наказанные. По результатам испытания этой партии подсистем процедура адаптации (наказания и поощрения) повторяется. Если некоторый признак случайно попадает и в самую лучшую и самую худшую подсистемы, то длина его участка остается неизменной. Если же он регулярно оказывается в составе самой информативной подсистемы, то длина его участка растет с каждым шагом адаптации. Точно так же признак, систематически попадающий в самую неинформативную подсистему, с каждым шагом сокращает длину своего участка и тем самым уменьшает вероятность включения в испытуемые подмножества признаков.

После некоторого количества (R) циклов поиска и адаптации процесс стабилизируется: участки удачливых признаков зани-

мают практически весь отрезок (0–1) и в испытуемую подсистему выбираются одни и те же признаки. Этот факт служит сигналом к окончанию процесса выбора n-мерной подсистемы наиболее информативных признаков.

Скорость сходимости и качество получаемого решения зависят от величины h. При больших значениях h процесс останавливается раньше, но качество полученного решения обычно хуже, чем при малых h. Малые значения h соответствуют более мягкой стратегии поощрений и наказаний. Это иллюстрирует рис. 24. Если исходить из того, что признак, наказываемый на всех R шагах адаптации, все еще сохраняет ненулевое значение ($l_{\min}$) длины своего участка, то величина h должна выбираться из соотношения $(1/g - Rh) \geq l_{\min}$. Практически приемлемые результаты получаются при $r = 10$ и $R = 10 \div 15$.

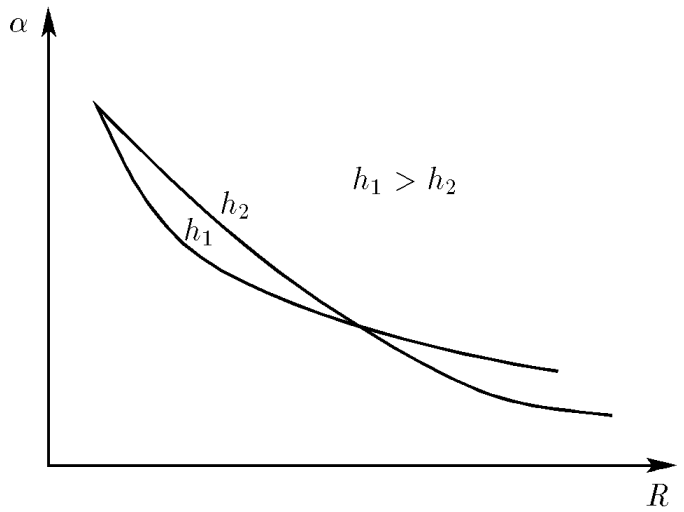


Рис. 24

Первые испытания алгоритма СПА проводились на задаче медицинской диагностики. Три образа были представлены обучающей выборкой из 250 реализаций в 17-мерном пространстве. Требовалось найти наиболее информативное подпространство размерности 3 и 6. Вначале методом полного перебора были найдены оптимальные решения этой задачи. Затем эти же наилучшие решения были найдены методом СПА. Оказалось, что время поиска решения методом СПА меньше времени полного перебора: для $n = 3$ в 5 раз и для $n = 6$ в 40 раз. Этот выигрыш по времени быстро растет с увеличением значений g и n.

В более сложных случаях, где полный перебор был невозможен, качество подсистемы признаков, выбранных методом СПА, сравнивались с качеством исходной системы из g признаков. Как правило, за приемлемое время алгоритм СПА выбирал подсистему, которая по информативности мало уступала исходной системе, что позволяет считать выбранную подсистему близкой к оптимальной. На одних и тех же примерах алгоритм СПА показывает лучшие результаты, чем описанные алгоритмы Del и Add.

§ 6. Направленный таксономический поиск признаков (алгоритм НТПП) [59, 82]

Если мы располагаем методом измерения близости (зависимости) между признаками, то можем сделать таксономию множества g признаков на n таксонов. И если при таксономии объектов в один таксон объединяются наиболее близкие, похожие друг на друга объекты, то в данном случае в таксоны объединяются признаки, одинаково проявляющие себя на объектах обучающей выборки. Так как в один таксон группируются признаки по принципу максимального сходства (зависимости), то между таксонами обеспечивается максимальное несходство (независимость). Выбрав затем по одному типичному признаку из каждого таксона, мы получим n-мерную подсистему, которая включает в свой состав признаки, отличающиеся максимальной независимостью друг от друга. Такая подсистема наилучшим образом соответствует решающему правилу, ориентированному на использование независимых признаков.

В реальных задачах может встретиться случай, когда в исходной системе имеются группы признаков, связанных между собой сильной зависимостью, но не несущих информации, полезной для распознавания заданных образов. Такие признаки объединятся в таксоны, их представители попадут в n-мерную подсистему, будут разрушать компактность образов и служить помехой при принятии решений. Чтобы избежать такой ловушки, в алгоритме НТПП предусмотрен следующий прием. Таксономия g признаков делается на n' таксонов ($g > n' > n$). Выбирается n' признаков и делается перебор $C_{n'}^{n}$ сочетаний из n' по n признаков. Эти сочетания сравниваются между собой по качеству распознавания и выбирается такое сочетание, которое приводит к наименьшей величине ошибок. Оно и принимается за наиболее информативную n-мерную подсистему признаков.

Глава 7

Заполнение пробелов и обнаружение ошибок в эмпирических таблицах

§ 1. Обзор работ по проблеме заполнения пробелов [93]

Реальные таблицы данных часто содержат пробелы: у некоторых объектов a_i значение того или иного признака x_j может отсутствовать. В результате на вход программ анализа данных подается таблица с одним или несколькими пустыми клеточками. Большинство известных методов анализа данных не рассчитано на обработку «некомплектных» таблиц, в связи с чем стали делаться попытки решать задачи заполнения содержащихся в них пробелов.

Решению этих задач посвящено большое число работ. Самые первые из них появились еще в докомпьютерное время (до 1960 года) и, начиная с классической работы С. Уилкса [150], носили в основном теоретический характер и были посвящены большей частью оценкам максимального правдоподобия (МП-оценкам) по некомплектным выборкам. На практике же в это время использовались примитивные способы борьбы с пробелами. Так, в одной из первых работ на эту тему [144] дается рекомендация при анализе табличных данных удалять те строки и те столбцы, в которых имеется хотя бы один пробел. Однако в практике встречаются таблицы, содержащие по несколько пробелов в каждой строке и в каждом столбце, и такие таблицы перед обработкой следовало

бы вычеркивать полностью и по несколько раз. Если это не желательно, то рекомендуется заполнять пробелы средними значениями величин, имеющихся в данном столбце. Очевидно, что это самый простой, но не самый точный метод заполнения пробелов. Полный обзор работ этого периода можно найти в [7].

С распространением ЭВМ были предложены более сложные машинные алгоритмы, основанные на методе наименьших квадратов: регрессионный метод [9, 151], метод главных компонент [40], пошаговая регрессия [157], метод многомерной линейной экстраполяции [137], метод прогностических переменных [61]. Учитывая тот факт, что оценки первых двух моментов полностью определяют оценки регрессии, многие авторы сосредоточились на проблеме оценивания ковариационной матрицы по данным с отсутствующими значениями [39, 160, 167].

Со второй половины 70-х годов особых успехов добилось направление, связанное с оценками максимального правдоподобия (МП-оценками), особенно в рамках нормальных распределений. Появились практические алгоритмы, вычисляющие МП-оценки пробелов, например [16, 148, 159]. В работе [46] предложена мощная вычислительная процедура: EM-алгоритм для решения общей задачи оценивания параметров в условиях некомплектной выборки. К настоящему времени эти методы интенсивно развиваются, созданы эффективные робастные варианты EM-алгоритма [114]. Возобладала тенденция поиска для всех классических статистических методов аналогов, способных работать с некомплектными данными, не заполняя пробелов [115, 161, 164]. Более полный обзор теории и практики содержится в монографиях [52, 113].

Методы, упомянутые выше, действуют глобально: в них предполагается, что зависимость заданного (например, линейного) типа реализована на всех объектах, поэтому и в оценивании зависимостей участвуют все строки и столбцы. Локальные алгоритмы, оценивающие зависимости по некомплектной выборке в некоторой окрестности предсказываемого объекта, были впервые предложены в работах [83, 85]. Постановку задачи предсказания значений пропущенных элементов можно пояснить на примере обработки таблицы размером $m \times n$, не содержащей пробелов.

Пусть в нашем распоряжении имеется набор различных стратегий (алгоритмов) $F = \langle f_1, f_2, \ldots, f_v, \ldots, f_t \rangle$, предназначенных для предсказания значений пропущенных элементов. Закроем в таблице известный элемент b_{11}, стоящий на пересечении строки

a_1 и столбца x_1, и предскажем его с помощью всех алгоритмов F поочередно. Каждый алгоритм f_v предскажет свое значение b_{11v}, которое будет отличаться от исходного («истинного») значения на величину $d_{11v} = |b_{11v} - b_{11}|$.

Восстановим в таблице элемент b_{11}, уберем элемент b_{12} и повторим процедуру. Получим отклонения d_{12v}. Проделав это по очереди со всеми элементами таблицы и просуммировав полученные отклонения, мы получим суммарную величину отклонений D_v для каждого алгоритма v. Наилучшим из них естественно считать такой алгоритм f'_v, который дает минимальную сумму отклонений:

$$v' = \arg\min_{v \in t} D_v.$$

Алгоритмы из набора F могут отличаться друг от друга лежащими в их основании эвристическими предположениями (гипотезами). Ниже описаны некоторые из этих гипотез и основанные на них алгоритмы двух семейств — ZET и WANGA.

§ 2. Базовый алгоритм ZET заполнения пробелов

В основе алгоритма ZET [72, 83] лежат три предположения. Первое (гипотеза избыточности) состоит в том, что реальные таблицы имеют избыточность, проявляющуюся в наличии похожих между собой объектов (строк) и зависящих друг от друга свойств (столбцов). Если же избыточность отсутствует (как, например, в таблице случайных чисел), то предпочесть один прогноз другому не возможно.

Второе предположение (гипотеза локальной компактности) состоит в утверждении, что для предсказания пропущенного элемента b_{ij} нужно использовать не всю таблицу, а лишь ее «компетентную» часть, состоящую из элементов строк, похожих на строку i, и элементов столбцов, похожих на столбец j. Остальные строки и столбцы для данного элемента неинформативны. Их использование лишь разрушало бы локальную компактность подмножества компетентных элементов и ухудшало точность предсказания.

Третье предположение (гипотеза линейных зависимостей) заключается в том, что из всех возможных видов зависимостей между столбцами (строками) в алгоритме ZET используются только

линейные зависимости. Если зависимости носят более сложный характер, то для их надежного обнаружения требуется такой большой объем данных, который в реальных задачах встречается нечасто. В работе алгоритма ZET можно выделить три этапа.

1. На первом этапе для данного пробела из исходной матрицы «объект-свойство», столбцы которой нормированы по дисперсии, выбирается подмножество компетентных строк и затем для этих строк — компетентных столбцов.

2. На втором этапе автоматически подбираются параметры в формуле, используемой для предсказания пропущенного элемента, при которых ожидаемая ошибка предсказания достигает минимума.

3. На третьем этапе выполняется непосредственно прогнозирование элемента по этой формуле.

Под компетентностью l-й строки по отношению к i-й понимается величина

$$L_{il} = r_{il} t_{il}.$$

Здесь $r_{il} = 1 - \rho_{il}$, ρ_{il} — евклидово расстояние между i-й и l-й строками, а t_{il} — коэффициент комплектности, равный числу свойств, значения которых известны как для i-й, так и для l-й строки. Компетентная строка не должна иметь пробела в j-м столбце.

Под компетентностью k-го столбца по отношению к j-му столбцу понимается величина

$$L_{jk} = r_{jk} t_{jk},$$

где r_{jk} — модуль коэффициента корреляции между j-м и k-м столбцами, а t_{jk} — коэффициент комплектности, равный числу объектов, у которых известны как j-е, так и k-е свойства. Компетентный столбец не должен иметь пробела в i-й строке.

По указанию пользователя программа выбирает компетентную подматрицу любого размера в пределах от 2×2 до $n \times m$. Обычно используется подматрица, содержащая от 3 до 7 строк и столбцов.

В процессе предсказания значения пробела с использованием зависимостей между j-м и всеми остальными (k-ми) столбцами вырабатываются «подсказки» b_k. Для их получения используется уравнение линейной регрессии между j-м и k-м столбцами (см. рис. 25). Если в подматрице было $(q+1)$ столбцов, то затем q подсказок усредняются с весом, пропорциональным компетентности соответствующего столбца. В итоге получается прогнозная

величина b_q, порожденная избыточностью, содержащейся в столбцах:

$$b_j = \sum_{k=1}^{q} b_k L_{jk}^{\alpha} \Big/ \sum_{k=1}^{q} L_{jk}^{\alpha} \ldots . \qquad (1)$$

Здесь α — коэффициент, регулирующий влияние компетентности на результат предсказания. При малых значениях α разница в компетентности сказывается мало, при больших α более компетентные столбцы влияют гораздо больше других. Выбор α и составляет суть этапа подбора формулы для прогнозирования: все известные элементы j-го столбца предсказываются при разных значениях α и затем выбирается такое значение α, при котором ошибка прогноза δ_j была минимальной.

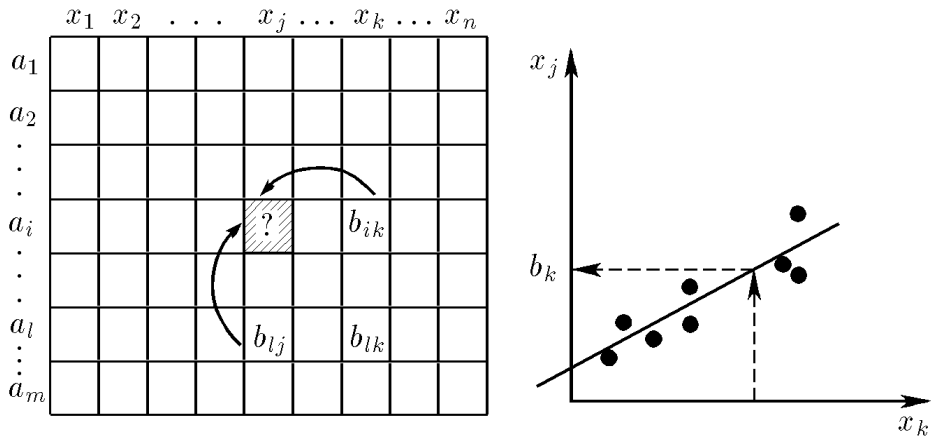


Рис. 25

По формуле (1) с выбранным значением α делается прогноз b_j величины пропущенного элемента, а полученная при выборе α минимальная величина δ_j в дальнейшем принимается в качестве оценки ожидаемой ошибки заполнения пробела по столбцам.

Процедура заполнения пробела с использованием связи между i-й строкой и всеми s другими (l-ми) строками $(1, 2, \ldots, l, \ldots, s)$ аналогична вышеописанной и выполняется по формуле

$$b_i = \sum_{l=1}^{s} b_l L_{il}^{\alpha} \Big/ \sum_{l=1}^{s} L_{il}^{\alpha} \ldots . \qquad (2)$$

Для выбора α здесь используются все известные элементы i-й строки, и выбор делается при минимальном значении ошибки δ_i их прогнозирования.

Общий прогноз b'_{ij} значения пропущенного элемента b_{ij} получается выбором либо прогноза b_j, если $\delta_j < \delta_i$, либо прогноза b_i, если $\delta_i < \delta_j$. Возможно и их усреднение с весом, обратно пропорциональным величине ожидаемой ошибки:

$$b'_{ij} = [b_j/(\varepsilon + \delta_j) + b_i/(\varepsilon + \delta_i)][(\varepsilon + \delta_j)(\varepsilon + \delta_j)]/[2\varepsilon + \delta_j + \delta_i]. \quad (3)$$

Здесь ε — константа, например, равная $0,01$, введенная для предотвращения деления на нуль.

Как отмечалось, оценка ожидаемой ошибки заполнения пробела (отклонения предсказанного значения от истинного) может быть получена в процессе подбора коэффициента α. О величине ожидаемой ошибки d_{ij} можно судить по ошибкам δ_j и δ_i предсказаний известных элементов i-й строки и j-го столбца при наилучшем значении α. Эксперименты показывают, что корреляция между средним значением этих ошибок $\delta^* = [\delta_j + \delta_i]/2$ и ошибкой d_{ij} всегда положительна.

Второй способ определения ожидаемой ошибки основан на оценке дисперсии «подсказок». Вычисляется дисперсия (dis) величин подсказок b_k и b_l, получаемых от всех k столбцов и l строк компетентной подматрицы. Большая дисперсия указывает на отсутствие устойчивой закономерной связи между элементом (ij) и другими элементами подматрицы, т. е. на отсутствие их компактности. Ясно, что в этих условиях рассчитывать на высокую точность предсказания величины b_{ij} не приходится. Эксперименты показали, что коэффициент корреляции между дисперсией dis и ошибкой предсказания d_{ij} достигает величины $+0,7$. Прогнозы ожидаемой ошибки заполнения по дисперсионному критерию оказались более надежными, чем по критерию, основанному на оценках ошибок δ.

Для различных прикладных задач были сделаны многочисленные модификации описанного выше базового алгоритма ZET, отличающиеся своим назначением и наборами разных режимов работы. Программы заполнения пробелов могут работать в одном из следующих режимов:

1. Заполнение всех пробелов.
2. Заполнение только тех пробелов, ожидаемая ошибка для которых не превышает заданной величины.

3. Заполнение пробелов только на базе информации, имеющейся в исходной таблице.
4. Заполнение каждого следующего пробела с использованием исходной информации и прогнозных значений ранее заполненных пробелов.

Для каждого из этих вариантов имеется несколько режимов выдачи промежуточных и окончательных результатов на печать.

§ 3. Некоторые варианты алгоритма ZET

3.1. Обнаружение грубых ошибок (алгоритм ZET–R). Алгоритм ZET–R используется для обнаружения грубых ошибок в исходной таблице данных (так называемый режим *редактирования таблиц*). Для этого программа по очереди предсказывает все элементы таблицы и сравнивает результаты предсказания с фактически имеющимися данными. Если предсказанное значение совпадает с исходным или мало отличается от него, то это означает, что элемент хорошо согласуется с закономерностями данной части таблицы данных. Если же обнаруживается большое расхождение, то выдается сигнал о необходимости проверки данного элемента. Если он отражает уникальный факт, выпадающий из общей закономерности, то его истинность нужно подтвердить. Если же он отражает ошибку, то ее нужно устранить. Таким путем удается обнаруживать грубые ошибки или умышленные искажения отдельных элементов таблицы данных.

3.2. Прогнозирование динамических рядов (алгоритм ZET–D). Алгоритм ZET–D работает с таблицами типа «время-свойство». На рисунке 1 такая таблица обозначена символами (ТВС/О); здесь более подробно представлена в табл. 3, *а*. T строк в этой таблице отражают значение n свойств $x_1, x_2, \ldots, x_j, \ldots, x_n$ некоторого объекта или процесса в последовательные моменты времени $t_1, t_2, \ldots, t_\tau, \ldots, t_T$. Таблицу можно переформировать, объединив в одну строку k соседних по времени строк. Первая строка такой новой таблицы отражает данные из k первых строк исходной таблицы $(t_1, t_2, \ldots, t_{k-1}, t)k$):

$$
\begin{array}{l}
b_{11}, b_{12}, \ldots, b_{1j}, \ldots, b_{1n}, \\
b_{21}, b_{22}, \ldots, b_{2j}, \ldots, b_{2n}, \\
\ldots\ldots\ldots\ldots\ldots\ldots \\
b_{k1}, b_{k2}, \ldots, b_{kj}, \ldots, b_{kn}.
\end{array}
$$

Во вторую строку новой таблицы поместим k строк, начинающихся с момента времени t_2, в третью — с момента времени t_3 и т. д. до строки, начинающейся с момента t_{T-k+2}. В результате получим таблицу, состоящую из kn столбцов и $T-k+2$ строк (см. табл. 3, *б*). Если строка t_τ соответствовала, например, свойствам объекта в τ-й год, то каждая строка новой таблицы будет соответствовать периоду в k лет.

Таблица 3

Преобразование таблицы методом «змейки»

	x_1	x_2	x_j	x_n
t_1	b_{11}	b_{12}	b_{1j}	b_{1n}
t_2	b_{21}	b_{22}	b_{2j}	b_{2n}
...	...	...	...	...
t_τ	$b_{\tau 1}$	$b_{\tau 2}$	$b_{\tau j}$	$b_{\tau n}$
...	...	...	...	...
t_T	b_{T1}	b_{T2}	b_{Tj}	b_{Tn}

а

	1	2	$k-1$	k
1	t_1	t_2	t_{k-1}	t_k
2	t_2	t_3	t_k	t_{k+1}
...	...	...	...	...
τ	t_τ	$t_{\tau+1}$	$t_{\tau+k-2}$	$t_{\tau+k-1}$
...	...	...	...	...
$T-k+2$	t_{T-k+2}	t_{T-k+1}	t_T	t_{T+1}

б

Все элементы новой таблицы известны, кроме элементов последнего сегмента t_{T+1}, в котором должны быть отражены свойства изучаемого объекта или процесса в момент времени $T+1$, следующий за последним моментом из отраженных в протоколе наблюдения.

Если каждую пустую j-ю клеточку последнего сегмента заполнить алгоритмом ZET, то получим прогноз свойств x_j в момент времени $t=T+1$. Описанный способ формирования длинных строк из сдвигаемых коротких и последующего прогнозирования элементов короткой строки в одной из недавних работ был назван методом «змейки».

В [56] описано несколько вариантов этого алгоритма для исходных таблиц разного характера. Есть вариант (алгоритм ZETMC), ориентированный на таблицы с фиксированным порядком следования свойств x_j. Примером такой таблицы может служить сводка ежемесячных показателей деятельности предприятия за T лет. Здесь роль свойств играют показатели в j-е месяцы, а τ-я строка — это данные за τ-й год. Прогнозирование делается не для всех месяцев года сразу, а последовательно для каждого

следующего месяца. Начало годового цикла — вещь условная, цикл можно начинать с любого месяца. Пусть таблица содержит данные за период с 1970 по 1995 годы. Возьмем первый столбец (данные за январи) и поставим его за последним столбцом (за декабрями). Если его сдвинуть на одну строку вверх, то в первой строке окажутся данные за год, начинающийся в феврале 1970-го и заканчивающийся в январе 1971-го года. В последней строке будет цикл, который начинается в феврале 1995-го и заканчивается январем 1996-го года. Данные за январь 1996-го года нам не известны, и эту пустую клеточку таблицы мы заполняем с помощью алгоритма ZET.

Затем мы можем перенести с первой позиции на последнюю столбец с данными за февраль. Годовые циклы будут начинаться с марта текущего года и заканчиваться в феврале следующего года. Заполнив новую пустую клеточку, мы предскажем отсутствующее значение февраля 1996-го года. Эту процедуру поочередного переноса первых столбцов на последнее место и прогнозирования очередного неизвестного значения можно продолжать сколь угодно долго.

Однако ясно, что с удалением прогнозируемого момента времени от момента последнего наблюдения точность прогноза будет падать, причем скорость нарастания ошибок зависит от характера наблюдаемого процесса и заранее предсказана быть не может. Для каждой конкретной таблицы рекомендуется метод ретроспективного анализа: на прошлом материале делаются прогнозы известных данных и фиксируется зависимость ошибок прогноза от длительности периодов упреждения. В результате можно предположительно говорить об ожидаемой ошибке прогноза при заданном периоде упреждения или о максимальном периоде упреждения при заданной допустимой величине ошибки прогноза.

Возможен и другой подход — оценивать ожидаемую ошибку по дисперсии подсказок, получаемых в процессе работы алгоритма ZET, как это описано в § 2 настоящей главы.

§ 4. Примеры применения алгоритмов семейства ZET

4.1. Применение в экономике. Много различных задач было решено в свое время по заданию существовавшего тогда Госплана Российской Федерации [56]. В одной из задач требовалось отредактировать таблицу (найти ошибки), в которой были

отражены характеристики зернового производства в областях и краях Российской Федерации. Описывающие свойства отражали данные о посевных площадях, количестве удобрений, тракторов, комбайнов и т. д. Целевые характеристики касались урожайности зерновых, валового сбора зерна и пр. Было выяснено, что некоторые характеристики не связаны с остальными, и это отражалось в больших погрешностях при их восстановлении. Так, например, обнаружился на первый взгляд странный факт, что количество удобрений не влияет на урожайность зерновых культур. Объяснение этому факту состояло в том, что удобрения использовались лишь при производстве овощей, а на зерновые их не хватало.

Были обнаружены и большие отклонения от общей закономерности для отдельных элементов таблицы. Некоторые из этих отклонений оказались работникам Госплана вполне понятными: «Мы давно подозревали, что ... область нас обманывает по этому показателю». Так это или нет — мы не знали и на этом эпизоде лишний раз убедились, что часто пользователь склонен переоценивать достоверность машинных решений. Нужно постоянно подчеркивать необходимость тщательной содержательной проверки получаемых программой решений, не воспринимать их в качестве бесспорной истины, а пытаться дополнить машинные ответы на вопрос «Что?» человеческими ответами на вопросы «Как?» и «Почему?».

Одна из таблиц содержала ежемесячные сведения о средних надоях молока в республике за период с 1946 по 1982 годы. Требовалось обучить программу делать прогнозы надоев на один год вперед. Использовалась описанная выше программа ZETMC. Выяснилось, что обнаруживаемые закономерности (похожести строк и похожести столбцов) позволяли получать такие прогнозы с достаточно высокой надежностью: ошибка годового прогноза не превышала 1,5 %. Практически прогноз вырабатывался на 14 месяцев вперед. Нам присылали данные о надоях с января по октябрь текущего года, и мы высылали в Госплан прогнозы на конец данного года и на все месяцы будущего года. В течение всего этого периода работники Госплана сообщали нам фактические данные за каждый месяц. Мы делали прогноз на оставшиеся месяцы прогнозируемого периода. Опыт показал, что такой метод скользящего уточняющего прогнозирования является наиболее адекватным для информационной поддержки процессов выработки управляющих решений.

4.2. Применение в геологии и медицине. Таблицы данных, которые содержат информацию, собранную несколькими разными геологическими экспедициями, обычно содержат большое количество пробелов. Экспедиции имели неодинаковый набор измерительной аппаратуры, какой-то прибор вышел из строя во время работы, какие-то данные оказались утерянными и т. д. И для того чтобы применить привычные методы анализа, нужно сначала попытаться заполнить пропущенные элементы такой сводной таблицы.

Аналогичная ситуация типична и для медицинских данных, полученных путем сведéния в одну таблицу сведений из историй болезни различных пациентов. При разных посещениях даже одного и того же врача фиксировались разные симптомы. Еще большее различие возникает при использовании документов от другого врача или другой поликлиники. Как правило, эти таблицы содержат не менее 30 % пробелов.

На таблицах такого рода отрабатывалась стратегия заполнения большого числа пробелов. Начальные условия для заполнения различных пробелов не одинаковы. Для некоторых пробелов удается выбрать компетентную подматрицу с высокой компетентностью строк и столбцов. Для других же этого сделать не удается, они оказываются менее обусловленными. Рекомендуемая стратегия состоит в том, что сначала нужно заполнить пробел с наилучшей обусловленностью. Затем, опираясь на все элементы, в том числе и на только что заполненный, найти самый обусловленный пробел из оставшихся. И такой процесс заполнения самого обусловленного элемента на каждом шаге продолжается до заполнения всей таблицы. На каждом шаге программа выдает информацию об ожидаемой ошибке прогнозирования значения заполняемого элемента. Процесс может быть остановлен при выходе в область, для которой ожидаемая ошибка превышает заданный порог.

Встретился нам и такой экзотический случай, когда в таблице было пропущено 82 % клеток. Вместе с тем материал для геологов был очень ценным, и нами была предпринята попытка заполнить пробелы в этой таблице. Имевшихся 18 % клеток хватило для заполнения всего нескольких пробелов. Для остальных пробелов нельзя было найти ни одной компетентной строки и ни одного компетентного столбца.

4.3. Применения в технике. В одной из задач данные

отражали известные характеристики телевизионных приемников различного типа. Фирмы, производящие телевизоры, указывают их технические характеристики, но наборы этих характеристик оказываются не полностью совпадающими. Сведе́ние таких данных в одну таблицу выявляет в ней пустые клеточки, которые было бы интересно заполнить, чтобы узнать некоторые характеристики, о которых изготовитель умалчивает. Выяснилось, что ряд свойств приемников связан с другими свойствами сильной зависимостью и предсказания таких хорошо обусловленных пробелов обычно подтверждаются. Вместе с тем обнаружились некоторые свойства, которые не зависят от значения других свойств и потому не могут быть хорошо предсказанными. Примером такого свойства может служить материал корпуса (металл, дерево или пластик), который не зависит от размера экрана, частоты развертки и т. д.

Многолетний опыт использования алгоритмов семейства ZET показал их высокую эффективность по сравнению с другими известными алгоритмами заполнения пробелов, редактирования таблиц и прогнозирования характеристик динамических (меняющихся во времени или пространстве) объектов.

Вместе с тем, по мере накопления опыта решения реальных задач возникли идеи дальнейшего совершенствования алгоритмов такого назначения. В процессе исследований изучается влияние различных способов нормировки столбцов, сравниваются разные стратегии выбора компетентных подматриц и различные способы прогнозирования пробелов по компетентным подматрицам. Делается также попытка создать алгоритм и приемлемую по машинному времени и памяти программу заполнения пробелов в так называемых трехвходовых таблицах или кубах данных типа «объект-свойство-время».

§ 5. Алгоритмы семейства WANGA

Алгоритмы семейства WANGA [85], как и семейства ZET, основаны на гипотезе локальной компактности, но предназначены для заполнения пробелов в таблицах с разнотипными переменными. Начнем с описания алгоритмов для таблицы, все n признаков в которой измерены в одной и той же шкале отношений.

5.1. Алгоритм WANGA–R. Пусть A — таблица с пропущенным элементом b_{ij}. Все другие элементы таблицы известны.

Для выбора компетентной подтаблицы размером в s строк и q столбцов воспользуемся следующей процедурой. Выберем элемент b_{lk}. На пересечениях i-й и l-й строк с j-м и k-м столбцами находятся четыре элемента таблицы: b_{lk}, b_{lj}, b_{ik} и неизвестный элемент $a(ij)$. Если признаки связаны сильной прямой зависимостью, то отношение двух элементов k-го столбца будет таким же или почти таким, как и отношение двух элементов j-го столбца. Тогда из предполагаемого равенства отношений $b_{lk}/b_{ik} = b_{lj}/b_{ij}$ можно получить вариант b' оценки («подсказки») заполняемого элемента: $b'_{lk} = b_{lj} \times b_{ik}/b_{lk}$. Повторив эту процедуру для всех других элементов таблицы, мы получим $(n-1) \times (m-1)$ вариантов подсказок:

$$b'_{11}, b'_{12}, \ldots, b'_{lk}, \ldots, b_{(n-1)(m-1)}.$$

Выделим из них подсказки, полученные с участием элементов l-й строки, и найдем их дисперсию d_l. Величину $L_l = 1/(d_l + 1)$ примем в качестве меры компетентности l-й строки. L_l достигает максимального значения единицы, если дисперсия d_l равна нулю, и убывает с ростом дисперсии, оставаясь положительной величиной. Аналогично по дисперсии d_k подсказок с участием всех элементов k-го столбца найдем его компетентность $L_k = 1/(d_{k+1})$. Сформируем компетентную подтаблицу A', включив в нее s самых компетентных строк и q самых компетентных столбцов.

Процесс заполнения пробела алгоритмом WANGA–R состоит в следующем. Присоединяем к таблице A' элементы j-го столбца и i-й строки. Перебираем все четверки элементов, которые находятся на пересечении двух столбцов: j-го и k-го и двух строк: i-й и l-й. Неизвестный элемент b_{ij}, входящий в состав всех этих четверок, вычисляем по описанному выше способу и получаем $(s \times q)$ вариантов подсказок: $b'_{11}, b'_{21}, \ldots, b'_{s1}, b'_{12}, b'_{22}, \ldots, b'_{lk}, \ldots, b'_{sq}$. Окончательное решение о значении пропущенного элемента получаем в виде средневзвешенной суммы подсказок:

$$b''_{ij} = \sum b'_{lk} L_{lk} \Big/ \sum L_{lk}$$

для всех $l = 1 \div s$ и $k = 1 \div q$. Здесь весовой коэффициент подсказки от элемента (lk) равен его компетентности: $L_{lk} = L_i \times L_k$. Степень доверия P к полученному решению можно оценить через величину дисперсии D всех $(s \times q)$ подсказок: $P = 1/(D + 1)$.

5.2. Алгоритм WANGA–I. Если данные в таблице A замерены в шкале интервалов, то минимальным подсказывающим элементом в алгоритме WANGA–I будет подматрица, состоящая из шести элементов, стоящих на пересечениях двух столбцов (j и k) и трех строк (i-, l- и t-й). Инвариантом шкалы интервалов является отношение разностей двух любых пар элементов одного и того же столбца. Основываясь на этом и на гипотезе о прямой связи между j-м и k-м столбцами, можно записать:

$$[b_{ij} - b_{lj}]/[b_{lj} - b_{tj}] = [b_{tj} - b_{lk}]/[b_{lk} - b_{tk}].$$

Отсюда получаем вариант подсказки пропущенного элемента:

$$b''_{klt} = b_{lj} + [(b_{lj} - b_{tj}) \times (b_{ik} - b_{lk})/(b_{lk} - b_{tk})].$$

Повторив эти операции с участием всех парных сочетаний из $(m-1)$ строк и всех $(n-1)$ столбцов, получим G подсказок, где

$$G = C^2_{m-1} \times (n-1).$$

Выделим подсказки, полученные с участием элементов l-й строки (их будет $(m-2) \times (n-1)$ штук), определим их дисперсию d_l и по ней — компетентность l-й строки $L_l = 1/(d_l + 1)$. Основываясь на таких оценках, выберем s наиболее компетентных строк.

Аналогичным способом найдем и q наиболее компетентных столбцов, сформировав в итоге компетентную подматрицу $s \times q$. Компетентность каждого элемента этой подматрицы $L_{lk} = L_l \times L_k$. Описанным выше методом найдем подсказки от элементов этой подматрицы и получим окончательный вариант заполнения пропущенного элемента, усреднив подсказки с их весами L_{lk}. О доверии к этому результату можно судить по величине $P = 1/(D+1)$, где D — дисперсия всех подсказок от компетентной подматрицы.

5.3. Алгоритм WANGA–0. Если все признаки в таблице A измерены в шкале порядка, то пробелы в ней заполняются алгоритмом WANGA–0. Преобразуем все столбцы, приведя их значения к шкале нормированных рангов. Инвариантным к преобразованиям шкалы порядка являются суждения такого рода: если $b_{lk} > b_{ik}$, то и $b_{lj} > b_{ij}$. Отсюда получается один из трех возможных вариантов подсказки: $b'_{ij} > b_{lj}$, если $b_{ik} > b_{lk}$; $a'_{ij} = b_{lj}$, если $b_{ik} = b_{lk}$, и $b'_{ij} < b_{lj}$, если $b_{ik} < b_{lk}$.

Использование всех элементов столбцов j и k даст $(m-1)$ вариант подсказок, которые могут не совпадать или даже противоречить друг другу. Оценивать общий результат будем по следующему правилу. Поставим в соответствие m ранговым номерам m накопителей $1, 2, \ldots, v, \ldots, m$. Если подсказка говорит, что неизвестный элемент равен v, то добавим единицу в v-й накопитель. Если подсказка говорит, что искомый ранг больше v, то в каждый накопитель с номером, бóльшим v, добавим величину, равную $1/(m-v)$. Если же подсказка $b'_{ij} < v$, то в ячейки от первой до $(v-1)$-й добавим величину, равную $1/(v-1)$. В итоге в каждой ячейке накопится величина, отражающая «число голосов» за принадлежность предсказываемого значения к тому или иному рангу. После нормировки суммы всех «голосов» к единице неопределенность подсказок можно оценить по энтропии

$$H = -\sum_{v=1}^{m} p_v \log p_v,$$

где p_v — доля голосов за ранг v. Компетентность L_k k-го столбца примем равной $1/(H+1)$. При «единодушном» голосовании за один и тот же ранг положительная величина L_k достигает максимального значения, равного единице.

Если в процессе работы со всеми столбцами накапливать подсказки, получавшиеся при участии элементов l-й строки, то описанным выше путем можно получить оценку ее компетентности L_l. Пользуясь этими оценками, можно выбрать компетентную подтаблицу A', содержащую s строк и q столбцов. Компетентность L_{lk} каждого элемента b_{lk} этой подтаблицы примем равной $L_i \times L_k$.

Повторим определение подсказок с участием всех $s \times q$ элементов из A'. Теперь для получения общего результата в накопители добавляем соответствующие доли не от единицы, а от величины L_{lk}. Распределение набранных рангами голосов позволяет найти окончательное решение: пропущенному значению присваивается ранг, набравший наибольшее количество голосов. Энтропия полученного распределения голосов H позволит нам оценить степень доверия P к этому результату: $P = 1/(H+1)$.

5.4. Алгоритм WANGA–N. Рассмотрим таблицу A, признаки в которой измерены в шкале наименований. Имя пропущенного элемента b_{ij} может быть одним из d имен $(1, 2, \ldots, f, \ldots, d)$,

содержащихся в j-м столбце, либо новым $(d+1)$-м именем x. Просмотрим все подсказывающие четверки элементов, стоящие на пересечении строк i и l и столбцов j и k. Подсказку будем искать, исходя из предположения: если i-й и l-й элементы k-го столбца названы одним и тем же именем, то и в j-м столбце элементы i-й и l-й строк должны иметь одинаковые имена, т. е. если $b_{ik} = b_{lk}$, то $b'_{ij} = b_{lj}$. И наоборот, если $b_{ik} \neq b_{lk}$, то $b'_{ij} \neq b_{lj}$. Найдем подсказки от всех $(m-1) \times (n-1)$ таких четверок и выберем подсказки, полученные с участием элементов k-го столбца. Подсчитаем среди них число подсказок, голосовавших за каждое из d имен $\left(w_f^+\right)$ и против каждого из них $\left(w_f^-\right)$. Вычтем голоса «против» из голосов «за»: $w_f = 1(w_f^+) - (w_f^-)$. Добавив к полученным результатам величину $w'_{f\,\min}$, получим d неотрицательных чисел w'_f. Умножим их на нормирующий коэффициент g такой, что $\sum w'_f \times g = 1$ при $f \div 1 - d$. Теперь можно найти энтропию значений H_k и через нее компетентность k-го столбца: $L_k = 1/(H_k + 1)$.

Собрав подсказки, полученные с участием всех элементов l-й строки, мы аналогичным способом найдем ее компетентность L_l. Таким путем выбирается подтаблица из s строк и q, имеющих наибольшие компетентности. Компетентность L_{lk} каждого элемента (lk) этой подтаблицы равна $L_l \times L_k$.

От каждого элемента подтаблицы получается своя подсказка, которая учитывается в счетчике голосов «за» и «против» с весом соответствующей компетентности. Если величины w_f для всех имен окажутся отрицательными, то делается вывод о том, что пропущенное имя не входит в состав d имеющихся имен. Среди имен, имеющих положительную величину w_f, выбирается имя с $w_{f\,\max}$, и это имя вставляется в пустую клеточку таблицы. Мерой доверия к принятому решению служит энтропия распределения голосов. Если в исходной таблице более чем один пропуск, то предсказывать можно каждый пропущенный элемент независимо от других (параллельная стратегия) или поочередно с использованием всех элементов как исходных, так и предсказанных на предыдущих шагах (последовательная стратегия). При последовательной стратегии нужно начинать с предсказания того элемента, для которого получаемая степень доверия максимальна.

Глава 8

Прогнозирование многомерных временных рядов

§ 1. Введение

Традиционно алгоритмы самообучения анализируют протоколы, в которых имеются описания входных воздействий и реакций изучаемой системы на эти воздействия. Раньше в кибернетической литературе задачи установления зависимостей между характеристиками входа и выхода назывались задачами анализа «черного ящика». Однако имеются еще более сложные ситуации, когда входные воздействия сами по себе остаются неизвестными и отражаются в протоколе лишь косвенно, через изменения выходных характеристик наблюдаемых объектов. Дальнейшим осложнением является наличие скрытых внутренних влияний характеристик одних объектов на характеристики других объектов. Такими свойствами обладают, например, протоколы наблюдений за характеристиками людей, взаимодействующих друг с другом в процессе решения неких внешних задач. Другим примером является протокол курса валют или ценных бумаг на бирже.

В таких ситуациях «абсолютно черного ящика» задача предсказания будущего состояния характеристик наблюдаемых объектов может быть решена лишь на базе двух гипотез: *гипотезы повторяемости* (в прошлом встречались такие же или аналогичные внешние воздействия) и *гипотезы адекватности реакции* (похожие воздействия вызывают похожие реакции), являющейся очевидной модификацией гипотезы локальной компактности.

В главе описывается обучающийся генетический алгоритм LGAP (Learning Genetic Algorithm for Prognosis) для извлечения закономерностей (знаний) из такого рода данных и использования этих знаний для прогнозирования будущих событий. Алгоритм LGAP существенно использует идеи алгоритмов ZET и WANGA для заполнения пробелов в эмпирических таблицах, которые были описаны в предыдущей главе.

§ 2. Обучающийся генетический алгоритм прогнозирования LGAP [80]

В алгоритме LGAP выделяется четыре этапа его работы: 1) формирование базовых элементов (базовых штаммов); 2) отбор компетентных штаммов; 3) выработка частных вариантов прогноза; 4) получение окончательного прогноза.

Пояснять алгоритм будем с помощью рис. 26.

2.1. Формирование базовых штаммов. Вместо трехвходовой таблицы «объект-свойство-время», обычно рассматриваемой в реальных задачах, будем пояснять алгоритм на примере двухвходовой таблицы «объект-время». Каждый элемент $b_{i,t}$ таблицы (протокола событий за T прошедших дней) отражает значение одной характеристики i-го объекта ($i = 1, 2, \ldots, Y$) в t-й момент времени. Моменты времени ($t = 1, 2, \ldots, T$) упорядочены в таблице по «возрасту»: самые свежие данные имеют индекс $t = 1$, данные за предшествующий день $t = 2$ и т. д. до дня с индексом $t = T$.

В алгоритме ZET компетентность элементов оценивалась по их похожести на некоторое базовое подмножество элементов, которые находятся в одной и той же строке или в одном и том же столбце. В данном алгоритме в состав базового множества могут входить G любых элементов таблицы, взятых из протокола за τ последних дней. Такие множества будем называть *базовыми штаммами мощности* G. В протоколе за τ дней имеется N элементов ($N = Y \times \tau$). Мощность базовых штаммов можно менять в пределах от единицы до N. При каждой заданной мощности G штаммы могут иметь разную «архитектуру», вследствие чего количество вариантов базовых штаммов мощности G равно числу сочетаний из N по G, а общее количество разных базовых штам-

мов выражается равенством

$$M = \sum_{G=1}^{N} C_N^G.$$

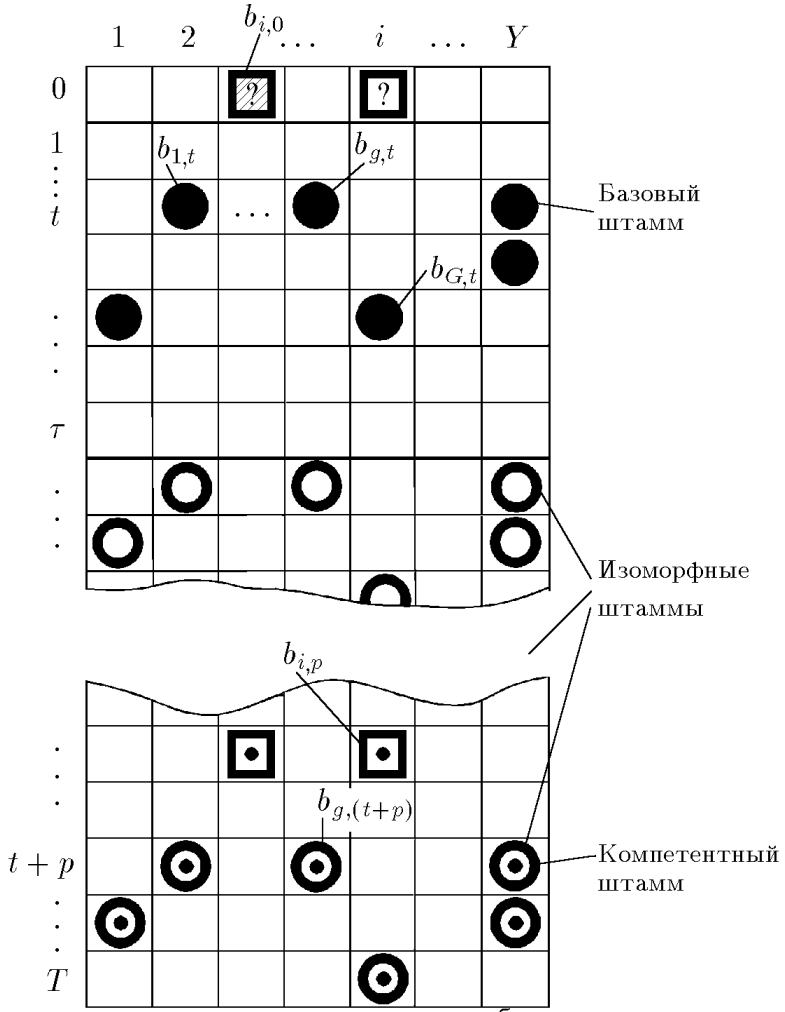


Рис. 26

Нетрудно видеть, что испытание всех базовых штаммов может превысить возможности компьютера рядового пользователя. Для сокращения перебора используется комбинация метода последовательного наращивания числа элементов (алгоритм Add,

описанный в главе 6) с методами генетического программирования [41, 156].

Алгоритм Add применительно к нашей задаче может выглядеть так. Вначале оценивается ожидаемая ошибка прогноза элемента $b_{i,0}$ при использовании в качестве базовых всех N элементов по одному ($G = 1$). Выбирается подмножество из n наилучших вариантов ($n < N$). Затем просматриваются базовые штаммы из пар, образованных каждым из n элементов, отобранных на первом шаге, и всеми остальными $(N-1)$ элементами. Таких вариантов имеется $n \times (N-1)$. Из них выбираются n лучших пар, к которым по очереди добавляем по одному все остальные $(N-2)$ элемента (здесь число вариантов около $n \times (N-2)$). Тем же путем отбираются лучшие тройки и т. д. до заданного количества элементов G в базовом штамме. При постоянной величине n общее число вариантов имеет значение порядка $G \times n \times N$, что, например, при $G = 12$, $n = 2$ и $N = 36$ меньше объема полного перебора в миллион раз. Описанный процесс можно считать некоторой моделью, сочетающей мутации (в виде появления новых элементов у потомков) с естественным отбором лучших из них для использования в качестве «родительских» штаммов в следующем поколении.

Однако известно, что этот метод обеспечивает получение лишь локально-оптимального решения. Чтобы попытаться улучшить найденные решения, применим на каждом шаге (т. е. при каждом новом значении G) процедуру «скрещивания». Для этого упорядочим элементы отобранных базовых штаммов по индексу времени. Если два элемента имеют одинаковый индекс t (т. е. находятся в одной строке таблицы), то упорядочим эти элементы по номеру объекта i (по номеру столбца). В соответствии с этим порядком присвоим каждому элементу в базовом штамме номер от единицы до G.

Из каждой пары родителей можно сформировать большое количество потомков путем замены k элементов первого родителя на k элементов второго родителя с одинаковыми порядковыми номерами этих элементов. При каждом значении k число новых потомков будет равно C_G^k. Можно ограничиться меньшим числом потомков, например двумя от каждой родительской пары, путем соединения первой половины одного родителя со второй половиной второго (и наоборот). Качество таких гибридных штаммов оценивается наряду с качеством породивших их родителей и из этого набора отбирается n наилучших штаммов для последую-

щего участия в мутациях и естественном отборе.

Упоминаемая постоянно процедура оценки качества базовых штаммов включает в себя этап поиска компетентных штаммов и этап оценки ожидаемой ошибки предсказания с использованием этих штаммов. Перейдем к описанию процедуры отбора множества компетентных штаммов.

2.2. Отбор компетентных штаммов. Элементы базового штамма q, начинающегося с элементов t-й строки, в массиве исходных данных помечены их порядковыми номерами g в штамме: $(1,t),(2,t),\ldots,(g,t),\ldots,(G,t)$. Набор адресов этих элементов описывает структуру (архитектуру) конкретного штамма, состоящего из G элементов. Если индекс t у всех элементов штамма увеличить на заданное число p, то мы получим штамм той же структуры, что и исходный, но только сдвинутый во времени на p шагов назад. Назовем такой штамм *изоморфным* данному базовому штамму.

Выделим среди изоморфных штаммов группу, состоящую из k «потенциально компетентных» штаммов (ГПК). В нее будем включать изоморфные штаммы с наибольшей похожестью на базовый штамм q. Меру похожести между штаммами можно оценивать разными способами.

Если ориентироваться на абсолютные значения характеристик их элементов, то можно использовать евклидово расстояние между штаммами:

$$R_{A,q,p} = \sqrt{\sum_{g=1}^{G} [b_{g,t} - b_{g,(t+p)}]^2}.$$

В ГПК отбирается k штаммов с наименьшими расстояниями $R_{A,q,p}$.

Если считать, что мы имеем дело с данными, измеренными в шкале отношений, то похожесть двух штаммов можно обнаружить через расстояние между соответствующими отношениями:

$$R_{o,q,p} = \sqrt{\sum_{g=1}^{G} \sum_{s=1}^{G} [b_{g,t}/b_{s,t} - b_{g,(t+p)}/b_{s,(t+p)}]^2}.$$

Инвариантами шкалы интервалов являются расстояния между отношениями интервалов:

$$R_{I,q,p} = \sqrt{\sum_{g=1}^{G}\sum_{s=1}^{G}\sum_{n=1}^{G}\sum_{m=1}^{G}\left\{\frac{[b_{g,t}-b_{s,t}]}{[b_{n,t}-b_{m,t}]}-\frac{[b_{g,(t+p)}-b_{s,(t+p)}]}{[b_{n,(t+p)}-b_{m,(t+p)}]}\right\}^2}.$$

Хорошими свойствами обладает мера Акаика [28], выражающая степень похожести P объектов через расстояние R между ними:

$$P_{q,p} = 1/(1 + R_{q,p}).$$

Для всех приведенных выше сильных шкал похожесть штаммов можно определять и по модулю коэффициента корреляции (cor) между значениями соответствующих элементов этих штаммов. Если в таблице представлены данные, измеренные в более слабых шкалах (порядка и наименований), то для измерения расстояния между штаммами можно воспользоваться мерами, которые будут описаны в следующей главе.

Теперь проверим, является ли данная ГПК на самом деле компетентной. Чтобы обосновать процедуру такой проверки, вспомним смысл гипотезы компактности H (см. гл. 3). Из нее следует, что сначала нужно удостовериться, что объекты множества (A, q) компактны в пространстве описывающих характеристик X. В нашем случае мы имеем дело с множеством A из k штаммов, входящих в ГПК. Пространство X имеет размерность, равную G. Каждая g-я характеристика этого пространства есть номер g-го элемента базового штамма. Значение этого элемента $b_{g,t}$ и есть значение g-й характеристики объекта (в данном случае базового штамма). Для изоморфного штамма, сдвинутого относительно базового на p моментов времени назад, значение g-й характеристики равно значению $b_{g,(t+p)}$ элемента, находящегося в клеточке с координатами $v_g(i, t + p)$. Таким образом, мы имеем дело с множеством A объектов в G-мерном пространстве X. Эти объекты были отобраны по критерию похожести на базовый штамм q. Если критерий похожести принять в качестве критерия компактности, то условие компактности $C^X_{A,q}$ для множества штаммов из ГПК выполняется автоматически.

Добавим к X еще одну — целевую характеристику z, т. е. еще один элемент таблицы. Пусть для базового штамма этим элементом будет тот, значение которого требуется предсказать,

т. е. $b_{i,0}$. Соответственно к каждому штамму из ГПК добавим по одному известному элементу $b_{i,p}$. Теперь появляется возможность проверить, выполняется ли условие компактности для целевой характеристики z у объектов множества A, входящих в ГПК. Условие компактности для значений одной характеристики может быть определено по-разному. Например, по величине дисперсии значений, по разности между $z_{\max}$ и $z_{\min}$ и т. д.

Если условие компактности для z на штаммах множества A не выполняется, то это означает, что по свойствам X они похожи друг на друга, а по свойству z не похожи. Следовательно, между свойствами X и z этих объектов закономерной связи нет, и нет оснований рассчитывать на успешное прогнозирование целевого свойства с опорой на штаммы из данной группы. Такая ГПК не включается в список компетентных групп, она и породивший ее базовый штамм из дальнейшего рассмотрения исключаются. Если же штаммы данной ГПК оказались компактными в пространстве свойств (X, z), то они совместно со своим базовым штаммом образуют группу компетентных штаммов.

Затем вся описанная последовательность процедур повторяется для другого штамма той же мощности G, но другой архитектуры. Таким способом множество базовых штаммов порождает коллектив из W групп компетентных штаммов мощности G.

2.3. Выработка частных вариантов прогноза. Стратегии прогнозирования зависят от того, какие отношения между элементами разных штаммов мы считаем инвариантными при переходе от одного штамма к другому (т. е. постоянными для разных моментов времени). Если мы считаем, что информативными являются абсолютные значения соответствующих элементов штаммов, то значение прогнозируемого элемента $b_{i,0}$ можно найти по абсолютным значениям k соответствующих элементов (предикторов), связанных с компетентными штаммами данной группы, т. е. с элементами $b_{i,p}$. Напомним, что количество разных компетентных базовых штаммов мощности G равно W, так что в прогнозировании элемента $b_{i,0}$ может принять участие $W \times k$ предикторов (W групп по k предикторов).

Если предполагается линейная зависимость между соответствующими элементами разных штаммов, то можно воспользоваться прогнозированием с помощью линейной регрессии. Строится линейная регрессия между G элементами базового и p-го компетентного штаммов. Подстановка элемента $b_{i,p}$ в уравне-

ние этой регрессии позволяет получить величину $b_{p,i,0}$ в качестве p-го варианта прогноза для элемента $b_{i,0}$. Аналогично получаются и все другие k вариантов прогноза для штаммов этой группы: $b_{1,i,0}, b_{2,i,0}, \ldots, b_{p,i,0}, \ldots, b_{k,i,0}$. Здесь, как и в предыдущем случае, общее число предикторов (и вырабатываемых ими частных прогнозов) равно $W \times k$.

Теперь предположим, что отношения между элементом $b_{i,p}$ и всеми g-ми элементами данного штамма $b_{g,(t+p)}$ такие же, как и отношения между элементом $b_{i,0}$ базового штамма и всеми его элементами $b_{g,t}$. Тогда можно получить G вариантов значения прогнозируемого элемента с опорой на каждый g-й элемент p-го компетентного штамма:

$$
\begin{aligned}
b_{1,i,0} &= (b_{1,t} \times b_{i,p})/b_{1,p},\\
b_{2,i,0} &= (b_{2,t} \times b_{i,p})/b_{2,p},\\
&\ldots\ldots\ldots\ldots\ldots\ldots\\
b_{g,i,0} &= (b_{2,t} \times b_{i,p})/b_{2,p},\\
&\ldots\ldots\ldots\ldots\ldots\ldots\\
b_{g,i,0} &= (b_{g,t} \times b_{i,p})/b_{g,p},\\
&\ldots\ldots\ldots\ldots\ldots\ldots\\
b_{G,i,0} &= (b_{G,t} \times b_{i,p})/b_{G,p}.
\end{aligned}
$$

В этом случае роль предиктора играет каждый элемент штамма в отдельности, так что каждый p-й штамм представляет собой набор из G предикторов, а все k штаммов из компетентной группы образуют коллектив из k групп по G предикторов. Общее число предикторов (частных прогнозов) для этого случая будет равным $W \times k \times G$ (W коллективов по k групп из G предикторов).

Нетрудно представить способ получения вариантов прогноза и для данных, измеренных в других шкалах. Эти способы могут представлять собой модификации основного содержания алгоритмов семейства WANGA, описанного в главе 7.

2.4. Получение окончательного прогноза. Частные прогнозы получены с помощью предикторов, организованных в иерархическую структуру: отдельные предикторы объединяются в группы, которые объединяются в коллективы и т. д. Фактически мы имеем дело с коллективно-групповым методом прогнозирования (КГМ) [64].

При свертке частных решений в общее имеет смысл обратить внимание на тот факт, что компетентность всех предикторов, породивших эти решения, превышает некоторый порог, но может быть неодинаковой. В результате разные группы предикторов вырабатывают разные наборы частных решений. Можно рассчитывать на то, что более компетентные группы будут выдавать более точные прогнозы.

Но компетентность групп проверялась по критерию компактности на множестве известных элементов A и отражала ситуацию в прошлом. Так делается обычно при ретроспективном выборе стратегий прогнозирования или распознавания: лучшей считается та стратегия, применение которой приводило к наименьшей сумме ошибок прогноза известных элементов. Однако в ходе экспериментов нередко обнаруживается, что в ряде случаев лучшие результаты дают некоторые другие стратегии, а не та, которая была лучшей «в среднем». По-видимому, они более точно соответствуют тем закономерностям процесса, которые проявляют себя в данный конкретный момент.

Какую же стратегию использовать для реального прогнозирования неизвестных величин? Где гарантия, что стратегия, чаще других в прошлом приводившая к успеху, будет лучшей и в этом конкретном случае?

Формулировка гипотезы компактности, приведенная в главе 3, позволяет ответить и на этот мучительный для прогнозистов вопрос. Из нее следует, что истинная компетентность группы предикторов должна проявляться не только в компактности объектов (A, q) в пространстве X и компактности объектов A в пространстве (X, z), но и в компактности вариантов прогноза характеристики z объекта q. Теперь у нас есть эти варианты, и мы можем оценить компактность прогнозов, полученных от группы предикторов f, например, через дисперсию их значений D_f. Как показали эксперименты, дисперсионный критерий оказался очень информативным: корреляция между D_f и ошибкой прогнозирования достигает величины $+0,7$. Опираясь на это, в качестве меры компетентности группы f будем принимать величину $L_f = 1/(1 + D_f)$.

Введение описанного выше дисперсионного критерия позволяет сформулировать следующий *принцип выбора* стратегий (в нашем случае — групповых предикторов): *применяй все имеющиеся в распоряжении стратегии и отдавай предпочтение той из них, которая дает варианты прогноза, обладающие наименьшей*

дисперсией.

Эта эмпирическая гипотеза отражает закономерность более высокого уровня (метазакономерность) по сравнению с традиционно используемой гипотезой о том, что хорошие результаты в прошлом обеспечивают хорошие результаты и в будущем.

С учетом сказанного выше процедура получения обобщенного решения состоит в следующем. Вначале на базе частных решений для каждой группы f вырабатывается групповое решение B_f. При прогнозировании в качестве группового решения может быть использовано среднеарифметическое значение частных прогнозов или их медиана. Вычисляется также характеристика компетентности группы L_f. Эти характеристики находятся для всех k групп предикторов из коллектива групп, участвовавших в прогнозировании.

Обобщенное решение (B) на следующем иерархическом уровне (на уровне коллектива) может быть получено с использованием параметрического семейства функций взвешенного усреднения групповых решений:

$$B = \sum_{f=1}^{k} \left(B_f \times L_f^{\alpha}\right) \Big/ \sum L_f^{\alpha}.$$

Здесь величина показателя степени α отражает стратегию учета влияния компетентности. Если $\alpha = 0$, то решения всех групп учитываются с равными весами. С ростом α растет влияние более компетентных групп. При очень больших значениях α в усреднении участвуют одна или несколько самых компетентных групп.

Если в прогнозировании участвовал один коллектив групповых предикторов, то величина коллективного решения (B) принимается в качестве окончательного значения прогнозируемой величины $b_{i,0}$. Если определить дисперсию групповых прогнозов, то можно получить представление о компетентности коллектива в целом и о величине ожидаемой ошибки данного прогноза.

Метод допускает использование не одного коллектива, а множества из W коллективов. Новые коллективы могут порождаться базовыми штаммами, мощность которых отличается от G. В этом случае по формуле, аналогичной вышеприведенной, делается взвешенное усреднение коллективных решений.

На рис. 27 представлена схема многоуровневого алгоритма коллективно-группового прогнозирования. На первом уровне ис-

пользуется базовый штамм мощности G_1 и архитектуры F_1. Находятся компетентные штаммы $C_1, C_2, \ldots, C_k$. Они образуют первую группу предикторов. С их использованием получаются частные прогнозы $b_1, b_2, \ldots, b_k$, вычисляются их дисперсия D_1 и групповой прогноз b'_1.

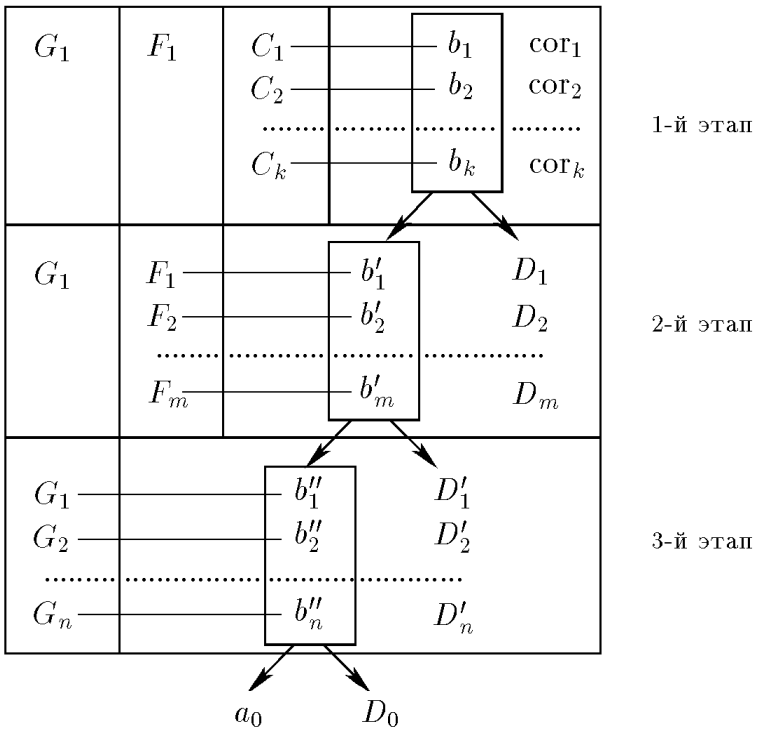


Рис. 27

На втором уровне эти процедуры повторяются для штаммов той же мощности G_1, но других архитектур: $F_2, F_3, \ldots, F_m$. В результате получаются групповые решения $b'_2, b'_3, \ldots, b'_m$ и по ним вычисляется коллективное решение a''_1 и дисперсия D'_1.

На третьем уровне меняется мощность базовых штаммов, и для каждой мощности $(G_2, G_3, \ldots, G_n)$ повторяются все предыдущие процедуры. В итоге получается множество из n коллективных решений: $b''_1, b''_2, \ldots, b''_n$ со своими значениями групповых дисперсий $D'_2, D'_3, \ldots, D'_n$. По ним находится окончательный прогноз b_0 и дисперсия коллективных прогнозов D_0, по которой можно судить об ожидаемой ошибке полученного прогноза.

§ 3. Критерии для оценки точности прогноза

Еще один сложный вопрос обычно возникает в процессе анализа результатов прогнозирования: как оценивать то, что мы называем точностью прогноза?

Часто берется абсолютное отклонение прогноза $b'_{i,0}$ от истинного значения $b_{i,0}$, деленное на истинное значение:

$$d' = \left|b'_{i,0} - b_{i,0}\right| / b_{i,0}.$$

Такая относительная величина мало чувствительна к ошибкам прогноза больших значений и чрезмерно чувствительна к ошибкам прогноза величин, близких к нулю. Кроме того, разница dd_i между минимальным и максимальным значениями может быть различной у разных наблюдаемых характеристик и одинаковая относительная ошибка d' будет приемлемой для принятия решений в одних случаях и не приемлемой в других.

В связи с этим предлагается судить о точности прогноза i-й характеристики по величине ошибки, нормированной по разнице dd_i:

$$d = \left|b'_{i,0} - b_{i,0}\right| / dd_i.$$

Такая мера обладает одинаковой чувствительностью к ошибкам прогноза для разных значений прогнозируемой характеристики. Ее чувствительность к ошибкам тем выше, чем в меньших пределах колеблется прогнозируемая характеристика, что представляется вполне логичным.

Иногда важно знать не абсолютную величину $b_{i,0}$ характеристики в будущем, а лишь то, будет ли она больше или меньше значения $b_{i,t}$ в данный момент времени. В таких случаях применима мера точности прогноза, учитывающая лишь совпадения знаков:

$$d^* = \begin{cases} 0, & \text{если } (b_{i,0} > b_{i,0}) \text{ и } \left(b'_{i,0} > b_{i,0}\right) \\ & \text{или } (b_{i,0} < b_{i,t}) \text{ и } \left(b'_{i,0} < b_{i,t}\right); \\ 0,5, & \text{если } (b_{i,0} = b_{i,t}) \text{ и } \left(b'_{i,0} \neq b_{i,t}\right); \\ 1 & \text{в других случаях.} \end{cases}$$

§ 4. Возможности распараллеливания алгоритма LGAP

Алгоритм LGAP легко распараллеливается на число предсказываемых элементов U. Каждая из U линий при заданном чи-

сле G элементов базового множества распараллеливается на M^G независимых процессов поиска компетентных штаммов и прогнозирования по ним. Обмены между процессами делаются на этапе выбора лучших базовых и создания новых штаммов путем мутаций или скрещиваний, после чего новые базовые штаммы запускаются в независимые параллельно протекающие процессы. Так что машинное время при решении этой задачи на многопроцессорной системе почти строго обратно пропорционально числу процессоров в системе.

§ 5. Экспериментальная проверка алгоритма LGAP

Экспериментальная проверка алгоритма проводилась на ежедневно публикуемых курсах различных валют на Московской межбанковской валютной бирже. Этот материал лишь косвенно отражает сложные внешние процессы, влияющие на формирование курсов валют, и потому задача прогнозирования курсов является заведомо сложной.

Были взяты данные за период $T = 545$ дней 1986–1987 гг. по шести валютам (доллар США, немецкая марка, фунт стерлингов, французский франк, японская иена и ЭКЮ). Вычислялись отношения курсов валют друг к другу, и эти кросскурсы нормировались по времени в диапазоне от нуля до единицы. Предсказывались значения кросскурсов в день t. В качестве базовых выбирались все неизоморфные штаммы мощностью G (G от трех до двенадцати), которые можно было построить из элементов τ дней, предшествовавших дню t (τ от одного до десяти дней). Компетентные штаммы определялись в оставшемся массиве в $(545 - T - t)$ дней. Кроме T, t и G в экспериментах варьировались количества компетентных предикторов на всех трех этапах их отбора.

Ввиду того, что после нормировки все кросскурсы изменяются в одинаковых пределах от нуля до единицы, в роли меры качества прогноза использовалась абсолютная величина ошибки в процентах.

Эксперименты подтвердили целесообразность использования дисперсионного критерия для оценки ожидаемой точности прогноза. Этот критерий оказался заметно более эффективным по сравнению с ранее использовавшимся в алгоритме ZET критерием аналогичного назначения.

Выяснилось, что наиболее компетентные штаммы концентрируются в непосредственной близости к прогнозируемому дню. Для одного и того же дня при прогнозе разных кросскурсов оказываются компетентными предикторы, порождаемые разными штаммами, так что настраивать программу на отдельные штаммы, которые в процессе ретроспективного прогнозирования давали наилучшие результаты, нельзя. Нужно использовать все штаммы и для каждого прогнозируемого элемента отбирать группы предикторов по дисперсионному критерию.

Ошибки прогноза для разных валют бывают разными. Наибольшую трудность представляет прогноз кросскурсов, в которых задействованы немецкие марки и/или французские франки. При наилучшей настройке параметров алгоритма прогноз 90 последних дней (на один день вперед) дал среднюю ошибку, равную 2,37 %. Знаки изменения кросскурсов прогнозируются с точностью 67 %. Эти результаты сравнивались с результатами прогнозов по так называемой нулевой стратегии, при которой предполагается, что завтра будет то же, что и сегодня. Ошибка кросскурса при этом была равной 4,012 %, а знаки предсказывались с точностью 54,3 %.

Если отобрать те дни, прогнозы в которые отличались малой дисперсией D'', то точность прогноза повышается. Так, если брать в среднем один прогноз из пяти, то ошибка для кросскурса уменьшалась до 1,102 %, а знак изменения кросскурса прогнозировался с точностью 81,1 %.

Алгоритм LGAP предполагается использовать для решения прикладных задач, которые традиционно решались нами с помощью алгоритма ZET: прогноз результатов деятельности предприятий, прогноз урожайности сельхозкультур, прогноз состояния наблюдаемых пациентов и т. д. Большой интерес представляют новые задачи прогнозирования процессов экономического, экологического и демографического характера, возникающие при изучении проблемы устойчивого развития ноосферы [79].

§ 6. Коллективно-групповые методы распознавания (класс алгоритмов КГМ [64])

Описанный выше подход к решению задачи прогнозирования применим и к задаче распознавания образов. Уже упоминались методы распознавания, использующие не одно, а несколько параллельно работающих решающих правил [116, 138]. Каждое

правило выдает свой частный вариант решения. Окончательное решение принимается на основании учета этих вариантов с помощью той или иной процедуры обобщения. Этот подход целесообразно расширить на случай, когда используется не одна группа решающих правил, а несколько, т. е. «коллектив» групп. Иерархия групп или коллективов может быть сколь угодно большой. На каждом уровне вырабатываются частные решения, по ним — обобщенные решения данного уровня, которые играют роль частных для следующего уровня и т. д.

Компетентность группы решающих правил может оцениваться по величине, характеризующей разброс получаемых результатов. При прогнозировании величин, измеряемых в сильных шкалах, для этой цели используется дисперсионный критерий. В задачах распознавания прогнозируется характеристика, измеряемая в шкале наименований, и здесь в качестве меры неодинаковости решений можно использовать энтропию h.

Если распознается S образов, то при равномерном распределении решений в пользу всех образов энтропия H_0 решений максимальна и равна $\ln S$. Если в пользу образа v высказалось p_v решающих правил из V, то энтропия предсказаний в группе f выражается формулой

$$h_f = -\sum_{v=1}^{S} \frac{p_v}{V} \ln \frac{p_v}{V}.$$

Компетентность L_f решающих правил группы f при распознавании образов можно принять равной $(1 - h_f/H_0)$. Если энтропия h_f решений равна нулю, то компетентность максимальна и равна единице. Если энтропия решений $h_f = H_0$, то компетентность такой группы предикторов естественно считать минимальной и равной нулю. По своему смыслу компетентность группы решающих правил эквивалентна их информативности. Но информативность оценивается не для всех возможных ситуаций, а каждый раз для данного распознаваемого объекта. Т. е. речь идет не о потенциальной информативности в среднем, как это обычно делается при оценке информативности признакового пространства, а об актуальной информативности системы «признаки-правила-объект» для каждого текущего случая.

Если соревнуются несколько групп решающих правил, то энтропийный критерий позволяет делать обоснованный выбор наиболее компетентной группы. Если же используется одна группа

(один коллектив решающих правил), то по величине энтропии можно судить об ожидаемой ошибке (или надежности) распознавания данного объекта.

Так же, как и в случае прогнозирования, энтропийный критерий приводит к формулировке *нового метода* выбора компетентных групп решающих правил: *если в распоряжении имеется несколько групп решающих правил, то нужно использовать все группы и затем выбирать ту из них, частные решения в которой отличаются наименьшей энтропией.*

На основании изложенного можно предложить следующую общую схему класса эффективных алгоритмов для решения задач распознавания образов с помощью коллективно-групповых решающих правил (класс алгоритмов КГМ).

Алгоритмы этого класса состоят в выполнении четырех последовательных этапов:

1) генерации групп решающих правил;

2) получения частных решений и оценки компетентности групп;

3) формирования обобщенного решения;

4) оценки ожидаемой ошибки.

I. Этап генерации групп решающих правил. На этом этапе можно использовать любые способы порождения семейств решающих правил. Общая схема этого процесса предложена Ю. И. Журавлевым [62,63].

Представим, что для распознавания используются линейные решающие правила в виде конечного набора из k гиперплоскостей. Если вместо констант в уравнениях этих плоскостей использовать непрерывно изменяемые параметры, то на базе каждой гиперплоскости можно породить группу из бесконечного числа гиперплоскостей, отличающихся друг от друга сдвигами и наклонами к координатным осям. Если использовать конечное число (G) дискретных значений изменяемых параметров, то каждая плоскость порождает группу из G различных гиперплоскостей. Средствами этого коллектива групп решающих правил можно получать $k \times G$ частных результатов распознавания.

Наряду с гиперплоскостями можно использовать и другие типы решающих правил, например наборы из квадратичных, таксономических или логических решающих правил. Каждый набор может породить свой коллектив решающих правил, и мы получим множество из W коллективов по k групп, состоящих из

G решающих правил. Это множество порождает $W \times k \times G$ частных результатов распознавания.

II. Этап получения частных решений и оценки компетентности групп. Техника получения частных решений правилами из группы f определяется видом решающих функций, входящих в эту группу. Результат распознавания по каждому отдельному правилу может быть категоричным («объект q принадлежит образу i») или вероятностным («объект q принадлежит образу i с вероятностью P_i, образу j с вероятностью $P_j \ldots$»). В любом случае сумма вероятностей принадлежности объекта q к тому или иному образу равна единице.

Если просуммировать «голоса», высказанные членами группы f в пользу каждого из S образов, и разделить полученные значения на количество членов этой группы G_f, то будет получено групповое решение о принадлежности объекта q к тому или иному образу:

$$P_{f1}, P_{f2}, \ldots, P_{fi}, \ldots, P_{fS}.$$

Сумма этих величин равна единице, а энтропия такого решения есть

$$h_f = -\sum_{i=1}^{S} P_{fi} \times \ln P_{fi}.$$

Отсюда компетентность L_f группы f равна $(1 - h_f/H_0)$.

Решение t-го коллектива получается путем взвешенного усреднения групповых решений. Для каждого образа суммируются голоса групп за этот образ с весом, равным компетентности группы:

$$P_{ti} = \sum_{f=1}^{k} \left(P_{fi} \times L_f^{\alpha}\right) \Big/ \sum L_f^{\alpha}.$$

В результате будет получено коллективное решение в виде вероятностей принадлежности объекта q одному из S образов:

$$P_{t1}, P_{t2}, \ldots, P_{ti}, \ldots, P_{tS}.$$

Теперь можно определить энтропию этого решения t-го коллектива и его компетентность L_t.

Аналогичные операции над всеми W коллективами дают материал для получения окончательного решения путем взвешенного усреднения коллективных решений:

$$P_i = \sum_{t=1}^{W} \left(P_{ti} \times Q_t^{\alpha}\right) \Big/ \sum Q_t^{\alpha}.$$

Если требуется получить одно категоричное решение, то выбирается образ i, имеющий наибольшее значение вероятности P_i.

По значению компетентности коллектива решающих правил (L) можно судить о надежности полученного результата распознавания объекта q.

Описанная схема выглядит и является на самом деле сложной для реализации. Ее можно упростить путем уменьшения величин G, k и W. Кроме того, алгоритмы класса КГМ очень хорошо распараллеливаются. На первом этапе параллельно могут работать $W \times k$ процессоров, генерирующих группы решающих правил; на втором, самом трудоемком — $W \times k \times G \times S$ процессоров, вырабатывающих частные решения для каждого из S образов. На третьем этапе можно использовать вначале $W \times k \times S$ процессоров, каждый из которых получает групповое решение для каждого образа, затем $W \times S$ процессоров, работающих над получением коллективных решений, и, наконец, S процессоров, вырабатывающих окончательное решение.

Глава 9

Согласование разнотипных шкал

§ 1. Расстояние между объектами в пространстве разнотипных признаков

До сих пор мы работали с объектами, все характеристики которых измерялись в одной из сильных шкал, и потому оценивать расстояние между объектами было несложно. Однако в реальных задачах часто встречаются таблицы со свойствами, измеренными в разных шкалах, в том числе в порядковых и номинальных. В этом случае возникает непростая проблема оценки меры расстояния, близости, похожести как между объектами (строками), так и между свойствами (столбцами).

Этой проблеме посвящены многие работы (см., например, [36, 60, 73, 98, 99, 163, 165]). Как правило, ищутся такие меры, которые удовлетворяли бы обычным аксиомам метрического пространства (непрерывности, симметричности и т. п.), были инвариантны к допустимым преобразованиям для данного типа шкалы и не зависели от состава изучаемых объектов. Итоги этих рассмотрений сводятся к тому, что меры, инвариантные к допустимым преобразованиям для многих шкал, можно указать, а мер, которые не зависели бы от состава выборки, не существует. Добавление к конечной выборке A или изъятие из нее какого-нибудь объекта может изменить прежние порядковые номера объектов i и l (для шкал порядка) или нормировку (для более сильных шкал), что приводит к изменению расстояния $d(i, l)$ между i-м и l-м объектами.

Какой же вывод нужно сделать из этих результатов? Не следует ли признать, что адекватных мер близости между объектами любой конечной выборки нет, а следовательно, нет и оснований верить результатам решения всех тех задач, в которых существенно используются меры близости или меры расстояния между объектами, т. е. задач таксономии, распознавания образов, корреляционного, регрессионного анализа и т. п.?

Не будем спешить соглашаться с таким пессимистическим заключением. Вспомним, что $d(i,l)$ меняется не при всяком изменении состава выборки A. Действительно, при нормировке сильных шкал по разности между самым большим и самым малым значением характеристики x в таблице, т. е. по $(x_{\max} - x_{\min})$, мера $d(i,l)$ будет сохраняться всегда, пока изменения состава объектов не коснутся объектов с $x_{\max}$ или $x_{\min}$. Для любых шкал нормированная мера $d(i,l)$ остается неизменной, если в таблице продублировать все объекты любое число раз. Если же встречаются другие ситуации, то это означает, что первоначальный состав выборки A плохо отражал свойства генеральной совокупности.

Таким образом, указанные выше трудности отражают фундаментальную для всех естественнонаучных дисциплин проблему представительности выборки. Формальными методами эту проблему решить невозможно. Исследователь должен либо знать, что выборка A включает полный набор изучаемых объектов, и тогда трудности, описанные выше, возникнуть не могут. Либо он должен верить в то, что выборка A представляет лишь часть генеральной совокупности G, но достаточно хорошо отражает ее закономерности, т. е. что выборка представительна. Тогда меры $d(i,l)$ будут одинаковыми для объектов i и l независимо от того, рассматриваем ли мы их на фоне выборки A или на фоне генеральной совокупности G. Выводы (т. е. таксономия, решающие правила, регрессионные уравнения и т. д.), сделанные на основании такой выборки, будут сохраняться и на генеральной совокупности. Некоторые отклонения от идеальной представительности можно частично компенсировать применением процедур, повышающих устойчивость $d(i,l)$ к случайным возмущениям. Например, нормировку делать не по крайним значениям характеристик, а по их дисперсии или медиане.

А если выборка A непредставительна, то никакие формальные ухищрения, в том числе и гарантии инвариантности $d(i,l)$ к допустимым преобразованиям шкал, не имеют смысла: из-за

непредставительности A индуктивные выводы для G все равно будут ложными.

В итоге вопрос о том, верить или нет мере расстояния $d(i,l)$, сводится к вопросу о том, представительна выборка A или нет. Эвристические способы получения некоторого представления о степени представительности выборки при решении задач распознавания образов обсуждались в § 10 главы 5. Если есть возможность, то малопредставительную выборку пополняют новыми объектами и тем самым увеличивают ее представительность. После того, как все такие возможности исчерпаны, вырабатывается оценка ожидаемой ошибки анализа и, если она устраивает пользователя, переходят к решению задачи анализа этих данных, полагаясь при этом на меры расстояния $d(i,l)$ между объектами, вычисляемые по данным из таблицы A.

Рассмотрим, какие меры расстояния можно использовать при обработке разнотипных шкал. Нам хотелось бы иметь меры $d(i,l)$, обладающие следующими очевидными свойствами:

а) непрерывности: мера $d(x_i, x_l)$ должна быть непрерывной функцией своих аргументов;

б) симметричности: предполагая пространство значений аргументов изотропным, потребуем, чтобы выполнялось соотношение $d(x_i, x_l) = d(x_l, x_i)$;

в) нормированности: мера $d(x_i, x_l)$ должна меняться в пределах от нуля до единицы, причем $d(x_i, x_l) = 0$, если $x_i = x_l$;

г) инвариантности: для преобразования f, допустимого в шкале данного типа, $d(x_i, x_l) = d\{f(x_i), f(x_l)\}$;

д) свойствам треугольника: для любых трех объектов a, b, c справедливо, что $d(ac) \leq \{d(ab) + d(bc)\}$.

Не для всех задач анализа данных нужны меры, которые удовлетворяли бы всем указанным выше требованиям. Часто достаточно, чтобы сохранялась информация о свойствах объектов лишь с точностью до порядка, так что требование г) можно было бы ослабить, а требование д) снять совсем. Однако мы попытаемся найти более универсальную меру, удовлетворяющую всем требованиям от а) до д). Не будем останавливаться на сильных шкалах (выше шкалы порядка). Для них свойствам а)–д) удовлетворяет, например, мера $d(i,l) = (x_i - x_l)/(x_{\max} - x_{\min})$. Оговоримся лишь, что для частного случая, когда $x_{\max} = x_{\min}$, неопределенное отношение $0/0$ принимается равным нулю.

Рассмотрим шкалу порядка. Напомним, что при всех допустимых преобразованиях для этой шкалы $f_{\text{п}}$ отношения из набора

$(<, >, =)$ между двумя числами x_i и x_l должны сохраняться и для чисел $f_{\text{п}}(x_i)$ и $f_{\text{п}}(x_l)$. Если мы построим матрицу размером $m \times m$ (где m — число объектов в выборке A), в которой для каждой пары объектов укажем их отношение в шкале порядка, то эта матрица не изменится при всех возможных преобразованиях группы $f_{\text{п}}$. В i-й строке этой матрицы представлена информация о том, в каких порядковых отношениях находится i-й объект ко всем остальным объектам таблицы A или какую порядковую роль играет он в этой таблице (матрице ролей). Естественно считать, что одинаковы два объекта, имеющие одинаковые порядковые отношения со всеми другими объектами. Различия (d) в отношениях i-го и l-го объектов к некоторому k-му объекту будем оценивать, анализируя содержание элементов на пересечении k-го столбца с i-й и l-й строками. При этом будем считать, что

$d(i,l)_k = 0$, если $(x_{ik} = \text{«<»}$ и $x_{lk} = \text{«<»})$
или $(x_{ik} = \text{«>»}$ и $x_{lk} = \text{«>»})$,
или $(x_{ik} = \text{«=»}$ и $x_{lk} = \text{«=»})$;

$d(i,l) = 1$, если $(x_{ik} = \text{«>»}$ и $x_{lk} = \text{«<»})$
или $(x_{ik} = \text{«=»}$ и $x_{lk} = \text{«>»})$;

$d(i,l) = 0,5$, если $(x_{ik} = \text{«=»}$ и $x_{lk} = \text{«<, >»})$
или $(x_{ik} = \text{«<, >»}$ и $x_{lk} = \text{«=»})$.

Суммарное различие между ролями объектов i и l во множестве A получаем из равенства

$$d_{\text{п}}(i,l) = \{1/(m-1)\} \sum_{k=1}^{m} d(i,l)_k.$$

Легко видеть, что если $x_i = x_l$, то $d(i,l) = 0$, и что для объектов i и l, имеющих максимально разные порядковые позиции, $d(i,l) = 1$. Очевидно выполнение и других требований к $d(i,l)$.

Напомним, что данные, измеренные в шкале порядка, можно без искажения содержания представить в шкале «нормированных рангов»: первому по порядку присваивается число 1, второму — число 2 и так до конца. Если встретятся t объектов с одинаковым порядковым номером (так называемые серии), то всем им присваивается номер «среднего» для них места: $x' = (1/t) \sum_{a=1}^{t} (a+h)$, где h — количество объектов, предшествовавших серии. После такой канонизации расстояние $d(i,l)$ находится по правилу

$$d_{\text{п}}(i,l) = \{1/(m-1)\} \left| x'_i - x'_l \right|.$$

В том, что эта мера равна мере, вычисленной выше по матрице ролей, легко убедиться на примере, приведенном в табл. 4. Здесь данные в шкале порядка имеют следующие значения: $x = 11;\ 6;\ 9;\ 11;\ 4;\ 109$. Те же данные в нормированных рангах принимают значения: $x' = 4,5;\ 2;\ 3;\ 4,5;\ 1;\ 6$.

Таблица 4

Матрица ролей в шкале порядка

x	11	6	9	11	4	109
11	$=$	$>$	$>$	$=$	$>$	$<$
6	$<$	$=$	$<$	$=$	$>$	$<$
9	$<$	$>$	$=$	$<$	$>$	$<$
11	$=$	$>$	$>$	$=$	$>$	$<$
4	$<$	$<$	$<$	$<$	$=$	$<$
109	$>$	$>$	$>$	$>$	$>$	$=$

Перейдем теперь к шкале наименований. Допустимые преобразования $f_{\text{н}}$ для шкал этого типа всегда сохраняют отношения «равно» и «неравно», так что при всех возможных переименованиях в матрице ролей $m \times m$ будут сохраняться значения отношений между всеми парами объектов из выборки A в виде символов «$=$» и «$\neq$».

Как и в предыдущем случае, будем в качестве меры расстояния между объектами i и l использовать разницу ролей, которую они играют среди объектов множества A, т. е. разницу их отношений ко всем остальным объектам из A. При этом будем пользоваться правилом

$$d_{\text{н}}(i,l) = (1/m)\sum_{k=1}^{m} d(i,l)_k,$$

где

$d(i,l)_k = 0$, если $(x_{ik} = \text{«}=\text{»}$ и $x_{lk} = \text{«}=\text{»})$
или $(x_{ik} = \text{«}\neq\text{»}$ и $x_{lk} = \text{«}\neq\text{»})$;
$d(i,l)_k = 1$, если $(x_{ik} = \text{«}=\text{»}$ и $x_{lk} = \text{«}\neq\text{»})$
или $(x_{ik} = \text{«}\neq\text{»}$ и $x_{lk} = \text{«}=\text{»})$.

Эта мера расстояния в шкале наименований удовлетворяет требованиям а)–д). Как отмечено в [163], такая мера $d(i,l)$ может

быть найдена и без построения матрицы ролей, а прямо через числа m_i и m_l, указывающие, сколько в выборке A имеется объектов с именем i и с именем l соответственно:

$$d_{\text{н}}(i,l) = (m_i + m_l)/m, \text{ если } i \neq l, \text{ и } d(i,l) = 0, \text{ если } i = l.$$

Пример, подтверждающий сказанное, приведен в табл. 5, в которой данные в шкале наименований имеют следующие значения: $x = a; b; a; a; c; b$, так что $m_a = 3$, $m_b = 2$, $m_c = 1$.

Таблица 5

Матрица ролей в шкале наименований

x	a	b	a	a	c	b
a	$=$	$\neq$	$=$	$=$	$\neq$	$\neq$
b	$\neq$	$=$	$\neq$	$\neq$	$\neq$	$=$
a	$=$	$\neq$	$=$	$=$	$\neq$	$\neq$
a	$=$	$\neq$	$=$	$=$	$\neq$	$\neq$
c	$\neq$	$\neq$	$\neq$	$\neq$	$=$	$\neq$
b	$\neq$	$=$	$\neq$	$\neq$	$\neq$	$=$

Легко видеть, что по приведенному правилу и по матрице ролей получаются одинаковые значения расстояний между объектами: $d_{\text{н}}(a,b) = 5/6$, $d_{\text{н}}(a,c) = 4/6$, $d_{\text{н}}(b,c) = 3/6$, $d_{\text{н}}(a,a) = 0$ и $d_{\text{н}}(b,b) = 0$.

Можно отметить, что при наличии в выборке A объектов только с двумя разными именами имеет место равенство $m_i + m_l = m$ и тогда $d(i,l) = 1$. Следовательно, для бинарных таблиц эта мера точно соответствует мере близости, вычисляемой через хеммингово расстояние.

Мерами $d_{\text{с}}$, $d_{\text{п}}$ и $d_{\text{н}}$ для шкал разных типов можно теперь пользоваться в многомерном случае, определяя расстояние $d_{\text{р}}$ по типу евклидова расстояния:

$$d_{\text{р}} = \sqrt{d_{\text{с}}^2 + d_{\text{п}}^2 + d_{\text{н}}^2}.$$

Меры такого типа удовлетворяют всем требованиям а)–д).

§ 2. Расстояние между разнотипными признаками

При корреляционном и регрессионном анализах, обработке групповых экспертных оценок и в других задачах анализа данных нужно уметь измерять расстояние между признаками (столбцами таблицы). В литературе известны методы измерения расстояния между однотипными признаками. Здесь мы опишем меру, пригодную для пар как однотипных, так и разнотипных признаков. Начнем с однотипных.

Если признаки x_1 и x_2 измерены в шкалах, более сильных, чем шкала порядка, то указанным выше требованиям а)–д) удовлетворяет мера расстояния $d(c,c) = 1 - r$, где r — модуль коэффициента линейной корреляции.

Среди многочисленных мер расстояния между двумя признаками, измеренными в шкале порядка, своей простотой и естественностью отличается мера Кенделла — Кемени [98,99]. Для ее определения нужно перебрать все C_m^2 парных сочетаний из m объектов и для каждой пары (i,l) сравнить порядковое отношение по признакам x_1 и x_2. Если порядковые отношения одинаковы, т. е.

если $(x_{1i} > x_{1l}$ и $x_{2i} > x_{2l})$ или $(x_{1i} < x_{1l}$ и $x_{2i} < x_{2l})$, или $(x_{1i} = x_{1l}$ и $x_{2i} = x_{2l})$,

то $d(i,l) = 0$. Если отношения порядка на этих признаках прямо противоположны, т. е.

если $(x_{1i} > x_{1l}$ и $x_{2i} < x_{2l})$ или $(x_{1i} < x_{1l}$ и $x_{2i} > x_{2l})$,

то $d(i,l) = 1$. В промежуточном случае, когда по одному признаку имеет место отношение «>» или «<», а по другому — отношение «=», считается, что $d(i,l) = 0,5$. Общее расстояние определяется как средняя мера «несогласия» двух признаков на всех парах объектов:

$$d_{\text{пп}} = \left(1/C_m^2\right) \sum_{i,l=1}^{m} d(i,l).$$

Если упорядочивания всех пар одинаковы, то $d_{\text{пп}} = 0$; если они на всех парах противоположны, то $d_{\text{пп}} = 1$. Если один признак (или эксперт x_1) устанавливает некоторый порядок объектов, а другой эти объекты считает одинаковыми (т. е. x_2 выдает серию длиной m), то мера $d_{\text{пп}} = 0,5$, что вполне естественно. Уместно отметить, что рекомендуемая во многих пособиях мера Спирмена [165] в этом случае дает $d_{\text{пп}} = \infty$.

Мера расстояния между признаками, измеренными в шкале наименований, определяется по правилу, аналогичному предыдущему: перебираются все сочетания пар объектов (i, l) и, если отношение по признакам x_1 и x_2 совпадают, т. е.

если $(x_{1i} = x_{1l}$ и $x_{2i} = x_{2l})$ или $(x_{1i} \neq x_{1l}$ и $x_{2i} \neq x_{2l})$,

то $d_{\text{нн}} = 0$. Если же эти отношения различны, т. е.

если $(x_{1i} = x_{1l}$, а $x_{2i} \neq x_{2l})$ или $(x_{1i} \neq x_{1l}$, а $x_{2i} = x_{2l})$,

то $d_{\text{нн}} = 1$. В итоге получаем, что

$$d_{\text{нн}} = \left(1/C_m^2\right) \sum_{i,l=1}^{m} d(i,l).$$

Величина $d_{\text{нн}}$ в точности равна величине хеммингова расстояния между матрицами смежности, одна из которых построена по признаку x_1, а вторая — по признаку x_2.

Перейдем теперь к разнотипным парам признаков. Оба признака можно сделать однотипными, если один из них «обеднить» до более слабого или второй «усилить» («оцифровать» [60]) до более сильного. Сделаем то и другое и для каждого случая найдем меру расстояния по описанным выше методам. Общую меру расстояния между двумя разнотипными признаками будем определять как среднюю величину двух этих частных расстояний.

Рассмотрим пару признаков (см. табл. 6), один из которых измерен в сильной шкале $(x_{\text{с}})$, а второй — в шкале порядка $(x_{\text{п}})$. Ослабление признака $x_{\text{с}}$ до шкалы порядка $\left(x_{\text{п}}^*\right)$ состоит в том, что мы теперь на его числовых значениях будем учитывать только отношение порядка. В результате для двух признаков в шкале порядка находим расстояние $d(\text{пп}^*)$ по методу, изложенному выше.

Усиление («оцифровка») шкалы порядка $(x_{\text{п}})$ до сильной шкалы $\left(x_{\text{с}}^*\right)$ делается так, чтобы: 1) значение порядка объектов по признаку $x_{\text{с}}^*$ совпадало с порядком по признаку $x_{\text{п}}$ и 2) числовые значения признака $x_{\text{с}}^*$ были максимально коррелированы со значениями признака $x_{\text{с}}$. Достигается это способом, показанным в табл. 6. Объекты упорядочиваются по возрастанию значений признака $x_{\text{п}}$. Если по признаку $x_{\text{п}}$ встречается серия, то всем t объектам, входящим в ее состав, приписываются значения $x_{\text{с}}^*$, равные среднеарифметическому значению их признака $x_{\text{с}}$:

$$x_{\text{с}}^* = (1/t) \sum_{i=1}^{t} x_{\text{с}i}.$$

Таблица 6

Пример усиления шкалы порядка $x_{\text{п}}$ до сильной шкалы $x^*_{\text{с}}$

$x_{\text{с}}$	$x_{\text{п}}$
3,5	12
2,3	6
10,0	8
2,1	1
11,2	50
15,6	153
13,6	50
7,0	10

а

Упорядочение ⟶

$x_{\text{с}}$	$x_{\text{п}}$
2,1	1
2,3	6
10,0	8
7,0	10
3,5	12
11,2	50
13,6	50
15,6	153

б

Усиление ⟶

$x_{\text{с}}$	$x^*_{\text{с}}$
2,1	2,1
2,3	2,3
7,0	6,83
10,0	6,83
3,5	6,83
11,2	12,4
13,2	12,4
15,6	15,6

в

Примечание. *а* — протокол в исходных шкалах; *б* — протокол, упорядоченный по $x_{\text{п}}$; *в* — протокол с усиленной шкалой $x^*_{\text{с}}$.

Затем, начиная с первого объекта таблицы, ищутся блоки инверсий, т. е. последовательности объектов, которые начинаются объектом i и заканчиваются самым далеким по порядку от него объектом l таким, что $x_{ci} \geq x_{cl}$ (блок в табл. 6). Каждому из t объектов блока инверсий приписывается числовое значение $x^*_{\text{с}}$, равное среднеарифметическому значению их признака $x_{\text{с}}$.

После этого через коэффициент корреляции вычисляем расстояние $d^*_{\text{сс}}$ и затем среднюю меру расстояния между признаками $x_{\text{с}}$ и $x_{\text{п}}$: $d_{\text{сп}} = \left[d^*_{\text{сс}} + d^*_{\text{пп}}\right]/2$. В нашем примере $d_{\text{сп}} = (0,0893 + 0,074)/2 = 0,0816$.

Рассмотрим сочетание признаков $x_{\text{с}}$ и $x_{\text{н}}$ (см. табл. 7). Ослабление сильной шкалы $x_{\text{с}}$ до шкалы наименований $x^*_{\text{н}}$ состоит в том, что всем различным значениям признака $x_{\text{с}}$ приписываются разные имена, а одинаковым значениям — одинаковые. Затем вычисляется расстояние в шкале наименований $d^*_{\text{нн}}$.

Таблица 7

Пример усиления и ослабления для шкал наименований $x_{\text{н}}$ и сильных шкал $x^*_{\text{с}}$

$x_{\text{с}}$	$x_{\text{н}}$
3,5	c
2,6	b
3,5	a
8,3	b
1,4	b
$-3,7$	a

а

Ослабление ⟶

$x^*_{\text{н}}$	$x_{\text{н}}$
k	c
c	b
k	a
a	b
p	b
n	a

б

Усиление ⟶

$x_{\text{с}}$	$x^*_{\text{с}}$
3,5	3,5
2,6	4,1
3,5	$-0,1$
8,3	4,1
1,4	4,1
$-3,7$	$-0,1$

в

Примечание. *а* — исходный протокол; *б* — протокол в шкале наименований; *в* — протокол в сильной шкале.

При усилении («оцифровке») номинального признака $x_{\text{н}}$ до $x^*_{\text{с}}$ объекту a приписывается величина $x^*_{\text{с}}(a) = x_{\text{с}}(a)$. Если одинаковое имя (например, b) имеют несколько (t) объектов, то всем им приписывается величина

$$x^*_{\text{с}}(b) = (1/t)\sum_{i=1}^{t} x_{ci}(b).$$

По полученным числовым значениям через корреляцию вычисляется мера $d^*_{\text{сс}}$, а затем и среднее расстояние $d(\text{сн}) = (d^*_{\text{нн}} + d^*_{\text{сс}})/2$. В нашем примере получается, что $d^*_{\text{сс}} = 0,34$, $d^*_{\text{н}} = 0,33$ и $d(\text{сн}) = 0,335$.

Наконец, для пары признаков $x_{\text{п}}$ и $x_{\text{н}}$ (см. табл. 8) обеднение $x_{\text{п}}$ до шкалы $x_{\text{н}}$ сводится, как и в предыдущем случае, к приписыванию разных имен объектам, имеющим разные порядковые номера, после чего находится расстояние $d^*_{\text{нн}}$.

Таблица 8

Пример усиления и ослабления для шкал наименований $x_{\text{н}}$ и порядка $x_{\text{п}}$

$x_{\text{п}}$	$x_{\text{н}}$
1	a
6	b
6	d
6	b
8	d

а

Ослабление ⟶

$x^*_{\text{н}}$	$x_{\text{н}}$
a	a
c	b
c	d
c	b
p	d

б

Усиление ⟶

$x_{\text{п}}$	$x^*_{\text{п}}$
1	1
3	3
3	4
3	3
5	4

в

Примечание. *а* — исходный протокол; *б* — протокол в шкале наименований; *в* — протокол в шкале порядка (нормированных рангов).

При усилении $x_{\text{н}}$ до $x^*_{\text{п}}$ признак $x_{\text{п}}$ канонизируется до нормированных рангов, а затем вместо имен $x_{\text{н}}$ объектам ставится в соответствие порядковые номера так же, как и в предыдущем случае: $x^*_{\text{п}}(a) = x_{\text{п}}(a)$ для одиночных объектов и

$$x^*_{\text{п}}(b) = (1/t)\sum_{i=1}^{t} x_{\text{п}i}(b)$$

для серии из t одинаково поименованных объектов. После нахождения расстояния $d^*_{\text{пп}}$ определяется $d(\text{пн}) = \left(d^*_{\text{пп}} + d^*_{\text{нн}}\right)/2$. В нашем примере $d^*_{\text{пп}} = 0,15$, $d^*_{\text{нн}} = 0,3$ и $d(\text{нп}) = 0,225$.

Отметим, что все использованные преобразования входят в группы допустимых преобразований для своих типов шкал, следовательно, величины полученных расстояний также инвариантны к допустимым преобразованиям оцениваемых признаков. Усиление и ослабление шкал вносят в некотором смысле симметричное искажение (добавление и потерю информации), так что усреднение получаемых частных мер после этих процедур можно считать оправданным. Применение указанных мер расстояния между объектами и между признаками позволяет использовать все то богатство математического обеспечения, которое было разработано для анализа таблиц данных с признаками, измеренными в сильных шкалах. При этом нужно к имеющимся программам добавить семантический блок, указывающий тип шкалы данного признака, и заменить блок определения расстояния в сильных шкалах на блок вычисления соответствующей меры из набора, описанного выше.

Глава 10

Алгоритмы таксономии в λ-пространстве

§ 1. Алгоритм λ-KRAB

Алгоритмы таксономии семейства FOREL, описанные в главе 4, оперируют евклидовыми расстояниями между точками. Однако было замечено, что при «ручной» таксономии человек обращает внимание не только на абсолютные расстояния, но и на отношения расстояний между несколькими соседними точками [54]. На этом основании была сформулирована гипотеза λ-компактности, изложенная в главе 3. Согласно этой гипотезе глаз человека хорошо улавливает различия в локальной плотности точек на плоскости и при прочих равных условиях предпочитает проводить границу между таксонами по участкам, где наблюдаются заметные изменения («скачки») этой плотности. Напомним, что локальное изменение плотности дискретного множества точек на границе между точками a и b можно описать нормированной величиной τ, а расстояние между этими точками — нормированной величиной d.

При «ручной» таксономии человек стремится к такому решению, при котором граница между таксонами проходила бы по участку с наибольшим значением характеристики $\lambda = \tau d$, которую мы называем λ-расстоянием. Если имеется несколько возможных вариантов таксономии с примерно одинаковыми значениями λ, то предпочтение отдается тому варианту, при котором таксоны включают в свой состав по возможности одинаковое ко-

личество объектов. Этот критерий «равномощности» таксонов хорошо отражает величина

$$h = k^k \prod_{i=1}^{k} \frac{m_i}{m},$$

где k — количество таксонов, m_i — число объектов в i-м таксоне, а m — общее число объектов.

Итоговый критерий, характеризующий качество таксономии, выражается величиной $F = h\tau d$. На максимизации фактически этого критерия основан алгоритм KRAB [54,69], давно и успешно применяемый при решении разнообразных задач таксономии.

Более поздние исследования, связанные с аккуратной формулировкой гипотезы λ-компактности, привели к заключению, что человек в процессе таксономии действительно использует все эти составляющие h, d и λ, но придает им разный вес. Была сформулирована задача идентификации модели следующего вида:

$$F = h^q \tau^s d^v.$$

Меняя параметры модели (v, s и q), можно на одном и том же множестве двумерных объектов получать разные варианты таксономии. Фиксируется тот вариант, который совпадает с вариантом, полученным экспертом при «ручной» таксономии. Запоминаются те сочетания параметров, при которых получался этот вариант. Затем программе и эксперту предъявляется другая выборка двумерных объектов (точек на плоскости) и эксперимент повторяется. После многочисленного повторения таких сравнительных экспериментов можно определить значения параметров, при которых машинная и экспертная таксономии совпадают.

Для проведения этих экспериментов была разработана программная система, с помощью которой на экране дисплея отображались сцены в виде точек, координаты которых задавались либо генератором случайных чисел, либо вручную, с помощью курсора и мыши. Эти сцены анализировались экспертами и программой таксономии, и экспериментатор, меняя значения параметров v, s и q, добивался совпадения результатов таксономии. В итоге такого исследования было установлено, что наилучшие результаты получаются при максимизации функционала F качества таксономии, в котором наибольший вес придается нормированному расстоянию d, затем характеристике скачка плотности τ и лишь потом характеристике равномощности таксонов h. Так как все эти

величины меняются в пределах от нуля до единицы, то уменьшение значимости того или иного параметра можно делать путем возведения его в степень, бóльшую единицы. С учетом этого значения параметров оказались такими: $v = 1$, $s = 2$ и $q = 4$. В окончательном виде критерий качества таксономии, положенный в основу алгоритма λ-KRAB, выглядит следующим образом:

$$F = h^4 \tau^2 d.$$

Работа алгоритма λ-KRAB начинается с нахождения пары точек с минимальным значением λ-расстояния между ними. Эти точки соединяются ребром графа. Затем соединяются следующие самые λ-близкие точки из числа не присоединенных к уже построенной части графа. Эта процедура повторяется до тех пор, пока все точки не окажутся соединенными ребрами этого графа. Такой граф не будет иметь петель, и суммарная длина всех его ребер будет минимальной. Граф, обладающий такими свойствами, называется *кратчайшим незамкнутым путем* (КНП) [135]. В нашем случае будем обозначать его через λ-КНП.

Теперь для разбиения множества A на два таксона необходимо разорвать одно ребро из ребер графа λ-КНП. Выберем ребро j с λ-длиной $\lambda_j = \tau_j^2 d_j$. Оставшимися ребрами λ-КНП соединяются два подмножества по m_i точек в каждом i-м подмножестве (таксоне). Эта информация позволяет для данного варианта разбиения вычислить характеристику равномощности таксонов h_j. Общая оценка качества F_j этого j-го варианта таксономии равна $\lambda_j h_j^4$. Вычисление величины F_j для всех $(m-1)$ ребер графа позволяет найти такой вариант таксономии, при котором достигается максимум критерия F.

Характеристика F инвариантна по отношению к количеству объектов m, числу таксонов k и абсолютным значениям длин ребер графа λ-КНП. Это позволяет сравнивать между собой качество таксономии различных множеств при разных количествах объектов m, разном числе таксонов k и разном среднем λ-расстоянии между объектами.

Алгоритмы семейства KRAB испытывались на многих модельных примерах и применялись для решения большого числа практических задач таксономии. В частности, успешно были решены все задачи, показанные на рис. 28, взятые из книги [12]. Там они были предназначены для доказательства теоремы несуществования: автор утверждал, что не может быть одного алго-

ритма, который смог бы решить все эти задачи. Однако если разные люди решают их одинаково, то, по-видимому, все они пользуются одним и тем же критерием. Следовательно, правильно было бы говорить не о том, что единого алгоритма нет и не может быть, а о том, что невозможно выявить интуитивные человеческие критерии качества таксономии. Описанный выше критерий является результатом попытки выявить именно эти человеческие критерии.

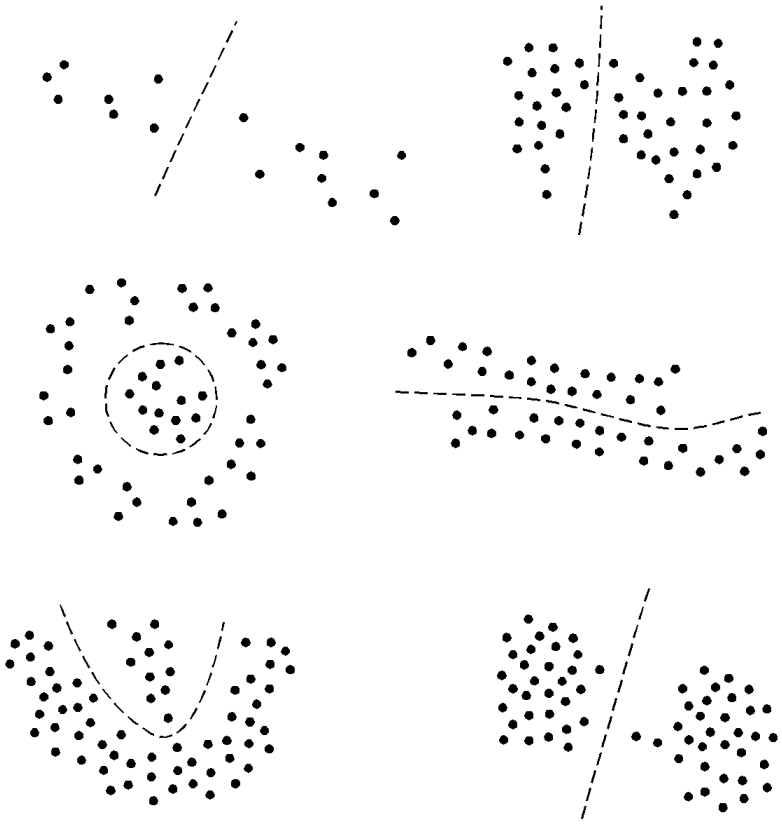


Рис. 28

Некоторое время казалось, что нам удалось решить поставленную задачу полностью. Однако затем были построены примеры, показанные на рис. 29, на которых таксономия (a) по критерию F не совпадала с человеческими решениями (b).

По-видимому, глаз человека обращает внимание не только на те характеристики, которые входят в состав критерия F, но также и на форму описания получаемых таксонов — на привычность этих фигур, их простоту. В примере из рис. 29, *а* человек выделяет два таксона простой сферической формы. Таксоны, выделяемые человеком в примере, показанном на рис. 29, *б*, описываются прямыми линиями.

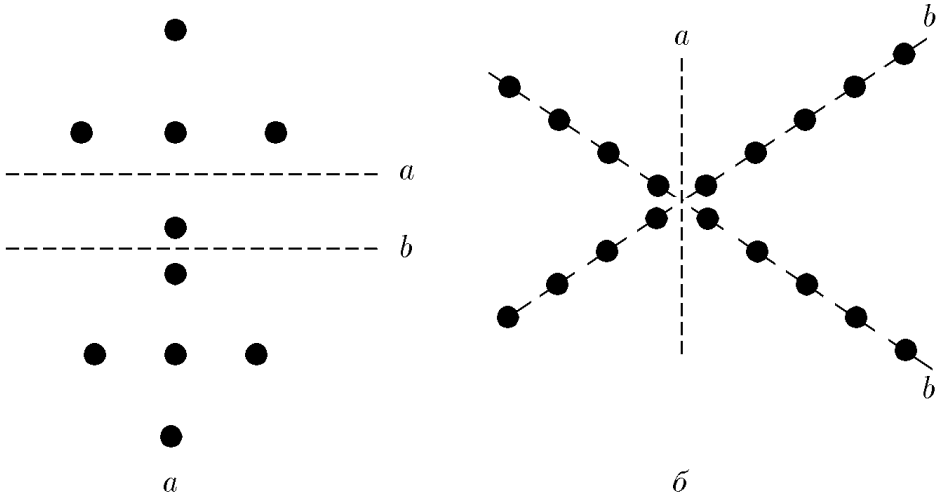


Рис. 29

Идея использования закономерностей, описывающих получаемые таксоны, в качестве дополнительного критерия качества таксономии принадлежит Р. Михальскому и Р. Степпу [127, 128]. Они ввели понятие *концепта*. Таксон — это не только набор похожих друг на друга объектов, но и некоторое новое понятие или концепт. В алгоритмах таксономии следует опираться на набор концептов в виде простых для восприятия понятий: сфера, эллипсоид, прямая линия, линейная разделяющая граница, логическое решающее правило и т. д.

В алгоритме λ-KRAB получаемые таксоны описываются наборами гиперсфер. Для этого применяется программа «Дробящиеся эталоны» (см. § 8 главы 5). В начале выбирается несколько вариантов таксономии на одно и то же число таксонов с наи-

большими значениями критерия F. В процессе сравнения этих вариантов предпочтение отдается тому из них, при котором потребовалось меньшее число эталонных гиперсфер. Если в число «финалистов» попали бы те два варианта таксономии, которые показаны на рис. 29, *а*, то для описания «человеческого» варианта было бы достаточно использовать два эталона, а для варианта по критерию F потребовалось бы три эталона (см. рис. 30). Алгоритм выбрал бы то же решение, что и эксперты.

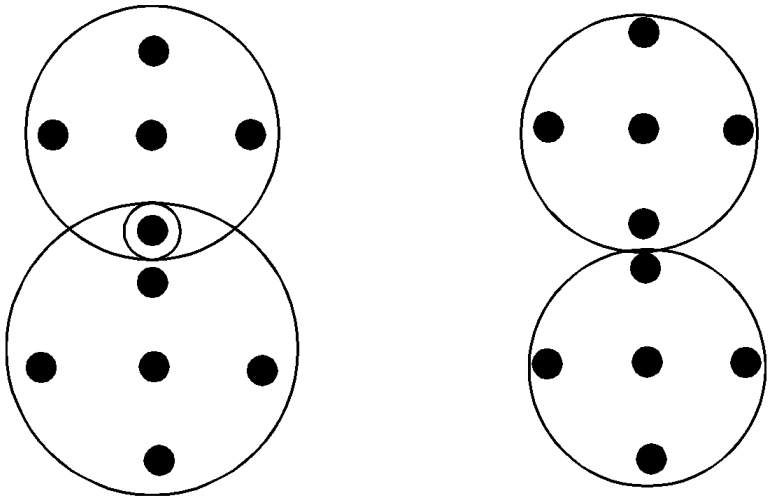


Рис. 30

Достичь такого же результата в примере из рис. 29, *б* с помощью сферических концептов не удастся. Здесь потребовался бы концепт «линейный объект». Из этого следует, что алгоритм, который мог бы получать человеческие решения во всех случаях, должен включать в свой состав такую же обширную библиотеку концептов, какую использует человек. Выявление состава этой библиотеки и формализация процедур использования разнородных концептов представляет собой интересную, но трудную и потому пока не решенную проблему.

§ 2. Алгоритм λ-KRAB–2

Сейчас уже имеются эффективные алгоритмы построения λ-КНП. Однако если количество исходных точек m превышает

несколько тысяч, этап построения λ-КНП становится затруднительным.

Дополнительные сложности возникают и в тех случаях, когда требуется получить большое число таксонов ($k \gg 2$). Тогда нужно разорвать не одно, а $(k-1)$ граничное ребро. Перебор всех возможных вариантов, количество которых равно числу сочетаний из $(m-1)$ по $(k-1)$, может потребовать неприемлемо больших машинных ресурсов. Для сокращения объема вычислений в этом случае можно воспользоваться модификацией базового алгоритма λ-KRAB — алгоритмом λ-KRAB–2.

Для ускорения процедуры на первом этапе в евклидовом пространстве делается таксономия множества из m объектов на k' таксонов простой сферической формы ($k' > k$) с помощью алгоритма FOREL. Затем центры этих таксонов используются в качестве исходных точек для построения кратчайшего незамкнутого пути алгоритмом λ-KRAB. На этапе вычисления характеристики равномощности h отличие от алгоритма λ-KRAB состоит лишь в том, что центр каждого мелкого таксона учитывается с весом, пропорциональным числу исходных точек, включенных в этот таксон.

На этапе выбора граничных ребер объем вычислений можно сократить, если из рассмотрения исключить короткие ребра λ-КНП, оставив для оценки F_j лишь t самых длинных ребер (например, $t = 3k$). Среди них практически всегда находятся ребра, разрыв которых обеспечивает получение наилучшей таксономии.

§ 3. Выбор числа таксонов

Если желательное число таксонов задано диапазоном от $k_{\min}$ до $k_{\max}$, то, наблюдая за функцией $F = f(k)$, можно в заданных пределах найти число таксонов, при котором F достигает максимума, что соответствует наиболее предпочтительной таксономии. На рис. 31, *а* показан пример множества A, таксономия которого на разное число таксонов характеризуется функцией, изображенной на рис. 31, *б*. Наиболее предпочтительное число таксонов равно 2. Если k требуется увеличить, то целесообразно выбрать 5 или 7 таксонов, но не 3, 4 или 6.

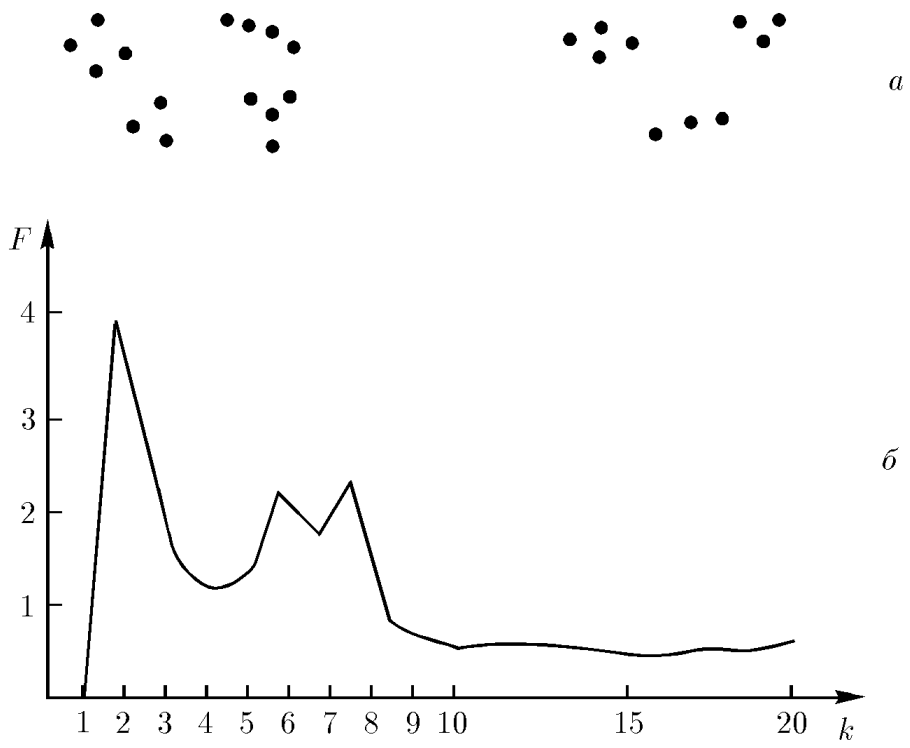


Рис. 31

Наличие такой возможности для объективного выбора наилучшего числа таксонов выгодно отличает алгоритмы семейства KRAB от многих других известных алгоритмов таксономии.

Глава 11

Методы распознавания образов в λ-пространстве

Напомним, что сформулированное в главе 3 условие компактности для решения задач распознавания образов является необходимым, но не достаточным. Условие, при котором точки разных образов (A и B) взаимно не компактны, т. е. сгустки точек разных образов не налагаются друг на друга, обозначим через $\lambda C^{\overline{-}X}_{A,B}$. С учетом этого гипотезу λ-компактности λH для распознавания образов можно записать в следующем виде:

$$\text{if } \left(\lambda C^{\overline{-}X}_{A,B} \& \lambda C^{X,z}_{A} \& \lambda C^{X}_{A,q}\right) \text{ then } \lambda C^{z}_{A,q}.$$

Руководствуясь этой гипотезой, рассмотрим ситуации, характерные для реальных задач распознавания: распределения образов неизвестны и обучающая выборка представлена конечным (небольшим) набором из m реализаций. При этом вырабатывается правило, по которому обучающая выборка распознается безошибочно или с ошибкой, не превышающей заданного порога. Из всех правил, удовлетворяющих этому условию, выбирается решающее правило, самое простое в своем классе.

Рассмотрим, как будут выглядеть λ-аналоги решающих правил, описанных в главе 3, в частности таких, как правило ближайшего соседа и таксономические решающие функции.

§ 1. Правило k ближайших соседей (алгоритм λ-NNR)

Этап обучения в алгоритме λ-NNR состоит из следующих процедур (см. рис. 32). Для каждого образа в отдельности строится полный граф, соединяющий точки его обучающей выборки. Среди ребер графа образа A находится самое длинное и обозначается через D_A. Для каждой точки i запоминается смежное ему ребро минимальной длины β_i. Вычисляется самое большое значение характеристики локального скачка плотности τ_A.

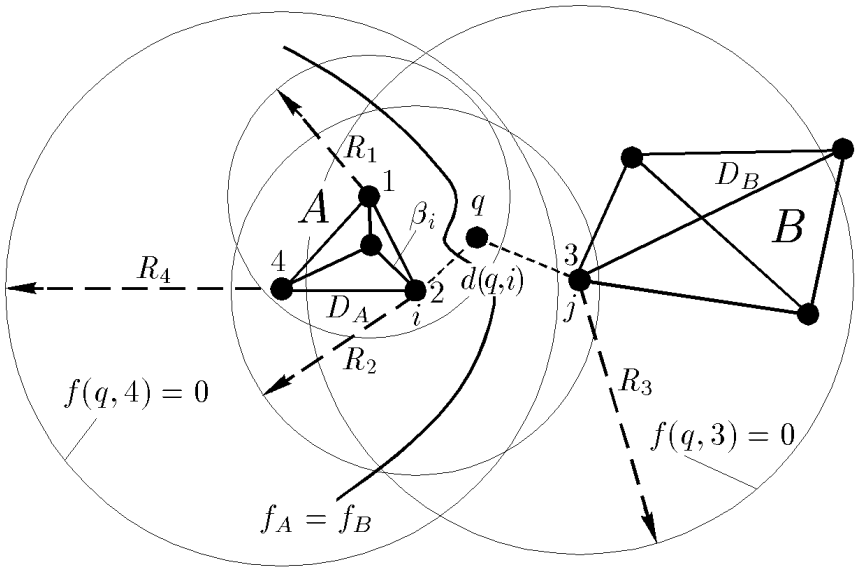


Рис. 32

На этапе распознавания определяется функция принадлежности распознаваемой точки q ко всем K образам поочередно. Среди точек образа A находится «ближайший сосед» — точка i, удаленная от точки q на минимальное расстояние $d(q,i)$. Нормированное расстояние d между точками q и i равно $d(q,i)/D_A$. Теперь можно определить величину $\tau^* = d(q,i)/\beta_i$ и нормированное значение этой величины $\tau = \tau^*/\tau_A$. Затем вычисляется характеристика $\lambda(A,q) = \tau^2 d$.

Введем понятие *функция принадлежности* $f(A,q)$ объекта q к образу A: $f(A,q) = 1 - \lambda(A,q)$. Аналогичным путем найдем значение функции принадлежности точки q ко всем другим образам: $f(B,q), f(C,q), \ldots, f(K,q)$. Точка q считается принадлежащей тому образу, функция принадлежности к которому имеет наибольшее значение. Можно ввести пороговое значение для f (например, $f = 0$) и считать, что точка не принадлежит ни одному из K образов, если все функции принадлежности меньше f.

Легко видеть, что такое правило в точности соответствует гипотезе локальной λ-компактности: в некоторой окрестности обучающей реализации некоторого образа могут появиться объекты того же образа. Границы этих окрестностей проходят по точкам, в которых функция f принимает нулевое значение (окружности на рис. 32). Функция f играет роль ядерной функции с центром в точках обучающей выборки, и параметры каждого ядра адаптивно меняются в зависимости от расстояния до соседних точек. Если некоторая точка пространства входит в окрестности точек, принадлежащих разным образам, то решение принимается в пользу того образа, которому соответствует большее значение функции принадлежности f. Решающая граница между i и j образами проходит по точкам, в которых $f(i,q) = f(j,q)$.

Для защиты от помех можно измерять f не до одного, а до $k > 1$ ближайших соседей из каждого образа. Решение при этом можно принимать по максимальному значению средней величины функции принадлежности к каждому образу.

Принятие решения по λ-расстоянию имеет определенные преимущества по сравнению с решениями, основанными на обычном евклидовом расстоянии. На рис. 33, *а* объект q эксперты уверенно присоединяют к образу A, хотя евклидово расстояние до ближайшего соседа относит его к образу B. При использовании же λ-расстояния этот объект распознается в качестве представителя образа A.

Если найти точки на плоскости, функции принадлежности которых к образам A и B одинаковы, то они образуют линию (λ-границу), показанную на рис. 33, *а*.

В главе 5 говорилось, что при неравных дисперсиях распределений образов оптимальная граница между ними представляет поверхность второго порядка. След такой поверхности показан на рис. 33, *б*. Очевидно сходство границ на этих рисунках.

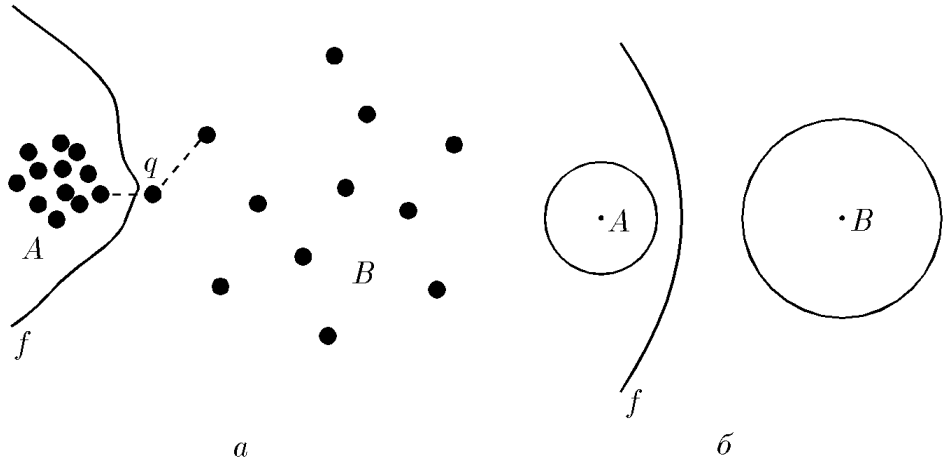


Рис. 33

Было бы интересно исследовать вопрос о том, стремится ли при увеличении объема выборки двух образов решающее правило, основанное на сравнении λ-расстояний до k ближайших соседей, к оптимальному статистическому решающему правилу. Обратим внимание на тот факт, что статистические решающие правила разработаны только для самых простых законов распределений образов, для которых оптимальными являются линейные или квадратичные разделяющие поверхности. При усложнении этих законов такие правила представляли бы собой поверхности r-го порядка, где $r > 2$. Как показано в [69], использование таких разделяющих границ требует быстро растущих в функции от r затрат машинных ресурсов. При этом эффективность решающих функций (отношение разделяющей способности к затратам ресурсов) быстро падает. Так, эффективность использования поверхности 5-го порядка в тысячу раз меньше эффективности использования набора поверхностей 2-го порядка.

Сложность же правила k ближайших соседей от вида распределения не зависит. С ростом объема обучающей выборки (m) машинное время и затраты памяти растут в зависимости от m линейно. Для уменьшения этих затрат необходимо предварительно сократить объем обучающих реализаций, оставив минимальный набор точек-прецедентов, достаточный для безошибочного распознавания всех точек обучающей выборки. С этой целью можно использовать алгоритм λ-STOLP.

§ 2. Выбор прецедентов (алгоритм λ-STOLP)

Так же, как и в алгоритме λ-NNR, для каждого из K образов находятся величины D_A, λ_A и β_i. Затем определяются значения функции принадлежности всех объектов i обучающей выборки образа A к своему образу $f(A)$ и ко всем другим образам $f(A^-)$. В этом процессе участвуют все объекты обучающей выборки. Среди объектов каждого образа находятся точки «максимального риска», т. е. такие объекты, у которых величина $R = f(A^-) - f(A)$ имеет наибольшее значение. Эти N объектов заносятся в список прецедентов.

Затем применяется стратегия пошагового уменьшения максимального риска. Для этого оценивается функция принадлежности всех объектов (кроме прецедентов) к своим и чужим образам с опорой только на имеющиеся точки-прецеденты. Находится один объект, имеющий самое большое значение функции риска R. Этот $(N+1)$-й объект пополняет список объектов-прецедентов. После этого процедура оценки величины R повторяется для всех оставшихся $(m - N - 1)$ объектов и самый «рискованный» из них включается в список прецедентов. Процесс продолжается до тех пор, пока самый большой риск ($R_{\max}$) для каждого объекта быть распознанным в качестве объекта чужого образа не станет меньше заданной пороговой величины R^* (например, $R^* = 0$).

Достаточность полученного списка прецедентов очевидна. Для проверки же необходимости всех прецедентов можно поочередно исключать их из списка и проверять, будет ли для всех обучающих точек выполняться условие $R_{\max} < R^*$. Если найдется прецедент, без которого это условие выполняется, то его можно исключить из списка. Если таких прецедентов окажется несколько, то исключается «самый ненужный», т. е. такой, при котором достигается минимум величины $R_{\max}$. После такого изменения списка процедура проверки оставшихся прецедентов повторяется. Проверка прекращается, если из списка нельзя удалить ни одного прецедента без нарушения условия $R_{\max} < R^*$.

Описанный алгоритм пошаговой оптимизации не может гарантировать точного решения. Однако получаемый список прецедентов, по-видимому, будет близок к оптимальному с точки зрения необходимости и достаточности.

Большие затраты машинного времени на выбор списка прецедентов окупятся в дальнейшем за счет существенного ускорения процесса распознавания контрольных объектов.

§ 3. Групповое распознавание

Встречаются ситуации, когда к распознаванию предъявляется не один, а сразу несколько (s) объектов. При этом можно выделить два случая. В одном из них заранее известно, что все они принадлежат одному и тому же образу, и нужно узнать какому именно. В другом случае такого ограничения нет: каждый объект может принадлежать любому образу. В статистической постановке первая задача рассматривается в [1]. Здесь мы опишем алгоритм λ-TRF, предназначенный для ее решения в условиях малых выборок при опоре на гипотезу локальной компактности (H_l). Алгоритм λ-GURAM при тех же предположениях решает вторую задачу.

3.1. Алгоритм λ-ТРФ. Напомним, что таксономические решающие функции демонстрируют свои особые преимущества в случае, когда к распознаванию предъявляется сразу несколько контрольных объектов. Пусть нам известно, что все они принадлежат одному и тому же образу. Например, контрольная выборка описывает свойства нескольких изделий из данной партии и требуется вынести решение о том, принадлежит ли эта выборка классу «хороших» или «плохих» изделий.

С помощью алгоритма Прима [135] построим λ-КНП для каждого из K образов. Затем построим λ-КНП, который соединяет образы друг с другом. Для этого найдем минимальные λ-расстояния между всеми парами образов. В качестве расстояния между двумя образами используется минимальное λ-расстояние между λ-ближайшими точками, принадлежащими разным образам. Чтобы граф без петель, соединяющий все образы, имел минимальную суммарную длину своих (пограничных) ребер, воспользуемся тем же алгоритмом Прима.

Оценим функционал качества разделения образов в виде величины F средней длины граничных ребер:

$$F = \{1/(K-1)\} \sum \lambda_j, \quad \text{где } j = 1 \div (K-1).$$

Теперь добавим контрольные объекты к обучающим объектам образа A и построим λ-КНП для этой смеси. После этого перестроим λ-КНП между образами и найдем оценку F_A качества разделения в предположении, что контрольные объекты принадлежат образу A. Затем повторим процедуру добавления контрольных объектов поочередно ко всем остальным образам и получения оценок F_i. Выберем тот (i-й) вариант, для которого оценка F_i

была максимальной. Присоединение контрольных объектов к i-му образу не ухудшило исходного качества F разделения образов или ухудшило его на минимальную величину. На этом основании принимается решение о принадлежности контрольных объектов этому i-му образу.

3.2. Алгоритм λ-GURAM. Теперь представим, что геологи вернулись из экспедиции и привезли коллекцию из s объектов, принадлежность каждого из которых тому или иному образу не известна.

Как и в алгоритме ТРФ (см. главу 5), вначале объединяем все реализации обучающей и контрольной выборок. Вычисляем λ-расстояния между всеми парами объектов этой смеси. Используя алгоритм Прима [135], строим кратчайший незамкнутый путь (λ-КНП).

Разные контрольные точки могут оказаться в разной ситуации (см. рис. 34). Некоторые контрольные точки (например, точка a) оказываются смежными только с другими контрольными точками. Назовем такие точки *внутренними*. Другие контрольные точки (например, b) связаны хотя бы одним ребром с точками обучающей выборки. Такие точки, а также ребра, соединяющие их с обучающими точками, называем *пограничными*. Распознавание контрольных точек делается последовательно (по одной точке). При этом каждая очередная распознанная контрольная точка добавляется к точкам обучающей выборки. Таким путем объем обучающей выборки растет в процессе распознавания, который выглядит следующим образом.

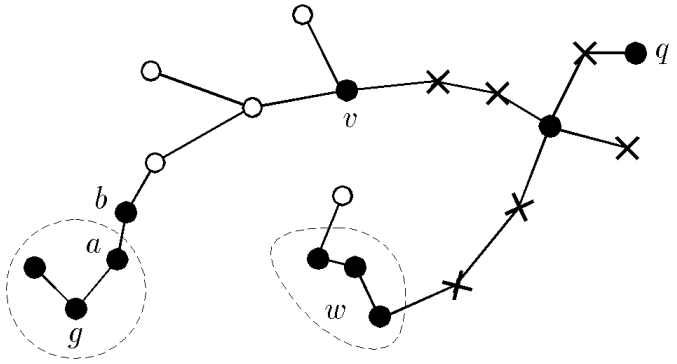


Рис. 34

Если некоторая пограничная точка q является конечной вершиной графа λ-КНП, т. е. имеет только одну смежную вершину, принадлежащую объекту i-го образа, то точка q также считается принадлежащей образу i. Если контрольная точка (например, v) связана ребрами λ-КНП с вершинами обучающих объектов нескольких образов, то ее следует относить к тому образу, λ-расстояние до которого минимально. Если среди контрольных точек обнаружится несколько внутренних, связанных только с одной пограничной, то все они должны быть отнесены к тому же образу, к которому отнесена эта пограничная точка. В таком положении находятся точки группы g на рис. 34. Если группа внутренних точек оказывается смежной с несколькими пограничными (например, группа w на рис. 34), то пограничная точка, которая имеет пограничное ребро минимальной λ-длины, относится к тому образу, с которым связана этим ребром. Смежная с ней внутренняя точка становится новой пограничной и вступает с другими пограничными точками в конкурентную борьбу за право быть распознанной на следующем шаге работы программы. Если на некотором шаге выявятся две пограничные точки с одинаковыми λ-расстояниями до своих образов, то сравнивается число обучающих реализаций у конкурирующих образов и первой распознается точка, примыкающая к более мощному образу. Эта стратегия соответствует хорошо известному правилу Байеса, согласно которому при равной апостериорной вероятности предпочтение отдается образу с большей априорной вероятностью его появления. Процесс присоединения контрольных объектов к своим образам продолжается до полного исчерпания списка пограничных точек.

Глава 12

Другие задачи анализа данных в λ-пространстве

§ 1. Критерии информативности λ-пространства

В предыдущей главе введено понятие функции принадлежности f_q q-го объекта своему образу A: $f_q(A) = 1 - \lambda(A, q)$, где $\lambda(A, q)$ — λ-расстояние между точкой q и ближайшей к ней точкой образа A. Аналогично определяется функция принадлежности к чужим образам: $f_q(A^-) = 1 - \lambda(A^-, q)$. Риск для точки q быть распознанной в качестве объекта чужого образа $R_q = f_q(A^-) - f_q(A)$. При скользящем контроле каждая точка обучающей выборки по очереди становится контрольной и распознается по всем остальным обучающим объектам с использованием правила ближайшего соседа. Если окажется, что точка q имеет величину $R_q > 0$, то она будет распознана с ошибкой.

Если среди m объектов обучающей выборки ошибочно были распознаны s объектов, то отношение $I = 1 - s/m$ можно считать мерой информативности данного признакового пространства. Действительно, если $s = 0$, то информативность признаков достаточна для безошибочного распознавания обучающей выборки, и величина I максимальна и равна единице. Если же $s = m$ (см. рис. 35), то это свидетельствует о том, что мы имеем случай «воды в губке», для которого не выполняется даже самая слабая гипотеза — гипотеза локальной λ-компактности λH_l, и рассчитывать на успешное распознавание контрольных объектов нет

никаких оснований

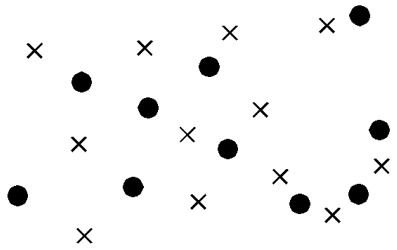


Рис. 35

Заметим, что алгоритм λ-STOLP для этого случая будет вынужден оставить в качестве прецедентов все m объектов обучающей выборки. Так что количество N необходимых прецедентов также говорит об информативности признакового пространства. Величина $J = 1 - (N - K)/(m - K)$ может служить еще одной мерой информативности признаков. Здесь K — количество образов. Информативность максимальна, если оказалось, что можно ограничиться одним прецедентом на каждый образ: $J = 1$. Если же $N = m$, то $J = 0$.

Величина s характеризует сложность стратегии природы, а величина t — сложность требуемого решающего правила. Мы видим, что эти две характеристики однозначно связаны друг с другом. Аналогичный факт был замечен Г. С. Лбовым и Н. Г. Старцевой при исследовании зависимости сложности статистических решающих правил от сложности стратегий природы [109].

Обычно рассматриваются два подхода [139]. При первом задается сложность природы и исследуются сложностные характеристики потребовавшихся решающих правил. При втором подходе задается сложность класса решающих правил и исследуются стратегии природы, с которыми справляются эти правила. Теперь становится очевидной полная эквивалентность этих подходов и их методологическое равноправие.

§ 2. Задачи заполнения пробелов

В описанных выше алгоритмах заполнения пробелов семейства ZET похожесть объектов друг на друга оценивается через евклидово расстояние между соответствующими строками таблицы данных. Для перехода к λ-пространству достаточно заменить

блок вычисления этого расстояния на блок определения λ-расстояния.

Исследования, которые подтверждали бы целесообразность такого перехода в этой задаче, еще предстоит провести. Пока можно лишь рассчитывать на то, что если бы человек видел картину взаимного расположения всех объектов в пространстве их характеристик и отбирал объекты, наиболее похожие на некоторый заданный объект, то он делал бы это, опираясь на закономерности зрительного восприятия. А эти закономерности, как мы убедились на описанных выше экспериментах, адекватно отражаются именно λ-расстояниями между объектами.

§ 3. Пакет прикладных программ ОТЭКС

Значительная часть изложенных в предыдущих разделах книги методов анализа данных реализована нами в виде программ. Эти программы объединены в пакете прикладных программ для обработки таблиц экспериментальных данных (ППП ОТЭКС) [82]. Первые версии этого пакета были сделаны более 20 лет назад. По мере появления новых вычислительных машин делались очередные версии пакета. При этом вносились изменения в состав программ, возникали новые программы, реализующие более эффективные алгоритмы, совершенствовалось сервисное сопровождение пакета.

Основной круг решаемых задач при этих изменениях оставался практически постоянным. В этот круг входят сейчас задачи, которые встречаются в практике обработки информации наиболее часто. К ним относятся следующие задачи.

1. Таксономия. В пакете ОТЭКС для решения этих задач имеются разные варианты программ из семейств FOREL и KRAB.

2. Выбор системы информативных признаков. В этом разделе есть программы, реализующие идеи алгоритмов NTPP и логических решающих правил.

3. Распознавание образов. Программы распознавания построены на базе разных вариантов гипотезы компактности: унимодальной, полимодальной, локальной и проективной унимодальной. Реализованы правила из классов линейных, логических и таксономических.

4. Заполнение пробелов и обнаружение ошибок в таблицах. Для решения этих задач имеются программы из двух

семейств: ZET и WANGA.

5. Прогнозирование. Продолжение динамических рядов осуществляется программами семейства ZET.

Количество объектов и свойств ограничивается только емкостью памяти компьютера. Число объектов может быть сравнимо с числом признаков. Признаки могут быть разнотипными, допускаются ошибки и пробелы в данных.

Примеры решаемых задач

Медицина.
Выделение информативной системы признаков (симптомов).
Выделение групп связанных признаков с их значениями (синдромов).
Формирование правил диагностики новых заболеваний.
Ранняя диагностика заболеваний.
Обнаружение ошибок в медицинских данных.
Прогнозирование процессов протекания заболевания.
Прогнозирование результатов терапевтических и других воздействий.

Геология.
Группировка объектов (территорий, месторождений, минералов и т. д.) по сходству их характеристик.
Прогнозирование месторождений.
Выявление наиболее важных для обнаружения месторождений характеристик геофизических и геохимических тестов.
Прогнозирование запасов полезных ископаемых в месторождении.
Обнаружение ошибок и заполнение пробелов в геологических данных.

Технология.
Выявление групп изделий с однотипной последовательностью технологических операций.
Поиск ближайшего аналога.
Выявление лимитирующих факторов, влияющих на качество продукции.
Адаптация технологического режима к свойствам сырья.
Прогнозирование качества изделий по результатам краткосрочных испытаний.

Социология.

Группировка объектов (населенных пунктов, регионов, анкетируемых людей) по сходству их характеристик.
Выявление факторов, определяющих исследуемый процесс (например, причин миграции населения).
Обнаружение ошибок и неверных ответов в результатах анкетирования.
Прогнозирование количественных характеристик исследуемых процессов.

Экономика.
Группировка предприятий по похожести профиля производства или кооперативных связей.
Выявление наиболее важных факторов, влияющих на экономическую эффективность предприятия.
Прогнозирование курса акций и валют на биржах.
Прогнозирование цен на мировом рынке.
Обнаружение ошибок и искажений в статистических данных.
Предсказание значений пропущенных данных.

Из этого перечня видно, что нет такой сферы деятельности, связанной с анализом данных, в которой нельзя было бы применить пакет ОТЭКС с пользой для дела.

Пользователи. Пакет ОТЭКС ориентирован на пользователя, не являющегося программистом. Научиться работать с ним очень просто. Результаты расчетов сопровождаются комментариями. Пакет хорошо документирован.

Разные версии ППП ОТЭКС в свое время были переданы для использования более чем в 100 организаций бывшего СССР. Пакет дважды отмечался серебряными медалями ВДНХ СССР. Сейчас пакет переводится из операционной среды DOS в среду WINDOWS, оснащается современным графическим сопровождением. К имеющимся добавляются программы, основанные на гипотезе λ-компактности и на дисперсионном критерии отбора информативных предикторов. С демо-версией пакета на русском и английском языках можно познакомиться по адресу:

www.math.nsc.ru/AP/oteks

Глава 13

Анализ данных и Data Mining

§ 1. Что такое Data Mining?

В последнее время в англоязычной литературе получил распространение термин Data Mining (DM) [127, 168], которым обозначается круг методов обработки данных, отличающихся от того, что авторы этого термина называют анализом данных. Попытки прямого перевода выражения Data Mining особым успехом не увенчались: вряд ли устроит вариант «горная промышленность данных» или «добыча данных». Чуть лучше было бы «обогащение данных». Внимательное чтение разъяснений этого термина показало*), что американское понимание термина «анализ данных» (обозначим это через AmAD) отличается от того, что под этим термином понимают французские и российские специалисты (эту версию интерпретации обозначим через FRAD).

Представители AmAD называют анализом данных классические дедуктивные процедуры математической статистики: корреляционный и регрессионный анализы, метод главных компонент, построение оптимальных решающих функций при известных законах распределения генеральных совокупностей и т. д. Схема действий при этом простая:

$$(\text{данные}) \Rightarrow (\text{программа AmAD}) = (\text{численный результат}).$$

*) Особенно после того, как обнаружилось, что авторы этого термина, поясняя его смысл, ссылаются на работы автора данной книги.

В основе программ лежат математические модели с известными параметрами. За адекватность модели и ее параметров изучаемому явлению AmAD-ист не отвечает. Он отвечает только за хорошую работу своей программы при заданных условиях.

Представители FRAD берутся за анализ явлений, для которых еще нет математических моделей. Есть только протоколы «стимул-реакция», представленные таблицами данных типа «объект-свойство-время». Конструирование моделей и определение параметров этих моделей является основным предметом внимания FRAD-истов. Они отвечают за привнесение эвристических гипотез о характере компактности, возможных формах (моделях) зависимостей, параметрах предполагаемых законов распределений и т. д. Наряду с дедуктивным аппаратом при решении этих задач используются индуктивные методы, реализованные в алгоритмах машинного обучения.

Вспомним классификацию задач прикладной математики, описанную в § 2 главы 1. Там выделялись задачи трех типов: вычислительная математика (ВМ), идентификация моделей (ИМ) и анализ данных (АД). Теперь можем сказать, что под AmAD подразумеваются задачи из области ВМ, в то время как FRAD полностью совпадает с АД . Задачи же, относящиеся к DM, охватывают область ИМ и АД, т. е. все то, что в прикладной математике отличается от ВМ.

Схематично получившуюся в результате классификацию задач можно представить так, как показано на рис. 36.

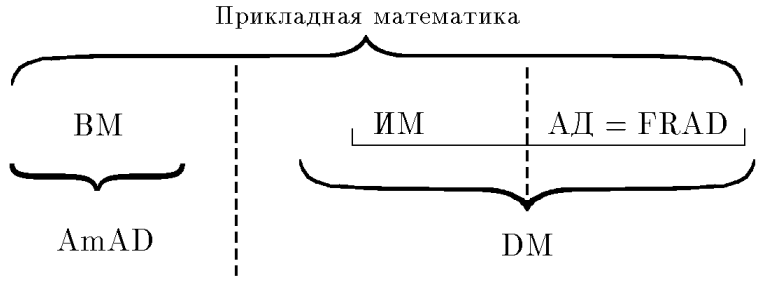


Рис. 36

Следует отметить одно важное методологическое отличие в подходах к решению задач между (ИМ&АД) и DM. В DM специально подчеркивается необходимость получения результата в таком виде, который удовлетворял бы двум требованиям: он дол-

жен быть понятным пользователю нематематику и вместе с тем быть пригодным для дальнейшей обработки компьютерными программами. Следовательно, не всякий формально правильный результат будет приемлемым, нужно выполнить еще и требования «прозрачности» для человека и машины.

Д. Мики формулирует этот тезис следующим образом [124].

1. Слабый критерий: система использует данные для выработки способов улучшения обработки будущих данных.

2. Сильный критерий: слабый критерий выполняется. Кроме того, система может представить эти способы в понятной символьной форме.

3. Сверхсильный критерий: слабый и сильный критерии удовлетворяются. Кроме того, система может представить эти способы в эффективной операционной символьной форме.

Некоторые методы АД обеспечивают получение таких результатов. В частности, закономерности в виде логических решающих правил представляют собой хороший пример: они легко интерпретируются человеком и удобны для дальнейшего машинного использования. Понятны человеку и описания таксонов, получаемых алгоритмами семейства FOREL.

В других же случаях результат, будучи приемлемым для машинного использования, оказывается трудно понимаемым человеком. Это касается всех тех случаев, когда результат решения задачи представлен перечнем каких-то элементов. Например, таксономия с помощью алгоритмов семейства KRAB дает результат в виде перечня объектов, входящих в тот или иной таксон. Машина может использовать такую информацию, а человек в длинный перечень фактов может поверить, но понять отражаемую этим перечнем закономерность может не всегда. Чтобы помочь человеку, а заодно упростить задачу и для машины, делается более краткое и простое описание результата в виде набора понятных человеку простых понятий или концептов (в данном случае сфер), покрывающих область каждого таксона.

То же самое происходит и при выборе информативных признаков, когда результат представлен их перечнем. Если список выбранных признаков большой, то сделать его понятным для человека можно группировкой отдельных признаков в небольшое число групп. В одну группу собираются признаки, связанные между собой взаимной зависимостью (коррелированные признаки). По такому принципу из признаков формируют факторы, а из симптомов — синдромы. Смысл каждой отдельной группы понятен

человеку, становится понятным и весь результат.

При распознавании с опорой на гипотезу локальной компактности формулы разделяющих поверхностей между разными образами в явном виде не выписываются: делать это было бы трудно и бесполезно из-за их громоздкости и непонятности. Чтобы объяснить человеку конкретное решение, сделанное по правилу ближайшего соседа, достаточно сказать, что контрольный объект оказался наиболее похожим на такой-то прецедент, и показать (или описать) его. Такое правило понятно человеку и легко реализуемо программой.

Не всегда результат, понятный машине, удается привести к понятному для человека виду. Например, квадратичная решающая функция в многомерном признаковом пространстве прозрачна для машины, но мало что говорит человеку. Замена ее набором гиперплоскостей также мало помогает делу — попробуйте представить себе даже одну гиперплоскость в пространстве с размерностью больше трех. То же, по-видимому, можно сказать и о результатах формирования вторичных признаков по методу МГУА [94], некоторых алгебраических методах принятия решений и т. д.

Объяснить процесс получения результата или сам результат в простых для человека терминах не удается, но использовать эти результаты для дальнейших машинных процедур можно. Человеку же придется удовлетвориться тем фактом, что эти результаты объективно правильны: признаки действительно являются информативными, объекты распознаются правильно, прогнозы сбываются.

Приведенные примеры показывают, что многие методы АД дают сразу прозрачные для человека и машины результаты, другие методы требуют дополнительных процедур для приведения результатов к прозрачному для человека виду. Важность и полезность прозрачности результата для человека признается всеми: домохозяйке нет дела до принципов работы телевизора, достаточно того, что она может управлять им с помощью небольшого количества кнопок. Но категорического условия делать все результаты не только правильными и прозрачными для машины, но и прозрачными для человека в АД обычно не выдвигается.

Идеология же Data Mining это условие считает обязательным. Чтобы сделать сложные результаты прозрачными, рекомендуется использовать широкий набор вспомогательных средств в виде простых для понимания заготовок (моделей, концептов).

Применение концептов делает результат психологически более приемлемым, переработанная таким способом информация становится для человека как бы более богатой. Может быть, на этом основании можно согласиться с термином «обогащение данных» (ОД) в качестве русского эквивалента термина Data Mining.

Один и тот же результат при разных концептуальных базах может получить разную интерпретацию, что не противоречит практике человеческого восприятия: на одной и той же картине один видит одно, другой другое. Снова вспомним высказывание Р. Фейнмана [155]: чем больше разных интерпретаций получает явление, тем глубже мы его понимаем.

Парадигма DM после этих пояснений может быть представлена такой схемой действий:

(данные+концепт)⇒(программа DM)=(прозрачный результат).

Использование заранее приготовленных концептов или моделей полезно не только для объяснения полученного результата, но и для самого процесса получения этого результата. Об этом говорит богатая история научных открытий, во многих из которых явно видны следы попыток такого типа: «А что если попробовать такую модель? А не образуют ли объекты такую структуру? А что если упорядочить объекты по такому правилу?..».

Эффективность такой технологии «обогащения данных» можно показать на примерах машинного переоткрытия некоторых законов природы.

§ 2. Переоткрытие некоторых законов природы

2.1. Закон Ома [37]. В верхней части (R0) табл. 9 представлен обучающий протокол, в котором признак X_i указывает значение тока в амперах, X_u - напряжения в вольтах и X_r — сопротивления в омах. Если бы мы позволили машине применить концепт такого типа, как $X_i \times X_r = X_u$ или $X_i = X_u/X_r$, то задача переоткрытия закона Ома была бы тривиальной. Мы разрешаем машине пользоваться более примитивными двух- $(a \geq b)$ и трехместными $(a + b \geq c)$ отношениями $P(2)_i$, $P(2)_r$, $P(2)_u$ и $P(3)_i$, $P(3)_r$, $P(3)_u$. Здесь отношение $P(2)_i$, например, имеет значение «истина», если ток в цепи a больше или равен току в

цепи b ($I_a \geq I_b$), а отношение $P(3)_r$ истинно, если сумма сопротивлений в цепях a и b не меньше, чем сопротивление в цепи c ($R_a + R_b \geq R_c$).

Таблица 9

Данные, использованные для переоткрытия закона Ома

X_i	X_r	X_u
66	8	528
83	7	581
90	19	1260
78	23	1794
59	26	1534
72	34	2448
87	35	3045
61	41	2501
72	52	3744
84	53	4452
97	48	4656
56	56	3136
74	64	4736
64	71	4544
73	110	8030 ($7488 \div 9472$)
90 ($72 \div 112$)	90	8100

Алгоритм ЭМП–1 состоит из процедур, выполняемых в три этапа. На первом этапе анализируется обучающая таблица и формируется обучающий протокол A. На втором генерируется множество B гипотетических протоколов и отбирается их подмножество B^*, наилучшим способом согласованное с обучающим протоколом. На третьем этапе выписываются явные значения предсказываемого элемента.

Этап I. Анализ двухместных отношений на первых двух строках (цепях) порождает элементарный протокол (подпрото-

кол) $\Pr(1,2) = \langle \overline{P}(2)_i, P(2)_r, \overline{P}(2)_u \rangle$. При очевидных обозначениях его можно записать более коротко: $\langle 0,1,0 \rangle$. Результат сравнения первой и третьей строк запишется как $\langle 0,0,0 \rangle$, а предпоследней и последней строк — как $\langle 1,0,1 \rangle$. В итоге получаем протокол A, состоящий из C_n^2 подпротоколов, где n — число строк (цепей).

Этап II. Теперь добавим $(n+1)$-ю строку, в которой известны значения двух характеристик (X_i, X_r) и попытаемся предсказать третью — X_u. Выпишем n подпротоколов, образованных этой строкой со всеми строками обучающей таблицы. Сравнение первой строки с $(n+1)$-й дает подпротокол $\langle 0,0,? \rangle$, второй — $\langle 1,0,? \rangle$, а n-й — $\langle 0,0,? \rangle$.

Заменим каждый из этих новых подпротоколов на два путем подстановки вместо знака «?» значений истинности (1) и ложности (0). Каждый вариант замены дает контрольный протокол B_j, а общее число получаемых контрольных протоколов равно 2^n.

Образуем протокол G_j, объединив в нем обучающий протокол A с одним из контрольных протоколов B_j. Посчитаем, сколько неизоморфных подпротоколов мощности 2 содержит протокол G_j. Запомним их количество m_j. В нашем случае неизоморфных подпротоколов (как мощности 2, так и мощности 3) может быть восемь: $\langle 0,0,0 \rangle$, $\langle 0,0,1 \rangle$, $\langle 0,1,0 \rangle$, $\langle 0,1,1 \rangle$, $\langle 1,0,0 \rangle$, $\langle 1,0,1 \rangle$, $\langle 1,1,0 \rangle$ и $\langle 1,1,1 \rangle$.

Повторим эту процедуру формирования объединенных протоколов G_j и подсчета числа неизоморфных подпротоколов m_j и оставим только те варианты B^*, в которых m_j оказалось минимальным. Если в B^* осталось больше, чем один протокол G_j, то для них переходим к подсчету числа неизоморфных протоколов мощности 3. Может оказаться, что некоторые протоколы G_j потребуют для своего покрытия большего числа неизоморфных протоколов, чем другие, и они будут исключены из дальнейшего рассмотрения.

Этап III. Оставшиеся протоколы считаются допустимыми и используются для получения предсказываемой величины. Если мощность протокола равна 2 и допустимый подпротокол, полученный при сравнении $(n+1)$-й строчки с первой, имеет вид $\langle 1,1,1 \rangle$, то получаем первый вариант прогноза: $X_u > 528$. Если среди допустимых есть протокол $\langle 0,1,1 \rangle$, порожденный сравнением с третьей строчкой, то появляется вариант $X_u > 1260$. Самое большое значение, предсказанное с помощью двухместных предикатов, есть $X_u > 4736$.

Если среди допустимых оказался протокол $\langle 0,1,1\rangle$ мощности 3, полученный при сравнении новой строки с двумя строками — первой и последней, то это означает, что $X_u > (4544 + 528)$. Протокол сравнения с двумя последними строками, если бы он имел вид $\langle 0,0,0\rangle$, дал бы вариант $X_u < 9280$. Программа, проанализировав все допустимые протоколы мощности 2 и 3, дает следующий прогноз: $7488 < X_u < 9472$. Истинное значение (8030) находится в этом промежутке и не намного отличается от его середины (8480).

Неизвестное значение тока при заданных напряжении и сопротивлении (вторая контрольная строка) предсказано программой также диапазоном, среднее значение которого (92) отличается от истинного (90) не более чем на 2,3 %. Для повышения точности нужно либо увеличить объем обучающего материала, либо анализировать протоколы большей мощности — 4, 5 и т. д.

2.2. Закон Менделя [70]. Обучающий материал, отражающий наблюдения того, какое количество розовых ($X_{\text{р}}$) и голубых ($X_{\text{г}}$) цветков имеется в разных поколениях ($X_{\text{п}}$) цветного горошка, представлено в табл. 10.

Таблица 10

Данные, использованные для переоткрытия закона Менделя

$X_{\text{п}}$	$X_{\text{р}}$	$X_{\text{г}}$
0	1024	0
1	768	256
2	640	384
3	576	448
4	544 $(540 \div 546)$	480
5	528	496
6	520	504
7	516	508

Мы знаем, что если в нулевом поколении были посеяны семена горошка с розовыми цветами, то согласно закону Менделя голубые цветы появятся уже в первом поколении и их количество будет расти от поколения к поколению, и через несколько поколений растений с голубыми цветами будет столько же, сколько

с розовыми. Если число растений с розовыми цветами в нулевом поколении было равно $X_{\text{р}}(0)$, то число растений с голубыми цветами в n-м поколении ($n > 0$) будет

$$X_{\text{г}}(n) = \sum_{i=1}^{n} X_{\text{р}}(0)/2^{(i+1)}.$$

Задание программе такого концепта было бы слишком большой подсказкой. Программа должна была научиться делать предсказания, опираясь на результаты анализа протоколов, построенных в виде двух-, трех- и четырехместных отношений: $P(2)_{\text{п}}$, $P(2)_{\text{р}}$, $P(2)_{\text{г}}$ типа $(aq \geq b)$, $P(3)_{\text{п}}$, $P(3)_{\text{р}}$, $P(3)_{\text{г}}$ типа $(a + b \geq c)$ и $P(4)_{\text{п}}$, $\text{P}(4)_{\text{р}}$, $P(4)_{\text{г}}$ типа $(a + b + c \geq d)$.

По описанному выше методу программа ЭМП–1 обучилась делать правильные предсказания любого элемента таблицы по двум другим элементам той же строки. В таблице приведен пример предсказания, сделанного программой для количества $X_{\text{р}}$ в четвертом поколении. Фактически алгоритм ЭМП–1 реализует гипотезу простоты, которая применительно к данному случаю выглядит так: «среди допустимых протоколов выбирай самый простой, т. е. тот, который описывается минимальным числом подпротоколов минимальной мощности».

2.3. Периодический закон Менделеева [65]. Согласно философии Data Mining обнаружение закономерностей одной из своих целей имеет отображение данных из исходного пространства описания X в некоторое другое пространство Y, более удобное для восприятия человеком.

Пространство описывающих характеристик X может быть произвольным — любой размерности, с любым числом объектов наблюдения. Пространство же восприятия должно отвечать некоторым условиям, вытекающим из ограниченных возможностей человеческой системы восприятия информации. Одно из таких ограничений связано с размерностью пространства Y: она не должна быть большой, так как объектами в многомерном пространстве человек оперирует с большим трудом. При большом количестве объектов восприятие облегчается, если эти объекты отобразить в малоразмерное пространство Y, в котором объекты оказываются упорядоченными по их свойствам. При этом можно обнаружить наиболее легко воспринимаемые человеком линейные зависимости свойств Y от свойств X. Если глобальной линейной

зависимости во всем диапазоне значений X добиться не удается, то можно попытаться воспользоваться суперпозицией локальных линейных зависимостей.

Сформулируем принцип *локальной линейной гладкости* L: «Свойства X в Δ-окрестностях точки f пространства Y меняются по линейному закону, так что свойства объекта в точке f равны среднему значению свойств его левого и правого соседей: $X(f) = \{X(f-\Delta) + X(f+\Delta)\}/2$». Пользуясь этим принципом, можно для некоторого множества A сконструировать пространство восприятия Y. Покажем это на простом числовом примере.

Пусть множество A состоит из семи объектов, свойство X которых указано в табл. 11. Построим одномерное пространство восприятия Y с целочисленными значениями координаты y. Поместим в точку y_0 любой объект, например a_7. Подберем к нему ближайшего левого соседа по свойству $X(-)$. Это, очевидно, будет объект a_1. Из приведенного выше соотношения можно найти ожидаемые (прогнозные) свойства правого соседа: $X(+) = 2X(a_7) - X(a_1) = 12 - 5 = 7$. Этим соседом оказывается объект a_4. Разместим найденные три объекта рядом друг с другом по оси Y (см. рис. 37).

Таблица 11

Объекты множества A в пространстве описания X

A	a_1	a_2	a_3	a_4	a_5	a_6	a_7
X	5	1	3,2	7	7,9	2,2	6

Теперь можно предсказать соседей для крайних из этих элементов: левого для a_1 и правого для a_4. Соседом a_4 должен быть объект со свойством $X(+) = 2 \times 7 - 6 = 8$. Объекта с точно таким значением X нет. Ближайшим к нему оказывается объект a_5. Условимся считать, что разница $8 - 7,9 = 0,1$ меньше допустимого порога нарушения гладкости (например, $0,3$), и поставим объект a_5 справа от объекта a_4.

Прогнозное значение свойства X у левого соседа объекта a_1 равно 4. Ближайшее к нему значение X у объекта a_3 равно $3,2$. Отклонение в $0,8$ превышает заданный порог, и мы приходим к заключению, что ближайший левый сосед для объекта a_1 в таблице отсутствует. Предположим, что в природе такой объект с $X = 4$ существует и он случайно не попал в таблицу. Поставим

его условно слева от a_1 (пунктир на рис. 37) и продолжим процедуру. Соседом слева от условного элемента является объект a_3, его левым соседом будет a_6, а его левым соседом a_2. Все объекты табл. 11 нашли на оси Y свое место. Теперь легко видеть, что если объект a_2 поместить в точку $y = 1$, то между свойствами X и Y существует простая закономерность: $x \approx y$.

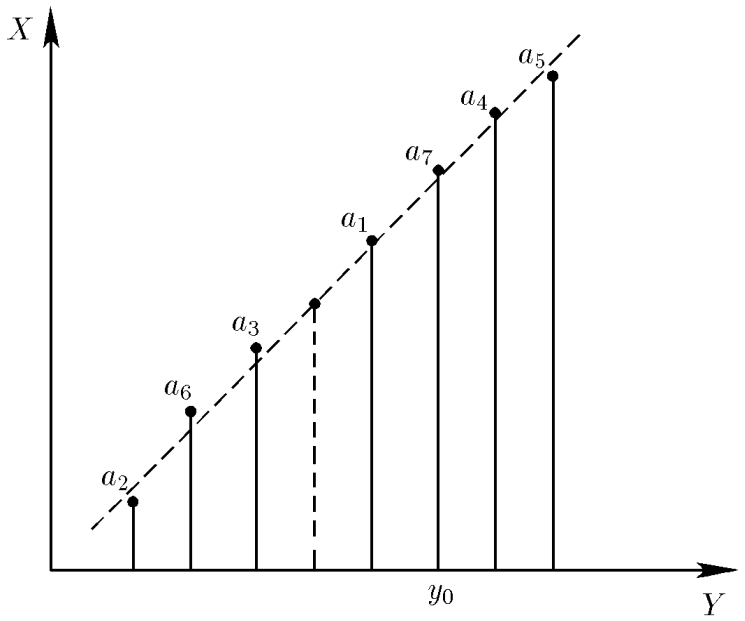


Рис. 37

Если объекты множества A обладают числом свойств, бóльшим чем одно, то прогноз и подбор соседей нужно делать по всем этим свойствам. В случае двумерного пространства восприятия Y соседями объекта a являются четыре объекта: сосед слева ($a_{\text{л}}$), справа ($a_{\text{п}}$), сверху ($a_{\text{в}}$) и снизу ($a_{\text{н}}$). Связь их свойств определяется соотношением

$$X(a) = \{X(a_{\text{л}}) + X(a_{\text{п}}) + X(a_{\text{в}}) + X(a_{\text{н}})\}/4.$$

Процедура двумерного линейного упорядочения аналогична той, что описана выше для одномерного случая: выбор начального элемента a и ближайших к нему двух соседей слева и сверху. Затем прогноз свойств двух соседей — справа и снизу, подбор объектов, свойства которых отличаются от прогнозных не более чем

на заданную величину, установка этих объектов на свои места и прогноз новых соседей. На каждом шаге устанавливается только один объект, самый близкий к прогнозу, и после этого прогнозы всех соседей повторяются.

На этом принципе построен алгоритм ПАМИР [65], предназначенный для решения задач многомерного упорядочения. С помощью программы ПАМИР был проведен ряд экспериментов по двумерному упорядочению объектов A различной физической природы. В табл. 12 и 13 показаны множества объектов с целочисленными значениями одного свойства X и наилучший их порядок на двумерной плоскости Y. Наилучшим считается такой вариант, при котором сумма локальных отклонений от линейной зависимости минимальна. Если истинные значения свойств объекта a_i равны $X(a_i)$, а его прогнозные свойства оказались равными $X\left(a'_i\right)$, то отклонения от линейной зависимости для всех m объектов имеют значения

$$Q = \sum_{i=1}^{m} \left| X(a_i) - X\left(a'_i\right) \right|.$$

Пусть нам дано множество A, описанное характеристикой $X(a)$: 4, 8, 3, 12, 9, 7, 6, 7, 10, 2, 8, 7, 11, 4, 9, 6, 7, 7, 5, 10, 6, 11, 8, 5, 4, 5, 9, 7, 3, 6, 9, 6, 10, 8, 9. Применение алгоритма ПАМИР позволило отобразить это множество в двумерное пространство восприятия Y, представленное в табл. 12.

Таблица 12

Двумерное упорядочение множества объектов A

y_2	y_1					
	1	2	3	4	5	6
1	2	3	4	5	6	7
2	3	4	5	6	7	8
3	4	5	6	7	8	9
4	5	6	7	□	9	10
5	6	7	8	9	10	11
6	7	8	9	10	11	12

В табл. 13 представлен результат аналогичной обработки множества объектов B, описанных свойством $X(b)$: 18, 30, 2, 8, 15, 12, 2, 4, 24, 10, 6, 16, 6, 9, 24, 15, 6, 20, 4, 18, 5, 12, 25, 4, 36, 12, 1, 10, 3, 3, 5, 12, 8, 30.

Таблица 13

Двумерное упорядочение множества объектов B

y_2	y_1					
	1	2	3	4	5	6
1	1	2	3	4	5	6
2	2	4	6	8	10	12
3	3	□	9	12	15	18
4	4	8	12	16	20	24
5	5	10	15	□	25	30
6	6	12	18	24	30	36

Закономерность, легко наблюдаемая в табл. 12, выражается формулой $x_i = y_{1i} + y_{2i}$, а в табл. 13 — формулой $x_i = y_{1i} \times y_{2i}$. Напрашивается вывод, что исходный набор объектов «не полон», отсутствующие объекты должны были бы занять места пустых клеточек в таблицах. Свойства этих «экообъектов» легко прогнозируются по приведенным выше интерполяционным формулам.

Возможности алгоритма ПАМИР были испытаны в экспериментах по переоткрытию периодического закона Менделеева.

Множество A было представлено химическими элементами первых семи рядов — с 1-го (водород) по 54-й (ксенон). Пространство описания включало в себя лишь три свойства элементов: x_1 — атомный вес, x_2 — валентность по кислороду и x_3 — валентность по водороду. В качестве пространства восприятия была использована плоскость с дискретными значениями переменных $y_1 = 1 \div 25$ и $y_2 = 1 \div 25$. Для каждого элемента определялось восемь соседей, как показано на рис. 38. Программа может анализировать гладкость по двум, четырем или восьми направлениям.

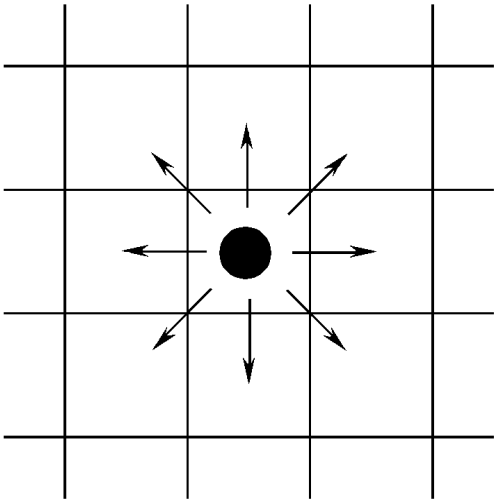


Рис. 38

В табл. 14 представлен результат работы программы ПАМИР. В качестве начального был выбран элемент с атомным весом 15 (фосфор). Рамкой показана граница таблицы в стандартном современном представлении [130]. Отклонения от стандартного вида имеют место для элементов длинного ряда VI группы VIII (элементы 44, 45 и 46).

Таблица 14

Двумерное упорядочение химических элементов

I	1							2		
II	3	4	5	6	7	8	9	10		
III	11	12	13	14	(15)	16	17	18		
IV	19	20	21	22	23	24	25	26	27	28
V	29	30	31	32	33	34	35	36	(45)	(44)
VI	37	38	39	40	41	42	43	(46)		
VII	47	48	49	50	51	52	53	54		

В табл. 15 показан результат упорядочения множества элементов таблицы с 1-го по 54-й за исключением крайних элементов длинных рядов (27, 28, 45, 46), а также элементов 12, 22, 35 и 39, которые были не известны во времена Менделеева. Программа, начав с элемента 17, расставила все элементы на те места, которые они занимают в таблице Менделеева, оставив пустые клетки в местах, соответствующих пропущенным элементам. Свойства пропущенных элементов легко можно найти по интерполяционным формулам, как это и делал Д. И. Менделеев.

Таблица 15

Результат упорядочения неполного множества элементов

I	1							2
II	3	4	5	6	7	8	9	10
III	11		13	14	15	16	17	18
IV	19	20	21		23	24	25	26
V	29	30	31	32	33	34		36
VI	37	38		40	41	42	43	44
VII	47	48	49	50	51	52	53	54

Как видно из этих примеров, если столбцам в табл. 14 и 15 сопоставить номера групп, а строкам — номера рядов, то результат программы ПАМИР почти совпадает с результатом, полученным Д. И. Менделеевым. Однако следует отметить существенные различия в пространствах X, использованных Менделеевым и программой ПАМИР. Д. И. Менделеев, кроме атомного веса и валентностей, использовал большое число других химических и физических свойств элементов: виды соединений с хлором, растворимость сернистых соединений, температуру плавления, цвет, запах летучих соединений и многое другое [97, 122]. Все эти глубокие знания облегчали ему поиск «соседей» при упорядочивании элементов. Следует отметить, что задача группировки ряда элементов решалась еще и предшественниками Менделеева. Так, было известно, что в одну группу следует относить элементы с современными номерами: 7–15–33–51; 8–16–34–52; 9–17–53; 20–38–56 и т. д. Подсказка такого рода могла бы существенно облегчить работу программы ПАМИР.

С другой стороны, машина использовала современные данные об атомных весах. До Менделеева атомные веса были известны неточно, ему приходилось исправлять их путем предсказания на основании всей таблицы. Ошибки в значениях атомных весов могут существенно исказить упорядочение, если использовать такое бедное описание свойств X, которое использовала программа.

Результаты данной работы можно сформулировать так: принцип локальной гладкости является эффективным средством отображения пространства описания X в пространство восприятия Y. Свойствами локальной гладкости обладают пространства восприятия, построенные для многих известных законов природы — законов Ома, Ньютона, Менделя, Менделеева и др. Этот принцип должен найти свое заметное место в ряду концептов, используемых в алгоритмах обнаружения закономерностей или DM.

Если пространство описания X наблюдаемых объектов множества A достаточно информативно, чтобы можно было построить пространство восприятия Y со свойством локальной гладкости, то сам процесс построения пространства Y с помощью ЭВМ принципиальных затруднений не вызывает. Это тем более справедливо для таких удачно найденных пространств X, для которых пространство Y удовлетворяет свойству глобальной линейной гладкости.

Часть III
Анализ знаний и структур

Глава 14
Метрика в пространстве знаний

§ 1. Меры близости между предикатами

Знания, которые используются в экспертных системах, часто бывают представлены в виде продукций типа «если $X_1 \& X_2 \& \ldots$ то A». При этом значения переменных могут задаваться разным способом, например: $X_1 = 7$; $X_2 = (2 \div 6)$; $X_3 = (a \vee b \vee c)$; $X_4 > 0$ и т. д. Такой специфичный вид представления знаний налагает большие ограничения на методы работы с ними. Методы логического вывода, опирающиеся на сравнение левых и правых частей двух продукций средствами языка PROLOG, рассматривают все переменные через призму шкалы наименований [146], и результат сравнения считается положительным, если имеет место точное совпадение значений сравниваемых предикатов. Величина отличия значений предикатов роли не играет, номинальная шкала не позволяет оперировать такими понятиями, как степень похожести, близости, аналогичности, т. е. понятиями, на которых основаны человеческие способы рассуждений по аналогии. Ясно, что для расширения логических возможностей экспертных систем

нужно научиться измерять степень похожести знаний или ввести метрику для измерения расстояний в пространстве знаний.

Такая метрика была введена в [81]. Можно считать, что каждый предикат отражает знание эксперта о распределении возможных значений данной характеристики. Утверждение $X_3 = (a \vee b \vee c)$ равносильно утверждению, что предикат X_3 с одинаковой вероятностью $(1/3)$ может принимать одно из трех значений, а условие $X_2 = (2 \div 5)$ означает, что X_2 с вероятностью 0,25 может быть равен одному из четырех значений в диапазоне от 2 до 5. Следовательно, расстояние между одноименными предикатами можно определять через расстояние между двумя распределениями вероятностей.

Предложенная мера для измерения этого расстояния $R = f(r \times h \times w)$ учитывает расстояние r от всех элементов одного распределения до всех элементов другого, энтропийную меру h, близкую по смыслу к дисперсии распределений, и степень w пересечения распределений (величину области «консенсуса»). Эти аргументы вычисляются следующим образом.

Если функции плотности вероятности $f_1(x)$ и $f_2(x)$ отражают суждения двух экспертов о значении предиката X на участке от $x_{\min}$, т. е. минимального из возможных значений, до $x_{\max}$ — максимально возможного значения этого предиката, тогда различия двух распределений измеряются величиной

$$r = 0,5 \int_{x_{\min}}^{x_{\max}} |f_1(x) - f_2(x)|\, dx.$$

В дискретном случае разделим ось X, отображающую мнение эксперта о распределении предиката, на m частей (квантилей) так, чтобы в каждой части была заключена плотность вероятности, равная $1/m$ (см. рис. 39, *а*). Правая граница первого квантиля находится в точке x_{11}, второго — в точке x_{12}, i-го — в точке x_{1i} и т. д. до x_{1m}. Аналогично границы квантилей распределения, указанного вторым экспертом, находятся в точках $x_{21}, x_{22}, \ldots, x_{2i}, \ldots, x_{2m}$. Расстояние выражается величиной

$$r = \sum_{i=1}^{m} |x_{1i} - x_{2i}| / (x_{\max} - x_{\min}).$$

Принимается, что чем больше область пересечения двух распределений (область консенсуса), тем меньше расстояние R. Определим величину w, связанную с областью консенсуса:

$$w = 0,5 \sum_{t=1}^{T} |P_{1t} - P_{2t}|,$$

где T — число делений, равномерно распределенных вдоль оси X, а P_{1t} и P_{2t} — указанные экспертами вероятности попадания оценок в t-ю градацию (см. рис. 39, *б*).

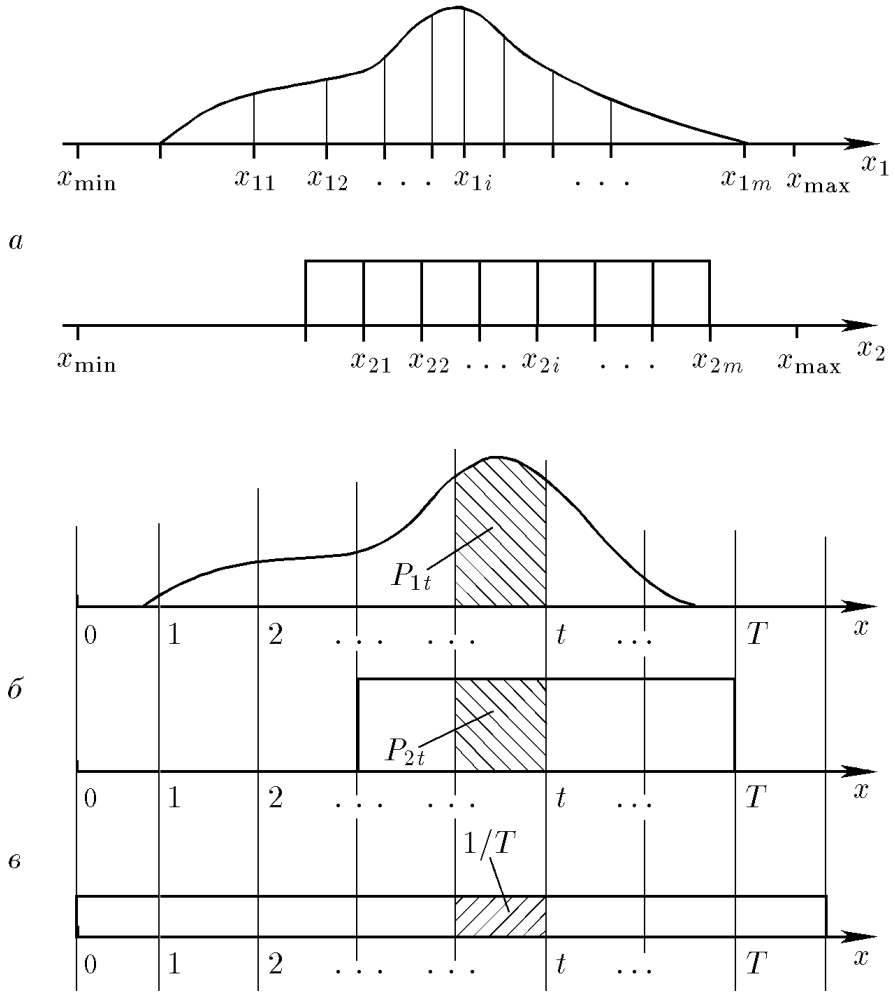


Рис. 39

Расстояние между суждениями экспертов зависит и от категоричности (h) их оценок (см. рис. 39, *в*). При одних и тех же расстояниях r и w мера R считается тем большей, чем больше распределения, отражающие мнения экспертов, отличаются от равномерного распределения по всему диапазону значений от $x_{\min}$ до $x_{\max}$. Величина h находится следующим образом:

$$h = 0,5(h_1 + h_2), \quad h_1 = 0,5 \sum_{t=1}^{T} |P_{1t} - 1/T|,$$

$$h2 = 0,5 \sum_{t=1}^{T} |P_{2t} - 1/T|.$$

Общая мера расстояния R между предикатами, отражающими знания двух экспертов о характеристике X, теперь принимается равной $r \times w \times h$. Эта мера удовлетворяет таким естественным аксиомам, как непрерывность, симметричность и транзитивность. Имеются способы вычисления меры R для порядковых и номинальных шкал.

Проверка правомочности применения описанной меры делалась путем экспертного оценивания. Были предъявлены различные пары распределений, и эксперты упорядочивали эти пары по степени их похожести, близости. Мера близости, найденная по приведенной формуле, сохраняла установленный экспертами порядок.

§ 2. Расстояние между знаниями

Зная расстояние между одноименными предикатами, можно построить меру расстояния и между двумя знаниями, включающими в себя несколько предикатов.

Если эксперт не высказывается о значении некоторой характеристики, то это означает, что он либо не знает этого значения, либо считает данную характеристику несущественной. В том и другом случае можно считать, что для него все значения характеристики X равновозможны. Это предположение позволяет находить расстояние между знаниями, если даже эксперты оперируют не полностью совпадающими наборами характеристик: распределения значений отсутствующих характеристик принимаются равномерными в диапазоне от $x_{\min}$ до $x_{\max}$.

Расстояние между двумя знаниями, представленными продукциями, состоящими из n предикатов, по аналогии с n-мерным евклидовым расстоянием находим по формуле

$$R = \sqrt{R_1^2 + R_2^2 + \cdots + R_n^2}.$$

Появление меры близости в пространстве знаний открывает большие возможности для совершенствования систем, оперирующих знаниями. Применительно к экспертным системам об этом будет сказано в главе 17.

Метрика в пространстве знаний позволяет формулировать и решать на материале базы знаний те же задачи, которые обычно решаются на материале базы данных. Появляется возможность для разработки методов анализа знаний (АЗ), аналогичных рассмотренным ранее методам анализа данных (АД). Перейдем к описанию методов анализа знаний.

Глава 15

Методы анализа знаний

§ 1. Таксономия знаний

В базах знаний (БЗ) систем искусственного интеллекта накапливается большое количество знаний, которые извлекаются из данных автоматически (с помощью логических решающих функций) либо получаются инженерами по знаниям от экспертов. Попытки разобраться в содержании этого хранилища опыта и мудрости естественно приводят к необходимости навести в нем некоторый предварительный порядок. В первую очередь возникает желание разобраться в структуре этого информационного массива, выделить в нем некоторые подмножества в чем-то похожих друг на друга знаний, найти типичных представителей каждого такого подмножества, т. е. ставится задача сделать таксономию знаний в БЗ.

С появлением меры расстояния между знаниями эту задачу можно решить с помощью описанных раньше алгоритмов таксономии. Нужно лишь предварительно вычислить расстояния между всеми парами имеющихся знаний. При этом можно учитывать расстояния между левыми частями продукций (условиями), правыми (следствиями) или продукциями целиком.

Применение алгоритмов семейства FOREL позволит получить структурные элементы (таксоны) простой и потому легко интерпретируемой формы. Для каждого таксона можно найти его центр и указать знание, наиболее близкое к нему. Центр можно использовать в качестве типичного представителя данного таксона знаний.

Таксономия при разных радиусах гиперсфер позволяет построить дерево таксонов. Его корневая вершина характеризует БЗ в целом, а вершины-листья соответствуют отдельным знаниям. На каждом иерархическом уровне дерева будет отражена своя система классификации знаний или система понятий (доменов [127]). Если фиксировать структуры, которые возникают на разных этапах наполнения БЗ, то по ним можно реконструировать историю становления данной отрасли знаний. Можно видеть, как возникали и быстро росли одни таксоны (понятия), как некоторые понятия становились чрезмерно перегруженными и делились на составные части, другие таксоны переставали развиваться и при очередной ревизии БЗ исчезали из-за потери актуальности.

Таксономия знаний алгоритмами семейства KRAB позволяет обнаруживать более сложные структурные элементы, которые отражают объективно имеющуюся структуру, но затрудняют понимание получаемых результатов человеком. Для облегчения понимания можно описать результаты набором простых фигур — гиперсфер или гиперпараллелепипедов — с помощью алгоритма ДРЭТ.

Как будет показано в главе 17, структуризация БЗ методами таксономии позволяет усовершенствовать ряд важных характеристик экспертных систем.

§ 2. Распознавание образов в пространстве знаний

Представим себе, что база знаний структурирована и на ее вход подается некоторое новое знание q. Требуется определить, к какому из имеющихся k таксонов (образов) следует отнести это новое знание. По своему смыслу это типичная задача распознавания образов.

Если исходить из гипотезы унимодальной компактности и нормального закона распределения знаний каждого образа, то в качестве эталонов можно оставить по одному (типичному) представителю на образ. В этом качестве может выступать либо одно из реально имеющихся знаний (самое близкое к центру таксона), либо искусственно синтезированное эталонное знание.

При создании искусственного эталона нужно для каждого предиката в отдельности найти его «среднее распределение», т. е. такое, расстояние от которого до соответствующих предикатов

всех знаний данного образа было бы минимальным. С этой целью нужно просуммировать плотности вероятностей для каждой из T градаций данного предиката всех m_i знаний i-го образа и затем разделить полученные суммы на m_i. В результате получится нормированное распределение предиката (см. рис. 40). Его плотность в каждой градации представляет собой среднеарифметическое значение плотностей этой градации у всех знаний образа, что обеспечивает минимум суммы расстояний от этого «центрального» предиката до соответствующих предикатов всех знаний данного образа. Синтезированное из таких искусственных предикатов знание будет приниматься нами за эталон образа. Распознавание принадлежности знания q к одному из k образов можно делать по минимуму расстояния до эталона.

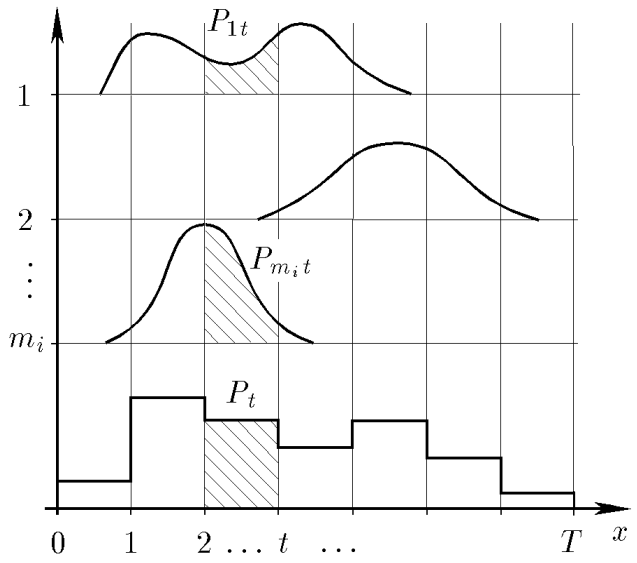


Рис. 40

Если решающее правило строить с опорой на прецеденты, то на этапе обучения для выбора необходимого и достаточного набора опорных знаний можно воспользоваться алгоритмом STOLP. На этапе распознавания можно применять правило одного или нескольких ближайших соседей.

Если ввести пороговое значение функции принадлежности, то знание q, которое по этому критерию не принадлежит ни к одному из k имеющихся образов, распознается в качестве пред-

ставителя нового $(k+1)$-го образа. Таким способом структура БЗ адаптируется к характеристикам изменяющегося потока новых знаний.

§ 3. Выбор информативного подмножества предикатов

Исходным множеством предикатов можно считать список тех предикатов, которые упоминаются в БЗ хотя бы один раз. Информативность предикатов в разных задачах анализа знаний может выглядеть по-разному. Мы рассматриваем здесь две задачи: задачу предсказания целевого предиката, заданного в непрерывной шкале (аналог задачи регрессионного анализа), и задачу предсказания целевого предиката, заданного в шкале наименований (задача распознавания).

Если считать, что предикат X не зависит от других предикатов, то его информативность в задаче регрессионного анализа можно было бы оценивать по корреляции его значений со значениями целевого предиката, если бы мы умели вычислять коэффициент корреляции между предикатами, заданными своими распределениями. Ввиду отсутствия такого аппарата опираемся на следующую гипотезу: если предикат X_p информативен, то его малые изменения вызовут малые изменения целевого предиката X_s, а большие изменения — большие. Величину изменения каждого предиката оцениваем по описанному выше расстоянию между распределениями. От корреляции между предикатами мы как бы переходим к корреляции между их первыми разностями. Если модуль этой корреляции высок, значит, предикат X_p сильно связан с целевым предикатом X_s и его следует считать важным, информативным.

Такой же подход можно использовать и для оценки информативности связки из двух, трех и т. д. предикатов с целевым предикатом X_s. Нужно вычислить расстояния между выбранными предикатами из левых частей всех пар знаний и расстояния между значениями целевого предиката для тех же пар. В результате получится две серии чисел, по модулю корреляции между которыми можно судить о влиянии выбранных предикатов на целевой предикат.

В случае задачи распознавания образов об информативности предиката X_p по отношению к целевому (номинальному) предикату X_s можно судить по тому, выполняется ли на этой паре

предикатов гипотеза компактности. Если она выполняется, то расстояния между предикатами X_p знаний, имеющих одно и то же значение целевого предиката (т. е. принадлежащих одному и тому же образу), должны быть малыми. Расстояния же между X_p из разных образов должны быть большими. Это значит, что знания одного и того же образа в пространстве X_p должны отображаться в «компактные сгустки», удаленные от сгустков представителей других образов. Если это так, тогда функции принадлежности всех знаний обучающей выборки к своим образам будут больше, чем к чужим. Количественно об информативности предиката X_p можно судить по тем же критериям, которые использовались и при распознавании данных: числу знаний, ошибочно распознаваемых по этому предикату. При использовании прецедентов о низкой информативности предиката будет говорить большое количество необходимых прецедентов.

Если условие компактности не выполняется, значит, предикат X_p, взятый в отдельности, не будет способствовать успешному распознаванию новых знаний и его следует считать неинформативным. Правда, этот предикат в группе с другими предикатами может оказаться информативным, но для проверки этого предположения нужно испытать его в составе таких групп.

Условие проверки на информативность для групп их двух, трех и большего числа предикатов то же, что и для одного: нужно проверять, выполняется ли для них гипотеза компактности.

Ясно, что большие вычислительные трудности, сопровождающие такого рода NP-полные переборные задачи, в данном случае усугубляются сложностью определения расстояний между знаниями. Дополнительные трудности могут возникнуть, если зависимости между предикатами носят нелинейный характер, который к тому же может меняться при разных значениях других предикатов (например, так меняется влияние содержания азота на рост растений при низких и высоких температурах). При этом придется пользоваться методами обнаружения кусочно-линейных зависимостей [105].

§ 4. Заполнение пробелов в базе знаний

Список знаний можно записать в виде таблицы той же формы, что и таблица «объект-свойство». Строку i в такой таблице занимает знание Z_i, а j-й столбец отражает мнения экспертов о значениях предиката X_j. Если информация о значении j-го предиката в строке i отсутствует, то это значение (P_{ij}) можно попытаться предсказать с помощью алгоритмов семейств ZET и WANGA.

В алгоритме ZET вначале отбирается компетентная подтаблица размером k строк и f столбцов. Строка Z_v $(v = 1, 2, \ldots, k)$ включается в число компетентных, если она содержит информацию о j-м предикате (P_{vj}) и входит в число k наиболее близких к строке Z_i по расстоянию $R_{i,v}$.

О компетентности столбца X_q $(q = 1, 2, \ldots, f)$, содержащего предикат P_{iq} в i-й строке, по отношению к столбцу X_j можно судить по критериям зависимости между предикатами, которые были описаны в предыдущем параграфе.

В отобранной подтаблице определяются расстояния $R_{i,v}$ от строки Z_i до всех остальных $(k-1)$ строк Z_v. Величину $L_v = (1 - R_{i,v})$ считаем мерой компетентности строки Z_v по отношению к строке Z_i. Затем синтезируется распределение P'_{ij} в виде некоторой функции от распределений j-го предиката во всех строках подтаблицы. Это прогнозное распределение должно обеспечивать минимум суммы S взвешенных расстояний от него до всех распределений, участвовавших в синтезе:

$$S = \sum_{v=1}^{k} \left(P'_{ij} - P_{vj}\right) L_v^{\alpha}.$$

Показателем степени α можно регулировать зависимость результата от компетентности L_v: при $\alpha = 0$ все распределения участвуют в синтезе прогноза с равными весами. При больших α доминируют распределения из самых близких строк.

Синтез прогнозного распределения будем делать почти так же, как делали эталонный предикат в распознавании образов (см. § 2) — путем механического усреднения взвешенных значений плотностей в каждой градации данного предиката. Если весь диапазон возможных значений предиката P_{vj} разделен на T одинаковых участков и вероятность того, что предикат принимает

значение t-й градации, равна P_{vjt}, то усредненное по всем строкам значение плотности в этой градации принимает значение

$$P'_{ijt} = \sum_{v=1}^{k} \left(P_{vjt} \times L_v^{\alpha}\right) \Big/ \sum L_v^{\alpha}.$$

Еще одно прогнозное распределение (P''_{ij}) можно получить, используя зависимости R_{jq} между j-м и всеми f остальными q-ми столбцами (предикатами) компетентной подтаблицы. Здесь суммировать с весами R_{jq} нужно распределения всех предикатов P_{iq} строки i. В качестве окончательного прогноза распределения пропущенного предиката P_{ij} можно принять усредненное по градациям значение двух полученных прогнозов: P'_{jl} и P''_{ij}.

Для оценки величины ожидаемой ошибки можно, как и в алгоритме ZET, применить метод контрольного прогнозирования известных значений предикатов в компетентной подтаблице.

Глава 16

Методы анализа структурных объектов

Объекты, с которыми мы имели дело до сих пор, были бесструктурными. Они описывались n-мерными векторами и представлялись точками в пространстве своих характеристик. Обратим теперь внимание на объекты, имеющие структуру. Примером таких объектов могут служить молекулы органических соединений, представленные графами с помеченными вершинами (атомами). Свойства молекул зависят не только от их атомного состава, но и от характера (пространственной структуры) связей между отдельными атомами. Свойства составных частей (атомов) позволяют описывать их в качестве бесструктурных, но свойства объекта в целом (молекулы) существенно зависят от структурной организации.

Анализ структурированных объектов включает в себя аппарат для описания информативных особенностей структур в виде их метрических и топологических свойств: диаметр графа, число вершин, наличие специфичных связей, клик, колец. Использование этого аппарата позволяет решать задачи нахождения изоморфных графов и определения наибольших общих частей двух графов, измерять расстояния между графами. На этой базе делается классификация молекул (таксономия графов), решаются, например, задачи обнаружения закономерных связей между структурными свойствами и биологической активностью веществ (задача распознавания), поиска структурных фрагментов, наиболее сильно влияющих на заданное свойство веществ (задача выбора информативных характеристик), предсказания свойств синтези-

руемых соединений (задача прогнозирования). Для решения этих задач разработаны эффективные алгоритмы, реализованные в программной системе СИСТРАН [92, 117].

Имеются объекты, важным свойством которых является не пространственная, а временная организация их составных частей. Это, например, тексты — литературные, генетические или музыкальные. Для их анализа разрабатываются меры близости между символьными последовательностями [131], что позволяет делать классификацию текстов, находить межтекстовые или межъязыковые аналогии. Решаются задачи обнаружения повторов, выявления структурных единиц из слитного текста, видов вариативности, иерархической организации текстов и пр. [10, 11, 44].

При распознавании речевых сигналов мы тоже сталкиваемся с проблемами анализа объектов, имеющих временну́ю структуру. Близкие особенности имеют объекты, представляющие собой динамические траектории развития процессов. При сравнении двух структурных объектов или динамических траекторий возникает проблема выравнивания их протяженности по оси времени. Простая схема линейного сжатия или растяжения обычно не дает хороших результатов. Например, одно и то же слово, состоящее из двух слогов, один диктор может произнести, растягивая первый слог, а другой — растягивая второй слог. Чтобы наилучшим образом совместить друг с другом одинаковые фонемы этих слов, необходимо для одного из них сделать нелинейное преобразование оси времени: одни его участки растянуть, а другие участки сжать. И только после такого наилучшего совмещения похожих частей друг с другом можно определить степень похожести этих слов. Описанная здесь нелинейная нормализация объектов по времени и определение расстояния между сравниваемыми объектами делается одним из двух методов: методом динамического программирования (ДП) или методом скрытых марковских процессов (СМП). Опишем суть этих методов.

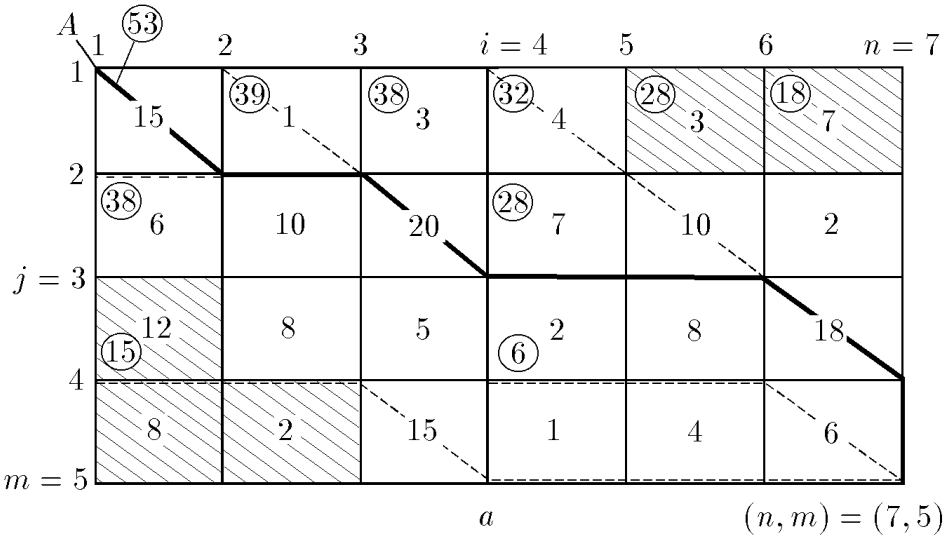
§ 1. Метод динамического программирования

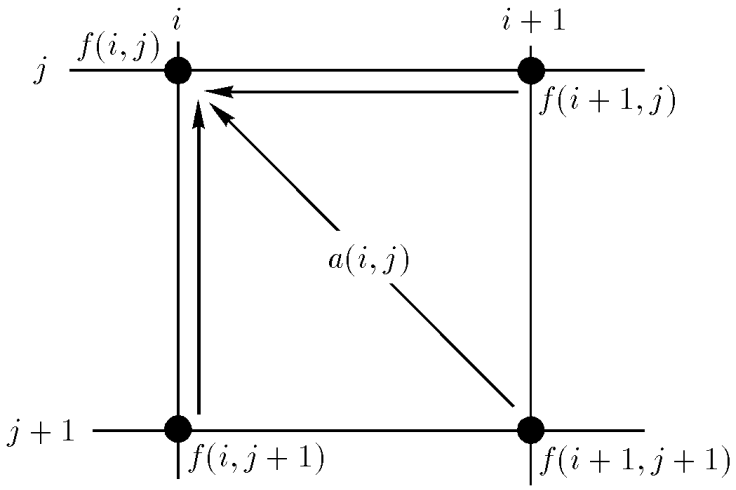
С общими идеями динамического программирования (ДП) можно познакомиться по работам [13, 30]. Здесь мы опишем его применительно к распознаванию слов по последовательности коротких отрезков речевого сигнала (сегментов), как это впервые было предложено в [134] и опробовано в [28, 32].

Каждый сегмент описывается вектором своих (например, спектральных) характеристик. Через евклидово расстояние между сегментами мы можем найти меры a_{ik} близости (сходства) каждого i-го сегмента со всеми k остальными. Пусть одно слово состоит из m сегментов, а второе — из n. Расположим последовательность сегментов первого слова по вертикали, а второго по горизонтали, как показано на рис. 41, *а*. Начала слов находятся в точке A. Сформируем матрицу размером $m \times n$, элементами которой служат меры a_{ik} сходства между i-м сегментом первого слова и k-м сегментом второго слова. Сходство между первыми сегментами равно a_{11}, сходство между последними равно a_{mn}. Обработка этой матрицы состоит из последовательного выполнения простой базовой процедуры над каждой ее ячейкой. Поясним эту процедуру с помощью рис. 41, *б*.

Пусть нам известны значения некоторой функции f во всех вершинах ячейки, кроме вершины (i, j). Тогда значение функции $f(i, j)$ принимается равной максимальному значению из трех следующих величин: $f(i, j+1)$, $f(i+1, j)$ и $f(i+1, j+1) + a(i, j)$, т. е. соседу справа, соседу снизу или соседу по диагонали, но с добавкой меры сходства, записанной в этой ячейке.

Примем, что значение функции у всех самых правых и всех самых нижних вершин матрицы равно нулю, а значения f во всех других вершинах нужно найти. Три вершины из четырех известны пока только для одной ячейки — самой правой и самой нижней. В нашем примере это ячейка $(4, 6)$. При выполнении базовой процедуры мы найдем, что наибольшей из трех соседних является диагональная величина, и потому функция $f(4, 6) = f(5, 7) + a(4, 6) = 0 + 6 = 6$. Отметим этот факт пунктирной диагональю, пересекающей ячейку $(4, 6)$. Теперь появились условия для выполнения базовой процедуры для двух соседних ячеек: $(3, 6)$ и $(4, 5)$. В ходе выполнения процедуры над вершиной $(4, 5)$ обнаружим, что максимальное значение функции f имеет ее правый сосед. Присвоим ей значение $f(4, 5) = 6$ и отметим этот факт горизонтальной пунктирной линией $(4, 5)$–$(4, 6)$. Аналогичная процедура над вершиной $(2, 5)$ даст результат в виде $f(3, 6) = 18$ и диагональной пунктирной линии через ячейку $(3, 6)$. После этого появляется возможность обрабатывать уже три соседних ячейки. И так до последней ячейки $(1, 1)$, для которой в нашем примере получаем значение функции $f(1, 1) = 53$.





б

Рис. 7.11

Легко видеть, что это значение является суммой мер сходства из тех ячеек, которые перечеркнуты пунктирными диагоналями. Движение от конечной вершины $(5,7)$ к начальной $(1,1)$ проходит по маршруту, обозначенному непрерывной линией. При этом движение по горизонтали и по вертикали величины f не меняет,

она растет только за счет ячеек, пересекаемых по диагонали. Переход по горизонтали означает, что мы как будто игнорируем соответствующий сегмент горизонтального слова, сокращаем его длительность на этом участке. Вертикальный переход, наоборот, эквивалентен растяжению данного участка горизонтального слова. Ломаная траектория между начальной и конечной точками как бы «нанизывает на шампур самые крупные кусочки» мер сходства.

Функция f является мерой похожести двух сравниваемых слов. Чтобы мера сходства не зависела от длины слов, разделим f на число сегментов более длинного из них. В итоге нормированной мерой похожести двух слов при самом лучшем варианте нелинейной деформации оси времени считаем величину $F = f/n$.

Описанный алгоритм требует больших затрат машинного времени: количество операций пропорционально квадрату длины слова (n^2). Если различия в темпах произнесения разных участков одного и того же слова будут не слишком большими, то линии наилучшего совмещения сегментов будут лежать в окрестностях диагонали, соединяющей вершины $(1,1)$ и (m,n). В расчете на это объем вычислений можно сократить, не проводя их для вершин, удаленных от диагонали на расстояние, большее некоторого порога v^*. Расстояние вершины от диагонали можно оценивать величиной $v = |i/n - j/m|$. В нашем примере при $v^* = 0,5$ базовую процедуру можно не делать для заштрихованных ячеек. Значения f для этих вершин принимаются равными нулю. Если сравниваются слова, содержащие по 30–40 сегментов, то такой прием без заметного риска потерять оптимальное решение позволяет сократить объем вычислений примерно вдвое.

В процессе распознавания нужно сравнить распознаваемое слово со всеми эталонами, что при большом словаре может оказаться неприемлемым по времени. Для значительного сокращения времени счета можно применить комбинированный метод принятия решений [67, 69], в котором сначала вычисляются приближенные меры сходства между словами, затем среди всех эталонов выбирается несколько наиболее сильных конкурентов и сравнение между ними делается описанным выше методом ДП.

В данной задаче приближенная мера сходства вычислялась следующим образом. В качестве эталонов использовалось по одной реализации на каждое распознаваемое слово. При проверке гипотезы о принадлежности контрольного слова к очередному эталонному слову совмещались их начальные сегменты и вычи-

слялись меры сходства для нескольких первых сегментов с одинаковыми номерами. По полученным мерам сходства начальных участков отбиралось 10 % наиболее правдоподобных гипотез, окончательный выбор из которых делался уже с применением ДП.

§ 2. Метод скрытых марковских процессов (СМП)

Пусть эталонное слово $C_э$ состоит из трех фонем, а распознаваемое слово $C_р$ может состоять из нескольких фонемоподобных сегментов — от одного до шести. Так может выглядеть короткое слово, если некоторые его соседние сегменты (два или больше) являются представителями одной и той же фонемы. С другой стороны, короткая последовательность $C_р$ может быть порождена и длинным словом, если некоторые его фонемы оказались отсутствующими («проглоченными» в скороговорке). Все возможные варианты событий показаны на рис. 42.

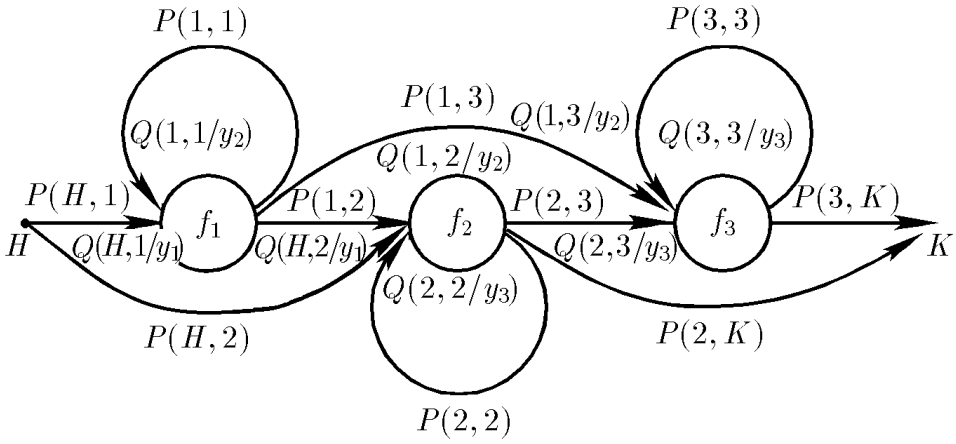


Рис. 42

Процесс, начатый в точке H, с вероятностью $P(H,1)$ может оказаться в вершине f_1. Здесь происходит сравнение характеристик y_1 первого сегмента с характеристиками эталона первой фонемы и выясняется, что вероятность принадлежности первого сегмента к первой фонеме равна $Q(H,1/y_1)$. Но процесс из начальной точки может с вероятностью $P(H,2)$ перейти сразу в вершину f_2. Для этого случая получается, что вероятность принадлежности первого сегмента второй фонеме равна $Q(H,2/y_1)$.

Из вершины f_1 процесс в форме второго сегмента y_2 может перейти снова в эту же вершину. Вероятность этого события равна $P(1,1)$, а вероятность принадлежности такого сегмента к первой фонеме равна $Q(1,1/y_2)$. Но процесс может перейти в вершину f_2 или, пропустив ее, оказаться в вершине f_3. Вероятности этих событий равны $P(1,2)$ и $P(1,3)$, а вероятности того, что этот сегмент будет распознан в качестве представителя второй и третьей фонемы, равны $Q(1,2/y_2)$ и $Q(1,3/y_2)$ соответственно. Из вершины f_2 есть три выхода: снова на себя, на вершину f_3 или на окончание слова (паузу). Из вершины f_3 имеется два допустимых перехода: цикл на себя и выход на паузу. Таким образом, самый короткий возможный путь от начала до конца слова пролегает через точки H–f_2–K. Самый длинный путь от H до K можно пройти многими маршрутами; они показаны на рис. 43. Здесь вдоль вертикальной оси размещены фонемы эталона, а вдоль горизонтальной оси — сегменты контрольного слова. Точка H обозначает начальную паузу перед словом, а точки K_1–K_6 — паузу после окончания слова соответствующей длины.

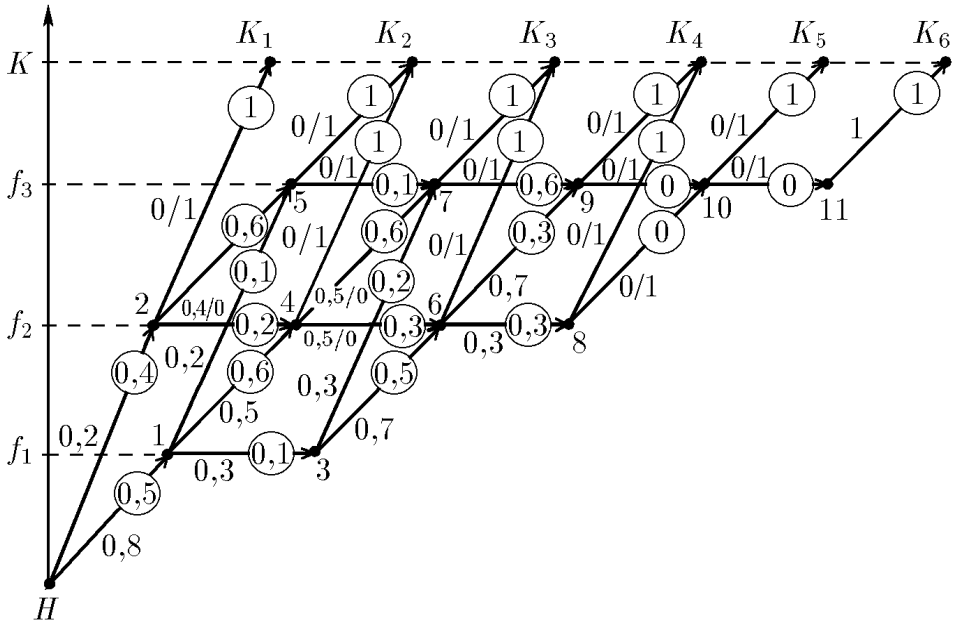


Рис. 43

Теперь вспомним, что марковской называется последовательность состояний (процесса или автомата), в которой текущее состояние зависит от последнего предыдущего и не зависит от будущего. Это значит, что решение о принадлежности сегмента y к той или иной фонеме нужно принимать не только по похожести характеристик y на эталон данной фонемы, но и с учетом того, какой фонеме принадлежал предыдущий сегмент. В соответствии с этой стратегией, если процесс в момент времени $(t-1)$ находился в состоянии f_i с вероятностью $F(i)$, а в момент t приобрел характеристики y, то вероятность того, что он находится в состоянии f_j, имеет значение

$$F(j) = F(i) \times P(i,j) \times Q(i,j/y).$$

Эталоны событий (фонем), по которым вычисляются величины $Q(i,j/y)$, а также вероятности переходов $P(i,j)$ определяются на материале обучающей выборки. Вероятности событий $F(i)$ вычисляются по мере продвижения процесса (т. е. поступления сегментов). Вероятность $F(H)$ того, что процесс начинается с точки H (с паузы), естественно считается равным единице. Аналогично, вероятности перехода от последнего сегмента к паузе $P(i,K)$ и возникновения паузы после последнего сегмента $Q(i,K/y)$ также приравниваются к единице. Этих исходных данных достаточно для начала процесса распознавания.

Поясним это на примере распознавания контрольного слова, состоящего из 4-х сегментов, путем сравнения его с эталонным трехфонемным словом. Таблица происходящих при этом событий приведена на рис. 43. Числа вдоль ребер соответствуют переходным вероятностям $P(i,j)$, а числа в кружочках — величинам $Q(i,j/y)$.

При появлении первого сегмента с характеристиками y_1 программа вычисляет его принадлежность Q к первой $(0,5)$ и второй $(0,4)$ фонеме. То, что их сумма меньше единицы, означает, что с ненулевой вероятностью сегмент y_1 принадлежит и некоторым другим эталонам. С учетом вероятностей переходов P $(0,8$ и $0,2)$ находятся вероятности событий $F(1)$ и $F(2)$: $F(1) = 0,8 \times 0,5 = 0,4$, $F(2) = 0,2 \times 0,4 = 0,08$.

Сумма вероятностей перехода из вершины 1 в вершины 3, 4 и 5 равна единице. Если для распознавания предъявлен всего один сегмент, возможен только один маршрут: H–2–K_1. В этом случае вероятность перехода из вершины 2 в конечную вершину K_1

равна единице, а вероятности других переходов равны нулю. Если же первый сегмент не является и последним, то тогда вероятность $P(2, K_1) = 0$, а сумма вероятностей $P(2,4)$ и $P(2,5)$ равна единице.

Второй сегмент y_2 сравнивается с эталонами всех трех фонем данного слова. Учет полученных значений Q, вероятностей переходов P и вероятностей событий F в предыдущий момент времени позволяет определить вероятности событий $F3$, $F4$ и $F5$:

$$\begin{aligned}
F(3) &= F(1) \times P(1,3) \times Q(1,3/y_2) = 0,012;\\
F(4) &= F(1) \times P(1,4) \times Q(1,4/y_2)\\
&\qquad + F(2) \times P(2,4) \times Q(2,4/y_2) = 0,1264;\\
F(5) &= F(1) \times P(1,5) \times Q(1,5/y_2)\\
&\qquad + F(2) \times P(2,5) \times Q(2,5/y_2) = 0,0368.
\end{aligned}$$

Если бы второй сегмент был последним, то переходные вероятности $P(4, K_2)$ и $P(5, K_2)$ приравнивались к единице. Но он не последний, и потому вероятность единственно возможного перехода из вершины 5 в вершину 7 равна единице. Единице же равна и сумма вероятностей перехода из вершины 4 в вершины 6 и 7. С учетом этого находим

$$\begin{aligned}
F(6) &= F(3) \times P(3,6) \times Q(3,6/y_3)\\
&\qquad + F(4) \times P(4,6) \times Q(4,6/y_3) = 0,02316;\\
F(7) &= F(3) \times P(3,7) \times Q(3,7/y_3) + F(4) \times P(4,7) \times Q(4,7/y_3)\\
&\qquad + F(5) \times P(5,7) \times Q(5,7/y_3) = 0,04232.
\end{aligned}$$

Четвертый сегмент сравнивается также с эталонами второй и третьей фонем, что позволяет найти вероятности событий $F8$ и $F9$:

$$\begin{aligned}
F(8) &= F(6) \times P(6,8) \times (6,8/y_4) = 0,0020844;\\
F(9) &= F(6) \times P(6,9) \times Q(6,9/y_4)\\
&\qquad + F(7) \times P(7,9) \times Q(7,9/y_4) = 0,0302556.
\end{aligned}$$

Последовавшая затем пауза означает, что процесс текущего посегментного распознавания окончен, и нужно переходить к оценке вероятности данного эталонного слова в целом. Вероятность

достижения конечной вершины K_4 равна $F(K_4) = F(8) \times 1 \times 1 + F(9) \times 1 \times 1 = 0,03234$. Чтобы можно было сравнивать такие вероятности для слов с разным числом сегментов, из полученного значения $F(K_n)$ нужно извлечь корень n-й степени, где n — максимальное число сегментов в эталонном или контрольном слове. В нашем случае более длинным оказалось контрольное слово, и потому из $F(K_4)$ нужно извлечь корень четвертой степени. В результате вероятность принадлежности слова к данному эталону оказывается равной $0,424$.

Впервые сетевой метод для распознавания устных слов по последовательности сегментов был применен в работах [33, 34]. В завершенном виде в форме описанного выше алгоритма Виттерби метод СМП приведен в [110, 142].

Сравнивая методы ДП и СМП, можно отметить, что метод СМП учитывает природу речевого сигнала более полно, чем ДП: переходные вероятности соответствуют закономерностям сочетания разных фонем в речевом языке; эталоны, зависящие от предшествующей фонемы, отражают коартикуляцию. Эксперименты показали, что метод СМП дает более высокие результаты. Однако эта полнота дается не даром. Для обучения по методу ДП достаточно однократного произнесения слов распознаваемого словаря. Для обучения же по методу СМП приходится анализировать большое количество реализаций каждого слова. После обучения нужно помнить матрицы вероятностей переходов и эталоны каждой фонемы в виде вариантов, зависящих от предшествовавшей фонемы. В итоге для СМП требуется гораздо больший объем обучающего материала, времени на обучение, памяти и времени на распознавание. В системах с подстройкой под диктора громоздкую процедуру обучения автомата приходится делать заново для каждого нового диктора. По этой причине в современных системах распознавания речи применяются как методы СМП, так и методы ДП.

§ 3. *D*-алгоритм для таксономии траекторий

Если в моменты времени $t = 1, 2, \ldots, T$ наблюдаются не один, а несколько (m) процессов или объектов, описываемых n характеристиками, то протокол наблюдений представляет собой «куб данных» размером $m \times n \times T$. Каждый объект за время T описывает в пространстве n свойств некоторую n-мерную траекторию. Если мы хотим выделить подмножества (таксоны) объек-

тов с одинаковыми или похожими траекториями, то нам нужно найти расстояния между всеми парами траекторий и применить один из алгоритмов таксономии. В простейшем случае в каждый момент времени можно вычислить евклидово расстояние между двумя объектами и сумму этих расстояний за все время T считать мерой расстояния между двумя траекториями.

Однако нередко встречаются случаи, когда похожие по своей природе процессы протекают с разной скоростью, и тогда простое совмещение их во времени даст большое значение расстояния. Иногда похожие процессы начинают наблюдаться в разные фазы их развития и не ясно, какие участки двух процессов совмещать друг с другом при вычислении расстояния между ними.

Именно для таких сложных случаев был разработан D-алгоритм [87]. Его основное отличие состоит в том, что при определении расстояния между траекториями применяется описанный выше метод динамического программирования, при котором сдвигается начало процесса и нелинейно растягивается или сжимается одна траектория по оси времени с целью получения наилучшего совмещения двух траекторий. Величина, пропорциональная усилию по деформации оси времени, служит мерой расстояния между траекториями. В результате среди кривых, приведенных на рис. 44, удается в один таксон поместить кривые 1 и 3, а кривые 2 и 4 — во второй. Это выглядит более естественно по сравнению с вариантом объединения в один таксон кривых 1 и 2, что получалось бы без динамического программирования.

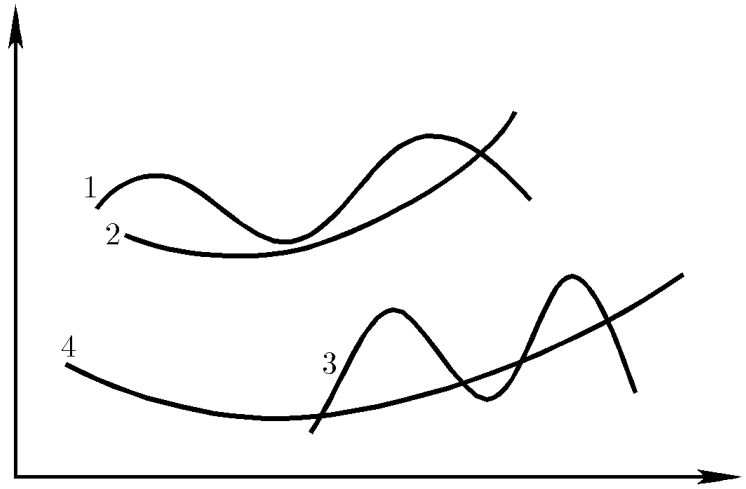


Рис. 44

D-алгоритм может быть полезным для анализа сложных динамических процессов, например процессов протекания различных заболеваний у групп больных, состояния которых наблюдаются в последовательные моменты времени. При этом одно и то же заболевание протекает у разных больных по-разному (с разной скоростью в разные фазы болезни), наблюдения за больными начинаются в разные фазы болезни. D-метод позволяет синхронизировать наблюдаемые процессы, определять фазу заболевания и прогнозировать наиболее вероятное состояние больного в следующие моменты времени.

§ 4. Иерархические структуры

В последнее время в области анализа данных отмечается рост интереса к анализу так называемых символьных объектов [50], с помощью которых описываются разного рода обобщенные характеристики некоторого массива исходных данных. Символьным объектом может быть обнаруженная в этом массиве логическая закономерность типа «если ... то ...», направленный граф, отражающий зависимость одних объектов от других и т. п. В частности, результаты иерархической таксономии выявляют структуру множества объектов, которую можно наглядно представить графически в виде иерархического дерева, начальные вершины (листья) которого отображают все объекты исходного множества, промежуточные вершины (ветви) описывают все более крупные таксоны (концепты), а конечная вершина (корень) представляет собой объединение всего исходного множества объектов в один таксон. Такую форму могут иметь, например, описания структур баз данных, технических систем, организационных структур предприятий. При изучении нескольких различных массивов данных может потребоваться сравнение между собой их внутренних структур, что приводит к необходимости измерять степень близости, похожести между иерархическими структурами.

В работах [76, 78] и предыдущей главе описаны методы анализа символьных объектов, имеющих форму конъюнкций типа «если ... то ...». Данная работа посвящена введению меры близости или расстояния на множестве символьных объектов типа иерархий [88].

Определим понятие «иерархия». Обозначим через W конечное множество объектов, $W = w_1, w_2, \ldots, w_l, \ldots, w_q$, а через H —

множество непустых частей множества W , называемых *таксонами* и обозначаемых через h. Теперь воспользуемся определением иерархии, данным в [50].

Иерархией H множества W называется множество подмножеств W таких, что:

1) $\forall w \in W\{w\} \in H$ (терминальные вершины (листья) — одноэлементные множества);

2) $W \in H$ (наибольший таксон (корень) содержит все элементы W);

3) для любых вершин $h, h' \in H$ мы имеем либо $h \cap h' = 0$, либо $h \subset h'$, либо $h' \subset h$.

Таким образом, иерархия — это многоуровневая структура, в которой объекты, находящиеся в одном таксоне на некотором (j-м) уровне, остаются в одном таксоне на $(j+1)$-м и всех других более высоких уровнях. Первому уровню соответствуют терминальные вершины (п. 1 в определении иерархии), а последнему, максимальному, уровню (обозначим его через m) — наибольший таксон, содержащий все элементы W; этот таксон можно обозначить тем же символом W (п. 2 в определении иерархии). На каждом уровне происходит или не происходит объединение таксонов (п. 3 в определении иерархии).

Обозначим точкой каждый таксон иерархии. Тогда отрезки, соединяющие эти точки (или вершины иерархии), передают порядок образования таксонов, который отвечает пп. 1–3 определения. На рис. 45 показана, например, иерархия H_1, которая содержит три уровня с шестью, тремя и одной вершиной на уровнях $j = 1, 2, 3$ соответственно. Иерархия H_2 содержит десять терминальных вершин $\{w_k\}$, $k = 1, 2, \ldots, l, \ldots, q$, $q = 10$, первого уровня, шесть таксонов на втором уровне, четыре таксона на уровне $j = 3$, два таксона на уровне $j = 4$ и один таксон на верхнем пятом уровне ($j = m_1 = 5$).

Можно представить себе и вырожденный случай иерархии, свернутую в одну изолированную вершину на первом уровне (иерархия H_3 на рис. 45). Каждая l-я вершина j-го уровня графа H_t (ljt) может характеризоваться структурным индексом a_{ljt}, равным числу примыкающих к ней ребер. В нашем примере $a_{1lt} = 1$ для всех терминальных вершин иерархий H_1 и H_2, индексы их других вершин принимают значения от 2 до 4: $a_{121} = 2$, $a_{132} = 3$, $a_{122} = 4$ и т. д., а индекс $a_{113} = 0$.

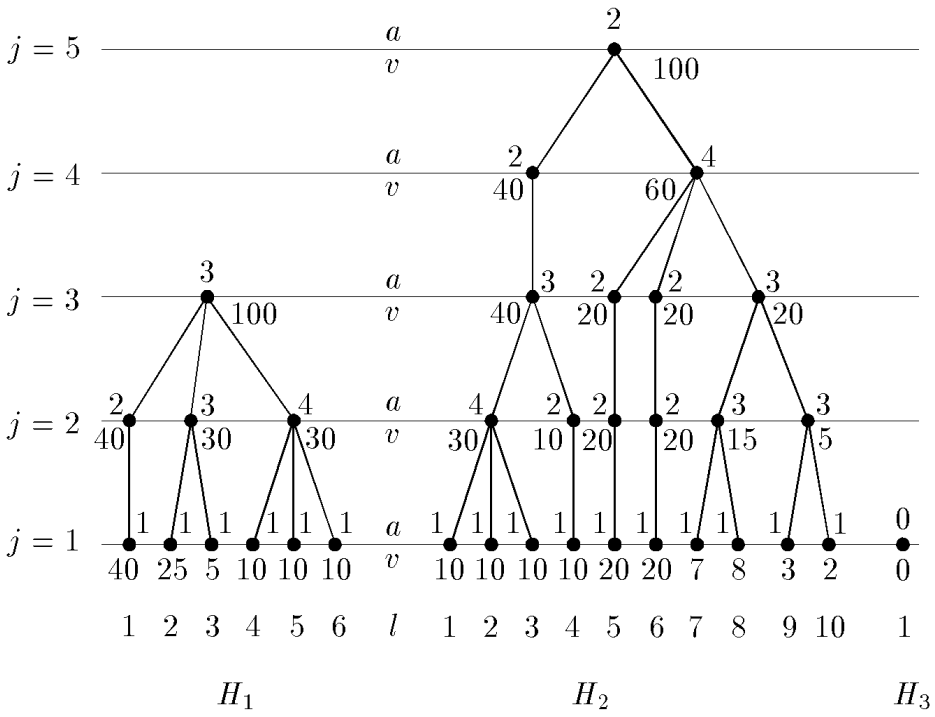


Рис. 45

Терминальные вершины могут быть равнозначными, но могут и отличаться по значимости (насыщенности, весу). Так может быть, например, когда листья представляют собой таксоны с разным числом входящих в них объектов. Весовой индекс вершины обозначим через v_{ljt}. Значения весовых индексов терминальных вершин, нормированных так, чтобы их сумма на каждом иерархическом уровне равнялась 100, указаны на рис. 45 в строке v (под чертой). Весовой индекс нетерминальных вершин уровня j равен сумме индексов вершин, входящих в эту вершину из предыдущего уровня $j-1$.

§ 5. Расстояние между иерархиями

Как определить расстояние между подобными иерархическими структурами? В работах [18, 19, 136] предлагается мера близо-

сти между такими графами с поименованными вершинами, списки вершин в которых совпадают или мало различаются. Здесь мы попытаемся предложить решение проблемы измерения расстояний между иерархиями с объектами произвольного состава.

Естественным образом возникает идея оценить расстояния между иерархиями через сложность превращения одной иерархии в другую, добавляя или убирая вершины и связи между ними, где это необходимо, т. е. применяя набор так называемых редакционных операций. Каждая операция имеет свою стоимость (c). Оптимальному переводу соответствует последовательность элементарных операций с минимальной суммарной стоимостью, которая носит название *редакционного расстояния* [131]. Связанную с ним переменную d — характеристику расстояния или различия во внешнем виде двух иерархических структур — назовем *расстоянием по виду структур*. С другой стороны, неплохо было бы учитывать и вес элементов, собираемых в таксоны на каждом уровне иерархий. Связанную с этим переменную — характеристику различия по насыщенности или весу таксонов двух иерархических структур — обозначим символом r.

Перейдем к математической постановке задачи нахождения характеристик расстояния d и r.

5.1. Расстояние по виду структуры. Пусть нам даны две иерархии H_1 и H_2 с числом уровней m_1 и m_2 соответственно. Будем рассматривать иерархию H_t как (упорядоченное) множество уровней с $j = 1$ по $j = m_t$, а каждый уровень — как совокупность h_{jt} расположенных на нем q вершин (таксонов) h_{ljt}:

$$h_{jt} = \{h_{1jt}, h_{2jt}, \ldots, h_{ljt}, \ldots, h_{qjt}\}.$$

В процессе превращения одной иерархии в другую потребуется вершину h_{lj1} заменить на вершину h_{lj2}. Считаем, что стоимость такой редакционной операции имеет значение $c(lj1, lj2) = a_{lj1} - a_{lj2}$. Для оценки стоимости замен всех q_1 вершин уровня j_1 первой иерархии на все q_2 вершины уровня j_2 второй иерархии используем простой алгоритм похожих пар. Для этого вначале сделаем число вершин в сравниваемых уровнях одинаковым и равным q, добавив к уровню с меньшим числом вершин f пустых вершин, где $f = |q_1 - q_2|$. *Пустой* называем вершину с индексом $\{a_{ljt} = 0\}$.

Затем для вершины h_{lj1} находится самая похожая на нее вершина h_{lj2}, т. е. такая, редакционное расстояние $c(1)$ до которой

минимально. Величина $c(1)$ добавляется в счетчик суммарного расстояния $c(j_1, j_2)$ между данными уровнями, и эта пара самых похожих вершин из дальнейшего рассмотрения исключается. Среди оставшихся вершин снова ищется самая похожая пара, величина их редакционного расстояния $c(2)$ добавляется к счетчику $c(j_1, j_2)$, а эта пара также исключается из дальнейшего анализа. Такая процедура нахождения на каждом (l-м) шаге самой похожей пары, добавления к счетчику $c(j_1, j_2)$ величины $c(l)$ и исключения l-й пары выполняется q раз. В итоге получается величина редакционного расстояния между двумя этими уровнями:

$$c(j_1, j_2) = \sum_{l=1}^{q} c(l).$$

Проведя сравнение всех m_1 уровней первой иерархии со всеми m_2 уровнями второй, мы получим матрицу $C(1,2)$ с номерами строк $1, 2, \ldots, j_1, \ldots, m_1$ и номерами столбцов $1, 2, \ldots, j_2, \ldots, m_2$. На пересечении строки j_1 и столбца j_2 будет стоять величина (частного) редакционного расстояния $c(j_1, j_2)$ между уровнями j_1 и j_2 сравниваемых иерархий (см. табл. 16).

Таблица 16

Редакционное расстояние между уровнями иерархий H_1 и H_2

j_1		j_2				
		1	2	3	4	m_2 5
	1	(4)	10	8	8	6
	2	1	(7)	(3)	3	7
m_1	3	11	13	7	(3)	(1)

Редакционным расстоянием d между иерархиями H_1 и H_2 называем стоимость не любого, а оптимального перевода уровней иерархии H_1 в соответствующие уровни иерархии H_2. Этот перевод будем искать с помощью метода динамического программирования [13,30]. В результате находим путь на матрице $C(1,2)$, соединяющий клеточку $(1,1)$ с клеточкой (m_1, m_2) и проходящий

через соседние клеточки либо по горизонтали слева направо, либо по вертикали сверху вниз, либо по диагонали вправо вниз. На каждом шаге прибавляем к счетчику расстояния $d(Q)$ величину $k \times c(j_1, j_2)$, взятую из той клеточки, через которую проходит путь. Весовой коэффициент k равен единице при переходе по диагонали и двум при переходе по горизонтали или вертикали (схема динамического программирования 2–1–2). Наша цель состоит в поиске такого пути Q, который набирает минимальную сумму $d(Q)$ стоимостей частных взвешенных редакционных расстояний. Этот путь показан в табл. 16. Он дает величину $d(Q) = 4 + 7 + 2 \times 3 + 3 + 2 \times 1 = 22$.

Для нормировки редакционного расстояния $d(Q)$ в пределы от нуля до единицы нужно $d(Q)$ разделить на *коэффициент нормализации* D, который представляет собой наибольшее редакционное расстояние от иерархий H_1 и H_2 до некоторой предельно отличающейся от них иерархии. В качестве таковой принимается вырожденная иерархия H_3. Она состоит из одной вершины первого уровня с нулевым числом входящих в нее ребер $\{a_{11} = 0\}$ и с нулевым индексом насыщенности $\{v_{11} = 0\}$.

Матрица частных редакционных расстояний для сочетания всех уровней иерархий H_1 и H_3 приведена в табл. 17, *а*, для иерархий H_2 и H_3 — в табл. 17, *б*.

Таблица 17

Матрица частных редакционных расстояний по виду структур между иерархиями (H_1, H_3) и (H_2, H_3)

j_3	j_1		
	1	2	3
1	(6)═(9)═(3)		

а

j_3	j_2				
	1	2	3	4	5
1	(10)═(16)═(10)═(6)═(2)				

б

Оптимальные пути здесь идут только по горизонтали, так что редакционное расстояние между H_1 и H_3 выражается величиной $D_{13} = 6 + 2 \times 9 + 2 \times 3 = 30$, а расстояние $D_{23} = 10 + 2 \times (16 + 10 + 6 + 2) = 78$. Следовательно, в качестве нормирующего коэффициента выбирается $D = 78$ и редакционное расстояние между иерархиями H_1 и H_2 по виду структуры имеет значение $d = d(Q)/D = 22/78 = 0,282$.

5.2. Расстояние по весовым индексам. Теперь опишем процесс нахождения другой характеристики расстояния (r) между иерархиями по весовым индексам входящих в их состав таксонов. Здесь также применяем метод динамического программирования, так как идея состоит в том же самом желании оптимально преобразовать все уровни одной иерархии в соответствующие уровни другой. Для оценки редакционных расстояний между уровнем j_1 первой иерархии и уровнем j_2 второй воспользуемся описанным выше алгоритмом похожих пар. Если число таксонов в данных уровнях не одинаково, т. е. если $q(1) \neq q(2)$, то устраняем этот «дефект» путем добавления к уровню с меньшим числом таксонов недостающего числа $f = |q(1) - q(2)|$ таксонов с нулевым весом $v = 0$. После этого находятся самые похожие пары вершин (таксонов) сравниваемых уровней и частные редакционные расстояния между этими вершинами суммируются в накопитель редакционного расстояния между рассматриваемыми уровнями:

$$c(j_1, j_2) = \sum_{l=1}^{q} |v_{lj1} - v_{lj2}|.$$

Как и в предыдущем случае, формируем матрицу (см. табл. 18) редакционных расстояний размером $j_1 \times j_2$ и ищем на ней оптимальный путь Q перевода одной иерархии в другую. Применяем такую же схему динамического программирования 2–1–2 и находим величину редакционного расстояния $r(Q)$ (в нашем примере оптимальный путь показан в табл. 18 и $r(Q) = 230$).

Таблица 18

Матрица редакционных расстояний
по весовым индексам между иерархиями H_1 и H_2

j_1	j_2				
	1	2	3	4	5
1	(50)	(30)	40	70	120
2	100	60	(40)	60	120
3	160	140	120	(80)	(0)

Наибольшая величина расстояния R была бы найдена при сравнении заданных иерархий H_1 и H_2 с наиболее на них не похожей пустой иерархией H_3. Частные расстояния r между уровнями иерархии H_1 и H_3 представлены в табл. 19, *а*, а между иерархиями H_2 и H_3 — в табл. 19, *б*.

Таблица 19

Частные расстояния между уровнями иерархии по весовым индексам между иерархиями (H_1, H_3) и (H_2, H_3)

j_3	j_1		
	1	2	3
1	100	100	100

а

j_3	j_2				
	1	2	3	4	5
1	100	100	100	100	100

б

Легко видеть, что расстояние от иерархии H_3 до любой иерархии H_t с числом уровней m_t имеет значение $R(t,3) = 100 + 2 \times 100 \times (m_t - 1)$. В нашем примере $R(1,3) = 500$, а $R(2,3) = 900$, так что редакционное расстояние между H_1 и H_2 по насыщенности таксонов находится следующим образом: $r = r(Q)/R = 230/900 = 0,256$.

Общее редакционное расстояние P между двумя иерархиями примем равной средней величине расстояний d и r: $P = (d + r)/2$. В нашем случае $P = (0,282 + 0,256)/2 = 0,272$.

§ 6. Таксономия иерархий

Использование описанной выше меры расстояния между иерархическими структурами позволяет формулировать и решать многие задачи того же типа, что решаются при анализе данных.

В частности, можно поставить задачу таксономии структурных объектов. Пусть имеются описания структур управления большого числа предприятий, и нам нужно понять, есть ли какие-либо структуры, которые встречаются чаще, чем другие, можно ли сформировать разумную классификацию структур или пространство структур заполнено ими равномерно и обнаружить их компактные сгустки не возможно.

Чтобы ответить на эти вопросы, достаточно вычислить расстояния между всеми парами структурных объектов и затем применить один из методов таксономии, описанных в этой книге.

В качестве типичных структур, представляющих k таксонов, можно для каждого таксона использовать по одной структуре, наиболее близкой к его центру. Сумма расстояний от этой структуры до остальных структур таксона будет минимальной.

§ 7. Распознавание иерархических структур

Если исходить из гипотезы локальной компактности, то можно применять метод ближайшего соседа, опираясь при этом на все реализации обучающей выборки или на предварительно выбранное множество прецедентов. Отбор прецедентов можно делать с использованием алгоритма STOLP.

Более сложные проблемы возникают при опоре на гипотезу унимодальной компактности. Здесь для выработки оптимального решающего правила может потребоваться синтезировать эталонную структуру, которая играла бы роль математического ожидания распределения структур каждого образа. Эффективные и прозрачные алгоритмы решения этой задачи пока не разработаны, и для этого случая можно рекомендовать использовать в качестве эталона одну из имеющихся иерархий данного образа. Роль эталонной должна играть иерархия, сумма расстояний от которой до всех остальных иерархий образа минимальна.

Глава 17

Анализ данных, знаний и структур в системах искусственного интеллекта

§ 1. Экспертные системы партнерского типа

Главная отличительная особенность интеллектуальной системы заключается в умении делать правильные предсказания. В процессе исследования некоторого объекта (или процесса) появляется протокол наблюдений за его поведением при тех или иных внешних воздействиях. Обычно протокол имеет либо форму текстовых записей типа «стимул-реакция», либо таблиц типа «объект-свойство». Интеллектуальная система анализирует протокол и обнаруживает устойчивые (закономерные) связи между различными характеристиками объекта. Если после этого системе предъявить некоторые (описывающие) характеристики, то она, используя обнаруженные закономерности, сможет предсказать правдоподобные значения других (целевых) характеристик.

На этом фундаменте — умении автоматически обнаруживать закономерности и использовать их для предсказания — строятся, и будут строиться, все системы искусственного интеллекта (ИИ). По мере совершенствования методов системы ИИ должны пройти три условных стадии своего развития.

На первой стадии они представляют собой системы со слабым интеллектом пассивного помощника: они не в состоянии сами обнаруживать закономерности и используют только те закономерности (знания), которые были получены от экспертов, перера-

ботаны «инженерами по знаниям» и введены в память системы (базу знаний) в удобном для машины виде. Система способна манипулировать этими знаниями, имитируя процесс строгого логического вывода, и выдавать ответы на запросы пользователя. При этом система не имеет механизмов, которые побуждали бы ее критически оценивать вводимые в ее память знания, обнаруживать в них противоречия и пробелы, инициировать вопросы к хозяину системы, обращая его внимание на несовершенство базы знаний. Пассивность системы-ассистента обнаруживается и в том, что она не пытается извлекать новые знания из данных, которые накапливаются в ее памяти в ходе работы, и не выдает пользователю по своей инициативе никакой информации, например о появлении новых закономерностей, об устаревании какой-то части прежних знаний.

Именно такими свойствами обладают экспертные системы (ЭС) первого поколения. Дальнейшее развитие интеллектуальных систем направлено на снятие указанных ограничений и превращение систем в активных интеллектуальных партнеров пользователя, т. е. в партнерские системы (ПС). Наряду со знаниями экспертов, вводимыми «инженерами знаний», ПС будут иметь средства самостоятельного извлечения знаний из данных, поступающих в систему в ходе ее создания и эксплуатации. На этом основании ПС будут способны обнаруживать противоречия между имеющимися и вновь поступающими знаниями и данными, выявлять в них ошибки и пробелы и обращаться по своей инициативе к пользователю с сообщением по поводу этих дефектов в информационной базе. Кроме того, ПС должны иметь более удобные средства общения с пользователем, которые повышали бы их «дружественность».

Следующий этап развития интеллектуальных систем должен привести к появлению таких систем, которые по отношению к пользователю могли бы выступать в качестве учителя или лидера, т. е. к появлению систем-лидеров (СЛ). Эти системы будут накапливать знания, получая их в ходе непосредственного диалога с экспертом без вмешательства «инженера знаний», извлекая их из протоколов экспериментов, в том числе проводимых под управлением СЛ. Они должны уметь читать и понимать статьи, книги, чертежи и схемы. Система должна строить модель изучаемой прикладной области, т. е. создавать ее теорию, строить модель пользователя (ученика) и модель самой себя, чтобы оптимизировать процесс формирования модели изучаемого мира

в сознании ученика.

По понятным причинам сейчас нет смысла пытаться строить более детальные предположения о свойствах и функциях систем-лидеров. Что же касается партнерских систем, то их особенности по сравнению с существующими сейчас экспертными системами целесообразно указать уже сейчас, так как работы над ПС ведутся во многих коллективах, и было бы хорошо в самом начале этих работ сформулировать в явном виде их цели. Ниже приводится описание свойств, которыми, по мнению автора, должна обладать партнерская система.

§ 2. Отличительные характеристики ЭС и ПС [75]

Приведем сводную таблицу характеристик экспертных и партнерских систем и расшифруем закодированные в ней обозначения (см. табл. 20).

1. Базы знаний ЭС включают в себя знания, представленные в какой-нибудь одной форме: в виде продукций, фреймов или семантических сетей. База знаний ПС должна иметь возможность работать одновременно с любой из этих форм представления знаний (библиотекой) и дополнительно включать в себя знания в форме программ, имитирующих поведение изучаемого или управляемого объекта в динамике при разных входных воздействиях на него.

2. Если в экспертных системах единственным источником знаний являются суждения экспертов, то партнерские системы должны иметь средства получения знаний и из данных, представленных в виде статистических или эмпирических таблиц «объект-свойство-время».

3. При извлечении знаний для ЭС приходится прибегать к помощи «инженера знаний». В ПС помимо этого можно использовать также программы автоматического обнаружения закономерностей (знаний), скрытых в базе данных.

4. Экспертные системы обычно имеют только базы знаний. Партнерские будут иметь также и базы данных, в частности такие, в которых данные представлены в форме трехвходовых таблиц типа «объект-свойство-время». Элементарной частью данных является запись типа: «У объекта A_i свойство X_j в момент времени T_t имеет значение Q_{ijt}. Достоверность этого факта равна P». Или короче: $(A_i, X_j, T_t, Q_{ijt}, P)$.

5. В отличие от ЭС, пользующейся одним (дедуктивным) способом логического вывода, например методом резолюций, заложенным в языке ПРОЛОГ, партнерская система должна обладать еще и средствами индуктивного вывода, в том числе средствами, имитирующими рассуждения по аналогии и немонотонные рассуждения.

6. Диалог экспертной системы включает в себя термины прикладной области, входящие в меню, или фразы жесткой конструкции. Партнерские системы должны быть оснащены лингвистическими процессорами, способными понимать высказывания пользователя на естественном проблемно-ориентированном языке.

7. Удобным для пользователя средством общения с системой был бы устный диалог. Это позволило бы общаться с ПС по телефону и открыло новые области применения систем искусственного интеллекта в службах сервиса и обучения.

8. Важной функцией партнерской системы, облегчающей процесс наполнения базы знаний, будет способность автоматического обнаружения противоречий между знаниями, уже имеющимися в БЗ, и новыми, поступающими от экспертов или от программ автоматического извлечения знаний из данных.

9. Аналогичное значение должны иметь средства автоматического обнаружения грубых ошибок в базе данных. ПС по своей инициативе должна информировать пользователя об обнаруженных неблагополучиях и предлагать варианты разрешения противоречий или исправления ошибок.

10. Партнерская система должна иметь средства автоматического прогнозирования значений величин, отсутствующих в БД (средства заполнения пробелов).

11. Отсутствующая в ЭС модель пользователя будет необходима ПС для планирования своего взаимодействия с ним. На первых порах эта модель может быть представлена программами адаптации системы к особенностям взаимодействия с нею данного конкретного пользователя (выявлением часто повторяющихся вопросов, часто решаемых им задач, наиболее предпочтительной формы получения ответов и т. д.).

Таблица 20

Характеристики экспертных и партнерских систем

Характеристики	Экспертная система	Партнерская система
1. Типы базы знаний	продукции, фреймы семантические сети	библиотека, имитационные модели
2. Источники знаний	эксперты	эксперты, базы знаний
3. Способы извлечения знаний	«инженер знаний»	«инженер знаний», автоматическое обнаружение в базе данных
4. Наличие базы данных	нет	да
5. Стратегия логического вывода	дедукция	набор (дедукция, индукция, немонотонные рассуждения)
6. Язык общения с пользователем	проблемный	проблемно-ориентированный естественный язык
7. Устный диалог	нет	ограниченный словарь
8. Обнаружение противоречий в базе знаний	нет	да
9. Обнаружение ошибок в базе знаний	нет	да
10. Заполнение пробелов	нет	да
11. Модель пользователя	нет	адаптация
12. Гомеостат	нет	да
13. Вид результата	число, рецепт	+гипотеза, модель

12. Партнерская система должна иметь средства автоматического поддержания и даже улучшения своих рабочих характеристик в ходе эксплуатации. Для этого она должна включать в свой состав программы обеспечения гомеостатического состояния.

13. Экспертная система способна давать ответы в форме численных значений запрашиваемых величин или стандартных рекомендаций. Партнерская система, анализируя данные, знания и модели, должна кроме этого давать сопровождающие пояснения и формулировать обнаруживаемые новые закономерности или тенденции в форме, легко понимаемой пользователем.

Нельзя строго обосновать необходимость каждой из указанных характеристик партнерской системы, так же как и достаточность их совокупности. Выбор именно этих свойств и их интерпретация были сделаны на основании представления автора о том, при каких условиях интеллектуальная система превращается из умного, но пассивного ассистента в разумного и активного партнера такого пользователя, который решает задачи планирования, проектирования, прогнозирования, распознавания или принятия других решений.

§ 3. Состояние разработок в области партнерских систем

Основой для разработки ряда подсистем партнерской системы могут служить результаты, имеющиеся в области методов обнаружения закономерностей, распознавания образов, анализа данных и знаний, математической лингвистики, а также опыт в создании экспертных систем. Отметим некоторые из этих предпосылок, опираясь на один из возможных вариантов блок-схемы партнерской системы (см. рис. 46). На этой схеме заштрихованы блоки, имеющиеся в наиболее распространенных экспертных системах. При описании некоторых блоков мы подразумеваем их реализацию в нашей инструментальной системе ЭКСНА (экспертная система наполняемая) [22, 23, 57, 58].

3.1. Блок диалога. Этот блок включает в себя подсистему генерации диалога для новой предметной области, лингвистический процессор и подсистему речевой связи.

Основу подсистемы диалога может составлять программа, которая в диалоге с пользователем непрограммистом выясняет

терминологию новой предметной области, формирует набор сообщений в «меню», определяет желательную форму выходных сообщений. С помощью этой же программы можно вносить изменения в сценарий уже существующего диалога.

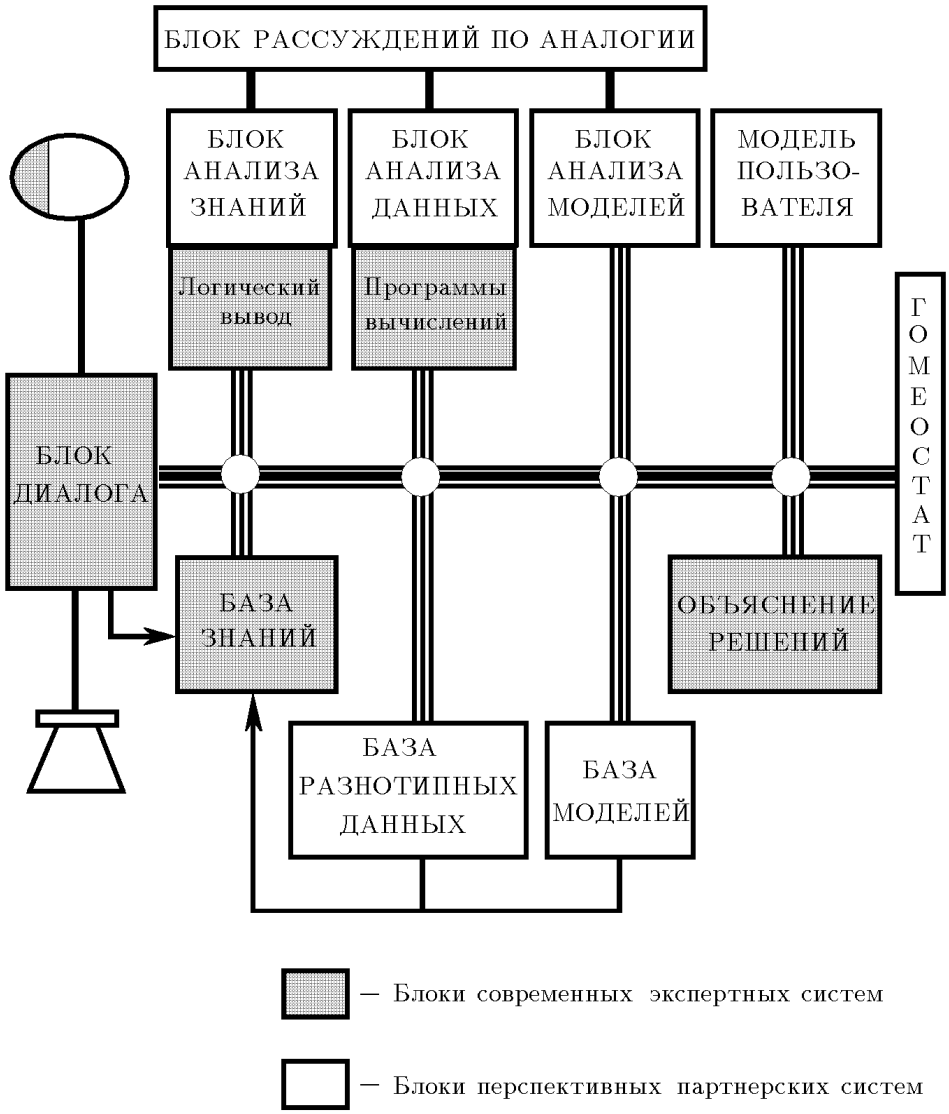


Рис. 46

В качестве прообраза лингвистического процессора может быть использована, например, программа, построенная на базе языка ДЕКЛ [106]. Эта программа может анализировать фразы естественного языка и строить соответствующие их семантическому содержанию сетевые структуры. Система задает пользователю вопросы, касающиеся новых для нее слов и отношений. В итоге она становится способной понимать смысл сообщений, формулируемых в естественной форме и синтезировать осмысленные фразы, относящиеся к данной предметной области.

Для создания подсистемы речевой связи можно использовать результаты многочисленных исследований и разработок в области речевой технологии. Для очень многих прикладных областей достаточно иметь программно-аппаратный комплекс, ориентированный на распознавание 200–300 устных команд [77]. Условия применения ПС во многих случаях позволяют использовать подстройку под диктора, что существенно упрощает задачу и повышает надежность распознавания. Существуют сейчас и синтезаторы речевых сигналов (текст-речь), вполне приемлемые по разборчивости и натуральности звучания речевых сообщений.

3.2. Блок логического вывода. Если знания в БЗ представлены в форме продукций типа «если ... то ...», то при сравнении двух продукций, например средствами языка ПРОЛОГ, их левые части (условия) и правые части (следствия) проверяются на точное совпадение. Если точного совпадения нет, то степень несовпадения роли не играет. Очевидно, что возможности блока логического вывода были бы существенно бóльшими, если бы можно было использовать информацию не только о совпадающих частях продукций, но и о близких, похожих, аналогичных частях. В главе 14 описан метод, позволяющий измерять степень похожести одной продукции на другую. Благодаря этому удалось построить механизм логического вывода, который имитирует человеческие способности рассуждений по аналогии [23].

Пусть, например, в БЗ имеется утверждение: «Если A и B, то C. Достоверность P». Поступает запрос: «Условия A' и B. Возможно ли следствие C?». Программа логического вывода обнаруживает, что условие в запросе не совпадает с условием базового знания: $A' \neq A$. Вычисление степени этого несовпадения показывает, что расстояние от A' до A не превышают заданного порога: $R(A', A) < R^*$. Это позволяет опереться на гипотезу, аналогичную гипотезе локальной компактности, которая

здесь выглядит так: «В непосредственной близости от известных продукций похожие условия влекут за собой похожие следствия». В результате можно сделать вывод, что «если A' и B, то C. Достоверность P'». Здесь P' меньше P на величину, зависящую от расстояния между условиями A' и A: $P' = P - f\{R(A', A)\}$.

Упомянутая мера близости между знаниями позволяет предварительно сделать таксономию знаний и тем самым структурировать базу знаний. Если для каждого таксона найти (или синтезировать) «эталонное» знание (см. гл. 14), то в процессе логического вывода отпадет необходимость проверять на совпадение или похожесть все знания БЗ. Достаточно найти таксон, эталон которого является самым близким к запросу, и дальше сравнивать запрос со всеми знаниями только этого таксона. Такой прием позволяет существенно ускорить процесс логического вывода, особенно в больших базах знаний.

3.3. Блок анализа данных. В блоке анализа данных должны быть программы обработки больших массивов разнотипных данных, представленных преимущественно в упоминаемой выше форме кубов данных типа «объект-свойство-время». В инструментальной системе ЭКСНА основу блока анализа данных составляют программы пакета ОТЭКС [57,58,82]. Напомним, что пакет содержит программы для решения задач четырех основных типов: таксономии, выбора информативных признаков, распознавание образов и заполнения пробелов.

Программы таксономии позволяют делать группировку (кластеризацию) наблюдаемых объектов по похожести их характеристик как в евклидовом, так и в λ-пространстве. Таксоны могут иметь как сферическую, так и произвольную форму. Признаки могут быть качественными, порядковыми и количественными. Имеются критерии для выбора наиболее предпочтительного числа таксонов.

Программы выбора признаков позволяют выделить наиболее информативное подмножество признаков из их исходного множества. Признаки могут быть разнотипными. Допускаются пробелы в таблицах данных.

Программы распознавания позволяют в процессе обучения обнаруживать закономерные связи между описывающими и целевыми характеристиками объектов. Эти закономерности используются затем в процессе распознавания новых объектов. Среди этих программ следует особо отметить программы, реализующие

алгоритмы таксономических и логических решающих функций. Первые позволяют повышать устойчивость решений в условиях малой обучающей выборки, а вторые обнаруживают закономерности в виде высказываний типа «если ... то ...». В партнерской системе это дает возможность автоматически извлекать знания из данных и пополнять ими базу знаний.

Программы заполнения пробелов предназначены для предсказания наиболее правдоподобного значения пропущенных элементов в таблицах данных. При этом используются закономерности, обнаруживаемые на данных, содержащихся в таблицах. Исходные данные могут содержать разнотипные признаки. Можно использовать эти программы и для обнаружения грубых ошибок или противоречий в таблицах данных. Имеются также программы для решения задач продолжения динамических рядов.

3.4. Блок анализа моделей. Представим себе, что запрос (его часть «если ...») касается таких режимов объекта, о которых нет сведений ни в базе данных, ни в базе знаний. Если ПС содержит модель объекта в виде программ, позволяющих имитировать его функционирование, то процесс логического вывода прерывается, запускается имитационная модель и фиксируется поведение объекта в требуемом режиме. В результате в базу знаний вносится знание о том, что последует за этим «если ...». Этот же результат включается в цепочку логического вывода и вывод продолжается. Работа с имитационной моделью требует больших затрат машинных ресурсов, потому к услугам модели можно прибегать изредка, только в том случае, когда встречаются запросы, не предусмотренные экспертами.

В системе ЭКСНА по такой схеме реализован выход из процесса логического вывода на блок вычислений по формулам или доопределения недостающих данных и знаний с помощью имитационных моделей [145].

3.5. Блок «Гомеостат». В свободное от обслуживания пользователя время партнерская система с помощью блока «Гомеостат» [22] анализирует вновь поступающие данные и знания, проверяет их на непротиворечивость, заполняет пробелы в данных, выявляет устаревшие и продублированные данные и знания и удаляет их в архив. При обнаружении новых закономерностей система вырабатывает сообщение пользователю, а при обнаружении противоречия или грубой ошибки формулирует ему соответствующие вопросы. Это может делаться с помощью программ

пакета ОТЭКС с использованием описанной выше метрики в пространстве знаний.

Помимо этого «Гомеостат» должен анализировать содержание сеансов работы с пользователем, выявляя часто повторяющиеся вопросы и строя на этой основе упрощенную модель пользователя. Такова функция программы «Частые вопросы» (ЧАВО) в инструментальной системе ЭКСНА.

3.6. Особенности системы ЭКСНА. Система ЭКСНА разработана Международной лабораторией интеллектуальных систем «СИНТЕЛ» и Институтом математики Сибирского отделения Российской академии наук (г. Новосибирск). Она реализована на IBM-совместимых компьютерах. Система хорошо документирована.

ЭКСНА представляет собой оболочку с набором разнообразных программных средств универсального назначения. Она предназначена для создания прикладных экспертных систем партнерского типа. Разработчику требуется организовать сбор и ввод в оболочку имеющейся у него содержательной информации — данных, знаний и моделей. Кроме того, с помощью подсистемы генерации диалога нужно сформировать удобную среду для общения с системой. Как показывает опыт, с помощью средств партнерской системы ЭКСНА пользователь генерирует свою прикладную версию ПС с базой знаний в 500 правил в течение одного месяца. В результате создается экспертная система партнерского типа, которая еще не реализует всех функций партнерских систем, но от экспертных систем первого поколения отличается следующими особенностями.

- ВСЕ экспертные системы используют знания экспертов.
- МНОГИЕ помогают пользователю сконструировать свой диалог.
- НЕКОТОРЫЕ системы включают в свой состав блок анализа данных.
- ОЧЕНЬ НЕМНОГИЕ системы умеют извлекать знания из данных.
- ТОЛЬКО ЭСПТ на базе ЭКСНА делает все это и, КРОМЕ ТОГО, умеет:
  - — рассуждать по аналогии;
  - — обнаруживать частичные и полные противоречия между знаниями;
  - — структурировать базу знаний;

— обрабатывать разнотипные данные в таблицах с пробелами;
— обнаруживать ошибки в базе данных;
— заполнять пробелы в таблицах данных;
— адаптироваться к особенностям пользователя;
— улучшать свои характеристики в процессе эксплуатации.

§ 4. Анализ данных, знаний и структур, связанных с изучением проблемы устойчивого развития [79]

Среди прикладных проблем, требующих глубокого анализа данных, знаний и структур с применением средств искусственного интеллекта, особое место занимает проблема устойчивого развития. Она стала привлекать общественное внимание после Конференции ООН по окружающей среде (Рио-де-Жанейро, 1992 г.), принявшей решение «Повестка дня на XXI век» [104, 153]. С этого времени термин «устойчивое развитие» стал очень популярным, однако его строгая формулировка пока отсутствует. По этой причине, прежде чем переходить к изложению задач анализа информации, связанной с этой проблемой, попытаемся ответить на следующие вопросы.

1. Об устойчивом развитии *чего именно* идет здесь речь?

2. Чем отличаются *процессы развития* от любых других процессов, происходящих в системе?

3. Чем отличаются *процессы устойчивого развития* от любых других процессов развития?

Резюме наших ответов на эти вопросы состоит в следующем:

а) нас интересует устойчивое развитие *ноосферы*;

б) к процессам развития системы (в отличие от процессов стагнации или деградации) мы относим такие процессы, которые *увеличивают ее способности к самосохранению*;

в) устойчивым мы считаем такой процесс, который неопределенно долго *остается процессом развития*.

Поясним пути получения этих выводов и определений.

Мы исходим из предположения о том, что увеличение функциональных возможностей системы, в частности возможности ее самосохранения, неизбежно связаны с ростом сложности ее организации. На всякую природную систему действуют разрушающие силы, причем чем сложнее устроена система, тем выше

риск ее разрушения. Если некая система оказывается в состоянии сохранять себя или даже усложнять свою структуру, это свидетельствует о том, что система имеет средства, которые могут противостоять естественным разрушительным процессам.

Такие средства реализуют функции типа «внешнее воздействие-адекватная реакция» и известны как механизм отражения или сознание, простейшие проявления которого присутствовали на самых ранних стадиях развития органического мира. Простейшие микроорганизмы противостоят энтропийным процессам с помощью высокопродуктивных способов размножения. В животном мире наблюдаются механизмы адаптации к изменениям среды обитания в виде пассивного гомеостатического приспособления к изменяющейся среде, рефлексов уклонения от угрозы и некоторых более сложных рефлексов. Для сохранения организации материи более высокого уровня сложности возможности этих механизмов были бы недостаточными. Требовалось появление способности материи к активному и упреждающему противодействию энтропии.

Эта способность получила свое воплощение в Разуме, с помощью которого носители разумной жизни могут не только приспосабливаться к среде обитания, но и изменять ее в благоприятном для себя направлении. При этом они опираются на накопленный опыт выживания (Знания) и используют три главные активные компоненты Разума:

— способность прогнозировать развитие ситуации и ставить перед собой цели (Мудрость);

— способность вырабатывать планы достижения выбранных целей (Ум) и

— способность организовывать действия по осуществлению выработанных планов (Воля).

Разум должен развиваться одновременно с развитием (усложнением) жизненных систем. Если развитие Разума отстает от роста сложности системы, то начинают преобладать силы ее энтропийного разрушения.

С развитием Разума у индивидуумов и по мере их социализации начинают проявляться результаты работы «Коллективного Разума» в виде постановок общих целей, коллективно вырабатываемых планов их достижения и организации совместных действий, направленных на реализацию этих планов. По мере роста Коллективного Разума росли масштабы коллективных усилий, результаты которых начали заметно сказываться на состо-

янии среды его обитания. Изменения природы Земли и ближнего космоса, вызываемые деятельностью людей, стали по своим масштабам сравнимыми с изменениями чисто природного характера. Эта веха в истории Земли осознается в качестве периода перехода к образованию вслед за геосферой и биосферой новой сферы ее развития — ноосферы [31, 129, 147].

Так же, как для отдельного человека главным средством самосохранения является его Разум, так для человечества и среды его обитания в эпоху ноосферы главным средством самосохранения является Коллективный Разум. Влияние несовершенного нарождающегося Коллективного Разума может порождать многочисленные процессы в природных, производственных, социальных или духовных областях, одни из которых объективно ведут к росту, а другие к ослаблению жизненного потенциала человечества. Их суммарный результат может носить нестабильный характер и в каждый момент времени проявляться в качестве процесса деградации, стагнации или развития. Выделим ту часть ресурсов Коллективного Разума, которая порождает процессы развития. Этому объекту наиболее близко соответствует используемый в русской философии термин «Соборный Разум» [154].

Таким образом, важнейшим средством самосохранения и развития ноосферы является не просто Разум населяющих ее индивидуумов и не просто та часть их интеллектуального и биоэнергетического потенциала, которая объединяется в Коллективный Разум, но та часть потенциала, которая образует Соборный Разум, ведущий к развитию ноосферы. Периоды преобладания процессов развития ноосферы над другими процессами будут занимать тем бóльшие отрезки времени, чем бóльшую мощность будет иметь Соборный Разум. Следовательно, пути достижения состояния устойчивого развития ноосферы совпадают с путями устойчивого роста мощности Соборного Разума.

В свете сказанного ясно, что при обсуждении проблем устойчивого развития ноосферы более конкретно речь должна идти об устойчивом развитии Соборного Разума. Соборный Разум, будучи высшим достижением эволюции природы, объединяет в себе и главную цель (его сохранение и развитие), и главное средство достижения этой цели.

Объективно нет оснований считать, что способность к самосохранению разумной жизни будет продолжаться бесконечно долго. Это нынешнее свойство природы следует рассматривать не как гарантию, а как шанс, которым можно воспользоваться,

а можно и потерять. Среди причин, которые могут привести к деградации ноосферы и Разума, чаще всего указывают на перенаселение Земли и загрязнение окружающей среды. Однако в гораздо большей степени развитию человечества угрожает несовершенство его социальной организации. Так, по оценкам биофизиков продуктов питания на земле достаточно для обеспечения более многочисленного населения, если Коллективный Разум сможет изменить взаимоотношения между отдельными людьми, человеческими коллективами и государствами. Мир погружается в атмосферу все большего числа конфликтов, возникающих на этнической, социальной и религиозной почве, с заметной тенденцией перерастания из локальных в глобальные. Так что наряду с вниманием к экологической ситуации следует обращать внимание и на состояние социальной и духовной сфер человечества. Предотвращение катастроф в этих сферах может быть достигнуто только усилиями Соборного Разума.

Следовательно, одна из основных целей, стоящих перед человечеством, состоит в объединении усилий Разума на поиске путей перехода процессов, протекающих в ноосфере, на траекторию устойчивого развития. В этой связи важной научной проблемой является проблема построения компьютерной модели, с помощью которой можно было бы имитировать процессы эволюции ноосферы. В завершенном виде модель должна состоять из иерархической системы элементов в виде индивидуумов, их малых групп, этносов, государств и человечества в целом вместе с производственной средой и средой земного и космического обитания, которые взаимодействуют между собой [45].

Элементы каждого уровня наделяются своими характеристиками и правилами взаимодействия с элементами своего уровня, а также с элементами других уровней. Так, например, индивидуум, обладающий своими личностными характеристиками, взаимодействует с другими индивидуумами, изменяя свои характеристики и влияя на характеристики других индивидуумов, а также взаимодействует со своим этносом, своим государством, всем человечеством и средой обитания в целом, оказывая в рамках имеющихся возможностей доступное ему влияние на характеристики всех этих элементов.

Наборы характеристик у элементов разных уровней различны и зависимости между ними могут иметь различный характер. Так, у каждого государства есть свои характеристики, которые не сводятся к средним характеристикам своих этносов, и свои

пути влияния на другие элементы иерархии, не являющиеся простой суммой влияний своих этносов на эти элементы.

Из сказанного выше следует, что для построения модели потребуется ввести несколько различных наборов характеристик, указать исходные («современные») значения этих характеристик, сформулировать функции влияния характеристик друг на друга и организовать динамический процесс, имитирующий развитие элементов модели во времени. На работающей модели можно будет вести наблюдение за ее поведением при разных условиях (т. е. значениях характеристик элементов и функций их взаимовлияния). Можно также определять «устойчивые» условия, т. е. условия, при которых ноосфера устойчиво развивается, и условия «стагнации», «деградации» и «катастрофы». Затем можно будет решать главную задачу: поиск технологий перевода некоторых заданных условий (например, существующих в настоящее время) в условия, обеспечивающие устойчивое развитие исследуемой системы при фиксированных ограничениях на время перевода и другие ресурсы. Если такие траектории будут обнаружены, можно будет предлагать некоторую глобальную целевую программу перехода развития на траекторию устойчивого развития и организовывать усилия на реализацию этой программы.

Решение описанной проблемы распадается на следующие составные части — отдельные задачи.

1. Определить множество свойств элементов модели различного ее уровня.

2. Разработать методы количественной оценки значений выбранных свойств у отдельных людей, этносов, государств, человечества и ноосферы в целом.

3. Провести измерения современных значений выбранных свойств. Организовать систему скрининга для периодического контроля состояния этих свойств.

4. Определить характер взаимного влияния свойств разного уровня друг на друга и формализовать эти функции влияния.

5. Построить имитационную модель эволюционного развития свойств во времени при заданном начальном векторе условий (т. е. начальных значениях этих свойств и параметров функций влияния).

6. Найти множество векторов условий, при которых реализуется устойчивое развитие ноосферы, и разработать методы и технологии перевода современного вектора условий в устойчивые векторы условия с учетом временных и других ресурсных огра-

ничений.

7. Организовать систему деятельности по практическому применению выбранной технологии формирования желательных значений свойств всех элементов ноосферы.

Многие из указанных задач сейчас могут решаться только методами групповых экспертных оценок с привлечением аксиологов, психологов, социологов, футурологов и т. д. Однако по некоторым элементам проектируемой модели уже сейчас имеется большой и постоянно растущий объем информации, касающейся процессов социально-экономического, технологического, демографического, экологического и т. п. характера. Эта информация имеет вид экспериментальных и статистических данных, хранящихся в разрозненных протоколах или организованных базах данных. Знания людей представлены технологическими инструкциями, сводами правил типа «если ... то ...», имитационными моделями, научными теориями. Вся эта информация записана на разных языках, зафиксирована на разных носителях, рассредоточена в разных местах, принадлежит разным агентам. Чтобы использовать ее в интересах устойчивого развития, потребуется обрабатывать все эти массивы информации в единой многоагентской системе. На пути к этому помимо организационных трудностей возникает проблема понимания системой разнородной информации.

Вновь появляющаяся информация все чаще фиксируется на машинных носителях. Что же касается информации, накопленной в прошлом, то для вовлечения ее в оборот необходимо разработать методы и средства, позволяющие переносить на машинные носители информацию из машинописных и рукописных текстов, считывать ее с фотоснимков, фонограмм и т. д. На этом этапе центральной научно-технической проблемой является проблема распознавания образов, широко представленная в данной книге.

Каждая новая таблица данных «объект-свойство-время» или новая база данных, попадающая на общий информационный склад, потребует предобработки. Нужно проверить эти данные на наличие в них грубых ошибок и противоречий с уже имеющимися данными, попытаться заполнить обнаруживаемые пробелы. Для этих целей можно использовать описанные в книге алгоритмы семейства ZET и WANGA. Целесообразно структурировать данные, выделив в таблицах кластеры похожих объектов, моментов времени или зависимых признаков. Для этого нужно

использовать алгоритмы таксономии. С их же помощью можно построить иерархическую структуру множества объектов, признаков и моментов времени. Выделение типичных представителей (прецедентов) каждого таксона позволит существенно ускорить многие процедуры дальнейшей обработки данных.

Аналогичная обработка должна делаться и на множестве вновь поступающих знаний. Здесь большую роль должны играть методы автоматического обнаружения грубых ошибок в знаниях и противоречий между знаниями. Очевидны области приложения также для методов таксономии знаний, выбора информативных предикторов, распознавания принадлежности новых знаний к ранее выделенным классам (образам) знаний. Совместный анализ данных и знаний позволит обнаруживать и устранять противоречия между знаниями и данными, пополнять базу знаниями, автоматически извлекаемыми из данных.

В процессе работы модели, имитирующей процессы развития, также потребуется постоянное использование описанных в книге методов анализа данных, знаний и структур для обнаружения статических и динамических закономерностей, прогнозирования многомерных процессов, распознавания возникающих ситуаций и т. д. Все такие процедуры удобно осуществлять средствами интеллектуальной оболочки в виде описанной выше экспертной системы партнерского типа. Так что область анализа данных и знаний будет играть большую роль в решении важнейшей задачи, стоящей перед современной наукой — задачи создания средств и методов перехода ноосферы на траекторию устойчивого развития.

Заключение

В заключение остановимся на нерешенных проблемах анализа данных и знаний. А таковых больше, чем решенных.

Действительно, достаточно сказать, что из 28 типов задач анализа данных, упомянутых в классификационной таблице 1 главы 2, в той или иной степени проработаны чуть больше десятка. Часть из них широко известны, алгоритмы их решения реализованы в виде программ, которые часто используются для решения практических задач. Для других же сформулированы лишь общие идеи методов их решения. Так что фронт работ по освоению задач новых типов и по совершенствованию методов и алгоритмов решения известных задач анализа данных очень большой.

Исследования в этой области должны быть направлены на преодоление трудностей решения реальных прикладных задач, проявляющихся в:

— многомерности признакового пространства;
— разнотипности измерительных шкал;
— большом числе образов;
— очень малых и очень больших объемах обучающих выборок;
— наличии ошибок и пробелов в таблицах данных;
— наличии неинформативных признаков;
— наличии структурированных объектов.

Для расширения областей применения методов распознавания в экономике, социологии, психологии и т.п. необходимо развитие методов теории измерений, ориентированных на измерение не физических, а «гуманитарных» характеристик.

Актуальной проблемой является разработка методов анализа данных, основанных на гипотезах, отличающихся от используе-

мой до настоящего времени гипотезы компактности.

Необходимо дальнейшее углубление связи между методами и системами искусственного интеллекта и системами распознавания образов и анализа изображений. В частности, актуальной является задача распространения методов анализа данных на решение аналогичных задач анализа знаний, что позволило бы поднять уровень развития искусственного интеллекта. В данной книге отмечены лишь некоторые первые шаги в этом перспективном направлении.

Погружение сферы анализа информации в мир многоагентских компьютерных сетей требует больших научно-организационных усилий для продвижения методов анализа данных и знаний в эту новую операционную среду. При этом могут оказаться полезными методы системного анализа [133].

Многие прикладники не знают, что из протекающего через их руки потока информации с помощью современных методов анализа данных можно было бы получить «сухой осадок» в виде в простых, «прозрачных» закономерностей или знаний. Эти знания могут помочь разобраться в структуре информации, выявить относительную важность ее различных частей, своевременно обнаружить факты, отклоняющиеся от общей закономерности, получить сигнал о первых предвестниках изменений в поведении наблюдаемого процесса. Поэтому распространение знаний о возможностях обсуждаемых здесь методов, включение такого рода материалов в учебные программы для студентов самого разного профиля также было бы своевременным и полезным.

Литература

1. **Абусев Р. А.** Групповая классификация. Решающие правила и их характеристики. Пермь: изд. Перм. ун-та, 1992.
2. **Автоматическое** распознавание образов / Ю. Л. Барабаш, Б. В. Варский, В. Т. Зиновьев и др. Киев: изд. КВАИУ, 1963.
3. **Айвазян С. А., Енюков И. С., Мешалкин Л. Д.** Прикладная статистика. Основы моделирования и первичная обработка данных. М.: Финансы и статистика, 1983.
4. **Айзерман М. А., Браверман Э. М., Розоноэр Л. И.** Теоретические основы метода потенциальных функций в задаче об обучении автоматов разделению входных ситуаций на классы // Автоматика и телемеханика. 1964. Т. 25, № 6. С. 917–936.
5. **Андерсон Т. В.** Введение в многомерный статистический анализ. М.: Физматгиз, 1963.
6. **Аркадьев А. Г., Браверман Э. М.** Обучение машины распознаванию образов. М.: Наука, 1964.
7. **Афифи А., Элашофф Р. (Afifi A. A., Elashoff R. M.)** Missing observations in multivariate statistics // J. Amer. Statist. Assoc. 1966. V. 61. P. 595–604.
8. **Афифи А., Эйзен С.** Статистический анализ. Подход с использованием ЭВМ. М.: Мир, 1982.
9. **Бак С. (Buck S. F.)** A method of estimation of missing values in multivariate data // J. Roy. Statist. Soc. Ser. B. 1960. V. 22. P. 202–206.

10. **Бахмутова И. В., Гусев В. Д., Титкова Т. Н. (Bachmutova I. V., Gusev V. D., Titkova T. N.)** Search for adoptions in song melodies // Comp. Music J. 1997. V. 21, N 1. P. 58–67.
11. **Бахмутова И. В., Гусев В. Д., Титкова Т. Н., Шиндин Б. А.** Об одном подходе к проблеме дешифровки древнерусских песнопений в невменной записи // Тр. Сибирской конф. по прикладной и индустриальной математике, Новосибирск, 1994. Т. 2. Новосибирск: Изд-во Ин-та математики, 1997. С. 1–10.
12. **Беккер Э. (Backer E.)** Cluster Analysis by Optimal Decomposition of Induced Fuzzy Sets. Delfts: Univ. Press, 1978.
13. **Беллман Р.** Динамическое программирование. М.: Изд-во иностр. лит., 1960.
14. **Бензекри Ж.-П. (Benzecri J.-P.)** L'analyse des Correspondaces. T. 2. Paris: Dunod, 1980.
15. **Береснев В. Л., Гимади Э. Х., Дементьев В. Т.** Экстремальные задачи стандартизации. Новосибирск: Наука, 1978.
16. **Бил Э., Литтл Р. (Beale E. M., Little R. J.)** Missing values in multivariate analysis // J. Roy. Statist. Soc. Ser. B. 1975. V. 37. P. 129–145.
17. **Благовещенский Ю. Н.** Оценивание по неполным выборкам. Ч. 1, 2 // Статистические модели и методы. М., 1984. Вып. 1: Труды ЦЭМИ. С. 4–32.
18. **Богарт К. (Bogart K. P.)** Preference structure, I // J. Math. Sociol. 1973. V. 3. P. 49–67.
19. **Богарт К. (Bogart K. P.)** Preference structure, II // SIAM J. Appl. Math. 1975. V. 29, N 2. P. 254–262.
20. **Бонгард М. М.** Проблема узнавания. М.: Наука, 1967.
21. **Боровков А. А.** Математическая статистика. Новосибирск: Наука, 1997.
22. **Бушуев М. В., Загоруйко Н. Г.** Инструментальная система ЭКСНА // Тр. Междунар. симп. «Экспертные системы в обучении». Прага: изд. Техн. ун-та, 1989. С. 26–32.
23. **Бушуев М. В., Ёлкина В. Н., Загоруйко Н. Г., Шемякина Е. Н.** Блок анализа знаний в инструментальной

экспертной системе ЭКСНА // Методы и системы искусственного интеллекта. Новосибирск, 1992. Вып. 145: Вычислительные системы. С. 29–79.

24. **Вапник В. Н.** Задача обучения распознаванию образов. М.: Знание, 1970.
25. **Вапник В. Н., Червоненкис А. Я.** Теория распознавания образов. М.: Наука, 1974.
26. **Васильев В. И.** Распознающие системы: Справочник. Киев: Наук. думка, 1983.
27. **Ващенко Н. Д.** Формирование понятий в семантической сети // Кибернетика. 1983. № 2. С. 101–107.
28. **Величко В. М., Загоруйко Н. Г.** Автоматическое распознавание ограниченного набора устных команд // Вычислительные системы. Новосибирск, 1969. Вып. 36. С. 101–110.
29. **Вентцель Е. С.** Теория вероятностей. М.: Наука, 1969.
30. **Вентцель Е. С.** Элементы динамического программирования. М.: Наука, 1964.
31. **Вернадский В. И.** Научная мысль как планетарное явление. М.: Наука, 1991.
32. **Винцюк Т. К.** Распознавание слов устной речи методами динамического программирования // Кибернетика. 1968. № 1. С. 81–88.
33. **Волошин Г. Я.** Об использовании языковой избыточности для повышения надежности автоматического распознавания речевых сигналов // Вычислительные системы. Новосибирск, 1967. Вып. 28. С. 21–48.
34. **Волошин Г. Я., Бахмутова И. В., Прокопенко А. А.** О сетевом алгоритме распознавания фонем по последовательности сегментов // Вычислительные системы. Новосибирск, 1969. Вып. 37. С. 44–47.
35. **Волошин Г. Я., Бурлакова И. А., Косенкова С. Т.** Статистические методы решения задач распознавания, основанные на аппроксимационном подходе. Ч. 1. Владивосток: изд. Дальневост. отд-ния РАН, 1992.
36. **Воронин Ю. А.** Введение мер сходства и связи для решения геолого-географических задач // Докл. АН СССР. 1971. Т. 199, № 5. С. 1011–1015.

37. Гаврилко Б. П., Загоруйко Н. Г. Универсальный алгоритм эмпирического предсказания // Вычислительные системы. Новосибирск, 1973. Вып. 55. С. 134–138.

38. Гладун В. П. Планирование решений. Киев: Наук. думка, 1987.

39. Глассер М. (Glasser M.) Linear regression analysis with missing observations among the independent variables // J. Amer. Statist. Assoc. 1964. V. 59. P. 834–844.

40. Глисон Т., Стелин Р. (Gleason T. C., Staelin R.) A proposal for handling missing data // Psychometrika. 1975. V. 40. P 229–252.

41. Гольдберг Д. (Goldberg D.) Genetic Algorithms in Search, Optimization and Machine Learning. Massachusetts: Addison; Wesley Reading, 1989.

42. Гренандер У. (Grenander U.) Lectures in Pattern Theory. N. Y.: Springer-Verl., 1976.

43. Губарев В. В. Вероятностные модели: Справочник. В 2-х ч. Новосибирск: изд. НЭТИ, 1992.

44. Гусев В. Д., Куличков В. А., Чупахина О. Н. Сложностной анализ геномов. I. Меры сложности и классификация выявленных структурных закономерностей // Молекулярная биология. 1991. Т. 25, вып. 3. С. 825–832.

45. Демин Д. В. Модель системы «человек-среда» и параметры катастрофы // Искусственный интеллект и экспертные системы. Новосибирск, 1997. Вып. 160: Вычислительные системы. С. 43–64.

46. Демпстер А., Лерд Н., Рабин Д. (Dempster A. P., Laird N. M., Rubin D. B.) Maximum likelihood from incomplete data via the EM-algorithm // J. Roy. Statist. Soc. Ser. B. 1977. V. 39. P. 1–38.

47. Денисов В. И. Математическое обеспечение системы «ЭВМ-экспериментатор». М.: Наука, 1977.

48. Денисов В. И., Лемешко Б. Ю. Вычисление оценок параметров распределений с использованием таблиц асимптотически оптимального группирования // Применение ЭВМ в оптимальном планировании и проектировании. Новосибирск: изд. НЭТИ, 1981. С. 3–17.

49. **Джамбу М., Лебо М.-О. (Jambu M., Lebeaux M.-O.)** Cluster Analysis and Data Analysis. Amsterdam: North-Holland, 1983.
50. **Дидэ Э. (Diday E.)** Symbolic Data Analysis. Paris: INRIA Roquencourt, 1995. P. 1–136.
51. **Дидэ Э., Лемер Ж., Пуже Ж., Тесту Ф. (Diday E., Lemaire J., Pouget J., Testu F.)** Elements d'analyse des Donnees. Paris: Dunod, 1982.
52. **Додж И. (Dodge Y.)** Analysis of Experiments with Missing Data. N. Y.: John Wiley & Sons, 1985.
53. **Ёлкин Е. А., Ёлкина В. Н., Загоруйко Н. Г.** О возможности применения методов распознавания в палеонтологии // Геология и геофизика. 1967. №9. С. 75–78.
54. **Ёлкина В. Н., Загоруйко Н. Г.** Количественные критерии качества таксономии и их использование в процессе принятия решений // Вычислительные системы. Новосибирск, 1969. Вып. 36. С. 29–46.
55. **Ёлкина В. Н., Загоруйко Н. Г., Куклин А. П., Комаровский Э. Д.** Типы ртутоносных и оловоносных территорий Чукотки // Колыма. Магадан, 1972. № 4. С. 37–40.
56. **Ёлкина В. Н., Загоруйко Н. Г., Новоселов Ю. А.** Математические проблемы агроинформатики. Новосибирск: изд. Ин-та математики СО РАН, 1987.
57. **Ёлкина В. Н., Загоруйко Н. Г.** Блок анализа данных в экспертной системе ЭКСНА // Экспертные системы и анализ данных. Новосибирск, 1991. Вып. 144: Вычислительные системы. С. 54–175.
58. **Ёлкина В. Н., Загоруйко Н. Г.** Блок анализа данных в экспертной системе ЭКСНА (окончание) // Методы и системы искусственного интеллекта. Новосибирск, 1992. Вып. 145: Вычислительные системы. С. 3–128.
59. **Ёлкина В. Н., Загоруйко Н. Г., Темиркаев В. С.** Алгоритмы направленного таксономического поиска информативных подсистем признаков (НТПП) // Вычислительные системы. Новосибирск, 1974. Вып. 59. С. 49–70.
60. **Енюков И. С., Кулакова Е. П.** Числовые метки для качественных признаков в дискретном анализе // Приклад-

ной многомерный статистический анализ. М.: Статистика, 1978. № 33: Ученые записки по статистике. С. 353–358.

61. Жанатауов С. У. Методы прогностических переменных // Машинные методы обнаружения закономерностей. Новосибирск, 1981. Вып. 88: Вычислительные системы. С. 151–155.

62. Журавлев Ю. И. Корректные алгебры над множествами некорректных (эвристических) алгоритмов. I–III // Кибернетика. Киев. 1977. № 4. С. 14–21; 1977. № 6. С. 21–27; 1978. № 2. С. 35–43.

63. Журавлев Ю. И. (Zhuravlev Yu. I.) An algebraic approach to recognition or classification problems // Pattern Recognition and Image Analysis. M., 1998. N 8(10). P. 59–100.

64. Журавлев Ю. И., Загоруйко Н. Г. Класс коллективно-групповых решающих правил, основанных на дисперсионном критерии компетентности предикторов // Анализ данных и сигналов. Новосибирск, 1998. Вып. 163: Вычислительные системы. С. 82–90.

65. Загоруйко А. Н., Загоруйко Н. Г. Эксперименты по переоткрытию закона Менделеева с помощью ЭВМ // Структурный анализ символьных последовательностей. Новосибирск, 1984. Вып. 101: Вычислительные системы. С. 75–81.

66. Загоруйко Н. Г. Линейные решающие функции, близкие к оптимальным // Вычислительные системы. Новосибирск, 1965. Вып. 19. С. 67–76.

67. Загоруйко Н. Г. Комбинированный метод принятия решений // Вычислительные системы. Новосибирск, 1966. Вып. 24. С. 22–31.

68. Загоруйко Н. Г. Какими решающими функциями пользуется человек? // Вычислительные системы. Новосибирск, 1967. Вып. 28. С. 69–79.

69. Загоруйко Н. Г. Методы распознавания и их применение. М.: Сов. радио, 1972.

70. Загоруйко Н. Г. Искусственный интеллект и эмпирическое предсказание. Новосибирск: изд. НГУ, 1975.

71. Загоруйко Н. Г. Таксономия в анизотропном пространстве // Эмпирическое предсказание и распознавание обра-

зов. Новосибирск, 1978. Вып. 76: Вычислительные системы. С. 26–34.

72. Загоруйко Н. Г. Классификация задач прогнозирования на таблицах «объект-свойство» // Машинные методы обнаружения закономерностей. Новосибирск, 1981. Вып. 88: Вычислительные системы. С. 3–8.

73. Загоруйко Н. Г. Согласование разнотипных шкал // Анализ разнотипных данных. Новосибирск, 1983. Вып. 99: Вычислительные системы. С. 3–14.

74. Загоруйко Н. Г. Гипотезы компактности и λ-компактности в методах анализа данных // Сиб. журн. индустр. математики. 1988. Т. 1, № 1. С. 114–126.

75. Загоруйко Н. Г. Партнерские системы // Анализ данных и знаний в экспертных системах. Новосибирск, 1990. Вып. 134: Вычислительные системы. С. 3–18.

76. Загоруйко Н. Г. Анализ данных и анализ знаний // Анализ последовательностей и таблиц данных. Новосибирск, 1994. Вып.150: Вычислительные системы. С. 3–17.

77. Загоруйко Н. Г. АРСО и речевые технологии // Прикладные системы искусственного интеллекта. Новосибирск, 1995. Вып. 153: Вычислительные системы. С. 3–31.

78. Загоруйко Н. Г. Алгоритмы редактирования базы знаний (алгоритмы семейства ZKB) // Искусственный интеллект и экспертные системы. Новосибирск, 1996. Вып. 157: Вычислительные системы. С. 3–12.

79. Загоруйко Н. Г. Исследование проблем, связанных с моделированием процессов развития ноосферы // Искусственный интеллект и экспертные системы. Новосибирск, 1997. Вып. 160: Вычислительные системы. С. 3–17.

80. Загоруйко Н. Г. Самообучающийся генетический алгоритм прогнозирования (GAP) // Искусственный интеллект и экспертные системы. Новосибирск, 1997. Вып. 160: Вычислительные системы. С. 80–95.

81. Загоруйко Н. Г., Бушуев М. В. Меры расстояния в пространстве знаний // Анализ данных в экспертных системах. Новосибирск, 1986. Вып. 117: Вычислительные системы. С. 24–35.

82. Загоруйко Н. Г., Ёлкина В. Н., Емельянов С. В., Лбов Г. С. Пакет прикладных программ ОТЭКС. М.: Финансы и статистика, 1986.

83. Загоруйко Н. Г., Ёлкина В. Н., Тимеркаев В. С. Алгоритм заполнения пропусков в эмпирических таблицах (алгоритм ZET) // Эмпирическое предсказание и распознавание образов. Новосибирск, 1975. Вып. 61: Вычислительные системы. С. 3–27.

84. Загоруйко Н. Г., Ёлкина В. Н., Полякова Г. Л. Полигон для сравнения алгоритмов таксономии // Обнаружение эмпирических закономерностей с помощью ЭВМ. Новосибирск, 1984. Вып. 102: Вычислительные системы. С. 120–126.

85. Загоруйко Н. Г., Ёлкина В. Н., Лбов Г. С. Алгоритмы обнаружения эмпирических закономерностей. Новосибирск: Наука, 1985.

86. Загоруйко Н. Г., Заславская Т. И. Применение методов распознавания образов в социологии. Новосибирск: Наука, 1968.

87. Загоруйко Н. Г., Кужелев О. В. Синхронизация многомерных динамических процессов // Анализ данных и знаний в экспертных системах. Новосибирск, 1990. Вып. 134: Вычислительные системы. С. 85–96.

88. Загоруйко Н. Г., Пичуева А. Г. Сравнение иерархических структур // Искусственный интеллект и экспертные системы. Новосибирск, 1996. Вып. 157: Вычислительные системы. С. 101–111.

89. Загоруйко Н. Г., Савельев Л. Я. Относительная мощность измерительных шкал // Структурный анализ символьных последовательностей. Новосибирск, 1984. Вып. 101: Вычислительные системы. С. 114–129.

90. Загоруйко Н. Г., Самохвалов К. Ф., Свириденко Д. И. Логика эмпирических исследований. Новосибирск: изд. НГУ, 1974.

91. Загоруйко Н. Г., Свириденко Д. И. Формализация процесса углубления понимания // Эмпирическое предсказание и распознавание образов. Новосибирск, 1976. Вып. 67: Вычислительные системы. С. 87–92.

92. Загоруйко Н. Г., Скоробогатов В. А., Хворостов П. В. Вопросы анализа и распознавания молекулярных структур на основе общих фрагментов // Алгоритмы анализа структурной информации. Новосибирск, 1984. Вып. 103: Вычислительные системы. С. 26–50.

93. Загоруйко Н. Г., Ульянов Г. В. Локальные методы заполнения пробелов в эмпирических таблицах // Экспертные системы и распознавание образов. Новосибирск, 1988. Вып. 126: Вычислительные системы. С. 75–121.

94. Ивахненко Г. И. Самообучающиеся системы распознавания и автоматического управления. Киев: Техника, 1969.

95. Ивахненко А. Г. Применение принципа самоорганизации для объективной кластеризации изображений, системного анализа и долгосрочного прогноза // Автоматика. 1986. № 1. С. 5–11.

96. Карнап Р. Философские основания физики. М.: Прогресс, 1971.

97. Кедров Б. М. Открытие галлия — первое химическое открытие нового типа // Прогнозирование в учении о периодичности. М.: Наука, 1976. С. 6–31.

98. Кемени Дж., Снелл Дж. Кибернетическое моделирование. М.: Сов. радио, 1972.

99. Кенделл М. Ранговые корреляции. М.: Статистика, 1975.

100. Кингсан Фу. (King-Sun Fu) The optimal secventional decisions. Lafayette: Purdue Univ. Press, 1967.

101. Ковалевский В. А. Корреляционный метод распознавания изображений // Журн. вычисл. математики и мат. физики. 1962. Т. 2, № 4. С. 584–592.

102. Ковер Т. (Cover T. M.) Classification and generalisation capabilities of linear Threshold units: Technical documentary report RADS–TDS–64–32, Febr. 1964. Rome: Air Development Center, 1964.

103. Ковер Т., Чарт П. (Cover T. M., Chart P. E.) Nearest-neighbor-pattern classification // IEEE Trans. Inform. Theory. 1967. V. IT-13, № 1. P. 21–27.

104. Коптюг В. А. Конференция ООН по окружающей среде

и развитию (Рио-де-Жанейро, июнь 1992 г.): Информационный обзор. Новосибирск: Наука, 1992.

105. Котюков В. И. Формирование решающих правил // Вычислительные системы. Новосибирск, 1971. Вып. 44. С. 37–48.

106. Кузнецов И. П. Язык декларативного типа ДЕКЛ. М.: Наука, 1986.

107. Лбов Г. С. Об ошибках классификации образов при неравных матрицах ковариации // Вычислительные системы. Новосибирск, 1964. Вып.14. С. 31–38.

108. Лбов Г. С. Методы обработки разнотипных экспериментальных данных. Новосибирск: Наука, 1981.

109. Лбов Г. С., Старцева Н. Г. Сложность распределений в задачах классификации // Докл. РАН. 1994. Т. 338, № 5. С. 592–594.

110. Левинсон С. Е. Структурные методы автоматического распознавания речи: Пер. с англ. // Тр. ТИИЭР. М., 1985. Т. 73, № 11. С. 100–129.

111. Леман Э. Проверка статистических гипотез. М.: Наука, 1979.

112. Лемешко Б. Ю. Программная система «Оценивание параметров распределений» // Тез. докл. Рос. науч.-техн. конф. «Информатика и проблемы телекоммуникаций». Новосибирск: изд. НЭТИ, 1994. С. 128–129.

113. Литтл Р., Рабин Д. (Little R. J., Rubin D. B.) Statistical Analysis with Missing Data. N. Y.: John Wiley & Sons, 1987.

114. Литтл Р., Смит П. (Little R. J., Smith P. J.) Editing and imputation for quantitative survey data // J. Amer. Statist. Assoc. 1987. V. 82. P. 58–68.

115. Литтл Р., Шлюстер М. (Little R. J., Schlusther M. D.) Maximum likelihood estimation for mixed continuous and categorical data with missing values // Biometrika. 1985. V. 72. P. 497–512.

116. Мазуров В. Д., Тягунов Л. И. Метод комитетов в распознавании образов // Метод комитетов в распознавании образов. Свердловск: изд. Уральск. отд-ния АН СССР. 1974. С. 10–40.

117. Макаров Л. И., Скоробогатов В. А. Комплекс программ для исследования зависимости «структура-свойство» химических соединений // Алгоритмический анализ графов и его применения. Новосибирск, 1988. Вып. 127: Вычислительные системы. С. 92–129.

118. Мальцева Н. И. Аппроксимация непрерывных распределений некоторыми смесями: Дис. ... канд. физ.-мат. наук. Казань, 1971.

119. Мамчур Е. А. Проблема выбора теории. М.: Наука, 1975.

120. Манохин А. Н. Методы распознавания, основанные на логических решающих функциях // Эмпирическое предсказание и распознавание образов. Новосибирск, 1976. Вып. 67: Вычислительные системы. С. 42–53.

121. Материалисты древней Греции. М.: Мир, 1957.

122. Менделеев Д. И. Периодический закон. Основные статьи. М.: Изд-во АН СССР, 1958.

123. Мерилл Т., Грин О. (Merill T., Green O. M.) On the effectiveness of receptors in recognition systems // IEEE Trans. Inform. Theory. 1963. V. IT–9. P. 11–17.

124. Мики Д. (Michie D.) Machine learning in the next five years // EWLS-88: Proc. 3-th Europ. working session on learning. Glasgow; London: Pitman, 1988.

125. Миллер Г. (Miller G. A.) The magical number seven, plus or minus two: some limits in our capacity for processing information // Psycholog. Rev. 1956. N 63. P. 81–97.

126. Михальски Р. (Michalski R. S.) Variable-valued logic: system VL1 // Proc. Symp. on multiple valued logic. Morgentown, 1974.

127. Михальски Р., Братко И., Кубат М. (Michalski R. S, Bratko I., Kubat M.) Machine Learning and Data Mining, Methods and Applications. N. Y.: John Wiley & Sons, 1998.

128. Михальски Р., Степп Р. (Michalski R. S., Stepp R.) Learning from obsevation: conceptual clustering // Machine learning: An artifitial intelligence approach. Morgan Caufmann, 1983.

129. Моисеев Н. Н. Алгоритмы развития. М.: Наука, 1987.

130. Некрасов Б. В. Основы общей химии. М.: Химия, 1973.

131. Немытикова Л. А. Методы сравнения символьных последовательностей // Методы обработки символьных последовательностей и сигналов. Новосибирск, 1989. Вып. 132: Вычислительные системы. С. 1–34.

132. Ниманн Г. (Niemann H.) Pattern Analysis and Understanding. Berlin e. a.: Springer-Verl., 1998.

133. Перегудов Ф. И., Тарасенко Ф. П. Основы системного анализа. Томск: Изд-во науч.-техн. лит., 1997.

134. Пирогов А. А., Слуцкер Г. С. К фонетической теории речи // Тез. докл. VI Всесоюз. акустической конф. М.: изд. Акустич. ин-та АН СССР, 1968. С. 62–64.

135. Прим З. Л. Кратчайшие связывающие сети и некоторые обобщения // Кибернетический сб. 1961. № 2. С. 95–107.

136. Раппопорт А. М., Шнейдерман М. В. Анализ экспертных суждений, заданных в виде структур // Прикладной многомерный статистический анализ. М.: Наука, 1978. С. 150–164.

137. Растригин Л. А., Пономарев Ю. П. Экстраполяционные методы проектирования и управления. М.: Машиностроение, 1986.

138. Растригин Л. А., Эренштейн Р. Х. Принятие решений коллективом решающих правил в задачах распознавания образов // Изв. АН СССР. Сер. Автоматика и телемеханика. 1975. № 9. С. 133–144.

139. Раудис Ш. Ю. Об определении объема обучающей выборки линейного классификатора // Вычислительные системы. Новосибирск, 1967. Вып. 28. С. 79–88.

140. Розенфельд А. Распознавание и обработка изображений с помощью вычислительных машин. М.: Мир, 1972.

141. Розин Б. Б., Котюков В. И., Ягольницер М. А. Экономико-статистические модели с переменной структурой. Новосибирск: Наука, 1984.

142. Романовский В. И. Дискретные цепи Маркова. М.; Л.: Гостехиздат, 1949.

143. Самохвалов К. Ф. О теории эмпирических предсказаний // Вычислительные системы. Новосибирск, 1973. Вып. 50. С. 3–35.

144. **Себестиан Г. С.** Процессы принятия решений при распознавании образов: Пер. с англ. Киев: Техника, 1965.
145. **Семенова Н. В.** Расширение функций блока анализа знаний в инструментальной системе ЭКСНА // Анализ последовательностей и таблиц данных. Новосибирск, 1994. Вып. 150: Вычислительны системы. С. 211–219.
146. **Супес П., Зинес Дж.** Основы теории измерений // Психологические измерения. М.: Мир, 1967. С. 117–132.
147. **Тейяр де Шарден П** Феномен человека. М.: Наука, 1987.
148. **Титтерингтон Д., Джанг Дж. (Titterington D. M., Jiang J. M.)** Recursive estimation procedures for missing data problems // Biometrika. 1983. V. 70. P. 258–267.
149. **Турбович И. Т.** Опознание образов. М.: Наука, 1968.
150. **Уилкс С. (Wilks S. S.)** Moments and distributions of estimates of population from fragmentary samples // Ann. Math. Statist. 1932. V. 3. P. 163–195.
151. **Уолш Дж. (Walsh J. E.)** Computer-feasible method for handling incomplete data regression analysis // J. Assos. Comput. Math. 1961. V. 18. P. 201–211.
152. **Уотсон Д. Д.** Двойная спираль. М.: Мир, 1969.
153. **Урсул А. Д.** Ноосферная стратегия перехода Российской Федерации на модель устойчивого развития // Научные и технические аспекты охраны окружающей среды. М.: ВИНИТИ, 1995. Вып. 10.
154. **Федоров Н. Ф.** Собрание сочинений в 4-х томах. М.: Прогресс, 1995.
155. **Фейнман Р.** Природа физических законов. М.: Мир, 1967.
156. **Фогель Д. (Fogel D. B.)** Evolutionary Computation: Toward a New Philosophy of Machine Intelligence. Piscataway; N. Y.: IEEE Press, 1995.
157. **Фрейн Г. (Frane G. M.)** Some simple procedure for handling missing values in multivariate analysis // Psychometrika. 1976. V. 41. P. 409–415.
158. **Харкевич А. А.** Борьба с помехами. М: Наука, 1965.
159. **Хартли Г., Хокинг Р. (Hartley H. O., Hocking R. R.)** The analysis of incomplete data // Biometrics. 1971. V. 27. P. 783–808.

160. Хасби Дж., Швертман Н., Аллен Д. (Huseby J. R., Schwertman N. C., Allen D. M.) Computation of the mean vector and dispersion matrix for incomplete multivariate data // Communs Statist. B. 1980. V. 9. P. 301–309.

161. Хокинг Р., Маркс Д. (Hocking R. R., Marx D. L.) Estimation with incomplete data: an improved computational method and the analysis of nested data // Comm. Statist. Theory Methods. A. V. 8. P. 1151–1181.

162. Хоменко Л. В. Методы параллельного формирования понятий на основе пирамидальных сетей // Экспертные системы и распознавание образов. Новосибирск, 1988. Вып. 126: Вычислительные системы. С. 24–35.

163. Черный Л. Б. Порождение мер связи между объектами с помощью мер связи между признаками // Проблемы анализа дискретной информации. Новосибирск: изд. ИЭиОПП СО АН СССР, 1975. С. 167–174.

164. Шривастава М. (Srivastava M. S.) Multivariate data with missing observations // Comm. Statist. Theory Methods. 1985. V. 14. P. 775–792.

165. Шусторович А. М. Об адекватных парных мерах сходства в задачах распознавания образов с разнотипными признаками // Вопросы обработки информации при проектировании систем. Новосибирск, 1977. Вып. 69: Вычислительные системы. С. 147–162.

166. Эверитт Б. (Everitt B.) Cluster analysis. London: Heinemann, 1981.

167. Энгельман Л. (Engelman L.) An efficient algorithm for computing covariance matrices from data with missing values // Comm. Statist. Theory Methods. B. 1982. V. 11. P. 113–121.

168. Proceeding of the Second International Conference on Knowledge Discovery and Data Mining. Portland, 1996.

Оглавление

Предисловие **3**

Часть I. **Введение в анализ данных**

Глава 1. **Основные понятия** **5**

§ 1. Чем отличаются «данные» от «знаний»? 5
§ 2. Что такое анализ данных? 7
§ 3. Принятие решений по прецедентам и моделям 9
§ 4. Что такое анализ знаний? 11
§ 5. Что такое закономерность? 11

Глава 2. **Классификация задач анализа данных** **16**

§ 1. Теория измерений 16
1.1. Типы измерительных шкал (16). 1.2. Сравнительная информативность шкал (20).
§ 2. Классификация задач анализа данных 22

Глава 3. **Базовые гипотезы, лежащие в основе методов анализа данных** **28**

§ 1. Гипотеза компактности 29
§ 2. Гипотеза λ-компактности 31

Часть II. **Методы анализа данных**

Глава 4. **Задачи таксономии** **36**

§ 1. Природа задач таксономии 36

§ 2. Алгоритмы таксономии класса FOREL 38
2.1. Алгоритм FOREL (38). 2.2. Алгоритм FOREL–2 (39). 2.3. Алгоритм SKAT (40). 2.4. Алгоритм KOLAPS (41). 2.5. Алгоритм BIGFOR (43). 2.6. Иерархическая таксономия (44).
§ 3. Динамичная таксономия 45
3.1. Алгоритм DINA (45). 3.2. Алгоритм SETTIP (46).
§ 4. Таксономия с суперцелью 47
4.1. Алгоритм ROST (48).
§ 5. Таксономия в анизотропном пространстве 48
§ 6. Сравнение алгоритмов таксономии 50
§ 7. Выбор числа таксонов 52
§ 8. Примеры решения практических задач 54
8.1. Задачи палеонтологии и геологической разведки (54). 8.2. Задачи социологии и экономики (55). 8.3. Задачи биологии (57). 8.4. Задачи океанологии (57). 8.5. Задачи распознавания речевых сигналов («кодовая книга») (58). 8.6. Другие области применения (59).
§ 9. Некоторые дополнительные замечания о таксономии 59

Глава 5. **Распознавание образов** **62**

§ 1. Алгоритмы построения решающих правил 63
§ 2. Статистические решающие правила 65
§ 3. Алгебраические методы построения решающих правил 69
§ 4. Распознавание большого числа образов 72
4.1. Метод отбора сильнейшего конкурента (МСК) (72). 4.2. Метод попутного разделения (ПОРА) (73). 4.3. Метод покоординатного вычеркивания (МПВ) (75).
§ 5. Оценка потерь 77
§ 6. Гипотеза компактности в распознавании образов .. 78
§ 7. Построение решающих правил по конечной выборке 80
§ 8. Решающие правила, опирающиеся на прецеденты . 82
8.1. Минимизация набора прецедентов (алгоритм STOLP) (84). 8.2. Метод «дробящихся эталонов» (алгоритм ДРЭТ) (87). 8.3. Таксономические решающие функции (алгоритм ТРФ) (90).
§ 9. Логические решающие правила 92
9.1. Алгоритм CORAL (93). 9.2. Алгоритм DW (95).

§ 10. Представительность выборки 98

Глава 6. **Выбор системы информативных признаков** **102**

§ 1. Постановка задачи 102
§ 2. Критерии информативности признаков 103
§ 3. Метод последовательного сокращения (алгоритм Del) 108
§ 4. Метод последовательного добавления признаков (алгоритм Add) 108
§ 5. Метод случайного поиска с адаптацией (алгоритм СПА) 110
§ 6. Направленный таксономический поиск признаков (алгоритм НТПП) 112

Глава 7. **Заполнение пробелов и обнаружение ошибок в эмпирических таблицах** **113**

§ 1. Обзор работ по проблеме заполнения пробелов 113
§ 2. Базовый алгоритм ZET заполнения пробелов 115
§ 3. Некоторые варианты алгоритма ZET 119
3.1. Обнаружение грубых ошибок (алгоритм ZET–R) (119). 3.2. Прогнозирование динамических рядов (алгоритм ZET–D) (119).
§ 4. Примеры применения алгоритмов семейства ZET .. 122
4.1. Применение в экономике (122). 4.2. Применение в геологии и медицине (123). 4.3. Применения в технике (124).
§ 5. Алгоритмы семейства WANGA 124
5.1. Алгоритм WANGA–R (125). 5.2. Алгоритм WANGA–I (126). 5.3. Алгоритм WANGA–0 (126). 5.4. Алгоритм WANGA–N (128).

Глава 8. **Прогнозирование многомерных временных рядов** **129**

§ 1. Введение 129
§ 2. Обучающийся генетический алгоритм прогнозирования LGAP 130
2.1. Формирование базовых штаммов (130). 2.2. Отбор компетентных штаммов (133). 2.3. Выработка частных вариантов прогноза (135). 2.4. Получение окончательного прогноза (136).

§ 3. Критерии для оценки точности прогноза 140
§ 4. Возможности распараллеливания алгоритма LGAP 140
§ 5. Экспериментальная проверка алгоритма LGAP 141
§ 6. Коллективно-групповые методы распознавания (класс алгоритмов КГМ) 142

Глава 9. **Согласование разнотипных шкал** **147**

§ 1. Расстояние между объектами в пространстве разнотипных признаков 147
§ 2. Расстояние между разнотипными признаками 153

Глава 10. **Алгоритмы таксономии в λ-пространстве** **158**

§ 1. Алгоритм λ-KRAB 158
§ 2. Алгоритм λ-KRAB–2 164
§ 3. Выбор числа таксонов 164

Глава 11. **Методы распознавания образов в λ-пространстве** **166**

§ 1. Правило k ближайших соседей (алгоритм λ-NNR) . 167
§ 2. Выбор прецедентов (алгоритм λ-STOLP) 170
§ 3. Групповое распознавание 171
3.1. Алгоритм λ-ТРФ (171). 3.2. Алгоритм λ-GURAM (172).

Глава 12. **Другие задачи анализа данных в λ-пространстве** **174**

§ 1. Критерии информативности λ-пространства 174
§ 2. Задачи заполнения пробелов 175
§ 3. Пакет прикладных программ ОТЭКС 176

Глава 13. **Анализ данных и Data Mining** **179**

§ 1. Что такое Data Mining? 179
§ 2. Переоткрытие некоторых законов природы 183
2.1. Закон Ома (183). 2.2. Закон Менделя (186).
2.3. Периодический закон Менделеева (187).

Часть III. **Анализ знаний и структур**

Глава 14. **Метрика в пространстве знаний** **195**

§ 1. Меры близости между предикатами 195
§ 2. Расстояние между знаниями 198

Глава 15. **Методы анализа знаний** **200**

§ 1. Таксономия знаний 200
§ 2. Распознавание образов в пространстве знаний 201
§ 3. Выбор информативного подмножества предикатов . 203
§ 4. Заполнение пробелов в базе знаний 205

Глава 16. **Методы анализа структурных объектов** **207**

§ 1. Метод динамического программирования 208
§ 2. Метод скрытых марковских процессов (СМП) 212
§ 3. D-алгоритм для таксономии траекторий 216
§ 4. Иерархические структуры 218
§ 5. Расстояние между иерархиями 220
5.1. Расстояние по виду структуры (221). 5.2. Расстояние по весовым индексам (224).
§ 6. Таксономия иерархий 225
§ 7. Распознавание иерархических структур 226

Глава 17. **Анализ данных, знаний и структур в системах искусственного интеллекта** **227**

§ 1. Экспертные системы партнерского типа 227
§ 2. Отличительные характеристики ЭС и ПС 229
§ 3. Состояние разработок в области партнерских систем 232
3.1. Блок диалога (232). 3.2. Блок логического вывода (234). 3.3. Блок анализа данных (235). 3.4. Блок анализа моделей (236). 3.5. Блок «Гомеостат» (236). 3.6. Особенности системы ЭКСНА (237).
§ 4. Анализ данных, знаний и структур, связанных с изучением проблемы устойчивого развития 238

Заключение **245**

Литература **247**

Предметный указатель **261**