

Десять лекций по статистическому и структурному распознаванию образов

МИХАИЛ ИВАНОВИЧ ШЛЕЗИНГЕР
Национальная Академия наук, Украина, Киев

ВАЦЛАВ ГЛАВАЧ
Чешский Технический Университет, Чешская республика, Прага

Kluwer Academic Publishers
Boston/Dordrecht/London

Оглавление

Предисловие к русскому изданию	1
Предисловие к английскому изданию	3
Письмо аспиранта Иржи Пехи до опубликования лекций	5
Письмо авторов аспиранту Иржи Пехе	6
Основные понятия и обозначения	8
Благодарность	11
Глава 1 Байесовские задачи распознавания	13
1.1 Вводные замечания	13
1.2 Формулировка байесовских задач	14
1.3 Два свойства байесовских стратегий	15
1.4 Два частных случая байесовских задач	20
1.4.1 Вероятность ошибочного решения о состоянии	20
1.4.2 Байесовская стратегия отказа от распознавания	22
1.5 Обсуждение	24
1.6 Библиографические заметки	38
Глава 2 Небайесовские задачи распознавания	39
2.1 Жесткая ограниченность байесовского подхода	39
2.1.1 Функция потерь	40
2.1.2 Априорные вероятности состояний	40
2.1.3 Условные вероятности наблюдений	42
2.2 Формулировка известных и новых небайесовских задач	43
2.2.1 Задача Неймана-Пирсона	43
2.2.2 Обобщенная задача Неймана-Пирсона для двух опасных состояний	46
2.2.3 Минимаксная задача	46
2.2.4 Задача Вальда	47
2.2.5 Статистические решения при неслучайных воздействиях	49

2.3	Двойственные задачи линейного программирования, свойства и решения	50
2.4	Решения небайесовских задач на основе теорем двойственности	56
2.4.1	Решение задачи Неймана-Пирсона	57
2.4.2	Решение обобщенной задачи Неймана-Пирсона для случая двух опасных состояний	61
2.4.3	Решение минимаксной задачи	62
2.4.4	Решение задачи Вальда для случая двух состояний	64
2.4.5	Решение задач Вальда для произвольного количества состояний	67
2.4.6	Испытание сложных случайных гипотез	69
2.4.7	Испытание сложных неслучайных гипотез	70
2.5	Заключительные замечания	71
2.6	Обсуждение	71
2.7	Библиографические заметки	90
Глава 3	Две статистические модели распознаваемого объекта	93
3.1	Условная независимость признаков	93
3.2	Гауссово распределение вероятностей	95
3.3	Обсуждение	98
3.4	Библиографические замечания	120
Глава 4	Обучение распознаванию образов	123
4.1	Мифы об обучении в распознавании	123
4.2	Три формулировки задач обучения в распознавании	124
4.2.1	Обучение как максимально правдоподобное оценивание	127
4.2.2	Обучение по неслучайному обучающему множеству	128
4.2.3	Обучение по минимизации эмпирического риска	129
4.3	Основные понятия статистической теории обучения	130
4.3.1	Неформальное описание проблем обучения распознаванию	131
4.3.2	Основы статистической теории обучения распознаванию по Вапнику и Червоненкису	135
4.4	Критический взгляд на статистическую теорию обучения	143
4.5	Набросок детерминированного обучения	145
4.6	Обсуждение	150
4.7	Библиографические заметки	159
Глава 5	Линейные дискриминантные функции	161
5.1	Вводные замечания о линейных дискриминантных функциях	161
5.2	Путеводитель по лекции	162
5.3	Задачи Андерсона	165
5.3.1	Эквивалентные формулировки задач Андерсона	165
5.3.2	Неформальный анализ задачи Андерсона	167

5.3.3	Определение вспомогательных понятий для задачи Андерсона	170
5.3.4	Решение исходной задачи Андерсона	172
5.3.5	Формальный анализ обобщенной задачи Андерсона	174
5.4	Линейное разделение конечных множеств точек	182
5.4.1	Формулировка задачи и ее анализ	182
5.4.2	Алгоритмы линейного разделения конечных множеств точек	186
5.4.3	Алгоритмы ϵ -оптимального разделения конечных множеств точек	190
5.4.4	Построение фишеровских классификаторов модифицированными алгоритмами Козинца и перцептронными алгоритмами	192
5.4.5	Другие модификации алгоритмов Козинца	195
5.5	Решение обобщенной задачи Андерсона	198
5.5.1	ϵ -решение задач Андерсона	199
5.5.2	Линейное разделение бесконечных множеств точек	203
5.6	Обсуждение	206
5.7	Связь с toolbox'ом	227
5.8	Библиографические заметки	228
Глава 6	Самообучение	231
6.1	Вводные замечания об особом характере лекции	231
6.2	Предварительное неформальное определение самообучения в распознавании	233
6.3	Самообучение в перцептроне	234
6.4	Эмпирический байесовский подход Роббинса	242
6.5	Квадратичная кластеризация и формулировка общей задачи кластеризации	248
6.6	Алгоритмы самообучения и их анализ	254
6.6.1	Задача распознавания	254
6.6.2	Задача обучения	254
6.6.3	Задача самообучения	256
6.6.4	Алгоритм самообучения	257
6.6.5	Анализ алгоритмов самообучения	258
6.6.6	Алгоритм решения задачи Роббинса и ее анализ	268
6.7	Обсуждение	270
6.8	Связь с TOOLBOX'ом	291
6.9	Библиографические заметки	291
Глава 7	Заключительные замечания по статистическому распознаванию и вводные замечания по структурному распознаванию	293
7.1	Общность статистического распознавания	293
7.2	Почему для распознавания изображений нужно структурное распознавание?	295

7.2.1	Множество наблюдений	295
7.2.2	Множество скрытых параметров изображений	298
7.2.3	Роль обучения в распознавании изображений	300
7.3	Основные понятия структурного распознавания	303
7.4	Обсуждение	308
7.5	Библиографические замечания	325

Глава 8 Распознавание марковских последовательностей **327**

8.1	Вводные замечания о последовательностях	327
8.2	Марковская модель распознаваемого объекта	329
8.3	Распознавание стохастического автомата	333
8.3.1	Распознавание стохастического автомата; формулировка задачи	333
8.3.2	Алгоритм распознавания стохастического автомата	334
8.3.3	Матричное представление вычислительной процедуры	335
8.3.4	Статистическая интерпретация матричных произведений	336
8.3.5	Распознавание марковских объектов при неполных данных	339
8.4	Наиболее вероятная последовательность скрытых параметров	341
8.4.1	Различие между распознаванием объекта в целом и распознаванием его составных частей	341
8.4.2	Формулировка задачи поиска наиболее вероятной последовательности состояний	342
8.4.3	Сведение задачи к поиску кратчайшего пути на графе	342
8.4.4	Поиск кратчайшего пути на графе	344
8.4.5	О формальном анализе задач	346
8.4.6	Обобщенные матрицы и их умножение	349
8.4.7	Распознавание подпоследовательности состояний	352
8.5	Поиск последовательности наиболее вероятных состояний	355
8.6	Марковские объекты с ациклической структурой	361
8.6.1	Статистическая модель объекта	361
8.6.2	Вычисление вероятности наблюдения	362
8.6.3	Наиболее вероятная совокупность скрытых параметров	366
8.7	Формулировка задач обучения и самообучения	367
8.7.1	Наиболее правдоподобное оценивание модели в режиме обучения	368
8.7.2	Минимаксная оценка модели	369
8.7.3	Настройка алгоритма распознавания	369
8.7.4	Задача самообучения	370
8.8	Построение наиболее правдоподобной модели в режиме обучения	371
8.9	Минимаксное оценивание марковской модели	377
8.9.1	Формулировка алгоритма и его свойств	377
8.9.2	Анализ задачи минимаксного оценивания	380

8.9.3	Минимаксное оценивание марковской модели	390
8.10	Настройка алгоритма распознавания последовательностей . .	390
8.11	Наиболее правдоподобное оценивание модели в режиме самообучения	392
8.12	Обсуждение	396
8.13	Связь с TOOLBOX"ом	421
8.14	Библиографические замечания	421

Глава 9	Регулярные языки и соответствующие задачи распознавания	423
9.1	Регулярные языки	423
9.2	Другие способы представления регулярных языков	425
9.2.1	Регулярные языки и автоматы	425
9.2.2	Регулярные языки и грамматики	426
9.2.3	Регулярные языки и регулярные выражения	427
9.2.4	Пример регулярного языка, представленного различными способами	428
9.3	Модификации регулярных языков; точное и наилучшее соответствие	430
9.3.1	Размытые автоматы и размытые языки	431
9.3.2	Штрафные автоматы и соответствующие языки	432
9.3.3	Простейшая задача на наилучшее соответствие	433
9.4	Заключение по первой части лекции	435
9.5	Левенштейновская аппроксимация последовательности	437
9.5.1	Предварительная формулировка задачи	437
9.5.2	Функции Левенштейна	437
9.5.3	Известный алгоритм вычисления левенштейновского отклонения	438
9.5.4	Свойства функций Левенштейна	441
9.5.5	Формулировка задачи и ее обсуждение	444
9.5.6	Формулировка основного результата и его обсуждение	445
9.5.7	Обобщенные свертки и их свойства	447
9.5.8	Конволюционная формулировка задачи и основного результата	454
9.5.9	Доказательство основного результата	456
9.5.10	Интерпретация основного результата	467
9.6	Обсуждение	470
9.7	Связь с TOOLBOX"ом	506
9.8	Библиографические замечания	506

Глава 10	Контекстно-свободные языки, их двумерные обобщения и соответствующие задачи распознавания	509
10.1	Вводные замечания	509
10.2	Неформальное объяснение двумерных грамматик и языков . .	510

10.3	Двумерные контекстно-свободные грамматики и языки	515
10.4	Задача на точное соответствие. Обобщенный алгоритм Кока-Янгера-Касами	517
10.5	Общая структурная конструкция	519
10.5.1	Структурная конструкция для определения множеств распознаваемых объектов	521
10.5.2	Основная задача структурного распознавания изображений	524
10.5.3	Вычислительная процедура решения основной задачи	525
10.6	Обсуждение	528
10.7	Bibliographical notes	536
	Список литературы	537
	Предметный указатель	544

Предисловие к русскому изданию

Известно, что введение к книге пишется после того, как книга уже написана, и то, что получилось, можно представить, как задуманное с самого начала. В этом отношении данное издание находится в лучшем положении. Мы пишем это введение после того, как книга уже вышла сначала в чешской, а затем в английской версии. Таким образом, мы уже имеем некоторые представления о том, что получилось, а что еще нет.

Во введении к научной работе автор как бы извиняется за ее написание, доказывает, что она кому-то нужна, и так или иначе избегает признать, что выполненная работа нужна прежде всего ему самому. Мы не видим в таком признании ничего зазорного. Наоборот, что-то оказывается кому-то нужным скорее, если это делается для себя, чем если это делается только лишь для продажи. Ведь не зря говорят "Сделайте, как для себя", когда надеются на добросовестную работу. Мы написали эту книгу прежде всего потому, что она дозарезу была нужна нам самим. При этом неизбежно достигаются еще и какие-то другие цели. Укажем некоторые из них.

1. Мы оба преподаем различные разделы распознавания образов в Национальном Техническом Университете Украины (Киевский политехнический институт), Чешском Техническом Университете в Праге, в Международном Научно-Учебном Центре Информационных технологий в Киеве. Отдельные курсы нами были прочитаны в Университете имени Масарика в Брно (Чехия) и Дрезденском Техническом Университете. Содержание этой книги в сильной степени пересекается с содержанием наших лекций.

Как и в любой студенческой группе, в наших группах оказывалась часть студентов, которые отсутствовали на лекциях по причинам, которые им казались уважительными. В конце семестра они вынуждены были освоить учебный материал, располагая только данной книгой. Многие из них справились с этим вполне успешно. Написанная книга, повидимому, вполне пригодна для самостоятельного изучения основ распознавания образов, начиная с самых азов.

2. Мы оба работаем в коллективах, каждый в своем, которые профессионально работают в распознавании образов. Время от времени к нам приходит студент старшего курса или недавний выпускник и выражает желание работать в нашей группе. Распознавание образов не входило в вузовский курс его обучения, но он много слышал об этой области знаний, и она показалась ему привлекательной. Он уверен, что сможет плодотворно работать в этой области. Однако мы такой уверенностью не обладаем. Мы знаем, насколько сильно отличается проза науки о распознавании от поэтических представлений об искусственных устройствах, наблюдающих и понимающих окружающий мир так же, как человек или другие живые существа. Различие здесь приблизительно такое же, как между любовью к музыке и желанием овладеть фортепианным мастерством.

Наша обязанность состоит в том, чтобы как можно раньше показать молодому исследователю внутреннюю кухню распознавания и определить, действительно ли его привлекает наука о распознавании настолько, что-

бы приложить к ней те усилия, которые она требует. Или она нечто совсем другое, чем он ожидал. Это ознакомление не должно предполагать предварительных знаний о распознавании, но должно вестись достаточно энергично, чтобы быстро вывести будущего исследователя на современный уровень проблем. Отсутствие монографии именно такого назначения привело к массе наших неудобств и потерям нашего времени. Но значительно больше потери молодого исследователя, научный темперамент которого позволял ему достигнуть успехов вне распознавания. Сейчас, когда у нас есть эта книга, потеря такого рода стало значительно меньше.

3. Книга написана не только для наших молодых коллег, но и для исследователей, имеющих опыт в распознавании образов. Если бы мы адресовали книгу только нашим опытным коллегам, мы бы назвали ее как-то вроде "Неизвестное об известном в распознавании образов". Мы оба работаем в распознавании образов так давно, что в этом уже неудобно признаться. Естественно, что мы получили некоторые новые результаты и это само по себе не является поводом для написания книги. Существенным является то, что некоторые из результатов удалось получить только после критического анализа некоторых общепринятых представлений и понимания того, что эти представления неоправданно сужают проблему, а иногда просто ошибочны. До тех пор, пока эти общепринятые представления не преодолены, изложение новых результатов будет связано с неудобствами.

К счастью, все эти предрассудки, несмотря на свою распространенность, не укоренились слишком глубоко, так как вся наука о распознавании еще сравнительно молода и не приобрела жесткие рамки. Преодоление этих предрассудков не вызовет какой-то болезненной ломки, и рано или поздно они отомрут сами собой. Мы написали эту книгу, чтобы это произошло раньше.

4. И наконец, мы укажем ту цель, которую мы ставили до написания книги, но о которой не можем сейчас с уверенностью сказать, что она этой книгой достигается.

Мы считаем, что круг прикладных проблем распознавания образов сейчас очерчен достаточно хорошо. Область приложения распознавания образов разумно сужена и сейчас в нее уже не включают, например, машинную игру в шахматы или решение проблем управления, как это было при ее зарождении полвека назад. Значительно менее четко сформирована чистая научная проблематика на уровне абстрактных формальных понятий, не связанных с тем или иным конкретным приложением. Мы считаем, что наука о распознавании находится сейчас на таком этапе, что для ее прогресса требуется анализ абстрактных математических моделей в большей степени, чем дальнейшее наращивание списка возможных приложений. Это требует привлечения новых сил в нашу науку, с несколько иным, если можно так сказать, научным темпераментом. К сожалению, этому не способствует общественное мнение внутри нашего же распознавания. Считается, например, неприличным опубликовать статью по проблеме распознавания, в которой во введении не указано конкретное приложение, а сама статья не завершается какой-то картинкой. Этим достигается лишь то, что плодо-

творная научная идея оказывается безнадежно замаскированной конкретным приложением и становится мало доступной тем профессионалам вне распознавания, которые эту идею могли бы обогатить. Мы все вместе как будто прикладываем специальные усилия к тому, чтобы нас как можно меньше поняли наши коллеги за пределами распознавания. Более того, мы как будто стараемся, чтобы специалист по распознаванию, скажем, рукописных текстов ни в коем случае не понял работу по стереозрению.

Область возможных приложений распознавания образов исключительно обширна и многообразна. Однако мы сможем говорить о распознавании образов, как о науке, только тогда, когда научимся решать прикладные задачи оптом, а не в розницу. Для этого необходимо отойти от них на некоторое расстояние. С этой целью мы взяли на себя смелость опубликовать монографию, полностью посвященную проблемам распознавания, сформулированным в чистом виде без ссылок на какое-либо приложение.

М.И.Шлезингер, В.Главач, февраль 2003

Предисловие к английскому изданию

Современная наука о распознавании характеризуется таким разнообразием методов и подходов, что создается впечатление о бесчисленных, не связанных друг с другом островках, рассеянных в океане нашего незнания. В этом океане трудно ориентироваться и легко заблудиться. В данной монографии показано, что некоторые группы известных в настоящее время задач допускают общую формулировку. Таким образом обнаруживается родство задач, которые до сих пор казались различными и решались независимо друг от друга. Эти частные задачи уже перестают казаться разрозненными островками, а обретают форму по крайней мере архипелага с ясно видимыми путями перехода от одного островка к другому.

Построение общей схемы дает, однако, значительно больше, чем просто ориентацию в известных методах. Представление известных методов, как частных случаев единой общей схемы, позволяет обнаружить и другие, ранее не замеченные, но важные частные случаи.

Два обширные направления в современном распознавании: статистическое и структурное - изложены в книге с единой методологической точки зрения. Исследование задач структурного распознавания с точки зрения статистического позволило сформулировать и решить новые задачи структурного распознавания и, как следствие, понять ограниченную применимость известных методов. Помимо этого моста, объединяющего два крупных направления в современном распознавании, обнаружены взаимосвязи и отдельных задач внутри статистического и внутри структурного распознавания. Это также привело к неожиданным результатам.

Монография адресована как нашим коллегам, которые уже обладают собственным опытом в распознавании образов, так и будущим коллегам, которые нуждаются во введении в проблему. Материал книги излагается, начиная от самых основ, так что для ее чтения не требуется каких-либо

предварительных знаний. Требуется лишь желание эти знания приобрести. Для чтения большей части книги желательны навыки чтения не очень сложных математических текстов, так как основные результаты изложены в виде математически точных утверждений и их формальных доказательств. Ключевые понятия и рекомендации объясняются и иллюстрируются с помощью доступных интеллектуальных упражнений.

Выбранный стиль изложения, можно надеяться, делает книгу доступной для студентов и аспирантов в самом начале их научной карьеры. Именно таким читателям мы старались показать на конкретных примерах все прелести, парадоксы и ловушки распознавания. Каждая лекция завершается ее обсуждением с одним из лучших наших студентов. Дискуссии с ним позволили углубить понятия, введенные в лекции, сделать их более прозрачными, уменьшить вычислительную сложность введенных в лекции процедур. Когда это оказывалось уместным, мы указывали нашему юному коллеге на ошибочные представления, которые встречаются в распознавании образов, увы, значительно чаще, чем хотелось бы.

Лекции построены в стиле, в котором принято излагать математические дисциплины. Это значит, что основные понятия формально определены, а результаты сформулированы в виде теорем, для которых приводится их полное доказательство. Задачи распознавания и идеи их решения оказываются таким образом очищенными от ненужных подробностей того или иного приложения. Мы надеемся, что такой стиль изложения привлечет в нашу науку исследователей с математическим складом ума. Наука о распознавании нуждается в таких исследователях и заслуживает их внимания. Большая часть современных публикаций, к сожалению, не способствует привлечению, так как вместо формулировки решаемой математической задачи читатель находит в ней формулировку задач криминалистики, или медицины, или генетики, или экологии или других. Широкая область прикладной применимости распознавания образов достаточно хорошо известна и в силу этого уже привлекает внимание специалистов в самых разнообразных областях практики. Вместе с тем научное содержание решаемых нами проблем значительно меньше известно нашим коллегам, работающим в других областях информатики, не в распознавании образов.

Сказанное никак не противоречит тому, что данные лекции написаны практиками и для практиков. Для представленных в лекциях алгоритмов точно описана их область определения и поведение. Это делает возможным использовать их нашими коллегами-практиками, как строительных блоков при построении сложных прикладных программных комплексов. Ряд алгоритмов, описанных в лекциях, реализован в виде toolbox-ов, работающих в MATLAB. Программное обеспечение доступно по WWW-адресам, указанным в книге. Программы написаны нашими студентами из Чешского Технического университета.

Оба автора работают в Восточной Европе (Украина и Чешская республика) и опираются на научные традиции своих национальных школ. В книге содержатся ссылки на работы по распознаванию образов, выполненные в наших странах. Многие из них неизвестны в других частях мира. Та-

ким образом, книга открывает перед широкой научной общественностью по крайней мере часть картины, представляющей распознавание образов в наших странах.

Мы заинтересованы в обратной связи от читателей. Без лишних сомнений посылайте нам сообщения по адресам (schles@image.kiev.ua, hlavac@fel.cvut.cz). Желаем Вам увлекательного и полезного чтения.

М.И.Шлезингер, В.Главач, май 2001

Письмо аспиранта Иржи Пехи до опубликования лекций

Глубокоуважаемый господин профессор Главач, Чески Крумлов, 25.11.1996
я аспирант первого года обучения на электротехническом факультете Чешского технического университета. Только сейчас я узнал, что в весеннем семестре 1996 года некий профессор с Украины прочитал на нашем факультете курс лекций по математической теории распознавания образов. К сожалению, я об этом узнал слишком поздно и поэтому не смог посетить эти лекции. Я сожалею об этом все больше и больше, потому что коллеги, посетившие лекции, часто на них ссылаются. Мне сказали, что эти лекции не опубликованы, но мне пришла мысль, что, возможно, у Вас остались какие-то записки лектора или его конспекты, относящиеся к лекциям. Если это так, то позвольте мне, пожалуйста, снять с них копию.

Я намерен выполнить свою аспирантскую работу в области распознавания образов и понимания изображений. Некоторый опыт в этом направлении я уже приобрел, и этот опыт, к сожалению, далеко не положительный. Я написал довольно сложную программу по распознаванию буквенных символов по их изображениям. Я обошелся без чтения литературы и без глубоких исследований. В основу работы я положил свои собственные соображения, которые представлялись мне естественными и разумными. Однако полученные результаты оказались настолько неожиданно плохими, что сначала я подумал, что допустил массу ошибок при программировании своих идей. После тщательной отладки программ я убедился, что они написаны правильно. Кстати, я программирую вполне прилично. Следовательно, ошибочными оказались мои идеи, казавшиеся мне почти очевидными.

Теперь я вижу, что распознавание образов - это необозримое поле, в которое трудно войти, а войдя в него, легко заблудиться. Я попытался восполнить недостаток своих знаний в распознавании по учебникам и журнальным статьям. Некоторые публикации, которые я просмотрел, скорее популяризуют проблему, чем описывают ее. С более глубокими публикациями у меня серьезные проблемы. При их чтении у меня создается ощущение, будто я читаю увлекательный роман, но читаю его не с самого начала, как положено, а откуда-то с середины, и мне остается только строить догадки, кто именно скрывается под тем или иным именем и какие взаимосвязи отдельных действующих лиц. Конечно, для чтения этих работ

мне нехватает каких-то элементарных, базовых знаний, на которые в данной публикации нет ссылки, так как они предполагаются общеизвестными. Это естественно, так как эти публикации не рассчитаны на таких новичков, как я.

От своих преподавателей я знаю, что математические средства современного распознавания значительно более обширны, чем тридцать лет назад, когда распознавание образов базировалось почти полностью на математической статистике в многомерных линейных пространствах. Современное распознавание является областью приложения результатов теории графов, формальных языков и грамматик, марковских цепей и полей, математического программирования, вычислительной и дискретной геометрии. Более того, новые алгебраические конструкции рождаются и внутри самого распознавания образов. Со всеми этими областями я знаком значительно хуже, чем это нужно для того, чтобы в них свободно ориентироваться.

Кстати, о ссылках, которые я нашел в попавшихся мне публикациях. Чаще всего это были ссылки на малодоступные публикации того же автора, либо ссылки на солидные математические монографии, иногда многотомные. В них невозможно найти краткое и простое определение того элементарного понятия, которое нужно для прочтения статьи. Я определенно прихожу к выводу, что в современной обширной литературе по теории и практике распознавания образов отсутствует монография, или учебник, или публикация какого-то иного рода, которая бы служила введением в проблему. В такой книге я бы хотел найти определение наиболее важных понятий, формулировку базовых задач и сведения о тех математических средствах, которые необходимы для дальнейшей моей самостоятельной работы или, по крайней мере, для чтения научных статей по распознаванию. Книга должна знакомить с предметом при минимальных предположениях об осведомленности читателя. Короче говоря, мне нехватает книги, написанной специально для меня.

Как мне сказал мой научный руководитель, искать такую книгу-учебник-монографию совершенно тщетно. Мне дали совет, что до некоторой степени отсутствие такой книги восполняется лекциями профессора из Киева. Возможно, Вы будете так любезны и пришлете мне записи этих лекций, чтобы я смог снять с них копию.

Заранее искренне благодарю Вас.

Иржи Пеха

Письмо авторов аспиранту Иржи Пехе

Дорогой коллега Пеха,

Киев, 5 января 1997

в значительной мере мы согласны с Твоей оценкой современного состояния распознавания образов. Распознавание образов зародилось в 50-ые годы и в 90-ые годы переживает новый всплеск своей популярности. Область возможных применений здесь так широка, что трудно сказать, где только

не пытаются применить распознавание образов. Эта область простирается от микромира до космоса.

Естественно, такая популярность не может не вдохновлять каждого, кто профессионально работает в этой области. В то же время она вызывает оправданную тревогу. На распознавание образов опять, как и полвека назад, возлагают такие надежды, которые вряд ли когда-нибудь сбудутся. Распознавание образов потихоньку опять начинает обретать репутацию палочки-выручалочки, с помощью которой одним махом решаются практические задачи без их детального и изнурительного изучения. Такая репутация вызывает здравый скептицизм трезвых практиков и раздражение здравомыслящих исследователей. Для более доверчивых такая репутация станет источником серьезных разочарований, когда опять станет ясно, что одно лишь использование слов "распознавание", "нейронные сети", "искусственное мышление" и тому подобных вовсе не гарантирует решение практических задач.

Сейчас наступило самое подходящее время, чтобы произвести своего рода инвентаризацию распознавания образов и отделить действительность от сладких грез о чудодейственных способностях распознающих систем. Распознавание образов уже выросло из младенческого возраста, когда требовало к себе бережного отношения. Сейчас оно уже не нуждается в снисхождении, и его авторитет может строиться не на мифах, а действительности.

Ты прав, что лекции, которые Ты пропустил, в какой-то мере содержат именно то, что Тебе нужно. Однако вряд ли Тебе окажутся полезными записки о лекциях, которые сейчас у нас есть. В них содержатся формулировки основных понятий и теорем, и это лишь незначительная часть того, что происходило в аудитории. В лекциях содержалась краткая мотивация для рассмотрения того или иного раздела распознавания, затем критический (зачастую довольно строгий) анализ некоторых известных подходов и предостережение от всевозможных ловушек. Имеются в виду так называемые разумные решения, вроде бы основанные на здравом смысле, но на самом деле ошибочные. Именно после таких комментариев, дополнений к основному материалу мы осознавали, как мало мы знаем даже о тех разделах распознавания, о которых думали, что разбираемся в них вполне прилично.

Мы неоднократно задавались над тем, не следует ли опубликовать все эти интересные, но не совсем академические соображения, и в свое время решили, что опубликовать их невозможно или по крайней мере очень трудно. Однако теперь мы все чаще сомневаемся в правильности нашего решения. Мы начинаем думать, что публикация этих лекций будет полезной не только для Тебя. Сейчас мы приводим эти лекции к виду, удобному для чтения. Однако это будет несколько иная книга, чем Ты себе представлял. Ты бы хотел иметь книгу типа справочника, в котором отдельные главы посвящены отдельным математическим дисциплинам, составляющим математический аппарат распознавания. Поначалу и мы думали написать именно такой справочник, но теперь изменили свои замыслы. Именно под влиянием Твоего письма мы поняли, что Ты нуждаешься не в математи-

ческом справочнике, а в чем-то другом.

Хотя Ты довольно скромно оцениваешь свои знания математических средств современного распознавания, Ты правильно догадываешься, что речь идет, как правило, о базовых, то есть самых элементарных сведениях, заимствованных распознаванием извне. Твои проблемы состоят не в том, что эти сведения так уж сложны, а в том, что их много, они чересчур разнообразны и почерпнуты из разных математических дисциплин. До того, как эти понятия были перенесены в распознавание, они не приходили в соприкосновение друг с другом. Новичка приводит в замешательство также то, что в рамках распознавания заимствованные понятия приобретают несколько иную окраску, а зачастую и иное название, чем в материнских математических дисциплинах. Современное распознавание состоит не только в использовании известных математических методов для решения той или иной практической задачи. Используемые в распознавании известные методы приходят в необычное для них соприкосновение, что приводит к новым понятиям, новым задачам и в конечном итоге к результатам, которые входят уже в науку о распознавании, а не в исходные математические дисциплины.

Лекции, которые мы сейчас приводим в порядок, посвящены проблемам распознавания, а не задачам линейного программирования, математической статистики или теории графов. Требуемые математические средства вводятся не изолированно друг от друга, а во взаимосвязи, в которую они должны вступить для решения той или иной задачи распознавания. Известные математические средства таким образом становятся составными частями единого механизма, который можно назвать математическим аппаратом распознавания. Нам кажется, что именно такого взгляда на распознавание Тебе сейчас не хватает и в этом корень Твоих затруднений.

К сожалению, сейчас мы только начали приводить лекции к виду, пригодному для опубликования. Работа потребует приблизительно два года. Мы понимаем, что Тебе не хотелось бы простаивать все это время, и предлагаем Тебе сотрудничество. Мы можем посылать Тебе тексты лекций по мере их написания. Было бы замечательно, если бы Ты сразу после получения части материала критически прокомментировал его, сообщил нам о своих идеях или вопросах, если они возникнут. Давай считать такую обратную связь необходимым условием нашего сотрудничества. Очередную порцию материала Ты получишь только после того, как мы получим Твой анализ предыдущей.

Сейчас мы высылаем Тебе первую лекцию и с нетерпением ожидаем, что же получится в результате нашего сотрудничества.

Шлезингер Михаил Иванович, Главач Вацлав

Основные понятия и обозначения

Прежде всего введем обозначения множеств. *Множество*, состоящее из небольшого количества элементов, будет представлено списком своих эле-

ментов, заключенного в фигурные скобки. Например, $\{A, B, C\}$ обозначает множество, состоящее из трех букв: A, B и C . Множества, состоящие из большого количества элементов, будут представляться в виде $\{x | \varphi(x)\}$, где x обозначает общий элемент, а $\varphi(x)$ определяет свойство тех и только тех элементов, которые входят в данное множество. Например, $\{x | 0 \leq x \leq 100\}$ есть множество чисел, удовлетворяющих условию $0 \leq x \leq 100$.

Конъюнкция (логическое умножение) обозначается как \wedge , а *дизъюнкция* (логическое сложение) - как \vee .

Обычно используемые *операции над множествами* будут обозначаться как $X \cup Y$ для объединения, $X \cap Y$ - для пересечения, $X \times Y = \{(x, y) | (x \in X) \wedge (y \in Y)\}$ - для декартова произведения и $X \setminus Y = \{x | (x \in X) \wedge (x \notin Y)\}$ - для разности двух множеств.

Пусть X и Y - два множества. Мы будем обозначать $f: X \rightarrow Y$ *функцию*, которая каждому элементу $x \in X$ ставит в соответствие элемент $f(x) \in Y$. Множество X есть область определения функции f , множество Y - область ее значений, а $f(x)$ - значение, которое функция f принимает в точке $x \in X$.

Пусть \mathbb{Z} - множество целых чисел, n - неотрицательное целое число, $I(n)$ - подмножество $\{i | 0 \leq i \leq n\}$, а X - некоторое множество. Функцию $x: I(n) \rightarrow X$ будем понимать, как *последовательность*. Ее i -ый элемент будет обозначаться $x(i)$ или x_i . Последовательность будем задавать упорядоченным списком ее элементов, например, x_1, x_2, \dots, x_n .

Как функцию, так и последовательность будем понимать как *совокупность* ее значений, которую будем обозначать $(x_y, y \in Y)$.

Помимо понятия множества полезным окажется понятие *мультимножества*. Конечное множество можно задать списком, в котором нет одинаковых элементов, то есть каждый элемент записан только один раз. Конечное мультимножество также можно записать в виде списка, однако тот или иной элемент может в этом списке появиться более одного раза. В отличие от списков, представляющих множества, списки, представляющие мультимножества, будут заключены не в фигурные, а круглые скобки. Пусть, к примеру, X - это множество $\{1, 2, 3, 4, 5, 6\}$, а $\varphi: X \rightarrow X$ - функция, определенная значениями $\varphi(1) = 1$, $\varphi(2) = 1$, $\varphi(3) = 2$, $\varphi(4) = 2$, $\varphi(5) = 6$, $\varphi(6) = 1$. Тогда $\{1, 6, 2\}$ - это множество $\{\varphi(x) | x \in X\}$, а $(1, 2, 6, 1, 2, 1)$ - это мультимножество $(\varphi(x) | x \in X)$.

Таким образом, мы будем различать три понятия: *множество*, *мультимножество* и *последовательность*. Множество задается списком элементов, которые попарно различны. Произвольная перестановка элементов в этом списке не меняет множество, представленное этим списком. Мультимножество также задается списком, в котором, однако, могут содержаться и одинаковые элементы. Так же, как и в множестве, порядок элементов в списке можно произвольным образом менять и это не влияет на мультимножество, представленное этим списком. Последовательность также может задаваться списком. В этом списке могут встречаться и одинаковые элементы. Однако изменение их порядка в списке меняет и последовательность, которую этот список представляет.

Пусть X - конечное множество, а \mathbb{R} - множество вещественных чисел. Функция $p: X \rightarrow \mathbb{R}$ будет называться *распределением вероятностей* на множестве X если $\sum_{x \in X} p(x) = 1$ и если неравенство $p(x) \geq 0$ выполняется для всех $x \in X$. Значение $p(x)$ будет называться вероятностью элемента x .

Почти везде в данной книге мы будем иметь дело с распределениями вероятностей $p: X \rightarrow \mathbb{R}$ на конечных множествах X . Мы идем на это сужение намеренно, чтобы избежать ненужных усложнений, которые возникли бы при рассмотрении бесконечных множеств и затуманили бы более важные идеи. Поскольку мы будем рассматривать в основном конечные множества, обозначение $|X|$ будет пониматься просто, как количество элементов в этом множестве.

Мы будем рассматривать распределения вероятностей, заданных для разнообразных областей определения. Все эти функции будут, как правило, обозначаться буквой p . Если возникнет необходимость различать вероятности, имеющие различные области определения, обозначение p будет дополняться нижним индексом, который будет указывать соответствующую область определения. Так, p_X есть функция $X \rightarrow \mathbb{R}$, а p_Y - функция $Y \rightarrow \mathbb{R}$.

Пусть X и Y - два множества, а p_{XY} - распределение вероятностей $X \times Y \rightarrow \mathbb{R}$. Число $p_{XY}(x, y)$ называется *совместной вероятностью* события, состоящего в том, что соответствующие переменные приняли значения x и y .

Обозначение $p_{X|Y}$ будет использовано для функции $X \times Y \rightarrow \mathbb{R}$, значение которой для пары значений $x \in X$ и $y \in Y$ определяется выражением

$$p_{X|Y}(x|y) = \frac{p_{XY}(x, y)}{\sum_{y \in Y} p_{XY}(x, y)}.$$

Определенную таким образом функцию $p_{X|Y}$ двух переменных будем называть *условным* (а также *апостериорным*) *распределением вероятностей*. Значение $p_{X|Y}(x|y)$ этой функции - это условная вероятность значения $x \in X$ при условии $y \in Y$.

В некоторых случаях окажется удобным представлять функцию $p_{X|Y}$ двух переменных x и y в виде совокупности $(p_{X|y}, y \in Y)$ функций одной переменной x . При этом функция $p_{X|y}$ в целом зависит от y как от параметра. Условная вероятность случайного значения $x \in X$ при условии $y \in Y$ будет в этом случае обозначаться $p_{X|y}(x)$. В других случаях будет считаться фиксированным значение x , а не y , а функция $p_{X|Y}$ будет пониматься как совокупность $(p_{x|Y}, x \in X)$ функций одной переменной y . Функция $p_{x|Y}: Y \rightarrow \mathbb{R}$ в целом зависит от значения x , как от параметра. Хотя число $p_{x|Y}(y)$ есть условная вероятность значения x при условии y , функция $p_{x|Y}: Y \rightarrow \mathbb{R}$ не является распределением вероятностей на множестве Y , так как сумма $\sum_{y \in Y} p_{x|Y}(y)$ не обязательно равна единице. Число $p_{x|Y}(y)$ будем называть *правдоподобием* значения y .

Для заданного совместного распределения вероятностей $p_{XY}: X \times Y \rightarrow \mathbb{R}$ *априорным распределением вероятностей* будем называть функцию

$p_X: X \rightarrow \mathbb{R}$, определенную как

$$p_X(x) = \sum_{y \in Y} p_{XY}(x, y).$$

Пусть X - множество, $p_X: X \rightarrow \mathbb{R}$ - распределение вероятностей, а $f: X \rightarrow \mathbb{R}$ - функция. Число

$$\sum_{x \in X} f(x) p_X(x)$$

называется *математическим ожиданием* случайной переменной $f(x)$ и обозначается $E(f)$.

Пусть X - множество, а $f: X \rightarrow \mathbb{R}$ - функция. Обозначение $\operatorname{argmax}_{x \in X} f(x)$ будет использовано для любого значения $x^* \in X$, для которого $f(x^*) = \max_{x \in X} f(x)$. Аналогичный смысл имеет обозначение $\operatorname{argmin}_{x \in X} f(x)$.

Скалярное произведение векторов a и b будет обозначаться $\langle a, b \rangle$. В описании некоторых алгоритмов будет использовано обозначение $::=$ для операторов присвоения. Формулировки теорем, лемм, примеров и замечаний заканчиваются символом \blacktriangle у правого поля текста. Окончания доказательств обозначены в тексте обозначены символом \blacksquare .

Книга состоит из десяти пронумерованных лекций. Лекции состоят из разделов (пронумерованных, к примеру, 2.3) и подразделов (например, 2.3.1). Обсуждение материала со студентом Иржи Пехой приведено в последнем разделе каждой лекции.

Благодарность

Этой книгой мы хотим выразить наше глубокое уважение и благодарность нашим учителям профессору В.А.Ковалевскому и профессору З.Котеку. Если появление этой книги - это заслуга, то это прежде всего их заслуга. Мы будем считать величайшей честью для себя, если наши учителя в число своих достижений включают и эту книгу.

Работа над чешской версией книги продолжалась значительно дольше, чем мы ожидали. Все это время нам оказывала неоценимую помощь Т.Н.Барабанюк.

Мы благодарны многим нашим коллегам и студентам за дискуссии, стимулирующие нашу работу в течение двух лет работы над чешской версией. Эти дискуссии не прерывались и в последующие почти два года, когда мы работали над английской версией.

Мы благодарны госпоже доц. М.Тлалковой за ее помощь при переводе с чешского языка на английский.

Рукопись книги прочитали гг. доктор И.Грим (рецензент чешского издания), доцент М.Навара, доктор И.Матас, профессор Б.Мелихар, доцент Ц.Матиска, профессор Х.Бишоф, аспиранты П.Бечварж, А.Фитч, Л.Янку, Б.Куркоски, Д.Пруша, Д.Бересфорд. Студенты В.Франц и Я.Дупач в качестве своих проектов разработали toolbox в Matlab[®], в котором реализовано

ваны многие алгоритмы, описанные в книге. Аспирант В.Зика оказал нам существенную помощь в типографской подготовке книги в \LaTeX .

Мы благодарим директора Издательства Чешского Технического Университета г-жу доктора И.Смолкову, за помощь в издании чешской версии, равно как и за передачу авторских прав на английское и русское издание. Мы благодарим профессора М.Вииргевера, ответственного редактора серии Машинное зрение Клуверовского издательства, который в 1999 году положительно воспринял идею об опубликовании книги, а затем непрерывно стимулировал наши усилия. Большую помощь нам оказали также редакторы серии доктор П.Роос, господин Дж.Финлей и их ассистент госпожа Инге Гардон.

Авторы благодарны друг другу за сотрудничество, которое вопреки географической удаленности, различию в возрасте, научном темпераменте и языку удалось довести до сегодняшнего результата.

Глава 1

Байесовские задачи распознавания

1.1 Вводные замечания

Байесовская теория принятия решений является одним из блоков, из которых строится статистическая теория распознавания. В этой лекции мы введем основные понятия, сформулируем байесовскую задачу принятия решений и укажем наиболее важные общие свойства этих задач. Понятия, введенные на этой лекции, будут неоднократно использоваться на протяжении всего курса. Эти понятия считаются общеизвестными, и мы были бы рады, если бы это было действительно так. В действительности же нередко рекомендации, которые прямо-таки противоречат выводам байесовской теории, хотя на первый взгляд представляются вполне естественными. Это свидетельствует о поверхностном или частичном понимании байесовской теории, а это хуже, чем полное незнание. Несомненно прав был тот, кто сказал, что лучше иметь дело с человеком, который не читает книг вообще, чем с тем, кто прочитал одну книгу.

Конечно, имеется много причин для разнобоя в понимании, что такое байесовская теория. Слишком уж знаменито имя Байес. Формула, по которой подсчитываются апостериорные вероятности, носит имя Байеса. Этим же именем называют рекомендацию принимать решение о значении случайной величины в пользу наиболее вероятного ее значения. В современной прикладной статистике появилось понятие "байесовские сети", как модели сложного случайного объекта. Поэтому вовсе нередка ситуация, когда оба собеседника спорят о байесовском методе распознавания и только после длительных недоразумений выясняют, что говорят о разных вещах.

Мы настоятельно просим читателя, не знакомого с этими понятиями, не переходить к чтению последующих глав до ознакомления с этой лекцией и с комментариями Иржи Пехи в конце лекции. С этой же просьбой мы обращаемся к читателю, знакомому с байесовской теорией: убедимся на всякий случай, что мы имеем в виду одно и то же.

1.2 Формулировка байесовских задач

Пусть объект характеризуется двумя параметрами: x и k . Первый параметр является наблюдаемым, а второй скрыт, то есть недоступен для непосредственного наблюдения. Параметр x будет называться *признаком объекта*, или *наблюдением* или наблюдаемым параметром. Признак x принимает значения из конечного множества X . Второй параметр будет называться *состоянием объекта* или *скрытым параметром*. Обозначим символом K конечное множество значений скрытого параметра k . Символ D будет обозначать *множество возможных решений*.

Здесь необходимо отметить, что скрытый параметр k иногда понимают, как подмножество наблюдений, называя его классом, а множество K понимают как конечную систему подмножеств наблюдений. Процесс распознавания в этом случае понимается, как классификация наблюдаемого объекта. Мы не будем себя ограничивать таким пониманием скрытого параметра k , потому что в ряде приложений такое понимание не совсем естественно. Например, скрытый параметр k может обозначать положение объекта в поле зрения робота, а наблюдение x - это изображение, полученное камерой. В этом случае несколько неестественно считать обнаружение объекта в поле зрения его классификацией.

На множестве $X \times K$ всех возможных пар наблюдений $x \in X$ и состояний $k \in K$ задано распределение вероятностей $p_{XK}: X \times K \rightarrow \mathbb{R}$ так, что для каждого состояния $k \in K$ и наблюдения $x \in X$ число $p_{XK}(x, k)$ обозначает совместную вероятность того, что объект находится в состоянии k , а признак этого объекта принял значение x .

Пусть $W: K \times D \rightarrow \mathbb{R}$ - это *штрафная функция*, такая, что $W(k, d)$, $k \in K$, $d \in D$, обозначает штраф, который выплачивается, когда объект находится в состоянии k и принято решение d . Функция W называется также функцией потерь. Пусть $q: X \rightarrow D$ обозначает функцию, которая каждому наблюдению $x \in X$ присваивает решение $q(x) \in D$. Функция q называется стратегией решения, или просто стратегией, или решающей функцией. Математическое ожидание штрафа, который выплачивается при использовании стратегии q , считается качеством этой стратегии и называется *риском*. Риск стратегии q обозначим $R(q)$.

Байесовская задача статистических решений состоит в том, чтобы для заданных множеств X , K , D , заданных функций $p_{XK}: X \times K \rightarrow \mathbb{R}$, $W: K \times D \rightarrow \mathbb{R}$ построить стратегию $q: X \rightarrow D$, которая минимизирует байесовский риск

$$R(q) = \sum_{x \in X} \sum_{k \in K} p_{XK}(x, k) W(k, q(x)).$$

Решением байесовской задачи является стратегия q^ , которая минимизирует риск $R(q)$. Она называется байесовской стратегией.*

Подчеркнем чрезвычайную общность приведенной постановки байесовской задачи. Мы ведь ничего еще не сказали о том, как устроены множества наблюдений X , состояний K и решений D , то есть не ввели никакие ограничения на то, какую математическую форму должны иметь элементы этих

множеств. Наблюдение $x \in X$, например, в зависимости от приложения, может быть числом или нечисловым математическим объектом. Это может быть символ в абстрактном алфавите, или вектор, или последовательность символов, это может быть функция одной переменной (скажем, процесс) или функция двух переменных (например, изображение). Это может быть функция с более сложной областью определения, чем множество значений одной или двух численных переменных. Например, это может быть функция, имеющая областью определения граф или какую-то другую алгебраическую структуру. Множества состояний K и решений D могут иметь столь же разнообразную форму.

В этом курсе мы подвергнем как можно более тщательному анализу различные конкретизации байесовской задачи, которые оказались плодотворными в распознавании образов. Но перед этим мы укажем наиболее общие свойства, которые имеют место для всего класса байесовских задач. Этих свойств не так уж много и они имеют достаточно простую формулировку. Однако в силу того, что они имеют место для любого частного приложения, они имеют характер законов, незнание которых приводит к грубым ошибкам.

1.3 Два свойства байесовских стратегий

Байесовская задача сформулирована как поиск детерминированной стратегии $q: X \rightarrow D$. Это значит, что с самого начала требуется, чтобы всякий раз при фиксированном значении x признака принималось одно и то же решение $d = q(x)$. При этом как бы игнорируется тот факт, что объект, излучающий одно и то же значение признака x , может находиться в самых разных состояниях. Создается впечатление, что детерминированный характер стратегии входит в противоречие со случайным характером состояния, в котором находится объект. Уж если наблюдаемый сию минуту объект может находиться случайно либо в том состоянии, либо в ином, то кажется, что и воздействовать на него следует случайным образом, один раз так, а другой раз иначе. Эти соображения стимулируют к тому, чтобы как-то расширить множество стратегий, среди которых должна быть найдена наилучшая. Множество стратегий следовало бы расширить так, чтобы оно включало не только все возможные функции вида $q: X \rightarrow D$, но и все возможные распределения вероятностей $q_r(d|x)$. Эти распределения понимались бы, как рандомизированные стратегии, при которых для каждого наблюдения x случайно выбирается подходящее решение d в соответствии с вероятностями $q_r(d|x)$.

Следующая теорема покажет, что бесполезно ожидать каких-то чудесных свойств от применения рандомизированных стратегий.

Теорема 1.1 Детерминированный характер байесовских стратегий.

Пусть X, K, D - конечные множества, $p_{XK}: X \times K \rightarrow \mathbb{R}$ - распределение вероятностей, $W: K \times D \rightarrow \mathbb{R}$ - штрафная функция. Пусть $q_r: D \times X \rightarrow \mathbb{R}$

- рандомизированная стратегия, риск которой равен

$$R_{\text{rand}} = \sum_{x \in X} \sum_{k \in K} p_{XK}(x, k) \sum_{d \in D} q_r(d | x) W(k, d). \quad (1.1)$$

В таком случае существует детерминированная стратегия $q: X \rightarrow D$, риск которой

$$R_{\text{det}} = \sum_{x \in X} \sum_{k \in K} p_{XK}(x, k) W(k, q(x))$$

не больше, чем R_{rand} . ▲

Доказательство. Перепишем равенство (1.1) в иной форме,

$$R_{\text{rand}} = \sum_{x \in X} \sum_{d \in D} q_r(d | x) \sum_{k \in K} p_{XK}(x, k) W(k, d).$$

Равенство $\sum_{d \in D} q_r(d | x) = 1$ выполняется для любого $x \in X$, а неравенство $q_r(d | x) \geq 0$ выполняется для любых $d \in D$ и $x \in X$. В силу этого справедливо неравенство

$$R_{\text{rand}} \geq \sum_{x \in X} \min_{d \in D} \sum_{k \in K} p_{XK}(x, k) W(k, d). \quad (1.2)$$

Обозначим $q(x)$ любое значение d , для которого

$$\sum_{k \in K} p_{XK}(x, k) W(k, q(x)) = \min_{d \in D} \sum_{k \in K} p_{XK}(x, k) W(k, d). \quad (1.3)$$

Определенная таким образом функция $q: X \rightarrow D$ есть детерминированная стратегия, которая не хуже, чем рандомизированная стратегия q_r . Действительно, подставим (1.3) в (1.2) и получим

$$R_{\text{rand}} \geq \sum_{x \in X} \sum_{k \in K} p_{XK}(x, k) W(k, q(x)).$$

В правой части этого неравенства записан риск детерминированной стратегии q . Таким образом, $R_{\text{det}} \leq R_{\text{rand}}$. ■

Мы видим, что с точки зрения математического ожидания потерь рандомизация стратегии не может улучшить детерминированную стратегию.

Второе важное свойство байесовских стратегий объясним сначала на простом примере, когда скрытый параметр принимает только два значения, $K = \{1, 2\}$. Предположим, что нам известны не все данные, которые нужны для построения байесовской стратегии $q: X \rightarrow D$. Предположим, что нам известны только условные вероятности $p_{X|1}(x)$ и $p_{X|2}(x)$, а априорные вероятности $p_K(1)$ и $p_K(2)$ и штрафы $W(k, d)$, $k \in \{1, 2\}$, $d \in D$, неизвестны. В этом случае байесовская стратегия не может быть построена. Однако можно показать, что уже не любая функция может служить байесовской стратегией. Стратегия должна принадлежать определенному классу функций, то есть обладать определенными свойствами.

Если бы априорные вероятности $p_K(k)$ и штрафы $W(k, d)$ были известны, надо было бы принять следующее решение $q(x)$ относительно наблюдения x ,

$$\begin{aligned}
 q(x) &= \operatorname{argmin}_d (p_{XK}(x, 1) W(1, d) + p_{XK}(x, 2) W(2, d)) \\
 &= \operatorname{argmin}_d (p_{X|1}(x) p_K(1) W(1, d) + p_{X|2}(x) p_K(2) W(2, d)) \\
 &= \operatorname{argmin}_d \left(\frac{p_{X|1}(x)}{p_{X|2}(x)} p_K(1) W(1, d) + p_K(2) W(2, d) \right) \\
 &= \operatorname{argmin}_d (\gamma(x) c_1(d) + c_2(d)).
 \end{aligned} \tag{1.4}$$

В последней строчке цепочки равенств (1.4) были введены обозначения $c_1(d) = p_K(1) W(1, d)$, $c_2(d) = p_K(2) W(2, d)$ и введено важное понятие: *отношение правдоподобия* $\gamma(x) = p_{X|1}(x)/p_{X|2}(x)$. По выражению (1.4) видно, что подмножество $X(d^*)$ наблюдений, для которых должно быть принято некоторое фиксированное решение d^* , есть множество решений системы неравенств

$$\gamma(x) c_1(d^*) + c_2(d^*) \leq \gamma(x) c_1(d) + c_2(d), \quad d \in D \setminus \{d^*\}.$$

Отношение правдоподобия $\gamma(x)$ входит линейно в каждое неравенство этой системы, в силу чего множество $X(d^*)$ соответствует выпуклому подмножеству значений отношения правдоподобия $\gamma(x)$. Величины $\gamma(x)$ - это вещественные числа, и поэтому этими выпуклыми подмножествами являются просто численные интервалы. Таким образом, мы можем сформулировать важное свойство байесовских стратегий для случая, когда скрытое состояние может принимать только два значения (а количество решений может быть и большим):

Любая байесовская стратегия делит ось вещественных чисел на $|D|$ интервалов $I(d)$, $d \in D$. По наблюдению $x \in X$ принимается решение d , если отношение правдоподобия $\gamma = p_{X|1}(x)/p_{X|2}(x)$ принадлежит интервалу $I(d)$.

В еще более частном случае, когда множество D состоит всего лишь из двух решений, $D = \{1, 2\}$, получается общеизвестный результат. В этом случае байесовская стратегия определяется единственным параметром - пороговым значением θ . По наблюдению x принимается то или иное решение в зависимости от того, превышает ли отношение правдоподобия $\gamma = p_{X|1}(x)/p_{X|2}(x)$ порог θ или нет.

Обобщим теперь этот общеизвестный вывод на случай, когда скрытый параметр принимает более чем два значения и понятие отношение правдоподобия в этом случае уже перестает иметь смысл. Для этого еще раз вернемся к уже рассмотренному случаю, когда $|K| = 2$ и $|D| \geq 2$, и представим байесовскую стратегию в ином виде. Каждое наблюдение $x \in X$ будем представлять точкой на плоскости. Координатами этой точки будут числа $p_{X|1}(x)$ and $p_{X|2}(x)$. Таким способом множество наблюдений X

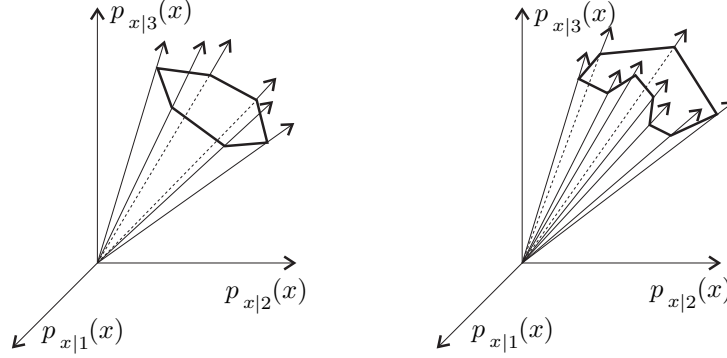


Figure 1.1 Convex cone (left) and non-convex one (right).

отображается в верхний правый квадрант плоскости, а каждое из подмножеств $X(d)$, $d \in D$, отображается в сектор, ограниченный двумя прямыми, проходящими через начало координат. Эти сектора, конечно же, являются выпуклыми множествами.

Рассмотрим теперь более общий случай, когда $|K| > 2$. Пусть Π есть $|K|$ -мерное линейное пространство. Подмножество Π' пространства Π называется *конусом*, если $\alpha\pi \in \Pi'$ для любого $\pi \in \Pi'$ и любого вещественного числа $\alpha > 0$. Если подмножество является конусом и, кроме того, является выпуклым, то оно называется *выпуклым конусом*, см. рис. 1.1.

Отобразим множество наблюдений X в положительный гиперквадрант пространства Π , то есть в множество точек с неотрицательными координатами, так, что наблюдению $x \in X$ соответствует точка $\pi(x)$ с координатами $p_{X|k}(x)$.

Любой байесовской стратегии соответствует разбиение положительного гиперквадранта пространства Π на $|D|$ выпуклых конусов $\Pi(d)$, $d \in D$, так, что по наблюдению x принимается решение d , когда $\pi(x) \in \Pi(d)$. Некоторые из конусов могут быть пустыми.

Выразим это свойство байесовских стратегий в виде следующей теоремы.

Теорема 1.2 Выпуклая форма классов в пространстве вероятностей.

Пусть X , K , D - конечные множества, а $p_{XK}: X \times K \rightarrow \mathbb{R}$, $W: K \times D \rightarrow \mathbb{R}$ - функции. Пусть $\pi: X \rightarrow \Pi$ - отображение множества X в $|K|$ -мерное линейное пространство Π ; $\pi(x) \in \Pi$ есть точка с координатами $p_{X|k}(x)$, $k \in K$.

Пусть любое разбиение положительного гиперквадранта пространства Π на $|D|$ выпуклых конусов $\Pi(d)$, $d \in D$, определяет q так, что $q(x) = d$ тогда и только тогда, когда $\pi(x) \in \Pi(d)$.

В таком случае существует такое разбиение $\Pi^(d)$, $d \in D$, что соответствующая стратегия минимизирует байесовский риск*

$$\sum_{x \in X} \sum_{k \in K} p_{XK}(x, k) W(k, q(x)). \quad \blacktriangle$$

Доказательство. Покажем, как нужно строить семейство конусов, существование которого требуется доказать. Пронумеруем все решения из D так, что $n(d)$ есть номер решения d . Укажем одну из стратегий, которая минимизирует риск. Это может быть стратегия, которая принимает решение d^* для таких и только таких наблюдений x , что

$$\sum_{k \in K} p_{X|k}(x) p_K(k) W(k, d^*) \leq \sum_{k \in K} p_{X|k}(x) p_K(k) W(k, d), \quad n(d) < n(d^*),$$

$$\sum_{k \in K} p_{X|k}(x) p_K(k) W(k, d^*) < \sum_{k \in K} p_{X|k}(x) p_K(k) W(k, d), \quad n(d) > n(d^*).$$

Эту систему неравенств можно представить с использованием координат точек $\pi(x) \in \Pi$, то есть чисел $\pi_k = p_{X|k}(x)$. Образует множества $\Pi(d^*)$ как множества точек π с координатами $\pi_k, k \in K$, которые удовлетворяют неравенствам

$$\begin{aligned} \sum_{k \in K} \pi_k p_K(k) W(k, d^*) &\leq \sum_{k \in K} \pi_k p_K(k) W(k, d), \quad n(d) < n(d^*), \\ \sum_{k \in K} \pi_k p_K(k) W(k, d^*) &< \sum_{k \in K} \pi_k p_K(k) W(k, d), \quad n(d) > n(d^*). \end{aligned} \quad (1.5)$$

Из приведенной системы неравенств видно, что выраженные таким образом множества $\Pi(d^*)$ являются конусами, так как если точка с координатами $\pi_k, k \in K$, удовлетворяет этим неравенствам, то точка с координатами $\alpha \pi_k, \alpha > 0$, также им удовлетворяет.

Неравенства (1.5) линейны относительно переменных $\pi_k, k \in K$, и поэтому множество $\Pi(d^*)$ их решений выпукло. ■

Из доказанного свойства байесовских стратегий следуют интересные выводы. Известно, что выпуклые множества, которые не пересекаются, разделены гиперплоскостью. Это значит, что существуют такие вектор α и число θ , что $\langle \alpha, \pi \rangle < \theta$ для всех точек π из первого множества и $\langle \alpha, \pi \rangle \geq \theta$ для всех точек второго множества. Теорема 1.2 утверждает не только, что подмножества в пространстве вероятностей, соответствующие различным решениям, выпуклы. Более того, они являются конусами. Это значит, что эти подмножества не просто разделены гиперплоскостью, а разделены гиперплоскостью, проходящей через начало координат. О таких множествах говорят, что они *линейно разделимы*, или, что существует линейная дискриминантная функция, разделяющая эти множества. Это свойство очень популярно в распознавании, и одну из следующих лекций мы посвятим линейным дискриминантным функциям. Доказанная теорема объясняет и до некоторой степени обосновывает эту популярность, так как доказывает, что в пространстве вероятностей любая байесовская стратегия реализуется линейной дискриминантной функцией.

1.4 Два частных случая байесовских задач

1.4.1 Вероятность ошибочного решения о состоянии

Отождествим множество D решений с множеством состояний K . Это значит, что если $q(x) = k$, то принято решение, что объект находится в состоянии k . Естественно, решение $q(x)$ не всегда совпадает с действительным состоянием k^* . Поэтому требуется его принимать так, чтобы вероятность ошибочного решения, то есть вероятность события $q(x) \neq k^*$ была как можно меньше.

Это требование можно выразить в виде частного случая байесовской задачи. Действительно, представим, что следует платить единичный штраф в случае $q(x) \neq k^*$ и не платить штраф вообще в любой другой ситуации. Это значит, что $W(k^*, q(x)) = 1$, если $q(x) \neq k^*$ и $W(k^*, q(x)) = 0$, если $q(x) = k^*$. Математическое ожидание

$$R = \sum_{x \in X} \sum_{k^* \in K} p_{XK}(x, k^*) W(k^*, q(x)) \quad (1.6)$$

в этом случае есть вероятность события $q(x) \neq k^*$. Байесовская задача состоит, как и в общем случае, в построении стратегии $q: X \rightarrow K$, которая минимизирует математическое ожидание, определенное выражением (1.6), то есть

$$q(x) = \operatorname{argmin}_{k \in K} \sum_{k^* \in K} p_{XK}(x, k^*) W(k^*, k). \quad (1.7)$$

Выполним несложные эквивалентные преобразования выражения (1.7),

$$\begin{aligned} q(x) &= \operatorname{argmin}_{k \in K} \sum_{k^* \in K} p_{XK}(x, k^*) W(k^*, k) \\ &= \operatorname{argmin}_{k \in K} p_X(x) \sum_{k^* \in K} p_{K|X}(k^* | x) W(k^*, k) \\ &= \operatorname{argmin}_{k \in K} \sum_{k^* \in K} p_{K|X}(k^* | x) W(k^*, k) \\ &= \operatorname{argmin}_{k \in K} \sum_{k^* \in K \setminus \{k\}} p_{K|X}(k^* | x) \\ &= \operatorname{argmin}_{k \in K} \left(\sum_{k^* \in K} p_{K|X}(k^* | x) - p_{K|X}(k | x) \right) \\ &= \operatorname{argmin}_{k \in K} (1 - p_{K|X}(k | x)) = \operatorname{argmax}_{k \in K} p_{K|X}(k | x). \end{aligned}$$

Полученная стратегия состоит в том, что следует вычислить *апостериорную* вероятность каждого состояния k при условии наблюдения x и принять решение в пользу наиболее вероятного состояния.

Несмотря на ясность и естественность приведенной стратегии, довольно часто пытаются как-то ее улучшить. Например, ожидается, что принятие случайного решения о состоянии k в соответствии с апостериорными вероятностями $p_{K|X}(k | x)$ может оказаться в каком-то смысле лучше. Как

правило, от авторов таких рацпредложений трудно добиться их обоснований. Предложения такого рода не выводятся из каких-либо однозначно сформулированных требований, а основаны на менее четких поэтических рассуждениях. Вроде того, что сестрица Алenuшка не всегда должна искать своего заблудившегося Иванушку только в том баре, где он бывает чаще всего, но иногда и там, где он бывает лишь изредка. Нет нужды подробно анализировать такие предложения, а достаточно сослаться на теорему 1.1, которой они противоречат. И тем не менее уделим внимание этому предложению, хотя бы с целью иллюстрации теоремы 1.1.

Предположим, что по текущему наблюдению было установлено, что объект находится в первом состоянии с вероятностью 0.9, а во втором - с вероятностью 0.1. Если в такой ситуации принимать решение в пользу наиболее вероятного состояния, то ошибочное решение будет приниматься в 10% случаев, именно, во всех случаях, когда объект находился во втором состоянии. При рандомизированной стратегии в 90% случаев будет приниматься решение в пользу первого состояния, а в 10% случаев - в пользу второго. При такой стратегии вероятность ошибки будет равна 0.1 в 90% случаев, а именно тогда, когда принимается решение в пользу первого состояния. В 10% же случаев вероятность ошибки будет равна 0.9. Суммарная вероятность ошибки окажется равной 0.18, то есть почти вдвое большей, чем в детерминированной байесовской стратегии.

Следующее псевдорешение встречается, к сожалению, еще чаще, чем предыдущее. Предположим, что по чьему-то заказу был разработан прибор или программа для реализации стратегии $q: X \rightarrow D$, которая по наблюдению x решает, в каком из четырех возможных состояний находится объект. Состояние k может, таким образом, принимать значения 1, 2, 3 и 4. Предположим, что реализована была стратегия, оптимальная с точки зрения вероятности ошибочного решения. Представим теперь, что требования к программе смягчились, так как оказалось, что нет нужды в столь детальной информации о состоянии объекта. Достаточно указать лишь, находится ли объект в состоянии, меньшем, чем 3, или нет. Однако это, более грубое решение тоже должно быть принято с как можно меньшей вероятностью ошибки. Очевидно, что назначение прибора, строго говоря, изменилось. Раньше множество D состояло из четырех решений: $k = 1$, $k = 2$, $k = 3$ и $k = 4$, а новое множество D' состоит уже только из двух решений: $k \in \{1, 2\}$ и $k \in \{3, 4\}$. Таким образом, предыдущую стратегию $q: X \rightarrow D$ следует заменить новой стратегией $q': X \rightarrow D'$. Может показаться, что задача упростилась и (ВНИМАНИЕ! СЕЙЧАС БУДЕТ ОШИБКА!) новую стратегию q' можно легко реализовать на основании уже реализованной ранее стратегии q . А именно, принимать решение, что состояние меньше, чем 3, если $q(x) = 1$ или $q(x) = 2$, и не меньше, чем 3, если $q(x) = 3$ или $q(x) = 4$. Теорема 1.2 заставляет серьезно сомневаться в такой опрометчивой рекомендации.

При решении первой задачи пространство вероятностей разбивалось на четыре подмножества: $\Pi(1)$, $\Pi(2)$, $\Pi(3)$ и $\Pi(4)$. Легкомысленно пред-

ложенная стратегия q' разбивает пространство Π на два подмножества: $\Pi(1) \cup \Pi(2)$ и $\Pi(3) \cup \Pi(4)$. Согласно теореме 1.2 все эти шесть подмножеств должны быть выпуклыми. Однако при принятом построении стратегии q' вполне может оказаться, что она делит пространство вероятностей на подмножества, которые не являются выпуклыми. Ведь объединение выпуклых множеств вовсе необязательно выпукло, см. рис. 1.2. Если окажется, что стратегия q' не делит пространство вероятностей на выпуклые подмножества, то она не только не минимизирует вероятность ошибки. Она не решает никакую байесовскую задачу, то есть не является оптимальной ни с какой точки зрения.

Рассмотрим на численном примере, как может оказаться, что построенная стратегия q' не минимизирует вероятность ошибочного решения. Предположим, что для некоторого наблюдения x апостериорные вероятности состояний 1, 2, 3 и 4 оказались равными 0.3; 0.3; 0.4 и 0.0 соответственно. Стратегия q решает, что объект находится в состоянии $k = 3$, и это наилучшее решение с точки зрения вероятности ошибки. На основании этого решения стратегия q' решает, что состояние объекта не меньше, чем 3. Это не лучшее решение. Вероятность ошибки в этом случае равна 0.6. При противоположном решении вероятность ошибки была бы 0.4, то есть меньше.

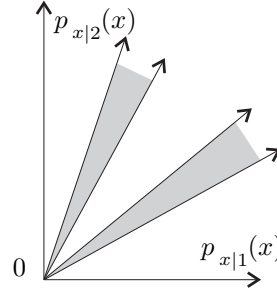


Figure 1.2 The union of two convex cones does not necessarily have to be a convex cone.

1.4.2 Байесовская стратегия отказа от распознавания

Обозначим $R(x, d)$ условное математическое ожидание штрафа, полученного при условии наблюдения x и решения d

$$R(x, d) = \sum_{k \in K} p_{K|X}(k|x) W(k, d). \quad (1.8)$$

Число $R(x, d)$ будем называть *частным риском*. Естественно, что для каждого наблюдения x решение должно быть выбрано так, чтобы частный риск был минимален. Однако может так случиться, что этот минимальный риск слишком большой. Было бы удобно, если бы именно ради таких ситуаций множество решений включало еще специальное решение, соответствующее ответу **not known**. Решение **not known** должно было бы приниматься в случаях, когда наблюдение x не содержит достаточную информацию для принятия другого решения с малым риском. Выразим приведенные соображения более точно, как частный случай байесовской задачи.

Пусть X и K - множества наблюдений и состояний, $p_{XK}: X \times K \rightarrow \mathbb{R}$ - распределение вероятностей, а $D = K \cup \{\text{not known}\}$ - множество решений.

Определим штрафы $W(k, d)$, $k \in K$, $d \in D$ следующим образом:

$$W(k, d) = \begin{cases} 0, & \text{если } d = k, \\ 1, & \text{если } d \neq k \text{ и } d \neq \text{not known}, \\ \varepsilon, & \text{если } d = \text{not known}. \end{cases} \quad (1.9)$$

Найдем байесовскую стратегию $q: X \rightarrow D$ для этой задачи. Для наблюдения x решение $q(x)$ должно минимизировать частный риск, то есть,

$$q(x) = \operatorname{argmin}_{d \in D} \sum_{k^* \in K} p_{K|X}(k^* | x) W(k^*, d), \quad (1.10)$$

или в эквивалентной записи

$$q(x) = \begin{cases} \operatorname{argmin}_{d \in K} R(x, d), & \text{если } \min_{d \in K} R(x, d) < R(x, \text{not known}), \\ \text{not known}, & \text{если } \min_{d \in K} R(x, d) \geq R(x, \text{not known}). \end{cases} \quad (1.11)$$

Минимальный частный риск решения $d \in K$ равен

$$\begin{aligned} \min_{d \in K} R(x, d) &= \min_{d \in K} \sum_{k^* \in K} p_{K|X}(k^* | x) W(k^*, d) \\ &= \min_{k \in K} \sum_{k^* \in K \setminus \{k\}} p_{K|X}(k^* | x) \\ &= \min_{k \in K} \left(\sum_{k^* \in K} p_{K|X}(k^* | x) - p_{K|X}(k | x) \right) \\ &= \min_{k \in K} (1 - p_{K|X}(k | x)) = 1 - \max_{k \in K} p_{K|X}(k | x). \end{aligned}$$

Частный риск решения **not known** равен

$$\begin{aligned} R(x, \text{not known}) &= \sum_{k^* \in K} p_{K|X}(k^* | x) W(k^*, \text{not known}) \\ &= \sum_{k^* \in K} p_{K|X}(k^* | x) \varepsilon = \varepsilon. \end{aligned} \quad (1.12)$$

Стратегия (1.11) приобретает таким образом вид

$$q(x) = \begin{cases} \operatorname{argmax}_{k \in K} p_{K|X}(k | x), & \text{если } 1 - \max_{k \in K} p_{K|X}(k | x) < \varepsilon, \\ \text{not known}, & \text{если } 1 - \max_{k \in K} p_{K|X}(k | x) \geq \varepsilon. \end{cases} \quad (1.13)$$

Это значит, что с самого начала нужно найти состояние с наибольшей апостериорной вероятностью. Если вероятность этого наиболее вероятного состояния больше, чем $1 - \varepsilon$, принимается решение в пользу этого состояния. В противном случае принимается решение **not known**. Такая стратегия представляется естественной и с неформальной точки зрения. Если

$\varepsilon = 0$, наилучшая стратегия состоит в том, чтобы никогда ничего не решать. Ведь штраф за ответ `not known` равен нулю. Если в этой ситуации будет принято определенное решение о состоянии, не исключена ошибка и, следовательно, и ненулевой штраф. И наоборот, если $\varepsilon > 1$, частный риск решения `not known` будет больше, чем частный риск любого другого решения. В этом случае ответ `not known` не будет дан никогда.

Эта правильная стратегия вполне естественная и прозрачная. Тем не менее всевозможные псевдорешения бытуют и здесь. Например, следующее: решение `not known` должно приниматься в том случае, когда наблюдается признак, маловероятный при любом состоянии объекта, то есть когда $p_{X|K}(x|k) < \theta$ для любого $k \in K$. Число θ - это порог, указывающий, какую вероятность наблюдения следует считать малой. Не стоит задумываться над тем, как выбирать этот порог. Более существенно, что такая стратегия не является решением нашей задачи ни при каком значении порога. Более того, она не решает никакую байесовскую задачу, так как противоречит теореме 1.2. Хотя множество $|K|$ -мерных векторов в пространстве вероятностей, которые соответствуют ответу `not known`, является выпуклым, но оно не является конусом. Эта ситуация иллюстрируется рис. 1.3. Серой областью представлено множество наблюдений с малой вероятностью. Эта область выпуклая, но она ограничена и поэтому не является конусом.

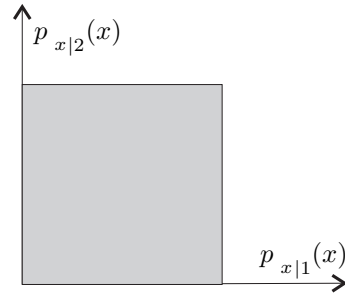


Figure 1.3 Decision `not known` for observations with small probability is not a cone and thus it does not correspond to any Bayesian strategy.

1.5 Обсуждение

В основном лекция оказалась такой, как я ожидал, и при этом сбьлись и мои пессимистические ожидания. Я уже не впервые знакомлюсь с байесовской теорией распознавания, и всякий раз мне представляется, что эта теория излагается в чересчур общем виде. На мой взгляд, только два частных случая задач, рассмотренные в конце лекции, имеют какое-то отношение к практике. Что касается всей предшествующей части, то я не вижу ее практическую применимость. Имело бы смысл ее рассказывать, если бы последние два частных случая невозможно было бы изложить иначе, как в рамках приведенной общей теории. Здесь же ситуация иная. Задачу о минимизации вероятности ошибки и об отказе от распознавания можно изложить, не прибегая к понятию штрафов и риска. Поэтому думаю, что вся предшествующая часть - это не более, чем интеллектуальное упражнение, общность ради общности. Могу ли я понимать материал лекции именно

так?

Мы согласны с Тобой, что в очень многих практических задачах должна минимизироваться именно вероятность ошибочного распознавания. Но не забывай, пожалуйста, что практические задачи неисчерпаемо разнообразны. Их невозможно втиснуть в формальную конструкцию, основанную на оптимизации какого-то одного всеобщего критерия, применимого во всех случаях жизни, даже такого общепризнанного, как вероятность ошибки. Уже в следующей лекции 2 мы расскажем о задачах, которые не укладываются в рамки байесовской конструкции во всей ее общности, и увидим, что байесовский подход в целом недостаточно богат. Таким образом, не может быть и речи о том, что его общность чересчур большая.

Ты же надеешься, что все практические задачи сможешь втиснуть в еще более узкие рамки, и ограничиться знанием о минимизации вероятности ошибки. Рано или поздно, думаем, что очень скоро, Ты выйдешь на практическую задачу, которая не вписывается в эти узкие рамки. Худшее, что Ты в этой ситуации сделаешь, это насильно втиснешь ее в те узкие рамки, которые Тебе известны. При этом Тебе придется деформировать исходную прикладную задачу до неузнаваемости. В конечном итоге Твой программный продукт будет решать совсем не ту задачу, что нужно. В описанной ситуации, конечно же, лучше было бы не втискивать задачу в рамки известного, а расширить эти рамки так, чтобы они охватывали и возникшую задачу без ее деформации. Так почему бы не расширить эти рамки уже сейчас, загодя?

Ты уверен, что все практические задачи распознавания сводятся к минимизации вероятности ошибки, то есть к нахождению апостериори наиболее вероятного значения скрытого параметра. Эта уверенность основана на неявном предположении, что все ошибки в каком-то смысле равноценны. Однако ситуация часто бывает иной. Ошибка может быть большей или меньшей, опасной или несущественной, некоторые ошибки можно не принимать во внимание вообще. Общий байесовский подход есть не просто интеллектуальное упражнение, а реалистический учет именно этих различий, выраженный функцией потерь. Игнорирование этих различий ведет к тому, что распознающая система будет стремиться достичь и те цели, которые в данном приложении не являются существенными.

Так что же плохого в том, что распознающая система будет выполнять нечто большее, чем нужно в данном конкретном приложении?

Стараясь избегать тех ошибок, которые в данном приложении не являются существенными, распознающая система неизбежно чаще будет допускать те ошибки, которые в данном приложении недопустимы. Мы видим, что пора рассмотреть конкретный поучительный пример такого рода.

Пусть X есть множество изображений, каждое из которых представляет цифру от 0 до 9. Номер цифры есть скрытый параметр изображения. Множество значений скрытого параметра, таким образом, есть $K = \{0,$

$1, 2, \dots, 9\}$. Предположим, что априорные вероятности на множестве K известны. Например, $p_K(k) = 0.1$ для всех $k \in K$. Кроме того, предположим, что и условные вероятности $p_{X|K}(x|k)$ известны, хотя они могут быть достаточно сложно устроены. Представим себе, например, что мы просто приобрели программу, которая при подаче на ее вход изображения $x \in X$ выдает 10 вероятностей $p_{X|K}(x|k)$. Кстати отметим, что эта программа реализует именно отображение множества изображений в пространство вероятностей, о котором мы говорили на лекции. Таким образом, имеется в распоряжении все, что нужно для решения популярной байесовской задачи на минимум вероятности ошибки. Для этого нужно указать цифру k с наибольшей апостериорной вероятностью, которая должна вычисляться по формуле Байеса

$$p_{K|X}(k|x) = \frac{p_K(k) p_{X|K}(x|k)}{\sum_{k=0}^9 p_K(k) p_{X|K}(x|k)} .$$

Проще говоря, решение может приниматься в пользу цифры k с наибольшим правдоподобием $p_{X|K}(x|k)$, так как априорные вероятности $p_K(k)$ одинаковы для всех цифр k .

Предположим теперь, что нам нужно распознать не одно-единственное изображение, а 20 взаимно независимых изображений. При этом целью распознавания является не определение 20 цифр k_1, k_2, \dots, k_{20} , представленных на изображениях x_1, x_2, \dots, x_{20} , а нечто значительно меньшее. Требуется лишь определить, чему равна их сумма.

Парень вроде Тебя, который во всей байесовской теории видит смысл лишь в формуле $k^* = \operatorname{argmax}_k p_{K|X}(k|x)$, действует решительно и немедленно. Для каждого $i = 1, 2, \dots, 20$ он вычисляет

$$k_i^* = \operatorname{argmax}_k p_{K|X}(k|x_i)$$

и решает, что искомая сумма равна $S = \sum_{i=1}^{20} k_i^*$.

Да я поступаю именно так и не вижу в этом ничего плохого.

Плохо то, что невозможно получить вразумительный ответ, почему нужно поступать именно так. Обычно скороговоркой, как нечто само собой разумеющееся, отвечают, что "из байесовской теории ведь следует, что такой алгоритм дает наилучшие результаты". Ты уже так не ответишь, потому что знаешь, что подобная ерунда никак не следует из байесовской теории. Напиши, пожалуйста, алгоритм, который из нее действительно следует.

Я только утвердился в своем первоначальном мнении, как нужно поступать в данной задаче. В то же время я понимаю, что от меня ожидается, и написал то, что следует из лекции.

Множество D решений есть, очевидно, $\{0, 1, \dots, 180\}$, так как искомая сумма может принимать лишь одно из указанных значений. Множество \tilde{X} наблюдений есть

$$\underbrace{X \times X \times \dots \times X}_{20 \text{ раз}},$$

то есть множество последовательностей, состоящих из 20 изображений. Множество \tilde{K} состояний есть

$$\underbrace{K \times K \times \dots \times K}_{20 \text{ раз}},$$

то есть множество последовательностей, состоящих из 20 цифр, каждая из которых может быть 0, 1, ..., 9. Обозначим \tilde{x} последовательность, которая предъявляется для распознавания. Обозначим \tilde{k} последовательность k_1, k_2, \dots, k_{20} цифр, которые действительно представлены на изображениях, подлежащих распознаванию. Вероятности $p_{\tilde{X}\tilde{K}}: \tilde{X} \times \tilde{K} \rightarrow \mathbb{R}$, очевидно, равны

$$p_{\tilde{X}\tilde{K}}(\tilde{x}, \tilde{k}) = \prod_{i=1}^{20} p_K(k_i) p_{X|K}(x_i | k_i),$$

потому что изображения в представленной последовательности взаимно независимы. Штрафная функция $W: \tilde{K} \times D \rightarrow \mathbb{R}$ принимает значения нуль или единица. Если $\sum_{i=1}^{20} k_i = d$, то $W(\tilde{k}, d) = 0$. Если $\sum_{i=1}^{20} k_i \neq d$, то $W(\tilde{k}, d) = 1$. Риск R стратегии $q: \tilde{X} \rightarrow D$ есть

$$R(q) = \sum_{\tilde{x} \in \tilde{X}} \sum_{\tilde{k} \in \tilde{K}} p_{\tilde{X}\tilde{K}}(\tilde{x}, \tilde{k}) W(\tilde{k}, d). \quad (1.14)$$

Совокупность этих данных определяет байесовскую задачу, которой соответствует стратегия

$$\begin{aligned} q(\tilde{x}) &= \operatorname{argmin}_{d \in D} \sum_{\tilde{k} \in \tilde{K}} p_{\tilde{X}\tilde{K}}(\tilde{x}, \tilde{k}) W(\tilde{k}, d) \\ &= \operatorname{argmin}_{d \in D} p_{\tilde{X}}(\tilde{x}) \sum_{\tilde{k} \in \tilde{K}} p_{\tilde{K}|\tilde{X}}(\tilde{k} | \tilde{x}) W(\tilde{k}, d) \\ &= \operatorname{argmin}_{d \in D} \sum_{\tilde{k} \notin \tilde{K}(d)} p_{\tilde{K}|\tilde{X}}(\tilde{k} | \tilde{x}) \\ &= \operatorname{argmin}_{d \in D} \left(1 - \sum_{\tilde{k} \in \tilde{K}(d)} p_{\tilde{K}|\tilde{X}}(\tilde{k} | \tilde{x}) \right) \\ &= \operatorname{argmax}_{d \in D} \sum_{\tilde{k} \in \tilde{K}(d)} p_{\tilde{K}|\tilde{X}}(\tilde{k} | \tilde{x}). \end{aligned}$$

В последних трех звеньях приведенной цепочки $\tilde{K}(d)$ обозначает множество последовательностей k_1, k_2, \dots, k_{20} , для которых $\sum_{i=1}^{20} k_i = d$. Далее полученную стратегию можно конкретизировать с учетом независимости

изображений, предъявленных для распознавания,

$$q(\tilde{x}) = \operatorname{argmax}_{d \in D} \underbrace{\sum_{k_1 \in K} \sum_{k_2 \in K} \cdots \sum_{k_{19} \in K} \sum_{k_{20} \in K}}_{k_1 + k_2 + \dots + k_{19} + k_{20} = d} \prod_{i=1}^{20} p_{K|X}(k_i | x_i), \quad (1.15)$$

где обозначение

$$\underbrace{\sum_{k_1 \in K} \sum_{k_2 \in K} \cdots \sum_{k_{19} \in K} \sum_{k_{20} \in K}}_{k_1 + k_2 + \dots + k_{19} + k_{20} = d}$$

понимается как суммирование по всем последовательностям $(k_1, k_2, \dots, k_{20})$, для которых $\sum_{i=1}^{20} k_i = d$.

Я уверен, что добросовестно выполнил все рекомендации, приведенные в лекции, и получил байесовскую стратегию решения задачи распознавания. Но теперь спрашивается, на что она пригодна. Она не может быть практически реализована. Проблема здесь вовсе не в выборе максимального числа из 181 чисел (181 - это количество возможных значений суммы $\sum_i k_i$). Что представляет собой фантастическую трудность, так это вычисление тех 181 чисел, из которых должно быть выбрано наибольшее. Для этого надо вычислить суммы слишком большого количества слагаемых. Грубо говоря, количество слагаемых равно количеству всех возможных последовательностей \tilde{k} , то есть 10^{20} слагаемых.

Теперь я лишь утвердился, что тот решительный парень, которого вы давеча насмешливо представили, поступает вполне разумно. Он пренебрегает выводами теории не потому, что их не знает, а потому, что он их уже очень хорошо знает. Он эти выводы просмотрел дальше, чем чистый теоретик, и увидел те трудности, которые не были заметны при абстрактном рассмотрении. Он предпочитает практически реализуемые рекомендации теоретически безукоризненным.

Ситуация с невообразимо большим количеством слагаемых не так страшна, если знать довольно распространенный прием. Мы будем и в дальнейшем его неоднократно использовать, и поэтому уделим ему сейчас некоторое внимание.

Обозначим те страшные 181 чисел, которые нужно вычислить, как $F(d)$. Тогда

$$F(d) = \underbrace{\sum_{k_1 \in K} \sum_{k_2 \in K} \cdots \sum_{k_{19} \in K} \sum_{k_{20} \in K}}_{k_1 + k_2 + \dots + k_{19} + k_{20} = d} \prod_{i=1}^{20} p_{K|X}(k_i | x_i). \quad (1.16)$$

Определим 20 вспомогательных функций, имеющих подобный вид,

$$F_1(d) = \underbrace{\sum_{k_1 \in K}}_{k_1 = d} \prod_{i=1}^1 p_{K|X}(k_i | x_i),$$

$$\begin{aligned}
F_2(d) &= \sum_{\substack{k_1 \in K \\ k_2 \in K \\ k_1+k_2=d}} \prod_{i=1}^2 p_{K|X}(k_i | x_i), \\
&\vdots \\
F_j(d) &= \sum_{\substack{k_1 \in K \\ k_2 \in K \\ \dots \\ k_j \in K \\ k_1+k_2+\dots+k_j=d}} \prod_{i=1}^j p_{K|X}(k_i | x_i), \\
F_{j+1}(d) &= \sum_{\substack{k_1 \in K \\ k_2 \in K \\ \dots \\ k_j \in K \\ k_{j+1} \in K \\ k_1+k_2+\dots+k_j+k_{j+1}=d}} \prod_{i=1}^{j+1} p_{K|X}(k_i | x_i), \\
&\vdots \\
F_{20}(d) &= \sum_{\substack{k_1 \in K \\ k_2 \in K \\ \dots \\ k_{19} \in K \\ k_{20} \in K \\ k_1+k_2+\dots+k_{19}+k_{20}=d}} \prod_{i=1}^{20} p_{K|X}(k_i | x_i).
\end{aligned}$$

Очевидно, что функция F_{20} совпадает с функцией F . Ясно также, что значения $F_1(d)$ легко вычислить. Фактически, их и вычислять не надо, так как они входят в исходные данные. Действительно, произведение $\prod_{i=1}^1 p_{K|X}(k_i | x_i)$ состоит из единственного сомножителя $p_{K|X}(k_1 | x_1)$, а сумма

$$F_1(d) = \sum_{\substack{k_1 \in K \\ k_1=d}} p_{K|X}(k_1 | x_1)$$

состоит из единственного слагаемого. Таким образом, $F_1(d) = p_{K|X}(d | x_1)$. Покажем, как следует вычислять числа $F_{j+1}(d)$, если числа $F_j(d)$ уже вычислены:

$$\begin{aligned}
F_{j+1}(d) &= \sum_{\substack{k_1 \in K \\ k_2 \in K \\ \dots \\ k_j \in K \\ k_{j+1} \in K \\ k_1+k_2+\dots+k_j+k_{j+1}=d}} \prod_{i=1}^{j+1} p_{K|X}(k_i | x_i) \\
&= \sum_{k_{j+1} \in K} p_{K|X}(k_{j+1} | x_{j+1}) \sum_{\substack{k_1 \in K \\ k_2 \in K \\ \dots \\ k_j \in K \\ k_1+k_2+\dots+k_j+k_{j+1}=d}} \prod_{i=1}^j p_{K|X}(k_i | x_i) \\
&= \sum_{k_{j+1} \in K} p_{K|X}(k_{j+1} | x_{j+1}) \sum_{\substack{k_1 \in K \\ k_2 \in K \\ \dots \\ k_j \in K \\ k_1+k_2+\dots+k_j=d-k_{j+1}}} \prod_{i=1}^j p_{K|X}(k_i | x_i)
\end{aligned}$$

$$= \sum_{k_{j+1} \in K} p_{K|X}(k_{j+1} | x_{j+1}) F_j(d - k_{j+1}). \quad (1.17)$$

Если исключить промежуточные звенья в приведенной цепочке, то получим

$$F_{j+1}(d) = \sum_{k_{j+1} \in K} p_{K|X}(k_{j+1} | x_{j+1}) F_j(d - k_{j+1}). \quad (1.18)$$

Вычисление одного числа $F_j(d)$ для одного значения j и одного значения d требует 10 умножений и 9 сложений. Для фиксированного значения j нужно вычислить не более, чем 181 таких чисел, соответствующих различным значениям d . Для получения функции F_{20} следует 19 раз преобразовать функцию F_j в функцию F_{j+1} . Таким образом, объем требуемых вычислений равен вовсе не 10^{20} , а не более, чем всего лишь $10 \times 181 \times 19$ умножений и приблизительно столько же сложений. Не стоит даже говорить о таких малых объемах.

Ловко! Догадываюсь, что преобразование (1.17) не просто ловкий трюк, а пример более общего метода оптимизации вычислений. Постараюсь его запомнить.

И тем не менее позвольте мне вернуться к своему настойчивому утверждению, что только поиск наиболее вероятного значения скрытого параметра имеет смысл. Ведь предыдущий, правда, очень интересный пример, никак это утверждение не опровергает. Если внимательно посмотреть на эти злополучные 181 чисел, то становится ясным, что числа $F(d)$ это ничто иное, как апостериорные вероятности того, что сумма случайных чисел k_1, k_2, \dots, k_{20} равна d . Кстати, рекуррентное выражение (1.18) не что иное, как известная формула для вычисления распределения вероятностей суммы двух случайных величин, которую мы изучали еще в школе. Одной из этих двух случайных величин есть число k_{j+1} , а другая имеет распределение вероятностей F_j . Я вижу, что мы не вышли за пределы поиска наиболее вероятного значения оцениваемого параметра. Мы избежали этого в одном месте, но зато пришли к этому же в другом.

К выводу, что решение следует принимать в пользу наиболее вероятного значения суммы мы не пришли вместе. Ты к этому пришел сам, когда без длительных размышлений определил штрафы $W(\tilde{k}, d)$, полагая, повидимому, что это наиболее естественно. Мы ничего не имеем против выбранной Тобой функции потерь. Мы лишь обращаем Твое внимание, что есть еще много других функций. В рассматриваемом примере выбранная Тобой функция потерь, пожалуй, не самая естественная. Согласись, что не очень естественно платить один и тот же штраф независимо от того, выдан ли ответ 99 или 25 вместо правильного ответа 100. Штраф во втором случае должен бы быть больше.

Разве может стратегия распознавания существенно зависеть от таких тонкостей?

Да! Просмотри хотя бы следующие три функции потерь, которые вполне уместны в нашем примере, и убедись, что им соответствуют различные стратегии.

Обозначим $d^*(k^*)$ действительное значение суммы, то есть число $\sum_{i=1}^{20} k_i^*$. Первая функция потерь может быть следующей: $W(k^*, d)$ равно нулю, если $|d^*(k^*) - d|$ не превышает допустимое отклонение Δ , и штраф равен единице, если разность $|d^*(k^*) - d|$ превышает допустимое отклонение Δ . Минимизация риска в этом случае обозначает минимизацию вероятности, что разность оценки d и действительного значения $d^*(k^*)$ превзойдет величину Δ .

Пусть вторая функция потерь будет разность $|d^*(k^*) - d|$, а третья - квадратичная разность $(d^*(k^*) - d)^2$.

Со второй функцией потерь пришлось немного повозиться, но думаю, что в конечном итоге я со всеми задачами справился. Общей частью всех трех алгоритмов является вычисление апостериорных вероятностей $F(d)$ каждого значения суммы. После их вычисления решение о значении суммы принимается действительно разными стратегиями. Самой простой оказалась задача с квадратичной функцией потерь. Решение d должно минимизировать частный риск

$$R(d) = \sum_{k^* \in \tilde{K}} p_{\tilde{K}|\tilde{X}}(k^* | \tilde{x}) \left(\sum_{i=1}^{20} k_i^* - d \right)^2 = \sum_{k^* \in \tilde{K}} p_{\tilde{K}|\tilde{X}}(k^* | \tilde{x}) (d^*(k^*) - d)^2.$$

Это значит, что искомое решение d находится из условия, что производная функции $R(d)$ равна нулю, то есть

$$\begin{aligned} 0 &= -2 \sum_{k^* \in \tilde{K}} p_{\tilde{K}|\tilde{X}}(k^* | \tilde{x}) (d^*(k^*) - d) \\ &= 2d - 2 \sum_{k^* \in \tilde{K}} p_{\tilde{K}|\tilde{X}}(k^* | \tilde{x}) d^*(k^*) \\ &= 2d - 2 \sum_{d^* \in D} \sum_{k^* \in \tilde{K}(d^*)} p_{\tilde{K}|\tilde{X}}(k^* | \tilde{x}) d^* \\ &= 2d - 2 \sum_{d^* \in D} d^* \sum_{k^* \in \tilde{K}(d^*)} p_{\tilde{K}|\tilde{X}}(k^* | \tilde{x}) \\ &= 2d - 2 \sum_{d^* \in D} d^* F(d^*). \end{aligned}$$

Из последнего выражения в цепочке следует, что $d = \sum_{d^* \in D} d^* F(d^*)$, как и следовало ожидать. Решение должно приниматься в пользу апостериорного математического ожидания оцениваемой суммы.

Во первой функции потерь штраф $W(k^*, d)$ равен единице или нулю в зависимости от того, превышает ли ошибка $|d^*(k^*) - d|$ порог Δ или нет. Обозначим $g(d^*, d)$ число, равное 0, если $|d^* - d| \leq \Delta$, и равное 1, если

$|d^* - d| > \Delta$. Решение о значении числа d должно быть

$$\begin{aligned}
& \operatorname{argmin}_{d \in D} \sum_{k^* \in \tilde{K}} p_{\tilde{K}|\tilde{X}}(k^*|\tilde{x}) W(k^*, d) \\
&= \operatorname{argmin}_{d \in D} \sum_{k^* \in \tilde{K}} p_{\tilde{K}|\tilde{X}}(k^*|\tilde{x}) g(d^*(k^*), d) \\
&= \operatorname{argmin}_{d \in D} \sum_{d^* \in D} \sum_{k^* \in \tilde{K}(d^*)} p_{\tilde{K}|\tilde{X}}(k^*|\tilde{x}) g(d^*, d) \\
&= \operatorname{argmin}_{d \in D} \sum_{d^* \in D} g(d^*, d) \sum_{k^* \in \tilde{K}(d^*)} p_{\tilde{K}|\tilde{X}}(k^*|\tilde{x}) = \operatorname{argmin}_{d \in D} \sum_{d^* \in D} g(d^*, d) F(d^*) \\
&= \operatorname{argmin}_{d \in D} \sum_{|d^* - d| > \Delta} F(d^*) = \operatorname{argmin}_{d \in D} \left(\sum_{d^* \in D} F(d^*) - \sum_{|d^* - d| \leq \Delta} F(d^*) \right) \\
&= \operatorname{argmin}_{d \in D} \left(1 - \sum_{|d^* - d| \leq \Delta} F(d^*) \right) = \operatorname{argmax}_{d \in D} \sum_{d^* = d - \Delta}^{d + \Delta} F(d^*).
\end{aligned}$$

Минимизация риска таким образом свелась к максимизации суммы

$$R'(d) = \sum_{d^* = d - \Delta}^{d + \Delta} F(d^*), \quad (1.19)$$

которая зависит от решения d . Эту сумму следует вычислить для каждого значения $d \in D$, то есть 181 раз. Решение должно приниматься в пользу значения d , для которого вычисленная сумма максимальна. Естественно, суммы $R'(d)$ должны вычисляться не по формуле (1.19), а по рекуррентной формуле

$$R'(d) = R'(d - 1) + F(d + \Delta) - F(d - \Delta - 1),$$

которая дает существенный выигрыш в вычислениях при больших значениях Δ .

Самой трудной для меня оказался случай с функцией потерь вида $W(\tilde{k}, d) = |d^*(\tilde{k}) - d|$. Я приступил к задаче, как вы говорите, решительно и скоро, и пришел к результату, который невозможно понять. Начинаю понемногу приучать себя к мысли, что работа в распознавании образов требует тщательности: легкомысленный шаг в сторону - и ты уже в трясине. Легкомысленное решение состоит в следующем. Для известных 181 чисел $F(d^*)$, $d^* = 0, \dots, 180$, должно быть найдено число d , которое минимизирует сумму

$$R(d) = \sum_{d^* \in D} F(d^*) |d^* - d|. \quad (1.20)$$

Я разбил множество $D = \{0, \dots, 180\}$ на три подмножества:

$$\begin{aligned} D^+ &= \{d^* \mid (d^* \in D) \wedge (d^* > d)\}, \\ D^- &= \{d^* \mid (d^* \in D) \wedge (d^* < d)\}, \\ D^= &= \{d^* \mid (d^* \in D) \wedge (d^* = d)\}. \end{aligned}$$

Риск $R(d)$ можно теперь записать в виде

$$R(d) = \sum_{d^* \in D^+} F(d^*)(d^* - d) - \sum_{d^* \in D^-} F(d^*)(d^* - d), \quad (1.21)$$

который (ВНИМАНИЕ! СЛЕДУЕТ ОШИБКА!) линейно зависит от d . Производная этого выражения равна

$$- \sum_{d^* \in D^+} F(d^*) + \sum_{d^* \in D^-} F(d^*).$$

Приравнивая эту производную нулю, получаю условие оптимальности значения d

$$\sum_{d^* \in D^+} F(d^*) = \sum_{d^* \in D^-} F(d^*), \quad (1.22)$$

которое утверждает, что минимальный риск достигается при таком значении d , при котором вероятность того, что случайная величина d^* станет меньше, чем d , равна вероятности того, что эта случайная величина будет больше, чем d . Такое значение называется медианой случайной величины.

Этот вывод трудно понять. Непонятно, что нужно делать, когда ни одно значение d не удовлетворяет условию (1.22). Это бы означало, что функция (1.21) не имеет минимума. Но исходя из здравого смысла это же бессмыслица. Величина d принимает конечное количество значений и одно из них должно быть не хуже всех остальных.

Ошибка была допущена при легкомысленном обращении с функцией (1.21), как с дифференцируемой. Эта функция зависит от d не только явно, но и через зависимость множеств D^+ и D^- от d . Очевидно, поиск минимума функций вида (1.21) требует более аккуратных рассуждений.

Функция -

$$R(d) = \sum_{d^* \in D} F(d^*) |d^* - d|$$

- это выпуклая функция. К тому же она линейна на любом интервале от d до $d + 1$, где d - целое число. Следовательно, ее минимум достигается при некотором целом значении d , для которого выполняется

$$R(d) - R(d + 1) \leq 0 \quad (1.23)$$

и

$$R(d) - R(d - 1) \leq 0. \quad (1.24)$$

Для $R(d)$ справедливы следующие два выражения

$$R(d) = \sum_{d^* \leq d} F(d^*)(d - d^*) + \sum_{d^* > d} F(d^*)(d^* - d), \quad (1.25)$$

$$R(d) = \sum_{d^* < d} F(d^*)(d - d^*) + \sum_{d^* \geq d} F(d^*)(d^* - d). \quad (1.26)$$

Для $R(d+1)$ справедлива цепочка

$$\begin{aligned} R(d+1) &= \sum_{d^* < d+1} F(d^*)(d+1 - d^*) + \sum_{d^* \geq d+1} F(d^*)(d^* - d - 1) \\ &= \sum_{d^* \leq d} F(d^*)(d+1 - d^*) + \sum_{d^* > d} F(d^*)(d^* - d - 1) \\ &= \sum_{d^* \leq d} F(d^*)(d - d^*) + \sum_{d^* > d} F(d^*)(d^* - d) + \sum_{d^* \leq d} F(d^*) - \sum_{d^* > d} F(d^*). \end{aligned}$$

Следовательно, условия (1.23) можно записать в виде

$$\sum_{d^* > d} F(d^*) - \sum_{d^* \leq d} F(d^*) \leq 0$$

и, учитывая, что

$$\sum_{d^* > d} F(d^*) + \sum_{d^* \leq d} F(d^*) = 1,$$

получаем

$$\sum_{d^* \leq d} F(d^*) \geq \frac{1}{2}. \quad (1.27)$$

Для $R(d-1)$ справедлива цепочка

$$\begin{aligned} R(d-1) &= \sum_{d^* \leq d-1} F(d^*)(d-1 - d^*) + \sum_{d^* > d-1} F(d^*)(d^* - d + 1) \\ &= \sum_{d^* < d} F(d^*)(d-1 - d^*) + \sum_{d^* \geq d} F(d^*)(d^* - d + 1) \\ &= \sum_{d^* < d} F(d^*)(d - d^*) + \sum_{d^* \geq d} F(d^*)(d^* - d) - \sum_{d^* < d} F(d^*) + \sum_{d^* \geq d} F(d^*). \end{aligned}$$

Следовательно, условие (1.24) можно записать в виде

$$\sum_{d^* < d} F(d^*) - \sum_{d^* \geq d} F(d^*) \leq 0$$

и, учитывая, что

$$\sum_{d^* < d} F(d^*) + \sum_{d^* \geq d} F(d^*) = 1,$$

получаем

$$\sum_{d^* \geq d} F(d^*) \geq \frac{1}{2}. \quad (1.28)$$

Именно условия (1.27) и (1.28) являются решением задачи, а не полученное ранее псевдорешение (1.22). Значение d^* , удовлетворяющее условиям (1.27) и (1.28), следует находить последовательным вычислением чисел $S(0) = F(0)$, $S(1) = S(0) + F(1)$, ..., $S(d^*) = S(d^* - 1) + F(d^*)$ и т.д. Первое число d^* в этой последовательности, для которого выполняется условие $S(d^*) \geq \frac{1}{2}$, есть $\arg \min_d \sum_{d^* \in D} F(d^*) |d - d^*|$.

Ты справился с упражнениями вполне удовлетворительно. Больше всего нас обрадовало, что при анализе задачи с квадратичной функцией потерь Ты мимоходом заметил, что, "как и следовало ожидать", решение должно приниматься в пользу апостериорного математического ожидания случайной величины. А ведь еще совсем недавно Ты и слышать ничего не хотел, кроме принятия решения в пользу значения с наибольшей апостериорной вероятностью. Тебя не смутил даже тот факт, что математическое ожидание может оказаться не целым числом. А ведь действительная сумма выражается только целым числом и поэтому только целые числа имеют ненулевую апостериорную вероятность. Теперь Ты сам убедился, что в некоторых задачах наилучшим может оказаться решение в пользу значения с нулевой вероятностью, а вовсе не с максимальной.

Я вижу, что все три задачи удалось решить только благодаря тому, что искомая сумма $\sum_{i=1}^{20} k_i$ принимает конечное, а главное, малое количество значений на множестве K . В силу этого оказалось возможным вычислить вероятности для каждого значения суммы, а затем эти вероятности обрабатывать.

Предположим, что нам придется иметь дело с несколько более сложной задачей, когда потребуется оценить не сумму 20 цифр, а 20-значное число, представленное этими цифрами. Это значит, что нам требуется оценить сумму

$$\sum_{i=1}^{20} a_i k_i = \sum_{i=1}^{20} 10^{i-1} k_i,$$

скажем, так, чтобы обеспечивался минимум математического ожидания квадратичских потерь. Усложнение задачи существенно связано с появлением коэффициентов a_i . Из-за них сумма $\sum_{i=1}^{20} a_i k_i$ может принимать на множестве \tilde{K} уже не 181 значение, а необозримо большое количество значений.

Рассмотрим эту задачу с самого начала, то есть с выражения (1.14) для риска. Коль скоро функцию F невозможно построить, постараемся обойтись без ее построения. Решение d^* о значении суммы $\sum_{j=1}^{20} a_j k_j$ должно минимизировать математическое ожидание квадратической ошибки

$(d - \sum_{j=1}^{20} a_j k_j)^2$. Таким образом,

$$\begin{aligned}
d^* &= \operatorname{argmin}_{d \in D} \sum_{\tilde{k} \in \tilde{K}} \prod_{i=1}^{20} p_{K|X}(k_i | x_i) \left(d - \sum_{j=1}^{20} a_j k_j \right)^2 \\
&= \sum_{\tilde{k} \in \tilde{K}} \prod_{i=1}^{20} p_{K|X}(k_i | x_i) \sum_{j=1}^{20} a_j k_j = \sum_{j=1}^{20} a_j \sum_{\tilde{k} \in \tilde{K}} k_j \prod_{i=1}^{20} p_{K|X}(k_i | x_i) \\
&= \sum_{j=1}^{20} a_j \sum_{k_1 \in K} \sum_{k_2 \in K} \dots \sum_{k_j \in K} \dots \sum_{k_{20} \in K} k_j \prod_{i=1}^{20} p_{K|X}(k_i | x_i) \\
&= \sum_{j=1}^{20} a_j \sum_{k_j \in K} p_{K|X}(k_j | x_j) k_j \sum_{k_1 \in K} \dots \sum_{k_{j-1} \in K} \sum_{k_{j+1} \in K} \dots \sum_{k_{20} \in K} \prod_{i=1, i \neq j}^{20} p_{K|X}(k_i | x_i) \\
&= \sum_{j=1}^{20} a_j \sum_{k_j \in K} k_j p_{K|X}(k_j | x_j) = \sum_{j=1}^{20} a_j \hat{k}_j.
\end{aligned}$$

Это доказывает общеизвестный факт, что математическое ожидание суммы случайных величин равно сумме математических ожиданий отдельных слагаемых. Мы видим, что алгоритм наилучшего в квадратическом смысле оценивания значения линейной функции $\sum_{i=1}^{20} a_i k_i$ в значительной своей части не зависит от коэффициентов a_i . Сначала вычисляются 20 апостериорных математических ожиданий \hat{k}_i , $i = 1, \dots, 20$. Это самая трудоемкая часть алгоритма, так как функции $p_{K|X}(k | x)$ могут быть довольно сложно устроены. При выполнении этой части алгоритма коэффициенты a_i не учитываются. Они используются только на завершающем этапе, когда решение о значении суммы $\sum_{i=1}^{20} a_i k_i$ вычисляется по формуле $d = \sum_{i=1}^{20} a_i \hat{k}_i$.

Оказывается, что квадратичная функция потерь существенно отличается от двух других рассмотренных нами функций. Только при этой функции потерь можно принимать решение о значении линейной функции от случайных переменных, толком не зная распределения вероятностей ее значений. Для двух других функций потерь это было бы невозможно.

Таким свойством обладают не только квадратичные функции потерь, а еще несколько других. Но мы согласны с Тобой, что в случае квадратических функций потерь отдельные задачи решаются особенно просто. Покажем, например, что стратегия решения некоторых задач распознавания вообще не зависит от функции потерь, если только известно, что эта функция квадратична.

Предположим, что целью распознавания является принятие решения не о сумме $\sum_{i=1}^{20} a_i k_i$, а о всей последовательности k_i , $i = 1, 2, \dots, 20$, при

функции потерь вида

$$W(\tilde{k}^*, \tilde{k}) = \left(\sum_{i=1}^{20} a_i k_i^* - \sum_{i=1}^{20} a_i k_i \right)^2. \quad (1.29)$$

Надеемся, для Тебя очевидно, что решение о последовательности $\tilde{k} = (k_1, k_2, \dots, k_{20})$ должно быть сформировано, как последовательность апостериорных математических ожиданий случайных величин k_i ,

$$\sum_{k_i \in K} p_{K|X}(k_i | x_i) k_i,$$

и это решение, как видно, не зависит от коэффициентов $a_i, i = 1, 2, \dots, 20$, вообще. Следовательно, математическое ожидание штрафа можно минимизировать даже не зная, о каком штрафе идет речь. Надо только знать, что он имеет вид (1.29). На основании этого понятного результата можно сделать следующий шаг. Поскольку стратегия, минимизирующая риск при штрафе вида (1.29) не зависит от коэффициентов $a_i, i = 1, 2, \dots, 20$, то она же будет оптимальной и для следующей функции потерь, определенной в виде суммы функций вида (1.29) так, что

$$W(\tilde{k}^*, \tilde{k}) = \sum_{j=1}^{20} \left(\sum_{i=1}^{20} a_{ij} (k_i^* - k_i) \right)^2. \quad (1.30)$$

При условии, что матрица коэффициентов a_{ij} неотрицательно определена, эта же стратегия будет оптимальной и для любой функции потерь вида

$$W(\tilde{k}^*, \tilde{k}) = \sum_{j=1}^{20} \sum_{i=1}^{20} a_{ij} (k_i^* - k_i) (k_j^* - k_j). \quad (1.31)$$

Действительно, любая неотрицательно определенная квадратичная функция вида (1.31) может быть представлена в виде (1.30).

Таким образом, если только известно, что функция потерь представлена в виде (1.31), то для построения оптимальной стратегии распознавания нет нужды полностью знать коэффициенты a_{ij} , так как оптимальная стратегия от этих коэффициентов не зависит. Достаточно знать только, что матрица этих коэффициентов неотрицательно определена.

Эти квадратичные функции потерь действительно прекрасны!

Ты далеко не единственный, кто так думает. Но не иди на поводу у этой красоты и не применяй их там, где они неуместны. А таких ситуаций очень много.

Мы закончили первое знакомство с байесовскими задачами распознавания. Тем временем мы подготовили уже следующую лекцию и высылаем ее Тебе.

Декабрь 1996.

1.6 Библиографические заметки

Байесовский подход назван в честь священника Томаса Байеса [Bayes, 1763], который в начале 18-го века предложил, как нужно оперировать с условными частотами (вероятностями). Литература по распознаванию довольно сильно насыщена статистическими соображениями. В статистической литературе задачи распознавания явно почти не просматриваются. Тем не менее, читателю, желающему увидеть, как основы байесовского подхода излагаются в математической статистике, можно порекомендовать публикацию [Wald, 1950], где сформулированы задачи в весьма общей форме. Близкими к распознаванию идеи изложены в учебнике [Anderson, 1958], в основной своей части посвященном многомерному статистическому анализу.

Большая часть статистических исследований в распознавании так или иначе связана с проблемами обучения. Исследованию статистических проблем распознавания вне их связи с проблемами обучения уделяется меньше внимания. В ряду первых работ в этом направлении следует указать работы В.А.Ковалевского [Kovalevski, 1965] и Чау [Chow, 1965].

В общеизвестном учебнике [Duda and Hart, 1973] содержится всестороннее рассмотрение статистического распознавания, и его мы рекомендуем любому читателю, работающему профессионально в распознавании образов. Недавно было опубликовано существенно переработанное новое издание этой книги [Duda et al., 2001].

Мы не будем касаться в этой книге проблемы выбора признаков и почти не будем рассматривать популярные методы поиска ближайших соседей. Эти проблемы детально описаны в [Devroye et al., 1996].

Различным аспектам распознавания посвящены также монографии [Devijver and Kittler, 1982], [Fukunaga, 1990], [Chen et al., 1993], [Nadler and Smith, 1993], [Pavel, 1993], [Young, 1994], [Bishop, 1996], [Theodoridis and Koutroumbas, 1999].

Глава 2

Небайесовские задачи распознавания

2.1 Жесткая ограниченность байесовского подхода

В первой лекции мы неоднократно подчеркивали большую общность байесовского подхода. Эта общность выражается в том, что байесовские задачи и основные свойства байесовских стратегий формулируются для множеств наблюдений X , состояний K и решений D с самыми разнообразными математическими свойствами. Очень желательно владеть всем тем богатством, которым обладает байесовский подход, и не отождествлять байесовский подход в целом с тем или иным его частным случаем. Мы уже знаем, что класс байесовских задач - это нечто большее, чем минимизация вероятности ошибочного решения.

При всей общности байесовского подхода существуют задачи, которые не могут быть выражены в его рамках. Это так называемые небайесовские задачи статистической теории решений. Небайесовские методы нужно знать, чтобы для каждого конкретного приложения выбирать наиболее уместную формализацию, а не наоборот. Исходную прикладную задачу не следует деформировать и неестественно втискивать ее в рамки заданного формализма, даже если этот формализм такой уважаемый, как байесовский подход.

Байесовский подход ограничивается сразу же после введения тех понятий, которые лежат в его основе. Во первых, это *функция потерь* $W: K \times D \rightarrow \mathbb{R}$. Во вторых, это *совместные вероятности* $p_{XK}(x, k)$, $x \in X$, $k \in K$, которыми вводятся два дополнительных понятия: *априорные вероятности* $p_K(k)$ ситуаций $k \in K$, которые должны быть распознаны, и *условные вероятности* $p_{X|K}(x|k)$, $x \in X$, $k \in K$ наблюдений x при условии ситуации k . Рассмотрим более тщательно, как введение этих понятий сужает круг прикладных задач, которые могут формализоваться в рамках байесовского подхода. Для начала, в подразделах 2.1.1–2.1.3, это рассмотрение будет неформальным.

2.1.1 Функция потерь

Как только произнесены слова "минимизация математического ожидания потерь", так сразу ограничивается характер множества значений, которые может принимать штраф. Это должно быть полностью упорядоченное множество, на котором, кроме отношений $<$ или \geq , должны быть определены сложение штрафов и умножение их на вещественное число. Для байесовского подхода существенно, что штрафы измеряются вещественными числами, и это ограничение не представляется обременительным при абстрактном рассмотрении формальных моделей. Однако существует ряд приложений, где эта неестественность бросается в глаза и сразу же видно, как отождествление штрафа с вещественными числами деформирует прикладное содержание задачи самым недопустимым образом. Эти прикладные задачи не должны решаться в рамках байесовского подхода. Они должны формулироваться без использования понятия штрафа.

Здесь можно вспомнить, правда, не слишком серьезно, известного героя русского фольклора: повернешь налево - коня потеряешь, повернешь направо - меч потеряешь, пойдешь прямо - потеряешь любимую. Знание байесовского метода здесь мало поможет нашему герою, так как никаким вразумительным способом ему не удастся определить, что больше: сумма p_1 лошадей и p_2 мечей или p_3 любимых. Здесь непонятно даже, что такое сумма p_1 коней и p_2 мечей, так же, как не имеет смысла сумма 30 метров и 80 секунд.

Здесь мы сталкиваемся с той нередкой ситуацией, когда функция потерь полностью задана, но она принимает значения, которые *не могут измеряться в одной системе единиц*. Вспомним, теперь уже чуть более серьезно, такую обширную область приложений, как диагностика сложного оборудования. На основании наблюдений следует определить, находится ли оборудование еще в исправном состоянии или в состоянии, которое уже опасно для окружающих. Каждая ошибочная диагностика связана с определенными потерями, которые известны. Но потери, когда исправное оборудование воспринимается, как опасное, совсем другого рода, чем потери в ситуации, когда проморгали опасную ситуацию. В первом случае потери составляют стоимость профилактических мероприятий, которые можно было бы и не проводить. Во втором случае потери могут быть совсем иного вида, при котором они не могут отождествляться ни с каким количеством профилактик. Например, может разрушиться что-то, что природа создавала миллионы лет. Несравнимость потерь от ошибок различного рода еще более выразительна в задачах, где объектом диагностики является человеческое общество или отдельно взятый человек, будь то медицинская диагностика или криминалистическая.

2.1.2 Априорные вероятности состояний

Чтобы сформулировать задачу распознавания в байесовских рамках, необходимо задать априорные вероятности $p_K(k)$ каждого состояния $k \in K$. Конечно же, может оказаться, что эти вероятности неизвестны, и эта труд-

ность лежит как-бы на поверхности. При более близком соприкосновении с прикладными задачами обнаруживаются и другие трудности, принципиально иного характера. Укажем три возможные степени возникающих ситуаций, от более легких к более трудным.

1. Состояние объекта случайно и априорные вероятности $p_K(k)$, $k \in K$, состояний известны. В этом случае задача может ставиться и решаться как байесовская.
2. Состояние объекта случайно, но априорные вероятности $p_K(k)$, $k \in K$, состояний неизвестны, так как объект распознавания еще не был достаточно хорошо исследован. Разработчик в этом случае имеет две возможности: (а) он может сформулировать задачу не в рамках байесовского подхода, а как-то иначе, так, чтобы априорные вероятности состояний не требовались в качестве исходных данных задачи; (б) он может более тщательно изучить объект распознавания, чтобы восполнить недостающие знания, необходимые для байесовского решения.
3. Состояние объекта не является случайным. В силу этого априорные вероятности $p_K(k)$, $k \in K$, не существуют и их невозможно получить никаким дополнительным исследованием объекта распознавания. В этом случае разработчик должен формулировать и решать свою задачу, как небайесовскую. Покажем пример такой ситуации, когда состояние объекта не случайно.

Пример 2.1 Задача, не входящая в байесовский класс. Пусть x - это сигнал, отраженный от наблюдаемого самолета. На основании этого сигнала нужно принять решение: свой ($k = 1$) или чужой ($k = 2$). Условные вероятности $p_{X|K}(x|k)$ могут очень сложно зависеть от наблюдения x . Тем не менее можно предположить, что по крайней мере существуют (хотя нам и неизвестны) такие числа $p_{X|K}(x|k)$, которые правильно отражают зависимость наблюдения x от состояния k . Что касается априорных вероятностей $p_K(k)$, то они неизвестны и не могут быть известны в принципе. Не существует такого числа α , $0 \leq \alpha \leq 1$, о котором можно было бы сказать, что это вероятность чужого самолета. Вероятности $p_K(k)$ просто не существуют, так как при сколь угодно длительном наблюдении частота появления чужого самолета не стремится к пределу, о котором можно было бы сказать, что это и есть вероятность. Иными словами, появление чужого самолета не является случайным событием. ▲

О неслучайном событии невозможно сказать, какова его вероятность, подобно тому, как невозможно сказать, какова температура звука или вкуса света. Такая характеристика, как вероятность, для неслучайного события просто не определена. Прикладные задачи, в которых требуется принять решение о неслучайном, но меняющемся состоянии объекта, не формализуются в рамках байесовского подхода. Они требуют иных теоретических конструкций, в которых понятие априорная вероятность не используется вообще.

Нас всегда удивляла та легкость, с которой незнание априорных вероятностей преодолевается, предполагая, что все эти вероятности равны между собой. В примере 2.1 это бы означало, что если нам неизвестна вероятность чужого самолета, то давайте считать, что чужой самолет так же вероятен, как и наш. Но это не согласуется со здравым смыслом, даже если принять, что появление чужого события есть случайное событие: хоть нам и неизвестна вероятность чужого самолета, но ясно, что она отличается от вероятности своего. До сих пор нам неизвестны веские доводы в пользу принятия гипотезы о равномерном распределении вероятностей в случае, когда это распределение неизвестно. Не исключено, что здесь имеет место одно из псевдорешений, когда логическое обоснование заменяется ссылкой на знаменитую личность. В данном случае, на К.Шеннона и общеизвестный факт, что энтропия достигает своего максимума именно при равномерном распределении вероятностей, несмотря на то, что к рассматриваемым вопросам этот факт не имеет никакого отношения.

2.1.3 Условные вероятности наблюдений

Рассмотрим следующую прикладную задачу. Пусть X - множество изображений, на которых представлены буквы. Имя буквы k есть скрытый параметр изображения, а K - множество имен букв. Предположим, что буквы пишутся тремя людьми. Обозначим эту тройку Z , и пусть переменная $z \in \{1, 2, 3\}$ определяет, кто написал ту или иную букву. В этой ситуации возникают две различные задачи: распознавание буквы независимо от того, кто ее написал, и распознавание, кто написал букву, вне зависимости от того, что это за буква. Мы будем говорить о первой задаче. В данной модели мы расширили круг основных понятий по сравнению с теми, которые мы ввели до сих пор. Мы сейчас уже говорим не о двух параметрах объекта: наблюдаемом признаке x и скрытом состоянии k , но еще и о третьем параметре z . Этот параметр скрыт от непосредственного наблюдения. Но в отличие от скрытого состояния k нас не интересует его значение. Целью распознавания является принятие решения о состоянии k , а не о значении параметра z . Несмотря на то, что параметр z не является ни признаком объекта, ни состоянием, которое следует распознать, его нельзя исключить из рассмотрения, потому что результат наблюдения x зависит не только от распознаваемого состояния k , но и от скрытого параметра z . Скрытый параметр z будем называть *мешающим* или вмешательством, чтобы отличить его от наблюдения и состояния.

В модели, пример которой мы сейчас рассмотрели, можно говорить о штрафах $W(k, d)$ и об априорных вероятностях $p_K(k)$ скрытых состояний, в данном случае, о вероятностях букв. Однако в данном приложении не определены условные вероятности $p_{X|K}(x | k)$. Причина в том, что вероятность того или иного изображения x зависит не только от буквы, которую это изображение представляет, но и от неслучайного вмешательства, то есть от того, кто эту букву написал. Здесь мы можем говорить только об условных вероятностях $p_{X|K,Z}(x | k, z)$, то есть о том, как может выгля-

деть та или иная буква, написанная тем или иным человеком. Если бы вмешательство было случайно и были бы известны вероятности $p_Z(z)$ для каждого значения z , можно было бы говорить и о вероятностях $p_{X|K}(x|k)$, потому что их можно было бы вычислить по формуле полной вероятности

$$p_{X|K}(x|k) = \sum_{z=1}^3 p_Z(z) p_{X|K,Z}(x|k, z).$$

Но условия применения алгоритма не дают основания для предположений о том, как часто тот или иной человек будет писать буквы, которые надо будет распознавать. Например, не исключается, что в течение всего периода эксплуатации алгоритма распознавания придется распознавать тексты, написанные одним-единственным человеком, но неизвестно каким. При таких статистически недоопределенных условиях требуется разработать стратегию, которая обеспечит требуемое качество распознавания букв вне зависимости от того, кто написал эту букву. Задача построения этой стратегии должна быть сформулирована так, чтобы понятие априорные вероятности $p_Z(z)$ вмешательства z не использовалось вообще, так как вмешательство не случайно и такое свойство как вероятность для него не определено.

Мы приведем формулировки наиболее известных небайесовских задач и их решения. Кроме того, мы укажем некоторые модификации известных задач. Мы покажем, что несмотря на большое разнообразие небайесовских задач они обладают общими чертами, которые их объединяют в один класс и позволяют решать их с помощью единого конструктивного механизма. Далее мы покажем, что класс небайесовских задач обладает общими чертами с классом байесовских задач. А именно, класс стратегий решения небайесовских задач почти полностью совпадает с классом байесовских стратегий. Любая небайесовская стратегия, как и байесовская, реализуется в виде разбиения пространства вероятностей на выпуклые конусы.

2.2 Формулировка известных и новых небайесовских задач

2.2.1 Задача Неймана-Пирсона

Пусть некоторый объект характеризуется признаком x , который принимает значения из множества X . Распределение вероятностей признака x зависит от состояния k , в котором объект находится. Этим состояний два : нормальное $k = 1$ и опасное $k = 2$. Множество K состояний, таким образом, есть $\{1, 2\}$. Известны условные вероятности $p_{X|K}(x|k)$, $x \in X$, $k \in K$.

Цель распознавания состоит в том, чтобы на основании наблюдения признака x решить, находится ли объект в нормальном состоянии или опасном. Стратегию принятия такого решения можно представить в виде разбиения множества X наблюдений на подмножества X_1 и X_2 . При наблюдении $x \in X_1$ принимается решение, что объект находится в нормальном состоянии, а при $x \in X_2$ - в опасном.

Поскольку одно и то же значение признака может иметь место как для объектов в нормальном состоянии, так и для объектов в опасном состоянии, безошибочная стратегия невозможна. Каждая стратегия будет характеризоваться двумя числами. Первое число обозначает условную вероятность принятия ошибочного решения при условии, что объект находится в нормальном состоянии. Это событие называется *ложная тревога*. Второе число обозначает условную вероятность ошибочного решения при условии, что объект находится в опасном состоянии. Это событие называется *пропуск цели*. Эти две ошибки известны также, как ошибки первого и второго рода. Условная вероятность ложной тревоги определяется суммой $\sum_{x \in X_2} p_{X|K}(x|1)$, а условная вероятность пропуска цели - суммой $\sum_{x \in X_1} p_{X|K}(x|2)$.

Задача Неймана-Пирсона [Neyman and Pearson, 1928; Neyman and Pearson, 1933] (мы будем дальше для краткости говорить задача Неймана) состоит в построении стратегии, то есть разбиения множества X на два подмножества $X_1 \subset X$ и $X_2 \subset X$, $X_1 \cap X_2 = \emptyset$, чтобы, во первых, условная вероятность пропуска цели не превышала заранее заданную величину ε ,

$$\sum_{x \in X_1} p_{X|K}(x|2) \leq \varepsilon, \quad (2.1)$$

и во вторых, при соблюдении указанного условия обеспечивалась минимальная вероятность ложной тревоги.

Это значит, что множества X_1 и X_2 должны обеспечивать минимальное значение суммы

$$\sum_{x \in X_2} p_{X|K}(x|1) \quad (2.2)$$

при условиях

$$\sum_{x \in X_1} p_{X|K}(x|2) \leq \varepsilon, \quad (2.3)$$

$$X_1 \cap X_2 = \emptyset, \quad X_1 \cup X_2 = X. \quad (2.4)$$

Основной результат Неймана-Пирсона состоит в том, что для множеств X_1 и X_2 , являющихся решением сформулированной задачи, существует порог θ , такой, что любое наблюдение $x \in X$, для которого *отношение правдоподобия*

$$\frac{p_{X|K}(x|1)}{p_{X|K}(x|2)}$$

меньше, чем θ , принадлежит X_2 ; и наоборот, любое наблюдение $x \in X$ с отношением правдоподобия, превышающим θ , принадлежит X_1 . Оставим без внимания вопрос о том, в какое множество входят наблюдения с отношением правдоподобия, в точности равным порогу. Этот вопрос очень интересен с теоретической точки зрения, но лежит вне тех вопросов, которые должны быть раскрыты в данной лекции. Учитывая, что точное равенство является достаточно редким событием, то из чисто прагматических соображений не стоит ему уделять слишком много внимания именно в данной лекции.

Доказательство основного результата, выполненное самим Нейманом, не является слишком простым и поэтому не пользуется широкой известностью. В силу этого знание этого основного результата является фактически не знанием, а верой, основанной на незыблемом авторитете Неймана. Этой веры достаточно при решении задач, в точности совпадающих с задачей Немана-Пирсона. Однако при столкновении с задачей, хоть чуть-чуть отличающейся от неймановской, одной веры уже недостаточно. Надо обладать техникой вывода этих результатов, чтобы для несколько модифицированной задачи пройти тот же путь, которым прошел Нейман в своей задаче. Рассмотрим пример такой незначительной модификации задачи Неймана-Пирсона.

Пусть количество состояний объекта равно трем, а не двум. Для каждого состояния $k \in \{1, 2, 3\}$ и для каждого наблюдения x заданы условные вероятности $p_{X|K}(x|k)$. Одно из состояний, состояние $k = 1$ является нормальным, а два других - опасные. Так же, как и в задаче Неймана-Пирсона, задача состоит в построении разумной стратегии, которая для каждого наблюдения решает, является ли состояние объекта нормальным или опасным.

На уровне фольклора здесь бытуют несколько рекомендаций по построению такой стратегии. Например, вычисляются два отношения правдоподобия,

$$\gamma_{12} = \frac{p_{X|K}(x|1)}{p_{X|K}(x|2)} \quad \text{и} \quad \gamma_{13} = \frac{p_{X|K}(x|1)}{p_{X|K}(x|3)}$$

и выбираются два пороговых значения θ_{12} и θ_{13} . Решение в пользу нормального состояния принимается, когда $\gamma_{12} > \theta_{12}$ и $\gamma_{13} > \theta_{13}$. Другие предложения представляют собой попытку изобрести какое-то обобщение отношения правдоподобия на случай отношения трех чисел, а не только двух, например, в виде

$$\frac{p_{X|K}(x|1)}{\max(p_{X|K}(x|2), p_{X|K}(x|3))}, \quad \text{или} \quad \frac{p_{X|K}(x|1)}{\sum_{k \in K} p_{X|K}(x|k)},$$

или подобные тому изобретения. Затем принимается решение в пользу нормального или опасного состояния на основе сравнения изобретенного "отношения правдоподобия" с порогом.

Предложения такого рода представляют собой попытки изобрести алгоритм решения некоторой интуитивно понимаемой задачи, минуя ее однозначную формулировку, или попытку натянуть конечный результат Неймана на такое приложение, к которому он, строго говоря, не относится. Поэтому такого рода предложения сами по себе неубедительны, а авторитетом Неймана уже не подкрепляются, так как он к этим рацпредложениям отношения не имеет.

При выходе на прикладную задачу, пусть даже мало отличающуюся от задачи Неймана-Пирсона, следует обобщать не алгоритм, обоснованный Нейманом, а задачу, решенную Нейманом. Для этой обобщенной задачи

следует пройти весь путь от ее формулировки до алгоритма ее решения. Тогда обобщение алгоритма Неймана получится как бы само собой. Но этот обобщенный алгоритм будет уже обладать убедительностью, так как будет известно, какую задачу он решает.

2.2.2 Обобщенная задача Неймана для двух опасных состояний

Обобщение задачи Неймана-Пирсона может иметь, например, такую формулировку. Множество наблюдений должно быть разбито на два подмножества X_1 и X_{23} , определяющие стратегию так, что:

- если $x \in X_1$, принимается решение в пользу нормального состояния;
- если $x \in X_{23}$, принимается решение в пользу нормального состояния.

Требуется построить стратегию, при которой как для объектов во втором состоянии, так и для объектов в третьем состоянии вероятность пропуска цели не превосходит наперед заданного уровня. При этом из всех стратегий, удовлетворяющих этому требованию, должна быть выбрана та, которая минимизирует вероятность ложной тревоги. Формально, искомые множества X_1 и X_{23} должны минимизировать сумму

$$\sum_{x \in X_{23}} p_{X|K}(x|1) \quad (2.5)$$

при условиях

$$\sum_{x \in X_1} p_{X|K}(x|2) \leq \varepsilon, \quad (2.6)$$

$$\sum_{x \in X_1} p_{X|K}(x|3) \leq \varepsilon, \quad (2.7)$$

$$X_1 \cap X_{23} = \emptyset, \quad (2.8)$$

$$X_1 \cup X_{23} = X. \quad (2.9)$$

Решение этой оптимизационной задачи мы рассмотрим позже, после того, как сформулируем ряд других небайесовских задач. Затем мы увидим, что существует конструктивная технология единообразного решения этого ряда.

2.2.3 Минимаксная задача

Пусть, как и раньше, X - это множество наблюдений, а K - множество состояний. Пусть заданы условные вероятности $p_{X|K}(x|k)$ для каждого наблюдения и каждого состояния. Пусть стратегия определяется разбиением множества наблюдений на подмножества $X(k)$, $k \in K$, так, что при наблюдении $x \in X$ принимается решение, что объект находится в состоянии k , если $x \in X(k)$. Каждая стратегия характеризуется совокупностью из $|K|$

чисел $\omega(k)$, $k \in K$, которые обозначают условную вероятность ошибочного решения при условии, что объект находится в состоянии k ,

$$\omega(k) = \sum_{x \notin X(k)} p_{X|K}(x|k).$$

В минимаксной задаче требуется найти такое разбиение множества наблюдений X на подмножества $X(k)$, $k \in K$, которое минимизирует число $\max_{k \in K} \omega(k)$.

Содержательный смысл этой задачи можно раскрыть на следующем примере. Пусть разработка алгоритма была заказана неким заказчиком, и заблаговременно было оговорено, что испытания алгоритма будут проходить в два этапа. На первом этапе заказчик испытывает разработанный алгоритм самостоятельно с целью определить состояние объекта, при котором происходит наибольшее количество ошибок, то есть найти наиболее трудно распознаваемое состояние. На втором этапе алгоритм по требованию заказчика испытывается только на объектах, находящихся в этом самом плохом состоянии. Результат именно этих испытаний будет записан в протокол, определяющий качество выполненной работы. Цель разработчика состоит в том, чтобы в заключительном протоколе испытаний был записан как можно лучший показатель.

Хорошо известно, что минимаксная задача в случае двух состояний, как и задача Неймана-Пирсона, решается сравнением отношения правдоподобия с порогом. Значительно менее известно доказательство этого факта, так как сложность его вызывает чувство безнадежности. Далее мы покажем простой вывод этого известного результата, основанного на теоремах двойственности линейного программирования.

2.2.4 Задача Вальда

Задача, которую мы здесь покажем, является лишь незначительной частью огромного поля, известного, как последовательный анализ Вальда [Wald, 1947; Wald and Wolfowitz, 1948].

При формулировке задачи Неймана-Пирсона бросается в глаза явная асимметрия требований относительно опасного и нормального состояний. Прежде всего требуется, чтобы вероятность пропуска опасного состояния была малой, и в этом смысле вероятность пропуска опасности является первостепенной характеристикой стратегии. Что касается вероятности ложной тревоги, то ее следует сделать как можно меньше, но не в ущерб первому требованию. В этом отношении вероятность ложной тревоги является как бы второстепенной характеристикой стратегии.

Было бы замечательно, если бы можно было построить такую стратегию, при которой обе эти вероятности не превосходили заданной величины ε . Однако эти требования могут оказаться противоречивыми. Это противоречие можно снять, если сформулировать задачу деления множества наблюдений X не на два подмножества X_1 и X_2 , соответствующих реше-

ниям в пользу первого или второго состояния, а на три подмножества X_0 , X_1 и X_2 имеющих следующий смысл:

- если $x \in X_1$, то принимается решение $k = 1$;
- если $x \in X_2$, то принимается решение $k = 2$; и, наконец,
- если $x \in X_0$, принимается решение, что текущее наблюдение не содержит достаточную информацию для надежного принятия решения о состоянии k .

Качество стратегии такого вида характеризуют четыре числа:

$\omega(1)$ - условная вероятность ошибочного решения при условии, что объект находится в состоянии $k = 1$,

$$\omega(1) = \sum_{x \in X_2} p_{X|K}(x|1);$$

$\omega(2)$ - условная вероятность ошибочного решения при условии, что объект находится в состоянии $k = 2$,

$$\omega(2) = \sum_{x \in X_1} p_{X|K}(x|2);$$

$\chi(1)$ - условная вероятность отказа от распознавания при условии, что объект находится в состоянии $k = 1$,

$$\chi(1) = \sum_{x \in X_0} p_{X|K}(x|1);$$

$\chi(2)$ - условная вероятность отказа от распознавания при условии, что объект находится в состоянии $k = 2$,

$$\chi(2) = \sum_{x \in X_0} p_{X|K}(x|2).$$

Для введенного класса стратегий требования $\omega(1) \leq \varepsilon$ и $\omega(2) \leq \varepsilon$ уже не являются противоречивыми при любом неотрицательном значении ε , так как стратегия $X_0 = X$, $X_1 = \emptyset$, $X_2 = \emptyset$, входит в рассматриваемый класс. Каждая стратегия, удовлетворяющая требованиям $\omega(1) \leq \varepsilon$ и $\omega(2) \leq \varepsilon$, характеризуется теперь еще тем, как часто она воздерживается от принятия решения, то есть характеризуется числом $\max(\chi(1), \chi(2))$.

В задаче Вальда требуется среди всех стратегий, удовлетворяющих требованиям $\omega(1) \leq \varepsilon$, $\omega(2) \leq \varepsilon$, найти стратегию, которая минимизирует число $\max(\chi(1), \chi(2))$. Известно, что решение этой задачи основано на сравнении отношения правдоподобия

$$\gamma(x) = \frac{p_{X|K}(x|1)}{p_{X|K}(x|2)}$$

с двумя определенным образом выбранными порогами θ_1 и θ_2 , $\theta_1 \leq \theta_2$. Решение в пользу первого состояния принимается, если отношение правдоподобия больше большего порога. Решение в пользу второго состояния

принимается, если отношение правдоподобия меньше меньшего порога. В оставшемся случае принимается решение отказаться от принятия решения. Далее мы покажем, как выводится такая стратегия, и обобщим ее для случая многих состояний.

2.2.5 Статистические решения при неслучайных воздействиях

Небайесовские задачи, рассмотренные в предыдущих подразделах, учитывали тот факт, что в некоторых приложениях невозможно определить понятие риска или априорные вероятности состояний. Статистические задачи при неслучайных мешающих воздействиях относятся к ситуациям, когда к тому же не существуют и условные вероятности $p_{X|K}(x | k)$, так как наблюдение зависит не только от состояния, но и от неслучайного воздействия.

Статистические задачи с неслучайными мешающими параметрами были сформулированы и исследованы Линником [Linnik, 1966]. Задачи относятся к ситуации, когда наблюдение x как случайная величина, зависит не только от состояния объекта, но и от некоторого дополнительного параметра z , воздействующего на объект. Этот параметр также не наблюдаем, но пользователя непосредственно его значение не интересует и поэтому нет нужды его оценивать. Тем не менее само существование параметра z должно учитываться, так как условные вероятности $p_{X|K}(x | k)$ не определены. Определены только условные вероятности $p_{X|K,Z}(x | k, z)$. Задачи распознавания объектов, подверженных неслучайным, но меняющимся воздействиям, известны как задачи *различение сложных гипотез* или *задачи Линника*.

Мы говорим об этих задачах во множественном числе, так как та или иная формулировка задачи зависит от того, является ли состояние k случайным или нет, определена ли функция потерь и т.п. Мы укажем два примера из этого обширного поля задач.

Различение сложных гипотез при случайном состоянии и неслучайных вмешательствах

Пусть X, K, Z - конечные множества возможных значений наблюдения x , состояния k и вмешательства z , $p_K(k)$ - априорные вероятности состояний k и $p_{X|K,Z}(x | k, z)$ - условные вероятности наблюдений x при фиксированном состоянии k и вмешательстве z . Пусть $X(k)$, $k \in K$, разбиение множества X наблюдений, которое определяет стратегию распознавания состояния k . Вероятность ошибочного решения ω зависит не только от стратегии, но и от вмешательства z ,

$$\omega(z) = \sum_{k \in K} p_K(k) \sum_{x \notin X(k)} p_{X|K,Z}(x | k, z).$$

Качество ω^* стратегии $(X(k), k \in K)$ определяется как вероятность ошибочного решения при наихудшем для этой стратегии вмешательстве, то есть

$$\omega^* = \max_{z \in Z} \omega(z).$$

Задача состоит в таком разбиении множества X наблюдений на подмножества $X(k)$, $k \in K$, чтобы минимизировалась величина ω^* ,

$$(X^*(k), k \in K) = \operatorname{argmin}_{(X(k), k \in K)} \max_{z \in Z} \sum_{k \in K} p_K(k) \sum_{x \notin X(k)} p_{X|K,Z}(x | k, z).$$

Различение сложных гипотез при неслучайном состоянии и неслучайном воздействии

Эта задача относится к случаю, когда ни состояние k ни вмешательство z не могут считаться случайными и, следовательно, априорные вероятности $p_K(k)$ не определены. В этом случае вероятность ошибочного решения ω , зависит не только от стратегии и неслучайного вмешательства z , но и от неслучайного состояния k ,

$$\omega(k, z) = \sum_{x \notin X(k)} p_{X|K,Z}(x | k, z).$$

Качество ω^* стратегии $(X(k), k \in K)$ определяется как

$$\omega^* = \max_{k \in K} \max_{z \in Z} \omega(k, z),$$

и задача формулируется как отыскание стратегии с наилучшим качеством, то есть отыскание разбиения

$$(X^*(k), k \in K) = \operatorname{argmin}_{(X(k), k \in K)} \max_{k \in K} \max_{z \in Z} \sum_{x \notin X(k)} p_{X|K,Z}(x | k, z).$$

2.3 Двойственные задачи линейного программирования, свойства и решения

Небайесовские задачи могут анализироваться в рамках единого математического формализма, именно, на основе теорем двойственности линейного программирования. Для удобства читателя мы не отсылаем его к учебникам, а вводим в данном разделе те сведения из линейного программирования, которые необходимы для решения небайесовских задач. Теоретическую основу линейного программирования составляют теоремы двойственности, известные в различных, но эквивалентных формулировках. Мы приведем их здесь в тех формулировках, от которых окажется наиболее близкий путь к основным задачам нашей лекции.

Как видно, данный раздел является вспомогательным. Кто достаточно свободно владеет техникой перехода от прямой задачи к двойственной, может данный раздел пропустить.

Пусть \mathbb{R}^m - m -мерное линейное пространство. Точка $x \in \mathbb{R}^m$ этого пространства имеет координаты x_1, x_2, \dots, x_m . Пусть I - множество $\{1, 2, \dots, m\}$ индексов (номеров координат). Пусть на некоторые координаты вектора x наложено ограничение, что они не могут принимать отрицательные значения. Обозначим $I^+ \subset I$ подмножество номеров тех

координат, которые могут принимать только неотрицательные значения. Подмножество $I^0 = I \setminus I^+$, таким образом, есть подмножество номеров тех координат, которые могут принимать любые значения, как положительные, так и отрицательные. Обозначим X подмножество точек пространства, которые удовлетворяют введенному ограничению, то есть, $X = \{x \in \mathbb{R}^m \mid x_i \geq 0, i \in I^+\}$.

Пусть \mathbb{R}^n - n -мерное линейное пространство, J - множество индексов (номеров координат), J^+ - подмножество номеров координат, которые могут принимать лишь неотрицательные значения, и $J^0 = J \setminus J^+$. Обозначим Y подмножество точек пространства \mathbb{R}^n , у которых координаты y_j с номерами $j \in J^+$ неотрицательны, то есть, $Y = \{y \in \mathbb{R}^n \mid y_j \geq 0, j \in J^+\}$.

Пусть на множестве $X \times Y$ определена функция двух переменных $f: X \times Y \rightarrow \mathbb{R}$. Эта функция не обязательно линейна, но функция $f(x^*, y)$ линейно зависит от переменной y при любом фиксированном значении $x^* \in X$. Подобным образом, $f(x, y^*)$ линейно зависит от переменной x при любом фиксированном значении $y^* \in Y$. Такие функции называются *билинейными* и имеют вид

$$f(x, y) = \sum_{i \in I} a_i x_i - \sum_{i \in I} \sum_{j \in J} b_{ij} x_i y_j + \sum_{j \in J} c_j y_j. \quad (2.10)$$

Точку $x' \in X$ назовем допустимой относительно заданной функции (2.10), если функция $f(x', y)$ одной переменной y ограничена сверху на множестве Y . Обозначим $\tilde{X} \subset X$ множество всех допустимых точек x . Из этого определения следует, что функция $\varphi: \tilde{X} \rightarrow \mathbb{R}$ со значениями

$$\varphi(x) = \max_{y \in Y} f(x, y)$$

определена на множестве \tilde{X} .

Прямая задача линейного программирования состоит прежде всего в определении, является ли множество \tilde{X} непустым. Если $\tilde{X} \neq \emptyset$, то следует определить, ограничена ли снизу функция $\varphi: \tilde{X} \rightarrow \mathbb{R}$. Если результат этих проверок положительный, то требуется найти точку $x^* \in \tilde{X}$, которая минимизирует φ . В этом случае говорят, что *задача линейного программирования имеет решение*.

Двойственная задача линейного программирования определяется симметрично. Множество \tilde{Y} определяется как множество тех точек $y' \in Y$, для которых функция $f(x, y')$ одной переменной x ограничена снизу на множестве X . Для каждой точки $y \in \tilde{Y}$ введем обозначение

$$\psi(y) = \min_{x \in X} f(x, y).$$

При решении двойственной задачи определяется прежде всего условие $\tilde{Y} \neq \emptyset$. Затем, если функция ψ ограничена сверху на множестве \tilde{Y} , то отыскивается точка $y^* \in \tilde{Y}$, максимизирующая ψ . В этом случае говорят, что *двойственная задача линейного программирования имеет решение*.

Взаимосвязь прямой и двойственной задач линейного программирования указывает следующая теорема, являющаяся частным случаем знаменитой теоремы Куна-Таккера.

Теорема 2.1 Первая теорема двойственности. *Если прямая задача линейного программирования имеет решение, то двойственная задача также имеет решение. При этом соблюдается равенство*

$$\min_{x \in \tilde{X}} \varphi(x) = \max_{y \in \tilde{Y}} \psi(y). \quad (2.11) \quad \blacktriangle$$

Доказательство. Нам неизвестно короткое доказательство этой теоремы. Различные доказательства можно найти в различных учебниках по линейной оптимизации. Доказательство, наиболее близкое к стилю наших лекций, содержится в [Zuchovickij and Avdejeva, 1967]. ■

Множество \tilde{X} и функцию $\varphi: \tilde{X} \rightarrow \mathbb{R}$ можно выразить явно, как это показывает следующая лемма.

Лемма 2.1 Каноническая форма задачи линейного программирования. *Множество \tilde{X} - это множество решений системы равенств и неравенств*

$$\sum_{i \in I} b_{ij} x_i \geq c_j, \quad j \in J^+, \quad (2.12)$$

$$\sum_{i \in I} b_{ij} x_i = c_j, \quad j \in J^0, \quad (2.13)$$

$$x_i \geq 0, \quad i \in I^+,$$

и для каждого $x \in \tilde{X}$ выполняется

$$\varphi(x) = \sum_{i \in I} a_i x_i. \quad (2.14) \quad \blacktriangle$$

Доказательство. Запишем функцию f в следующем виде

$$f(x, y) = \sum_{i \in I} a_i x_i + \sum_{j \in J^0} y_j \left(c_j - \sum_{i \in I} b_{ij} x_i \right) + \sum_{j \in J^+} y_j \left(c_j - \sum_{i \in I} b_{ij} x_i \right) \quad (2.15)$$

Покажем сначала, что если некоторая точка x' не удовлетворяет системе равенств (2.12) и неравенств (2.13), то $x' \notin \tilde{X}$. Другими словами, что в этом случае функция $f(x', y)$ одной переменной y не ограничена сверху.

Предположим, что данная точка x' не удовлетворяет одному из неравенств системы (2.12), скажем, j' -ому неравенству. Функция f в таком случае может принимать сколь угодно большое значение. Для этого достаточно, чтобы переменная $y_{j'}$ стала достаточно большой, а этому ничто не препятствует: ведь на нее наложено лишь ограничение, что она не должна быть отрицательной.

Предположим, что данная точка x' не удовлетворяет некоторому, скажем j'' -ому равенству в системе равенств (2.13). В таком случае функция f может опять достигать сколь угодно большие значения. Для этого достаточно, чтобы абсолютная величина переменной $y_{j''}$ стала достаточно большой, а знак ее совпадал со знаком разности $c_{j''} - \sum_{i \in I} b_{ij''} x_i$, которая

не равна нулю. Таким образом, мы видим, что любая точка $x \in \tilde{X}$ есть решение системы неравенств (2.12) и равенств (2.13).

Покажем теперь, что если x' удовлетворяет условиям (2.12) и (2.13), то функция $f(x', y)$ одной переменной y ограничена на множестве Y , то есть, $x' \in \tilde{X}$.

Функция $f(x', y)$ состоит из трех сумм, см. (2.15). Первая сумма не зависит от y . Вторая сумма также не зависит от y . Более того, она равна нулю, так как в силу (2.13) разности $c_j - \sum_{i \in I} b_{ij} x'_i$ равны нулю для всех $j \in J^0$. Третья сумма неположительна, так как все слагаемые этой суммы неположительны. Действительно, для $j \in J^+$ по определению выполняется $y_j \geq 0$ и по условию (2.12) выполняется $c_j - \sum_{i \in I} b_{ij} x'_i \leq 0$. Следовательно, третья сумма в выражении (2.15) ограничена сверху нулем, а все выражение (2.15) в целом ограничено сверху числом $\sum_{i \in I} a_i x'_i$.

Таким образом доказано, что множество \tilde{X} есть множество решений системы неравенств (2.12) и равенств (2.13).

Очевидно, что верхний предел функции $f(x', y)$, то есть число $\sum_{i \in I} a_i x'_i$ достигается на множестве Y , так как $f(x', 0) = \sum_{i \in I} a_i x'_i$, а $0 \in Y$. Это значит, что $\max_{y \in Y} f(x', y) = \sum_{i \in I} a_i x'_i$ и равенство (2.14) доказано. ■

Таким образом, прямая задача линейного программирования состоит в решении следующей задачи на условный экстремум: найти

$$\min_{x \in X} \sum_{i \in I} a_i x_i \quad (2.16)$$

при условиях

$$\sum_{i \in I} b_{ij} x_i \geq c_j, \quad j \in J^+, \quad (2.17)$$

$$\sum_{i \in I} b_{ij} x_i = c_j, \quad j \in J^0, \quad (2.18)$$

$$x_i \geq 0, \quad i \in I^+. \quad (2.19)$$

В подобном стандартном виде можно представить и двойственную задачу линейного программирования. Коротко повторим для двойственной задачи основные моменты доказательства леммы 2.1 и покажем, что в ней требуется найти

$$\max_{y \in Y} \sum_{j \in J} c_j y_j \quad (2.20)$$

при условиях

$$\sum_{j \in J} b_{ij} y_j \leq a_i, \quad i \in I^+, \quad (2.21)$$

$$\sum_{j \in J} b_{ij} y_j = a_i, \quad i \in I^0, \quad (2.22)$$

$$y_j \geq 0, \quad j \in J^+. \quad (2.23)$$

Функцию f можно представить в виде

$$f(x, y) = \sum_{i \in I^0} x_i \left(a_i - \sum_{j \in J} b_{ij} y_j \right) + \sum_{i \in I^+} x_i \left(a_i - \sum_{j \in J} b_{ij} y_j \right) + \sum_{j \in J} c_j y_j. \quad (2.24)$$

Если для некоторой точки y' не выполняется i' -ое неравенство в системе (2.21), то вторую сумму в выражении (2.24) можно сделать сколь угодно отрицательной выбором достаточно большого положительного значения переменной $x_{i'}$. Если для точки y' не выполняется i'' -ое равенство в системе (2.22), то первую сумму в правой части (2.24) можно сделать сколь угодно отрицательной приемлемым выбором большого по абсолютной величине значения $x_{i''}$.

Очевидно, что для любой точки y , которая удовлетворяет условиям (2.21) и (2.22), значение $f(x, y)$ на множестве X ограничено снизу числом $\sum_{j \in J} c_j y_j$. Эта граница снизу достигается в точке $x = 0$, которая принадлежит X .

Выражения (2.16)–(2.23) вполне однозначно и лаконично определяют пару взаимно двойственных задач линейного программирования и правила перехода от прямой задачи к двойственной. Тем не менее стоит запомнить эти правила и в следующем словесном выражении. Оно не так однозначно, как формальное представление (2.16)–(2.23), и не такое краткое, однако в таком словесном выражении оно иногда легче запоминается.

1. В прямой задаче требуется минимизировать линейную функцию многих переменных при линейных ограничениях на эти переменные. В двойственной задаче требуется максимизировать линейную функцию от других переменных при линейных ограничениях на эти переменные.
2. Каждому ограничению прямой задачи соответствует переменная в двойственной задаче, и каждой переменной в прямой задаче соответствует ограничение в двойственной задаче. Количество переменных в прямой задаче равно количеству ограничений в двойственной задаче. Количество ограничений в прямой задаче равно количеству переменных в двойственной задаче.
3. Ограничения в прямой задаче имеют вид равенств или неравенств вида \geq . Ограничения в двойственной задаче имеют вид равенств или неравенств вида \leq .
4. Некоторые переменные как в прямой, так и в двойственной задаче могут принимать как положительные, так и отрицательные значения. Это *неограниченные переменные*. Другие переменные в прямой и двойственной задачах могут принимать только неотрицательные значения. Это *ограниченные переменные*.
5. Ограничению-равенству прямой задачи соответствует неограниченная переменная в двойственной задаче. Ограничению-неравенству прямой задачи соответствует ограниченная переменная в двойственной задаче. Свободной переменной в прямой задаче соответствует ограничение в виде равенства в двойственной задаче. Ограниченной переменной в

прямой задаче соответствует ограничение в виде неравенства в двойственной задаче.

6. Коэффициенты a_i в линейной функции, которую следует минимизировать в прямой задаче, являются пороговыми значениями в правых частях ограничений в двойственной задаче. Пороговые значения c_j в правых частях ограничений прямой задачи являются коэффициентами линейной функции, которую следует максимизировать в двойственной задаче.
7. Коэффициент b_{ij} при i -ой переменной в j -ом ограничении прямой задачи в точности равен коэффициенту при j -ой переменной в i -ом ограничении двойственной задачи.

Следующая теорема указывает важные свойства точек x^* и y^* , являющихся решением прямой и двойственных задач линейного программирования.

Теорема 2.2 Вторая теорема двойственности или теорема о дополняющей жесткости. Пусть $x^* = (x_i^*, i \in I)$ - решение прямой, а $y^* = (y_j^*, j \in J)$ - решение двойственной задач линейного программирования.

Если некоторая координата x_i^* в точке x^* не равна нулю, то в точке y^* i -ое ограничение двойственной задачи выполняется как равенство.

Если j -ое ограничение прямой задачи в точке x^* выполняется как строгое неравенство, то переменная y_j^* двойственной задачи равна нулю. ▲

Доказательство. Теорема фактически утверждает, что для всех $i \in I$ выполняется равенство

$$x_i^* \left(a_i - \sum_{j \in J} b_{ij} y_j^* \right) = 0, \quad (2.25)$$

а для всех $j \in J$ - равенство

$$y_j^* \left(c_j - \sum_{i \in I} b_{ij} x_i^* \right) = 0. \quad (2.26)$$

Очевидным является равенство

$$\sum_{i \in I} a_i x_i^* + \sum_{j \in J} y_j^* \left(c_j - \sum_{i \in I} b_{ij} x_i^* \right) = \sum_{i \in I} x_i^* \left(a_i - \sum_{j \in J} b_{ij} y_j^* \right) + \sum_{j \in J} c_j y_j^*. \quad (2.27)$$

Первая теорема двойственности утверждает, что $\sum_{i \in I} a_i x_i^* = \sum_{j \in J} c_j y_j^*$, откуда следует

$$\sum_{j \in J} y_j^* \left(c_j - \sum_{i \in I} b_{ij} x_i^* \right) = \sum_{i \in I} x_i^* \left(a_i - \sum_{j \in J} b_{ij} y_j^* \right). \quad (2.28)$$

Поскольку x^* и y^* - это решения прямой и двойственной задач, они удовлетворяют ограничения (2.17)–(2.19) и (2.21)–(2.23), и таким образом, для

всех $j \in J$ выполняется

$$y_j^* \left(c_j - \sum_{i \in I} b_{ij} x_i^* \right) \leq 0, \quad (2.29)$$

и для всех $i \in I$ выполняется

$$x_i^* \left(a_i - \sum_{j \in J} b_{ij} y_j^* \right) \geq 0. \quad (2.30)$$

Таким образом, равенство (2.28) утверждает, что сумма некоторых неположительных чисел равна сумме некоторых неотрицательных чисел. Это может быть только тогда, когда все слагаемые в обеих суммах равны нулю. Это доказывает равенства (2.25) и (2.26). ■

2.4 Решения небайесовских задач на основе теорем двойственности

Мы покажем сейчас, как небайесовские задачи следует решать в рамках теории двойственности линейного программирования.

До сих пор мы представляли стратегию в виде разбиения множества наблюдений X на подмножества $X(k)$, $k \in K$. Для каждой пары различных подмножеств $X(k')$ и $X(k'')$ выполнялось $X(k') \cap X(k'') = \emptyset$, и, кроме того, $\bigcup_{k \in K} X(k) = X$. Это значит, что каждое наблюдение $x \in X$ принадлежало какому-то одному и только одному подмножеству $X(k)$. Эти подмножества представляли стратегию в том смысле, что при $x \in X(k)$ принималось решение, что наблюдаемый объект находится в состоянии k .

Эту же стратегию можно выразить и с помощью такой функции $\alpha: X \times K \rightarrow \mathbb{R}$, что $\alpha(x, k)$ - это неотрицательные числа и, кроме того, $\sum_{k \in K} \alpha(x, k) = 1$ для любого $x \in X$. Любое разбиение множества X , то есть любую *детерминированную стратегию*, можно представить как целочисленную функцию из указанного класса так, что, если $x \in X(k)$, то $\alpha(x, k) = 1$.

Множество нецелочисленных функций α указанного класса значительно шире множества всех детерминированных стратегий. Нецелочисленную функцию α следует понимать как *рандомизированную стратегию*, такую, что число $\alpha(x, k)$ есть условная вероятность принятия решения в пользу состояния k при условии наблюдения x .

Мы решим небайесовские задачи, не ограничиваясь заранее рассмотрением только лишь детерминированных стратегий, то есть только лишь целочисленных функций α . Но мы увидим, что решение задач в их оптимизационной постановке само собой окажется почти всегда детерминированным. Так же как и при решении байесовских задач, стратегии решения небайесовских задач выражаются как разбиение пространства вероятностей на выпуклые конусы. Внутренние точки конусов соответствуют наблюдениям, при которых решение должно быть детерминированным. Необ-

ходимость рандомизации может возникнуть (но необязательно) только при наблюдениях, которые соответствуют точкам на границе двух конусов.

2.4.1 Решение задачи Неймана-Пирсона

Исходная формулировка задачи Неймана-Пирсона выражена требованиями (2.2)–(2.4). Сформулируем эту же задачу, представляя стратегию в виде функции $\alpha: X \times K \rightarrow \mathbb{R}$, а не множеств X_1 и X_2 , как это было раньше. Сама задача при этом не меняется: как и прежде, она состоит в минимизации ложной тревоги при условии, что вероятность пропуска опасности не превышает заданной величины ε . Вероятность ложной тревоги есть сумма

$$\sum_{x \in X} \alpha(x, 2) p_{X|K}(x|1), \quad (2.31)$$

а вероятность пропуска опасности есть сумма

$$\sum_{x \in X} \alpha(x, 1) p_{X|K}(x|2). \quad (2.32)$$

Для каждого наблюдения $x \in X$ искомые значения функции α должны удовлетворять условиям $\alpha(x, 1) + \alpha(x, 2) = 1$, $\alpha(x, 1) \geq 0$, $\alpha(x, 2) \geq 0$ для всех $x \in X$. Эти требования можно выразить как задачу линейного программирования.

Введем формат лаконичной записи задач линейного программирования, которые мы будем исследовать далее. Задача будет выражена в виде нескольких линейных выражений. Выражение в первой строке будет представлять функцию, которую следует максимизировать или минимизировать. Варьируемые переменные, по которым осуществляется оптимизация, не будут записываться в явном виде. Имеется в виду, что в каждой задаче осуществляется оптимизация по всем несвязанным переменным. Впрочем, в каждом конкретном случае станет достаточно ясно, по каким переменным происходит оптимизация. В строчках, следующих после первой строки, будут записаны равенства и неравенства, выражающие ограничения на переменные в данной задаче. Слева от каждого ограничения будет записан идентификатор двойственной переменной, соответствующей данному ограничению. Двойственные переменные будут отделены от ограничений вертикальной чертой. Подобным образом в двойственной задаче, слева от ограничений будут записаны идентификаторы соответствующих переменных прямой задачи.

Величины $\alpha(x, k)$, $x \in X$, $k \in K$, определяющие стратегию, будут трактоваться как переменные в последующих задачах. Именно эти величины должны будут определяться в процессе решения задачи, то есть при оптимизации выражения, записанного в первой строчке. Иногда, в частности, при рассмотрении минимаксных проблем, будут вводиться вспомогательные переменные. Их смысл будет объяснен, когда до этого дойдет очередь.

Здесь мы считаем необходимым обратить внимание на то, что величины x и k вовсе не являются варьируемыми переменными, по которым

производится оптимизация. Это переменные, по которым производится суммирование или которые служат номерами ограничений в задаче.

В этом смысле величины x и k в предстоящих записях имеют тот же смысл, что индексы i и j в предыдущем разделе 2.3, где задачи линейного программирования анализировались в общем виде. Различные вероятности $p_{X|K}(x, k)$, $p_K(k)$ и т.п. - это константы в каждой конкретной задаче. Они играют ту же роль, что и мультипликативные коэффициенты в линейных функциях, которые следует оптимизировать или которые являются ограничениями в задаче.

Задача минимизации (2.31) при условии (2.32) и других очевидных условиях выражается следующей задачей линейного программирования:

$$\left. \begin{aligned} \min \sum_{x \in X} \alpha(x, 2) p_{X|K}(x|1), & \quad (\text{a}) \\ \sum_{x \in X} \alpha(x, 1) p_{X|K}(x|2) \leq \varepsilon, & \quad (\text{b}) \\ \alpha(x, 1) + \alpha(x, 2) = 1, \quad x \in X, & \quad (\text{c}) \\ \alpha(x, 1) \geq 0, \quad x \in X, & \quad (\text{d}) \\ \alpha(x, 2) \geq 0, \quad x \in X. & \quad (\text{e}) \end{aligned} \right\} \quad (2.33)$$

В оптимизационной задаче (2.33) переменными являются числа $\alpha(x, 1)$, $\alpha(x, 2)$ для всех $x \in X$. Константами являются числа ε и вероятности $p_{X|K}(x|k)$ для всех $x \in X$, $k = 1, 2$.

Перепишем задачу в другой форме, чтобы представить неравенство (2.33b) в виде отношения \geq , как это требуется в стандартном виде прямой задачи. Слева от каждого ограничения запишем соответствующую двойственную переменную. При этом примем во внимание, что в строке (2.33c) представлено не одно ограничение, а $|X|$ ограничений, соответствующих различным наблюдениям x . Каждому ограничению из этого семейства, то есть каждому наблюдению x , поставлена в соответствие двойственная переменная $t(x)$.

$$\left. \begin{aligned} \min \sum_{x \in X} \alpha(x, 2) p_{X|K}(x|1), & \quad (\text{a}) \\ \tau \left| - \sum_{x \in X} \alpha(x, 1) p_{X|K}(x|2) \geq -\varepsilon, & \quad (\text{b}) \right. \\ t(x) \left| \alpha(x, 1) + \alpha(x, 2) = 1, \quad x \in X, & \quad (\text{c}) \right. \\ & \quad \alpha(x, 1) \geq 0, \quad x \in X, \quad (\text{d}) \\ & \quad \alpha(x, 2) \geq 0, \quad x \in X. \quad (\text{e}) \end{aligned} \right\} \quad (2.34)$$

Этой прямой задаче соответствует следующая двойственная задача

$$\left. \begin{array}{l} \max \left(\sum_{x \in X} t(x) - \varepsilon \tau \right), \quad (\text{a}) \\ \alpha(x, 1) \left| \begin{array}{l} t(x) - \tau p_{X|K}(x|2) \leq 0, \quad x \in X, \quad (\text{b}) \\ t(x) \leq p_{X|K}(x|1), \quad x \in X, \quad (\text{c}) \\ \tau \geq 0. \quad (\text{d}) \end{array} \right. \end{array} \right\} \quad (2.35)$$

На примере этой простейшей и первой рассматриваемой нами небайесовской задачи мы подробно объясним технику преобразования прямой задачи в двойственную. В последующих задачах эти объяснения будут все более и более краткими.

В строке (2.35а) записана линейная функция от двойственных переменных τ и $t(x)$, $x \in X$. Каждая переменная $t(x)$, $x \in X$, умножается на коэффициент 1, так как 1 стоит в правой части ограничений (2.34с), которым переменные $t(x)$ соответствуют. Переменная τ умножается на $-\varepsilon$, потому что порог $-\varepsilon$ стоит в правой части неравенства (2.34b), которому соответствует переменная τ .

В строке (2.35b) записаны $|X|$ ограничений. Каждое из них соответствует переменной $\alpha(x, 1)$ прямой задачи при том или ином значении x . Ограничение выражено в виде неравенства, потому что переменная $\alpha(x, 1)$ объявлена в прямой задаче, как неотрицательная. В правой стороне ограничения записан 0, потому что переменная $\alpha(x, 1)$ не присутствует в линейной функции (2.34а), которую следует оптимизировать в прямой задаче, то есть присутствует с коэффициентом 0. Левая сторона ограничения состоит из двух слагаемых, потому что переменная $\alpha(x, 1)$ присутствует только в двух ограничениях прямой задачи: в ограничении (2.34b) и в одном ограничении из семейства (2.34с). Переменная $t(x)$ в ограничении (2.35b) умножена на 1, потому что переменная $\alpha(x, 1)$ в ограничении (2.34с) умножена на 1. Переменная τ в ограничении (2.35b) умножена на $-p_{X|K}(x|2)$, потому что переменная $\alpha(x, 1)$ в ограничении (2.34b) умножена на $-p_{X|K}(x|2)$.

В строке (2.35с) представлены $|X|$ ограничений, соответствующих группе переменных $\alpha(x, 2)$, $x \in X$. В правой стороне ограничения, соответствующего конкретному значению переменной x , записана вероятность $p_{X|K}(x|1)$, так как именно эта вероятность служит коэффициентом при переменной $\alpha(x, 2)$ в линейной функции (2.34а), которую следует минимизировать в прямой задаче. В левой части ограничения (2.35с) записана только переменная $t(x)$, так как переменная $\alpha(x, 2)$ присутствует только в одном ограничении (2.34с) прямой задачи. Переменная $t(x)$ в ограничении (2.35с) умножена на 1, так как переменная $\alpha(x, 2)$ в ограничении (2.34с) умножена на 1. Ограничение (2.35с) имеет вид неравенства, так как переменная $\alpha(x, 2)$ объявлена в прямой задаче, как неотрицательная, см. (2.34е).

Ограничение (2.35d) объявляет переменную τ неотрицательной, так как она соответствует ограничению-неравенству (2.34b) в прямой задаче. Двой-

ственные переменные $t(x)$, $x \in X$, могут принимать любые значения, так как они соответствуют ограничениям-равенствам в прямой задаче.

Мы видим, что преобразование прямой задачи (2.34) в двойственную задачу (2.35) выполняется по четким формальным правилам. После приобретения определенных навыков этот переход выполняется почти автоматически, и приведенные выше подробные объяснения становятся излишними. Действительно, понятие двойственных задач определено настолько однозначно, что переход от одной задачи к другой может выполнить даже механическое устройство, например, вычислительная машина.

После того, как задача Неймана-Пирсона представлена в виде пары двойственных задач, ее решение легко находится на основании второй теоремы двойственности о дополняющей жесткости (Теорема 2.2). Решением задачи (2.35) не могут быть такие значения τ и $t(x)$, $x \in X$, при которых оба неравенства (2.35b) и (2.35c) выполняются строго. В этом случае на основании второй теоремы двойственности должно было бы выполняться $\alpha(x, 1) = \alpha(x, 2) = 0$, что противоречило бы ограничению (2.34c). Следовательно, по крайней мере одно из неравенств (2.35b) или (2.35c) должно обращаться в равенство, то есть

$$t(x) = \min (p_{X|K}(x|1), \tau p_{X|K}(x|2)) .$$

Это значит, что если

$$p_{X|K}(x|1) < \tau p_{X|K}(x|2) \quad (2.36)$$

то $t(x) < \tau p_{X|K}(x|2)$ и строго выполняется неравенство (2.35b). Следовательно, $\alpha(x, 1) = 0$ и в силу ограничения (2.34c) $\alpha(x, 2) = 1$: состояние k должно оцениваться, как опасное.

Если

$$p_{X|K}(x|1) > \tau p_{X|K}(x|2) \quad (2.37)$$

то $t(x) < p_{X|K}(x|1)$. Поскольку неравенство (2.35c) выполняется строго, то $\alpha(x, 2) = 0$, $\alpha(x, 1) = 1$ и решение принимается в пользу нормального состояния. Условия (2.36) и (2.37) можно представить в известной форме, при которой отношение правдоподобия

$$\gamma(x) = \frac{p_{X|K}(x|1)}{p_{X|K}(x|2)} \quad (2.38)$$

сравнивается с порогом τ .

Мы видим, таким образом, что задачу Неймана-Пирсона можно записать в виде пары двойственных задач (2.34) и (2.35), после чего ее решение легко получить на основании второй теоремы двойственности. Подобным образом можно выводить решения и других небайесовских задач, как известных, так и не известных.

2.4.2 Решение обобщенной задачи Неймана-Пирсона для случая двух опасных состояний

Покажем теперь кратко, как вывести стратегию для модифицированной задачи Неймана-Пирсона, которая была сформулирована в подразделе 2.2.2 в виде требований (2.5)–(2.9).

Пусть объект может находиться в одном из трех состояний, $k = 1, 2$ или 3 . Состояние $k = 1$ считается нормальным, а два другие состояния - опасными. Требуется определить числа $\alpha_1(x)$ и $\alpha_{23}(x)$, определяющие стратегию следующим образом. Если для наблюдения x выполняется $\alpha_1(x) = 1$, то принимается решение, что состояние объекта нормальное. Если $\alpha_{23}(x) = 1$, то принимается решение, что объект находится в опасном состоянии, не уточняя при этом, в каком именно, $k = 2$ или $k = 3$. Числа $\alpha_1(x)$ и $\alpha_{23}(x)$ должны минимизировать сумму

$$\sum_{x \in X} \alpha_{23}(x) p_{X|K}(x|1) \quad (2.39)$$

при условиях

$$\sum_{x \in X} \alpha_1(x) p_{X|K}(x|2) \leq \varepsilon, \quad (2.40)$$

$$\sum_{x \in X} \alpha_1(x) p_{X|K}(x|3) \leq \varepsilon, \quad (2.41)$$

$$\alpha_1(x) + \alpha_{23}(x) = 1, \quad x \in X, \quad (2.42)$$

$$\alpha_1(x) \geq 0, \quad x \in X,$$

$$\alpha_{23}(x) \geq 0, \quad x \in X. \quad (2.43)$$

Выражение (2.39) представляет вероятность ложной тревоги. Условие (2.40) обозначает, что вероятность пропуска опасного состояния $k = 2$ должна быть малой, а условие (2.41) выражает то же требование для состояния $k = 3$. Ограничения (2.42)–(2.43) выражают стандартные требования, которым должна удовлетворять любая стратегия. Запишем приведенные требования к стратегии в виде, стандартном для пары двойственных задач.

Прямая задача:

$$\left. \begin{array}{l} \min \sum_{x \in X} \alpha_{23}(x) p_{X|K}(x|1), \\ \tau_2 \left| - \sum_{x \in X} \alpha_1(x) p_{X|K}(x|2) \geq -\varepsilon, \right. \\ \tau_3 \left| - \sum_{x \in X} \alpha_1(x) p_{X|K}(x|3) \geq -\varepsilon, \right. \\ t(x) \left| \begin{array}{l} \alpha_1(x) + \alpha_{23}(x) = 1, \quad x \in X, \\ \alpha_1(x) \geq 0, \quad x \in X, \\ \alpha_{23}(x) \geq 0, \quad x \in X. \end{array} \right. \end{array} \right\} \quad (2.44)$$

Двойственная задача:

$$\left. \begin{array}{l} \alpha_1(x) \left| \begin{array}{l} \max \left(\sum_{x \in X} t(x) - \varepsilon(\tau_2 + \tau_3) \right), \\ t(x) - \tau_2 p_{X|K}(x|2) - \tau_3 p_{X|K}(x|3) \leq 0, \quad x \in X, \quad (\text{a}) \\ t(x) \leq p_{X|K}(x|1), \quad x \in X, \quad (\text{b}) \\ \tau_2 \geq 0, \\ \tau_3 \geq 0. \end{array} \right. \end{array} \right\} (2.45)$$

Из условий (2.45a), (2.45b) и требования, что числа $\alpha_1(x)$ и $\alpha_{23}(x)$ не могут одновременно равняться нулю, следует, что для $x \in X$ переменная $t(x)$ должна принимать значение

$$t(x) = \min(p_{X|K}(x|1), \tau_2 p_{X|K}(x|2) + \tau_3 p_{X|K}(x|3)). \quad (2.46)$$

Это значит, что если

$$p_{X|K}(x|1) < \tau_2 p_{X|K}(x|2) + \tau_3 p_{X|K}(x|3), \quad (2.47)$$

то $\alpha_1(x)$ должно равняться 0, $\alpha_{23}(x)$ должно равняться 1, и при таком наблюдении x должно приниматься решение об опасном состоянии. Если для наблюдения x выполняется неравенство

$$p_{X|K}(x|1) > \tau_2 p_{X|K}(x|2) + \tau_3 p_{X|K}(x|3), \quad (2.48)$$

то должно приниматься решение, что состояние объекта нормальное. Таким образом стратегия определена с точностью до двух неотрицательных весовых коэффициентов τ_2 и τ_3 . Если они заданы, то вычисляется отношение правдоподобия

$$\gamma(x) = \frac{p_{X|K}(x|1)}{\tau_2 p_{X|K}(x|2) + \tau_3 p_{X|K}(x|3)}, \quad (2.49)$$

и решение о нормальном или опасном состоянии принимается в зависимости от того, превышает ли оно 1 или нет.

2.4.3 Решение минимаксной задачи

Мы будем пользоваться стандартным приемом, который позволяет преобразовать минимаксные задачи в задачи минимизации, избегая рассмотрения вспомогательной операции максимизации. Покажем этот прием на примере.

Пусть $f_j(x)$, $j \in J$, - совокупность функций одной переменной и следует найти значение x^* , при котором число $\varphi(x) = \max_{j \in J} f_j(x)$ достигает минимальное значение. Эта оптимизационная задача может быть выражена в виде оптимизации не по одной переменной, а двух: первоначальной

переменной x и новой, вспомогательной переменной c . В этой новой формулировке должна быть найдена пара (x, c) с минимальным значением c при условиях $c \geq f_j(x)$, $j \in J$. Задача

$$\min_x \max_{j \in J} f_j(x) \tag{2.50}$$

таким образом представлена в виде

$$\left. \begin{aligned} \min c, \\ c - f_j(x) \geq 0, \quad j \in J. \end{aligned} \right\} \tag{2.51}$$

Этот прием иллюстрируется на рис. 2.1 для случая $J = \{1, 2, 3\}$. Затемненная область на рисунке указывает множество пар (x, c) , которые удовлетворяют условиям $c - f_1(x) \geq 0$, $c - f_2(x) \geq 0$, $c - f_3(x) \geq 0$. В задаче (2.51) требуется найти точку в этой области с наименьшей

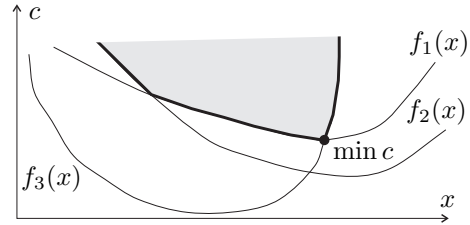


Figure 2.1 Minimax task for three functions f . Очевидно, это точка, отмеченная черным кружочком на рис. 2.1. Функция $\max(f_1(x), f_2(x), f_3(x))$ представлена на рисунке в виде жирной линии. В задаче (2.50) требуется найти на этой линии точку с минимальной координатой c . Очевидно, что это та же самая точка, обозначенная черным кружочком.

На основе эквивалентности задач (2.50) и (2.51) минимаксная задача, сформулированная в подразделе 2.2.3, может быть выражена как следующая задача линейного программирования. Переменными в этой задаче являются числа $\alpha(x, k)$ и вспомогательное число c .

$$\left. \begin{aligned} \min c, \\ c - \sum_{x \in X} \left(\sum_{k^* \neq k} \alpha(x, k^*) p_{X|K}(x|k) \right) \geq 0, \quad k \in K, \quad (a) \\ \sum_{k \in K} \alpha(x, k) = 1, \quad x \in X, \quad (b) \\ \alpha(x, k) \geq 0, \quad x \in X, \quad k \in K. \end{aligned} \right\} \tag{2.52}$$

В силу (2.52b) сумма $\sum_{k^* \neq k} \alpha(x, k^*)$ равна $1 - \alpha(x, k)$, и следовательно, неравенство (2.52a) можно записать как $c + \sum_{x \in X} \alpha(x, k) p_{X|K}(x|k) \geq 1$. Решаемая задача окончательно записывается в виде следующей пары двойственных задач:

Прямая задача:

$$\left. \begin{array}{l} \min c, \\ \tau(k) \left| \begin{array}{l} c + \sum_{x \in X} \alpha(x, k) p_{X|K}(x|k) \geq 1, \quad k \in K, \\ \sum_{k \in K} \alpha(x, k) = 1, \quad x \in X, \\ \alpha(x, k) \geq 0, \quad x \in X, \quad k \in K. \end{array} \right. \end{array} \right\} \quad (2.53)$$

Двойственная задача:

$$\left. \begin{array}{l} \max \left(\sum_{x \in X} t(x) + \sum_{k \in K} \tau(k) \right), \\ \alpha(x, k) \left| \begin{array}{l} t(x) + \tau(k) p_{X|K}(x|k) \leq 0, \quad x \in X, \quad k \in K, \quad (\text{a}) \\ \sum_{k \in K} \tau(k) = 1, \\ \tau(k) \geq 0, \quad k \in K. \end{array} \right. \end{array} \right\} \quad (2.54)$$

Очевидно, что при фиксировании значений всех переменных, кроме $t(x)$, $x \in X$, переменные $t(x)$ должны достигать как можно больших значений. В силу условий (2.54a) эти значения равны

$$\min_{k \in K} (-\tau(k) p_{X|K}(x|k)).$$

Для некоторых значений $x \in X$ и $k^* \in K$, а именно, тех, которые удовлетворяют условию

$$k^* \neq \operatorname{argmax}_{k \in K} (\tau(k) p_{X|K}(x|k)),$$

переменные $\alpha(x, k^*)$ должны равняться нулю. Это значит, что ненулевыми могут быть только те числа $\alpha(x, k^*)$, для которых

$$k^* = \operatorname{argmax}_{k \in K} (\tau(k) p_{X|K}(x|k)).$$

2.4.4 Решение задачи Вальда для случая двух состояний

Задача Вальда была сформулирована в подразделе 2.2.4. Мы выразим стратегию решения вальдовской задачи в виде трех функций α_0 , α_1 и α_2 , определенных на множестве X . Равенство чисел $\alpha_1(x)$ или $\alpha_2(x)$ единице понимается как решение в пользу первого или второго состояния. Равенство $\alpha_0(x) = 1$ обозначает, что стратегия воздерживается от решения в пользу какого-либо состояния. Задачу Вальда можно выразить парой двойственных задач, которые далее решаются уже проторенным нами способом.

Прямая задача:

$$\left. \begin{array}{l}
 \min c \\
 q_1 \left| \begin{array}{l} c - \sum_{x \in X} \alpha_0(x) p_{X|K}(x|1) \geq 0, \\
 q_2 \left| \begin{array}{l} c - \sum_{x \in X} \alpha_0(x) p_{X|K}(x|2) \geq 0, \\
 \tau_2 \left| \begin{array}{l} - \sum_{x \in X} \alpha_1(x) p_{X|K}(x|2) \geq -\varepsilon, \\
 \tau_1 \left| \begin{array}{l} - \sum_{x \in X} \alpha_2(x) p_{X|K}(x|1) \geq -\varepsilon, \\
 t(x) \left| \begin{array}{l} \alpha_0(x) + \alpha_1(x) + \alpha_2(x) = 1, \quad x \in X, \\
 \alpha_0(x) \geq 0, \quad \alpha_1(x) \geq 0, \quad \alpha_2(x) \geq 1, \quad x \in X. \end{array} \right. \end{array} \right. \end{array} \right. \end{array} \right. \end{array} \right. \end{array} \right\} \quad (2.55)$$

Двойственная задача:

$$\left. \begin{array}{l}
 \max \left(\sum_{x \in X} t(x) - \varepsilon(\tau_1 + \tau_2) \right), \\
 \alpha_1(x) \left| \begin{array}{l} t(x) - \tau_2 p_{X|K}(x|2) \leq 0, \quad x \in X, \quad (a) \\
 \alpha_2(x) \left| \begin{array}{l} t(x) - \tau_1 p_{X|K}(x|1) \leq 0, \quad x \in X, \quad (b) \\
 \alpha_0(x) \left| \begin{array}{l} t(x) - q_1 p_{X|K}(x|1) - q_2 p_{X|K}(x|2) \leq 0, \quad x \in X, \quad (c) \\
 c \left| \begin{array}{l} q_1 + q_2 = 1, \\
 q_1 \geq 0, \quad q_2 \geq 0, \quad \tau_1 \geq 0, \quad \tau_2 \geq 0. \end{array} \right. \end{array} \right. \end{array} \right. \end{array} \right. \end{array} \right\} \quad (2.56)$$

Из условий (2.56a), (2.56b), (2.56c) и вида функции, которую следует максимизировать в двойственной задаче, следует, что переменная $t(x)$ должна равняться наименьшему из трех чисел $\tau_1 p_{X|K}(x|1)$, $\tau_2 p_{X|K}(x|2)$ и $q_1 p_{X|K}(x|1) + q_2 p_{X|K}(x|2)$. Рассмотрим *три случая*, соответствующих тому, какое из указанных трех чисел наименьшее.

Случай 1. Если

$$\left. \begin{array}{l}
 \tau_1 p_{X|K}(x|1) < \tau_2 p_{X|K}(x|2), \\
 \tau_1 p_{X|K}(x|1) < q_1 p_{X|K}(x|1) + q_2 p_{X|K}(x|2), \end{array} \right\} \quad (2.57)$$

то неравенства (2.56a) и (2.56c) выполняются строго. Это значит, что $\alpha_1(x) = 0$, $\alpha_0(x) = 0$ и, следовательно, $\alpha_2(x) = 1$.

Случай 2. Если

$$\left. \begin{array}{l}
 \tau_2 p_{X|K}(x|2) < \tau_1 p_{X|K}(x|1), \\
 \tau_2 p_{X|K}(x|2) < q_1 p_{X|K}(x|1) + q_2 p_{X|K}(x|2), \end{array} \right\} \quad (2.58)$$

то $\alpha_1(x) = 1$, так как в силу строгого выполнения неравенств (2.56b) и (2.56c) должно выполняться $\alpha_2(x) = 0$ и $\alpha_0(x) = 0$.

Случай 3. Наконец, если

$$\left. \begin{aligned} q_1 p_{X|K}(x|1) + q_2 p_{X|K}(x|2) < \tau_1 p_{X|K}(x|1), \\ q_1 p_{X|K}(x|1) + q_2 p_{X|K}(x|2) < \tau_2 p_{X|K}(x|2) \end{aligned} \right\} \quad (2.59)$$

то $\alpha_0(x) = 1$, так как в этом случае неравенства (2.56a) и (2.56b) выполняются строго и поэтому $\alpha_1(x) = 0$ и $\alpha_2(x) = 0$.

Условия (2.57)–(2.59) в явном виде указывают стратегию решения вальдовской задачи. Однако приведем ее к тому общеизвестному виду, в котором она известна по основополагающей работе Вальда [Wald, 1947].

Легко можно увидеть, что решение зависит от того, какая из трех величин

$$\tau_1 p_{X|K}(x|1), \quad \tau_2 p_{X|K}(x|2), \quad q_1 p_{X|K}(x|1) + q_2 p_{X|K}(x|2)$$

наименьшая, или, что то же самое, какая наименьшая из следующих трех величин:

$$\frac{\tau_1}{\tau_2} \gamma(x), \quad 1, \quad \frac{q_1}{\tau_2} \gamma(x) + \frac{q_2}{\tau_2} \quad (2.60)$$

где $\gamma(x)$ - это отношение правдоподобия $p_{X|K}(x|1)/p_{X|K}(x|2)$. На рис. 2.2

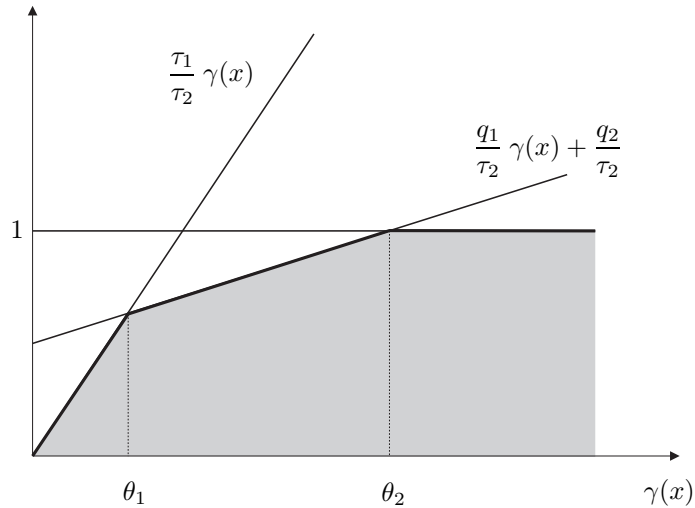


Figure 2.2 On the solution of Wald task.

показано, как эти три числа (2.60) зависят от отношения правдоподобия. Очевидно, что это - прямые линии. На оси абсцисс показаны два порога θ_1 , θ_2 . Видно, условие $\gamma(x) < \theta_1$ эквивалентно условию (2.57), условие $\gamma(x) > \theta_2$ совпадает с условием (2.58) и, наконец, условие $\theta_1 < \gamma(x) < \theta_2$ совпадает с условием (2.59). В первом случае принимается решение в пользу второго

состояния, во втором случае - в пользу первого состояния, а в третьем случае стратегия воздерживается от принятия решения. Это в точности решение вальдовской задачи, сформулированное Вальдом.

По рис. 2.2 выясняется еще одно интересное свойство вальдовских решений. В некоторых задачах подмножество X_0 может оказаться пустым. Это значит, что стратегия решения этой задачи никогда не выдаст ответ **not known**, даже если возможность такого ответа была предусмотрена при конструировании стратегии.

Обобщение задачи Вальда в случае, когда количество состояний объекта больше двух, и решение этой обобщенной задачи не столь прозрачно. Возможно, поэтому здесь имеется много различных рацпредложений, подобных попыткам обобщить неймановские стратегии. В следующем подразделе мы укажем одно из возможных разумных обобщений именно задачи Вальда, а не стратегии, полученной Вальдом для теперь уже частного случая. Мы увидим, что хотя в случае многих состояний понятие отношения правдоподобия теряет смысл, сформулированная обобщенная задача тоже может быть решена.

2.4.5 Решение задач Вальда для произвольного количества состояний

Когда $K = \{1, 2, \dots, n\}$, стратегию распознавания можно выразить числами $\alpha(x, k)$, где либо $k \in K$, либо $k = 0$. Качество такой стратегии характеризуется $2|K|$ вероятностями $\omega(k)$ и $\chi(k)$, $k \in K$. Число $\omega(k)$ есть условная вероятность ошибочного решения при условии, что объект находится в состоянии k . Число $\chi(k)$ есть условная вероятность ответа **not known**, при условии, что объект находится в состоянии k .

Искомая стратегия должна удовлетворять требованию $\omega(k) \leq \varepsilon$, $k \in K$, для заранее заданного значения ε предельно допустимой вероятности ошибки. Из множества стратегий, удовлетворяющих этому безусловному требованию, должна быть выбрана стратегия, которая минимизирует величину $\max_{k \in K} \chi(k)$.

Вероятности $\omega(k)$, $k \in K$, выражаются суммами

$$\begin{aligned} \omega(k) &= \sum_{x \in X} \left(\sum_{k^* \in K \setminus \{k, 0\}} \alpha(x, k^*) \right) p_{X|K}(x | k) \\ &= \sum_{x \in X} \left(1 - \alpha(x, k) - \alpha(x, 0) \right) p_{X|K}(x | k) \\ &= 1 - \sum_{x \in X} \left(\alpha(x, k) + \alpha(x, 0) \right) p_{X|K}(x | k), \end{aligned}$$

а вероятности $\chi(k)$, $k \in K$, - суммами

$$\chi(k) = \sum_{x \in X} \alpha(x, 0) p_{X|K}(x | k).$$

Задача Вальда, обобщенная на случай многих состояний, выражается следующей парой двойственных задач.

Прямая задача:

$$\left. \begin{array}{l} \min c, \\ q(k) \left| \begin{array}{l} c - \sum_{x \in X} \alpha(x, 0) p_{X|K}(x|k) \geq 0, \quad k \in K, \\ \tau(k) \left| \begin{array}{l} \sum_{x \in X} (\alpha(x, k) + \alpha(x, 0)) p_{X|K}(x|k) \geq 1 - \varepsilon, \quad k \in K, \\ \alpha(x, 0) + \sum_{k \in K} \alpha(x, k) = 1, \quad x \in X, \\ \alpha(x, 0) \geq 0, \quad \alpha(x, k) \geq 0, \quad k \in K, \quad x \in X. \end{array} \right. \end{array} \right. \end{array} \right\} \quad (2.61)$$

Двойственная задача:

$$\left. \begin{array}{l} \max \left(\sum_{x \in X} t(x) + (1 - \varepsilon) \sum_{k \in K} \tau(k) \right), \\ \alpha(x, 0) \left| \begin{array}{l} t(x) + \sum_{k \in K} \tau(k) p_{X|K}(x|k) - \sum_{k \in K} q(k) p_{X|K}(x|k) \leq 0, \quad x \in X, \\ \alpha(x, k) \left| \begin{array}{l} t(x) + \tau(k) p_{X|K}(x|k) \leq 0, \quad x \in X, \quad k \in K, \\ \sum_{k \in K} q(k) = 1, \\ q(k) \geq 0, \quad \tau(k) \geq 0, \quad k \in K. \end{array} \right. \end{array} \right. \end{array} \right\} \quad (2.62)$$

По записи двойственной задачи очевидно, что для каждого наблюдения x величина $t(x)$ должна быть равной наименьшему из следующих $|K| + 1$ чисел: первые $|K|$ чисел - это числа $-\tau(k) p_{X|K}(x|k)$, а $(|K| + 1)$ -ое число равно $\sum_{k \in K} (q(k) - \tau(k)) p_{X|K}(x|k)$. В зависимости от того, какое именно из этих чисел будет наименьшим, определится решение: либо $\alpha(x, k) = 1$ для некоторого $k \in K$, либо $\alpha(x, 0) = 1$. Более точно, следует вычислить величину

$$\max_{k \in K} (\tau(k) p_{X|K}(x|k)) .$$

Если

$$\max_{k \in K} (\tau(k) p_{X|K}(x|k)) < \sum_{k \in K} (\tau(k) - q(k)) p_{X|K}(x|k),$$

то принимается решение $k^* = 0$. В противном случае

$$k^* = \operatorname{argmax}_{k \in K} (\tau(k) p_{X|K}(x|k)) .$$

Мы видим, что стратегия решения такой задачи Вальда не так проста, как стратегия решения в частном случае, рассмотренном в предыдущем подразделе. Вряд ли такую стратегию можно угадать, изобрести на основе одного лишь так называемого здравого смысла. В то же время точная формулировка требований к стратегии, выраженная в паре двойственных задач

(2.61), (2.62), сводит ее решение к несложному математическому упражнению.

2.4.6 Испытание сложных случайных гипотез

Задача испытания сложных случайных гипотез была сформулирована в подразделе 2.2.5. Состоит она в построении стратегии, которая минимизирует вероятность $\omega(z)$ ошибочного распознавания при наихудшем для этой стратегии вмешательстве $z \in Z$. Это значит, что стратегия должна минимизировать величину $\max_{z \in Z} \omega(z)$. Вероятность $\omega(z)$ ошибочного распознавания, которую обеспечивает стратегия $\alpha(x, k)$, $k \in K$, $x \in X$, при вмешательстве $z \in Z$, есть сумма

$$\begin{aligned}
 \omega(z) &= \sum_{k \in K} p_K(k) \sum_{x \in X} \left(\sum_{k^* \neq k} \alpha(x, k^*) \right) p_{X|K,Z}(x|k, z) \\
 &= \sum_{k \in K} p_K(k) \sum_{x \in X} \left(1 - \alpha(x, k) \right) p_{X|K,Z}(x|k, z) \\
 &= \sum_{k \in K} p_K(k) \left(1 - \sum_{x \in X} \alpha(x, k) p_{X|K,Z}(x|k, z) \right) \\
 &= 1 - \sum_{k \in K} p_K(k) \sum_{x \in X} \alpha(x, k) p_{X|K,Z}(x|k, z). \tag{2.63}
 \end{aligned}$$

Требование минимизации величины $\max_{z \in Z} \omega(z)$ можно выразить следующей парой двойственных задач.

Прямая задача:

$$\left. \begin{array}{l}
 \min c, \\
 \tau(z) \left| \begin{array}{l}
 c + \sum_{k \in K} p_K(k) \sum_{x \in X} \alpha(x, k) p_{X|K,Z}(x|k, z) \geq 1, \quad z \in Z, \\
 \sum_{k \in K} \alpha(x, k) = 1, \quad x \in X, \\
 \alpha(x, k) \geq 0, \quad k \in K, \quad x \in X.
 \end{array} \right.
 \end{array} \right\} \tag{2.64}$$

Двойственная задача:

$$\left. \begin{array}{l}
 \max \left(\sum_{x \in X} t(x) + \sum_{z \in Z} \tau(z) \right), \\
 \alpha(x, k) \left| \begin{array}{l}
 t(x) + \sum_{z \in Z} p_K(k) \tau(z) p_{X|K,Z}(x|k, z) \leq 0, \quad x \in X, \quad k \in K, \\
 \sum_{z \in Z} \tau(z) = 1, \\
 \tau(z) \geq 0, \quad z \in Z.
 \end{array} \right.
 \end{array} \right\} \tag{2.65}$$

По виду функции, которая должна максимизироваться в двойственной задаче, и ограничениям на числа $t(x)$ видно, что число $t(x)$ должно равняться наименьшему из следующих $|K|$ чисел,

$$- \sum_{z \in Z} p_K(k) \tau(z) p_{X|K,Z}(x|k, z), \quad k \in K.$$

Решение о состоянии $k \in K$ объекта зависит от того, какое из этих $|K|$ чисел наименьшее, или, что то же самое, какое из чисел

$$\sum_{z \in Z} p_K(k) \tau(z) p_{X|K,Z}(x|k, z), \quad k \in K,$$

наибольшее. Решение k^* должно быть равным

$$\operatorname{argmax}_{k \in K} \sum_{z \in Z} p_K(k) \tau(z) p_{X|K,Z}(x|k, z).$$

2.4.7 Испытание сложных неслучайных гипотез

Задача испытаний сложных неслучайных гипотез была сформулирована в подразделе 2.2.5. Эту задачу представляет следующая пара двойственных задач.

Прямая задача:

$$\left. \begin{array}{l} \min c, \\ \tau(z, k) \left| \begin{array}{l} c + \sum_{x \in X} \alpha(x, k) p_{X|K,Z}(x|k, z) \geq 1, \quad z \in Z, \quad k \in K, \\ \sum_{k \in K} \alpha(x, k) = 1, \quad x \in X, \\ \alpha(x, k) \geq 0, \quad k \in K, \quad x \in X. \end{array} \right. \end{array} \right\} \quad (2.66)$$

Двойственная задача:

$$\left. \begin{array}{l} \max \left(\sum_{x \in X} t(x) + \sum_{z \in Z} \sum_{k \in K} \tau(z, k) \right), \\ \alpha(x, k) \left| \begin{array}{l} t(x) + \sum_{z \in Z} \tau(z, k) p_{X|K,Z}(x|k, z) \leq 0, \quad x \in X, \quad k \in K, \\ \sum_{z \in Z} \sum_{k \in K} \tau(z, k) = 1, \\ \tau(z, k) \geq 0, \quad z \in Z, \quad k \in K. \end{array} \right. \end{array} \right\} \quad (2.67)$$

Из приведенной записи задач следует, что если для некоторого наблюдения $x \in X$ и некоторого $k^* \in K$ и всех остальных $k \in K \setminus \{k^*\}$ выполняется неравенство

$$\sum_{z \in Z} \tau(z, k^*) p_{X|K,Z}(x|k^*, z) > \sum_{z \in Z} \tau(z, k) p_{X|K,Z}(x|k, z),$$

то $\alpha(x, k^*) = 1$.

2.5 Заключительные замечания

Мы рассмотрели ряд небайесовских задач, который сейчас можно было бы и продолжить. Мы увидели, что известные к настоящему времени небайесовские задачи уже не предстают перед нами как распыленные и изолированные друг от друга островки знаний. Их объединяет то, что они формулируются как специфические задачи линейной оптимизации. При аккуратной формулировке этих задач в виде пары двойственных задач линейного программирования сравнительно легко выводится стратегия решения той или иной задачи, по крайней мере, с точностью до небольшого количества параметров. В силу этой общности сейчас можно заполнять новыми задачами те пробелы, которые до сих пор отделяли известные задачи друг от друга. Решение этих новых задач может оказаться возможным на основе теорем двойственности без привлечения изобретательности и интуиции. В ряде случаев, по крайней мере в рассмотренных нами примерах, обоснование той или иной стратегии не выходит за рамки математического упражнения.

Хотелось бы сформулировать еще один вывод, хотя мы к нему непосредственно и не вели при изложении лекции. Мы видим, что байесовские и небайесовские задачи не отделены друг от друга каким-то непроходимым разрывом. Байесовские задачи, так же как и небайесовские, можно сформулировать как специальные задачи линейной оптимизации. Правда, рассмотренные в первой лекции частные задачи были настолько просты, что для их решения не понадобилось использовать теорию двойственности. В этом смысле можно говорить, что байесовские и небайесовские задачи вместе образуют определенный класс специальных задач линейной оптимизации, и байесовские задачи в этом классе составляют простейшие задачи. Существенным является то, что добавление к этим простейшим задачам более сложных не привело к расширению класса стратегий. Решение ни одной из рассмотренных нами задач не привело к появлению какой-то новой стратегии, такой, которая бы не была стратегией решения некоторой байесовской задачи. Любая построенная нами стратегия решения небайесовской задачи так же, как и байесовская, представляла собой разбиение пространства вероятностей на выпуклые конусы.

2.6 Обсуждение

Лекция очень сложна. Почти все задачи, рассмотренные в лекции, оказались новыми для меня. До сих пор я знал только задачу Неймана-Пирсона и не особенно задумывался над обоснованием стратегии для ее решения. Теперь передо мной раскрылись обширные и ранее мне неизвестные горизонты небайесовских задач с объяснением, что все эти задачи являются задачами линейного программирования. Однако это мне ничего не объясняет, потому что я, к сожалению, незнаком с методами и техникой линейного программирования.

Невероятно! Разве Ты не изучал линейное программирование в школе?

Конечно, изучал, но совсем не в том духе, как об этом говорится в лекции. Основной упор делался на вычислительные процедуры, в частности, на симплекс-метод, который я знаю достаточно хорошо. Здесь же требуется совсем другая техника линейного программирования, не техника написания программ для решения задач на компьютере, а техника анализа задач с карандашом в руках.

Сейчас я вижу две области, в каждой из которых я мало разбираюсь, но которые, как видно, тесно связаны, и поэтому, сориентировавшись в одной из них, я смог бы проникнуть в другую. Но что я должен делать? Самостоятельно погрузиться в размышления о небайесовских задачах или в изучение обширной литературы по линейной оптимизации?

И то, и другое, но в первую очередь второе. Вне зависимости от того, будешь ли Ты в распознавании или нет, Ты должен хорошо ориентироваться в теоретических основах математического программирования, именно в теории, а не только в вычислительных методах. В любой области прикладной информатики, в которой Ты можешь оказаться в будущем, эти знания Тебе необходимы.

Однако мы не хотим оставлять Тебя наедине с Твоими проблемами в самом начале наших контактов. Задумайся над своим непониманием и определи ту часть в лекции, которая является основным препятствием для понимания. Конечно, это очень трудно, когда вся лекция в целом непонятна. Но все же попробуй.

Мне кажется, что я понимаю задачи линейного программирования в той канонической постановке, которая приведена в лекции. Я сам придумывал всякие механические или гидравлические модели, анализ которых сводился к решению задачи линейного программирования. Я понимал содержательный смысл переменных в терминах моих моделей. Однако я не смог понять содержательный смысл формально вводимых двойственных переменных так, чтобы они были как-то согласованы с исходными переменными в моей модели. Поэтому я не смог понять содержательный смысл двойственной задачи, а теоремы двойственности оказались уже чистой абстракцией. Для меня было бы полезно, если бы вы указали какой-то пример, в котором переменные как в прямой, так и в двойственной задаче имели бы ясный содержательный смысл.

Ну Ты великолепен! Человек, который так ясно может изложить свое непонимание, как правило, может добиться понимания без посторонней помощи. Уверены, что Ты бы со временем придумал эти двойственные переменные сам. Но мы Тебе поможем, и это будет быстрее.

Только, пожалуйста, пусть это не будет пример из экономики. Там ситуация с прямой задачей мне тоже понятна. Но когда начинают говорить о двойственных переменных, как о спросе, установлении цен, конъюнктуре, то у меня складывается впечатление, что мне стараются объяснить что-то, чего я не понимаю, с помощью чего-то, чего я тоже не понимаю.

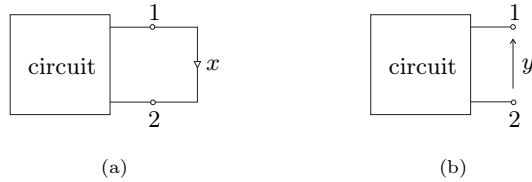


Figure 2.3 Electrical analogy. (a) Current x , (b) voltage y .

Будь спокоен. Мы так же, как и Ты, разбирались во всевозможных механических и гидравлических моделях, чтобы прочувствовать линейное программирование. Но сейчас мы объясним Тебе основные понятия на примере электрической модели, и только потому, что ее легче рисовать. Ничего сложного там не будет. От Тебя требуется лишь понимание, что мощность в электрической цепи есть произведение тока на напряжение. И тем не менее, ее рассмотрение отнимет у нас некоторое время.

Не сомневайтесь в моем терпении и желании разобраться в двойственных задачах.

Если Тебе будет что-то непонятно, без колебаний прерывай нас.

Числа в нашей модели будут представлены переменными токами в определенных ветвях электрической цепи и переменным напряжением между какими-то точками этой цепи. Например, величина x представлена током $x \cdot \sin t$ в ветви на рис. 2.3а. Стрелка на этой ветви показывает, какое направление тока считается положительным. Здесь возможны два случая: (а) Величина $x = 3$, например, соответствует переменному току $3 \sin t$, то есть, току, который протекает от точки 1 к точке 2, когда $\sin t \geq 0$, и току, который протекает от точки 2 к точке 1 при $\sin t \leq 0$. (б) Величина $x = -3$ будет представлена током $-3 \sin t$, который течет от точки 2 к точке 1 при $\sin t \geq 0$, и от точки 1 к точке 2 при $\sin t \leq 0$.

Подобным образом некоторые величины представлены напряжением, то есть разностью потенциалов между определенными точками цепи. Например, величина y будет представлена разностью $\varphi_1 - \varphi_2 = y \sin t$ потенциалов между точками 1 и 2 цепи на рис. 2.3б. Стрелка между точками 1 и 2 обозначает, что положительной считается именно разность $\varphi_1 - \varphi_2$, а не разность $\varphi_2 - \varphi_1$.

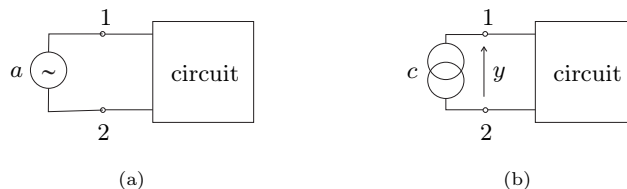


Figure 2.4 (a) Voltage sources. (b) Current sources.

Токи и напряжения возникают в цепи благодаря источникам энергии двух типов: (идеальному) источнику напряжения и (идеальному) источ-

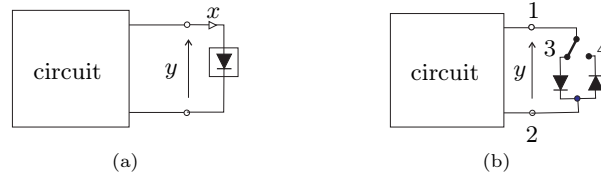


Figure 2.5 (a) Phase rectifier. (b) Possible implementation of a phase rectifier.

нику тока. Идеальный источник напряжения a (см. рис. 2.4a) производит напряжение $\varphi_1 - \varphi_2 = a \sin t$ независимо от нагрузки, подключенной к этому току. От этой нагрузки зависит только ток в ветви, подключенной к этому источнику. Идеальный источник тока c (см. рис. 2.4b) посылает во внешнюю ветвь ток, который равен $c \sin t$, независимо от нагрузки, подключенной к этому источнику. От нагрузки зависит только напряжение между точками подключения источника к цепи.

Следующий элемент, который нам будет нужен, это фазовый выпрямитель. При его включении в ветвь (см. рис. 2.5a) он препятствует появлению в этой ветви тока $x \sin t$ с отрицательным значением x и напряжения $\varphi_1 - \varphi_2 = y \sin t$ между точками 1 и 2 с положительным y .

Стоп! Извините, я вынужден прервать вас. Должен ли я понимать, что величины x и y сейчас представляются уже не синусоидальными функциями, а как-то иначе? Ведь если я включу в ветвь на рис. 2.5a выпрямитель (я имею в виду обыкновенный диод), то ток в этой ветви уже не может быть синусоидальным. Для напряжения на диоде справедливо то же самое.

Ты понял нас неправильно, но правильно сделал, что прервал нас. Речь здесь, конечно же, не идет об обыкновенном диоде. Ты же видишь, что мы настойчиво называем этот прибор фазовым выпрямителем, а не просто выпрямителем. Речь идет о приборе, который пропускает через себя синусоидальный ток $x \sin t$, но только, когда $x \geq 0$. Здесь нет ничего необычного. Это может быть реализовано с помощью цепи, представленной на рис. 2.5b. Управляемый ключ подключает точку 1 попеременно к точке 3, когда $\sin t \geq 0$, и к точке 4, когда $\sin t < 0$.

А те диоды, которые представлены на рис. 2.5b, это обыкновенные диоды или тоже что-то особенное?

Да это уже диоды в обычном смысле этого слова.

Понятно. Через ветвь на рис. 2.5b может проходить только синусоидальный ток $x \sin t$, $x \geq 0$, и только синусоидальное напряжение $y \sin t$, $y \leq 0$, может возникнуть на концах этой ветви.

Последний элемент, который понадобится для кристирования нашей модели, это трансформатор в обычном смысле этого слова, то есть некоторые

катушки на общем железном сердечнике. Каждая катушка характеризуется количеством витков и их направлением, которое может быть положительным и отрицательным. Например, трансформатор на рис. 2.6 состоит из 5 катушек. Количества витков в этих катушках слева направо равны соответственно 3, 3, 0, -4 и 1. Нулевое количество витков в третьей катушке обозначает, что провод этой катушки не охватывает сердечник вообще. Количество витков -4 в четвертой катушке обозначает, что витки этой катушки намотаны на сердечник в направлении, противоположном тому, как на этот сердечник намотаны первая, вторая и пятая катушки. Количества витков для этих катушек считаются положительными.

В силу взаимодействия магнитных полей, которые создают в сердечнике токи, протекающие в катушках, для этих токов x_1, x_2, \dots, x_m обязательно выполняется равенство

$$\sum_{i=1}^m x_i b_i = 0,$$

где b_i - количество витков в i -ой катушке, учитываемое со знаком + или - в зависимости от направления витков. К примеру, в катушках на рис. 2.6 могут возникнуть только такие токи x_1, x_2, \dots, x_5 , которые удовлетворяют ограничению

$$3x_1 + 3x_2 - 4x_4 + x_5 = 0.$$

Второе свойство трансформатора состоит в том, что напряжения на катушках пропорционально количеству витков. Не забывай, что это количество может быть как положительным, так и отрицательным. Для примера на рис. 2.6 напряжения y_1, y_2, y_3, y_4 однозначно определяются напряжением y_5 , то есть, $y_1 = 3y_5, y_2 = 3y_5, y_3 = 0, y_4 = -4y_5$.

Сейчас мы знаем все элементы, из которых мы построим модель задачи линейного программирования. Это - источники тока, источники напряжения, фазовые выпрямители и трансформаторы.

Мне кажется, что я понимаю эти элементы вполне отчетливо.

Тогда придумай схему из этих элементов, в которой могут возникать только такие токи $x_1 \sin t, x_2 \sin t, x_3 \sin t, x_4 \sin t, x_5 \sin t$, которые являются решением следующей системы равенств и неравенств

$$x_1 \geq 0, \quad x_2 \geq 0, \quad x_3 \geq 0, \quad (2.68)$$

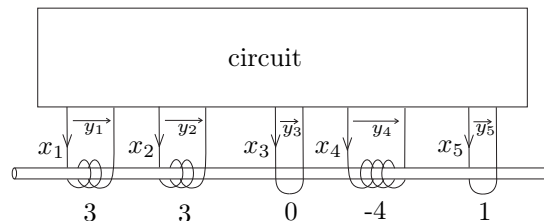


Figure 2.6 A transformer diagram for varying number of turns in the coil and directions of winding.

$$\left. \begin{aligned} 2x_1 - 3x_2 + 4x_3 + 5x_4 - 2x_5 &\geq c_1, \\ x_1 + 2x_2 + x_3 + x_5 &\geq c_2, \\ -x_1 + 3x_3 &\geq c_3, \\ x_1 + x_2 + x_3 + x_4 + x_5 &= c_4, \\ -x_1 + 2x_2 + 3x_3 + x_4 - x_5 &= c_5, \\ 3x_1 + 2x_2 - 4x_3 - 2x_4 + 3x_5 &= c_6. \end{aligned} \right\} \quad (2.69)$$

Эту систему можно понимать, как ограничения в некоторой задаче линейного программирования. Давай не будем пока принимать во внимание линейную функцию, которую надо минимизировать.

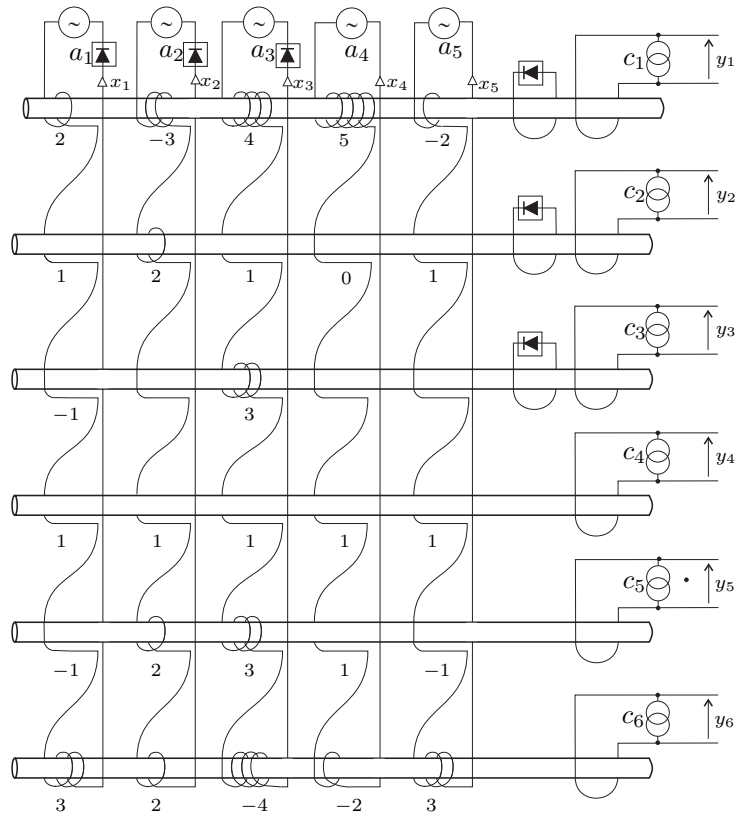


Figure 2.7 The equivalent electrical circuit for the linear programming task.

Это было ужасно. Ужасно трудно было справиться с тремя условиями в системе (2.69), которые имеют вид неравенств. Но в конечном итоге я с этим справился, и результат представлен на рис. 2.7. Приведенная на рисунке схема гарантирует, что токи x_1, \dots, x_5 в этой схеме будут удовлетворять условиям (2.68) и (2.69). Я уверен, что вы ожидали именно этот

результат, потому что он исключительно наглядный. Изображение схемы прямо по своей форме похоже на систему ограничений, которую эта схема должна реализовать. Моя схема состоит из шести трансформаторов, которые соответствуют шести ограничениям в системе (2.69), и пяти ветвей, токи в которых представляют пять переменных x_1, \dots, x_5 . Ветвь с номером $i = 1, \dots, 5$ (считая номера слева направо) содержит b_{ij} витков в j -ом трансформаторе, $j = 1, 2, \dots, 6$, причем это число в точности равно коэффициенту b_{ij} при переменной x_i в j -ом ограничении системы (2.69).

Условия (2.68) я легко обеспечил просто тем, что включил фазовый выпрямитель в первую, вторую и третью ветви, в силу чего по этим ветвям не могут протекать отрицательные токи. Три последние ограничения в системе (2.69), которые имеют вид равенств, я обеспечил тем, что добавил в 4-ый, 5-ый и 6-ой трансформаторы дополнительные катушки, по одной в каждый трансформатор, с количеством витков, равным -1 . Сами же катушки подключил к источникам токов, которые посылают в эти катушки ток c_j . В силу этого токи в катушках j -ого трансформатора, $j = 4, 5, 6$, удовлетворяют условиям

$$\sum_{i=1}^5 b_{ij}x_i - c_j = 0,$$

которые эквивалентны трем ограничениям-равенствам в системе (2.69).

Первые три ограничения-неравенства в системе (2.69) обеспечиваются следующим способом. Я поставил дополнительную катушку с произвольным, но отрицательным количеством витков в j -ый, $j = 1, 2, 3$, трансформатор и включил в эту катушку фазовый выпрямитель. Таким способом для $j = 1, 2, 3$ обеспечиваются равенства

$$\sum_{i=1}^5 x_i b_{ij} - x_{0j} - c_j = 0, \quad x_{0j} \geq 0,$$

которые эквивалентны первым трем ограничениям-неравенствам в системе (2.69). И это, кажется, все.

Ты сделал все замечательно и даже забежал немного вперед. На рис. 2.7 мы видим еще источники напряжения a_1, \dots, a_5 , которые фактически не нужны для обеспечения ограничений (2.68) и (2.69). Зачем же Ты их ввел?

Это только интуиция и больше ничего. Мне кажется, что в силу этих источников напряжения в схеме установятся не любые токи, удовлетворяющие ограничениям (2.68) и (2.69), но именно те токи, которые минимизируют сумму $\sum_{i=1}^5 a_i x_i$. Действительно, сумма $\sum_{i=1}^5 a_i x_i$ - это мощность, которую источники напряжений a_1, \dots, a_5 потребляют из окружающей схемы. Однако, данные источники являются источниками энергии, а ее потребителями. Они стремятся с максимальной скоростью отдавать энергию во

внешнюю схему. Эта скорость как раз есть мощность $-\sum_{i=1}^5 a_i x_i$. Это значит, пока что интуитивно, что в схеме на рис. 2.7 устанавливаются токи, которые максимизируют скорость $-\sum_{i=1}^5 a_i x_i$ отдачи энергии в схему.

Ты убедишься в правильности своей интуиции чуть позже.

Но теперь я бы хотел вернуться к вопросу, ради которого мы придумали эту схему. Где здесь двойственные переменные, двойственная задача и теоремы двойственности?

Насколько мы Тебя уже знаем, сейчас Ты это увидишь. Ты проанализировал токи x_1, \dots, x_5 , возникающие в Твоей схеме. Но в схеме есть еще и напряжения, и не только напряжения на источниках напряжения a_1, \dots, a_5 , которые посылают в схему токи x_1, \dots, x_5 , но и напряжения на источниках токов c_1, \dots, c_6 . Обозначим их y_1, \dots, y_6 .

Я понял! Как я это сам не заметил! Это и есть двойственные переменные.

Ты можешь доказать это?

Конечно! В силу свойств трансформатора на катушке j -го трансформатора, включенной в i -ую ветвь, возникает напряжение $b_{ij}y_j$, где y_i - это напряжение на источнике тока c_j . Известно, что сумма напряжений в любом контуре равна нулю. Отсюда немедленно следует, что напряжения на катушках, включенных в пятую ветвь, удовлетворяют равенству

$$-2y_1 + y_2 + y_4 - y_5 + 3y_6 = a_5 .$$

Аналогично, для напряжений на четвертой ветви выполняется

$$5y_1 + y_4 + y_5 - 2y_6 = a_4 .$$

В первую, вторую и третью ветви включены выпрямители, напряжения на которых не могут быть положительными. Поэтому сумма $\sum_{j=1}^6 b_{ij}y_j$ напряжений на катушках i -ой ветви, $i = 1, 2, 3$, не может превышать напряжение a_i . Для первой ветви выполняется

$$2y_1 + y_2 - y_3 + y_4 - y_5 + 3y_6 \leq a_1 ,$$

для второй -

$$-3y_1 + 2y_2 + y_4 + 2y_5 + 2y_6 \leq a_2$$

и для третьей -

$$4y_1 + y_2 + 3y_3 + y_4 + 3y_5 - 4y_6 \leq a_3 .$$

Раньше я поставил дополнительные катушки с выпрямителями на первый, второй и третий трансформаторы с единственной целью обеспечить три

условия-неравенства в системе (2.69). Сейчас же я вижу, что наличие этих же катушек приводит к тому, что напряжения y_1, y_2, y_3 на соответствующих источниках тока не могут стать отрицательными.

Фактически произошло вот что. При составлении схемы, которая должна генерировать токи x_1, x_2, x_3, x_4, x_5 , удовлетворяющие условиям прямой задачи

$$\left. \begin{aligned} x_1 \geq 0, \quad x_2 \geq 0, \quad x_3 \geq 0, \\ 2x_1 - 3x_2 + 4x_3 + 5x_4 - 2x_5 \geq c_1, \\ x_1 + 2x_2 + x_3 \quad \quad \quad x_5 \geq c_2, \\ -x_1 \quad \quad + 3x_3 \quad \quad \quad \geq c_3, \\ x_1 + x_2 + x_3 + x_4 + x_5 \geq c_4, \\ -x_1 + 2x_2 + 3x_3 + x_4 - x_5 \geq c_5, \\ 3x_1 + 2x_2 - 4x_3 - 2x_4 + 3x_5 \geq c_6, \end{aligned} \right\} \quad (2.70)$$

я, даже не сознавая это, составил схему, генерирующую напряжения $y_1, y_2, y_3, y_4, y_5, y_6$, которые удовлетворяют условиям двойственной задачи

$$\left. \begin{aligned} y_1 \geq 0, \quad y_2 \geq 0, \quad y_3 \geq 0, \\ 2y_1 + y_2 - y_3 + y_4 - y_5 + 3y_6 \leq a_1, \\ -3y_1 + 2y_2 \quad \quad + y_4 + 2y_5 + 2y_6 \leq a_2, \\ 4y_1 + y_2 + 3y_3 + y_4 + 3y_5 - 4y_6 \leq a_3, \\ 5y_1 \quad \quad \quad + y_4 + y_5 - 2y_6 = a_4, \\ -2y_1 + y_2 \quad \quad + y_4 - y_5 + 3y_6 = a_5. \end{aligned} \right\} \quad (2.71)$$

Я начинаю предполагать, что любая физическая модель прямой задачи линейного программирования есть одновременно и моделью двойственной задачи. Для этого требуется, чтобы система представляла два вида физических величин, произведение которых есть энергия или мощность, и содержала в себе два вида источников энергии или мощности. Такими двумя видами физических величин могут быть ток и напряжение, как в нашем случае, сила и координата в механических системах, давление и скорость жидкости или газа в гидравлических или пневматических моделях. Все это - примеры взаимно двойственных переменных. Правда, я до сих пор не доказал сам себе, что составленная мною электрическая схема решает пару двойственных задач. Моя уверенность в этом основана лишь на интуиции. Можете ли вы помочь мне разобраться в этом до конца?

Можем, причем с большим удовольствием. Однако сейчас, когда Ты уже немного освоился с возможным физическим содержанием прямых и двойственных переменных, позволь нам ненадолго перейти на более отвлеченный уровень. Мы докажем одно вспомогательное утверждение в абстрактной форме, а потом посмотрим, что оно выражает в нашей схеме.

Пусть $x = (x_1, x_2, \dots, x_m)$ - вектор, удовлетворяющий условиям

$$\sum_{i=1}^m b_{ij} x_i \geq c_j, \quad j = 1, \dots, n^*, \quad (2.72)$$

$$\sum_{i=1}^m b_{ij}x_i = c_j, \quad j = n^* + 1, n^* + 2, \dots, n, \quad (2.73)$$

$$x_i \geq 0, \quad i = 1, \dots, m^*, \quad (2.74)$$

где m, n, m^*, n^* - целые числа, $m^* \leq m$, $n^* \leq n$. Пусть также $y = (y_1, y_2, \dots, y_n)$ - вектор, удовлетворяющий условиям

$$\sum_{j=1}^n b_{ij}y_j \leq a_i, \quad i = 1, 2, \dots, m^*, \quad (2.75)$$

$$\sum_{j=1}^n b_{ij}y_j = a_i, \quad i = m^* + 1, m^* + 2, \dots, m, \quad (2.76)$$

$$y_j \geq 0, \quad j = 1, \dots, n^*. \quad (2.77)$$

В таком случае справедливо, что

$$\sum_{i=1}^m a_i x_i \geq \sum_{j=1}^n c_j y_j. \quad (2.78)$$

Доказательство этого утверждение достаточно просто. Из неравенств (2.72) и (2.77) следует, что

$$\sum_{j=1}^{n^*} \left(\sum_{i=1}^m b_{ij}x_i - c_j \right) y_j \geq 0.$$

Из равенства (2.73) следует, что

$$\sum_{j=n^*+1}^n \left(\sum_{i=1}^m b_{ij}x_i - c_j \right) y_j = 0.$$

Если сложить два последние выражения, получается

$$\sum_{j=1}^n \left(\sum_{i=1}^m b_{ij}x_i - c_j \right) y_j \geq 0. \quad (2.79)$$

Из неравенств (2.75) и (2.74) следует, что

$$\sum_{i=1}^{m^*} \left(a_i - \sum_{j=1}^n b_{ij}y_j \right) x_i \geq 0,$$

а из равенства (2.76) -

$$\sum_{i=m^*+1}^m \left(a_i - \sum_{j=1}^n b_{ij}y_j \right) x_i = 0.$$

Суммируя два последние выражения, получаем

$$\sum_{i=1}^m \left(a_i - \sum_{j=1}^n b_{ij} y_j \right) x_i \geq 0, \quad (2.80)$$

а суммируя неравенства (2.79) и (2.80), получаем

$$\sum_{i=1}^m a_i x_i - \sum_{j=1}^n c_j y_j \geq 0,$$

что является лишь другой формой неравенства (2.78), которое требовалось доказать.

В силу того, что все ограничения (2.72)–(2.77), касающиеся схемы на рис. 2.7, выполняются, выполняется и неравенство (2.78).

Для нашей схемы неравенство (2.78) можно легко понять и без доказательства. Суммарная мощность $\sum_{j=1}^n c_j y_j$, передаваемая источниками токов c_j , $j = 1, \dots, n$, в схему, не может быть больше суммарной мощности $\sum_{i=1}^m a_i x_i$, получаемой источниками напряжения a_i , $i = 1, \dots, m$, из сети. Ведь в сети нет никаких элементов, где бы могла накапливаться или рассеиваться энергия.

На основании чисто физических соображений отношение (2.78) можно существенно усилить в виде равенства

$$\sum_{i=1}^m a_i x_i = \sum_{j=1}^n c_j y_j, \quad (2.81)$$

которое эквивалентно первой теореме двойственности (см. теорему 2.1). Вы говорили, что ее доказательство достаточно сложно. Равенство (2.81) обязательно выполняется, так как иначе бы нарушался закон сохранения энергии. Равенство (2.81), таким образом подтверждает мою интуитивную догадку, что в схеме могут быть только такие токи x_i , которые соответствуют решению прямой задачи линейного программирования, и только такие напряжения y_i , которые соответствуют решению двойственной задачи.

Осталось сделать только небольшое усилие, чтобы понять, что обозначает вторая теорема двойственности в Твоей схеме.

Это так ясно, что любое доказательство может эту ясность только затуманить.

Если через выпрямитель протекает ненулевой ток, то напряжение на нем обязательно равно нулю. Если напряжение на выпрямителе не равно нулю, то ток через этот выпрямитель обязательно равен нулю.

Это, кажется все, чем мы можем сегодня помочь Тебе в неформальном понимании двойственных задач.

Большое спасибо. Я об этом еще много буду думать. Я попросил бы вас сейчас помочь мне понять уместность или неуместность байесовских или небайесовских методов в определенной ситуации.

Валяй!

Я понял из лекции, что в определенных практических ситуациях понятие штрафа за ошибку не имеет смысла и это одна из причин для формализации этих ситуаций в виде небайесовских задач. Другая причина для их формулировки возникает при отсутствии тех или иных априорных вероятностей. В рассмотренных в лекции примерах представлены ситуации, когда неизвестно ни то, ни другое. Мне бы хотелось рассмотреть ситуацию, когда статистическая модель распознаваемого объекта полностью известна, но неизвестна или даже имеет смысла функция потерь. Зачем в этом случае формулировать задачу непременно, как небайесовскую? Почему нельзя решать задачу на минимум вероятности ошибки, просто не произнося слов "штраф" и "риск"? Ведь понятие вероятности ошибки само по себе понятно, и без ссылки на тот факт, что это частный случай риска при определенной функции потерь.

Конечно, Ты можешь решать задачу на минимум вероятности ошибки. Но при этом Ты должен знать, что этому формальному критерию соответствует что-то вразумительное в терминах Твоей прикладной задачи. Не можем утверждать строго, но все же скажем, что если в той или иной прикладной задаче функция потерь не имеет смысла, то как правило, и вероятность ошибки не имеет смысла, либо выражает нечто совсем несущественное в данном приложении.

Рассмотрим в качестве примера задачу ранней диагностики онкологических заболеваний. Ты, повидимому, знаешь, как важна эта задача, так как при ранней диагностике больного еще можно спасти. К сожалению, эта задача и достаточно трудная. Представь себе, что Ты уже стал крупным менеджером, или министром здравоохранения, или богатым филантропом и мы пришли к Тебе с алгоритмом диагностики раковых заболеваний и утверждаем, что этот алгоритм дает ошибочный ответ не чаще, чем в 1% случаев. Ты знаешь, что эксперт-медик ошибается чаще, и немедленно покупаешь наш алгоритм. Ты надеешься (ВНИМАНИЕ! СЛЕДУЕТ ОШИБКА!), что, пользуясь этим алгоритмом, Ты сможешь спасти 99% людей, которые бы погибли без Твоей помощи. Твой энтузиазм, однако, быстро испаряется, когда Ты узнаешь, что наш алгоритм действует абсолютно примитивно. Он принимает решение, что все пациенты здоровы, даже не анализируя их симптомов, а 1% ошибочных решений достигается лишь благодаря тому, что больных-то в сто раз меньше, чем здоровых.

Ты видишь, что нам удалось провести Тебя на мякине, потому что Ты свои правильные, но интуитивно понимаемые требования выразил в виде неадекватного формального критерия. В данном случае качество стратегии характеризует не одна вероятность ошибки, а две вероятности. Эти

вероятности формально можно складывать, так как это просто вещественные числа. Однако их сумме не соответствует ничего вразумительного в терминах Твоего конкретного приложения. Ты должен формулировать задачу, чтобы подобные неинтерпретируемые величины не появлялись в ней вообще.

Например, если в Твоей задаче требуется, чтобы вероятность пропуска больного не превышала 1%, а вероятность ложной тревоги о здоровом человеке была, какой уж будет, лишь бы поменьше, то уместно Твою задачу формализовать как задачу Неймана-Пирсона. Если Тебе нужно, чтобы как вероятность пропуска больного, так и вероятность ложной тревоги не превышала 1%, то следует решать задачу Вальда. Да мало ли какие здесь возможны разумные постановки. Ведь Ты сейчас уже владеешь общим подходом к решению небайесовских задач, и Ты не станешь втискивать приложение в тесные рамки своих знаний, а будешь расширять свои знания по мере соприкосновения с прикладными задачами. Например, в Твоем приложении может потребоваться, чтобы вероятности обеих ошибок не превышали 1%, условная вероятность отказа при наблюдении больного человека не превышала 2%, а условная вероятность отказа при наблюдении здорового человека была как можно меньше. Хотя такую конкретную задачу мы не рассматривали в лекции, надеемся, что она не застанет Тебя врасплох и Ты с ней справишься.

Конечно, теперь уже справлюсь. Кстати, о вальдовскрй задаче. Я согласен, что стратегия, выведенная формально для случая многих состояний, априори далеко не очевидна. Вряд ли ее можно было бы изобрести на основе одной лишь интуиции. Однако неочевидность этой стратегии является в каком-то смысле и ее недостатком. Очень рискованно пользоваться результатами, которые интуитивно никак не ожидалось. Уж слишком большая вера должна быть в безупречность логических выкладок, которые к этому результату привели. Я ожидал здесь совсем другую стратегию. Позвольте мне ее описать и задать затем вопрос, где же я ошибся.

Формально обоснованный алгоритм можно представить в следующей форме. Стратегия зависит от $2|K|$ параметров $\tau(k)$, $q(k)$, $k \in K$, располагая которыми должны вычисляться следующие величины

$$\gamma_k(x) = \frac{\tau(k) p_{X|K}(x|k)}{\sum_{k \in K} \tau(k) p_{X|K}(x|k)},$$

которые напоминают апостериорные вероятности состояний k при условии наблюдения x . Затем самая большая из этих величин сравнивается с порогом $\theta(x)$. Этот порог определяется по формуле

$$\theta(x) = 1 - \frac{\sum_{k \in K} q(k) p_{X|K}(x|k)}{\sum_{k \in K} \tau(k) p_{X|K}(x|k)}$$

и различен для различных наблюдений. Именно здесь - самый трудный для понимания момент. Ведь в байесовской задаче с отказом этот порог был одним и тем же для всех наблюдений. Конечно, это не довод, но я ожидал, что стратегия будет иметь вид

$$k^* = \begin{cases} \operatorname{argmax}_{k \in K} \gamma_k(x), & \text{если } \max_{k \in K} \gamma_k(x) > (1 - \delta), \\ 0, & \text{если } \max_{k \in K} \gamma_k(x) < (1 - \delta). \end{cases}$$

Где я ошибся?

На этот раз Ты нас тоже удивил. Ты задал бессмысленный вопрос. Твое предложение о некоторой стратегии не может ни правильным, ни неправильным. Правильным или неправильным может быть лишь утверждение о соответствии некоторого алгоритма и некоторой задачи. Отсутствие одной составляющей в этой паре еще не есть утверждение, о котором можно спорить, правильное оно или нет. Только после того, как Ты сформулируешь задачу, на решение которой претендует предлагаемый алгоритм, можно ставить вопрос, ошибся ли Ты или нет.

А что же мне делать, если я знаю какой-то алгоритм, но не знаю задачу, которую он решает? Ведь такая ситуация вовсе не редкая в прикладном распознавании.

В этом случае Ты должен молчать, пока Ты не будешь в состоянии сформулировать, для решения какой задачи Твой алгоритм предназначен. Попробуем вместе сделать ретроспективный анализ Твоего алгоритма и найти ту задачу, которую он решает.

Стратегия, которую Ты предлагаешь, была бы достигнута, если бы на переменные $\tau(k)$ и $q(k)$, $k \in K$, было наложено ограничение, что их отношение

$$\frac{\sum_{k \in K} q(k) p_{X|K}(x|k)}{\sum_{k \in K} \tau(k) p_{X|K}(x|k)} = \delta$$

одно и то же для всех $x \in X$. Это ограничение можно выразить в линейной форме

$$-\sum_{k \in K} q(k) p_{X|K}(x|k) + \delta \sum_{k \in K} \tau(k) p_{X|K}(x|k) = 0 .$$

Включим эти линейные ограничения в двойственную задачу (2.62) и получим новую двойственную задачу.

Двойственная задача:

$$\left. \begin{aligned}
 & \max \left(\sum_{x \in X} t(x) + (1 - \varepsilon) \sum_{k \in K} \tau(k) \right) \\
 & t(x) + \sum_{k \in K} \tau(k) p_{X|K}(x|k) - \sum_{k \in K} q(k) p_{X|K}(x|k) \leq 0, \quad x \in X, \quad (c) \\
 & t(x) + \tau(k) p_{X|K}(x|k) \leq 0, \quad x \in X, \quad k \in K, \\
 & \delta \sum_{k \in K} \tau(k) p_{X|K}(x|k) - \sum_{k \in K} q(k) p_{X|K}(x|k) = 0, \quad x \in X, \quad (a) \\
 & \sum_{k \in K} q(k) = 1, \quad (b) \\
 & q(x) \geq 0, \quad \tau(k) \geq 0.
 \end{aligned} \right\} (2.82)$$

Эта задача отличается от задачи (2.62), рассмотренной в лекции, наличием $|X|$ дополнительных ограничений (2.82a). Исходная задача таким образом несколько деформировалась. Мы построим прямую задачу, для которой задача (2.82) является двойственной, чтобы увидеть, в чем же состояла эта деформация. Сначала эквивалентно преобразуем задачу (2.82). Во первых, на основе (2.82a) ограничения (2.82c) могут быть записаны в виде

$$t(x) + (1 - \delta) \sum_{k \in K} \tau(k) p_{X|K}(x|k) \leq 0, \quad x \in X. \quad (2.83)$$

Во вторых, пара ограничений (2.82a) и (2.82b) эквивалентна паре

$$\left. \begin{aligned}
 & \delta \sum_{k \in K} \tau(k) p_{X|K}(x|k) - \sum_{k \in K} q(k) p_{X|K}(x|k) = 0, \quad k \in K \\
 & \sum_{k \in K} \tau(k) = \frac{1}{\delta}.
 \end{aligned} \right\} (2.84)$$

Действительно, из ограничения (2.82a) следует

$$\delta \sum_{x \in X} \sum_{k \in K} \tau(k) p_{X|K}(x|k) = \sum_{x \in X} \sum_{k \in K} q(k) p_{X|K}(x|k).$$

На основе ограничения (2.82b) и равенства $\sum_{x \in X} p_{X|K}(x|k) = 1$ получаем

$$\delta \sum_{k \in K} \tau(k) = \sum_{k \in K} q(k) = 1.$$

Подставим это выражение в функцию, которую следует максимизировать, и можем записать

$$\sum_{x \in X} t(x) + \frac{1 - \varepsilon}{\delta}.$$

Максимизация этой функции эквивалентна максимизации функции

$$\sum_{x \in X} t(x).$$

В конечном итоге получаем эквивалентное представление задачи (2.82) в виде

$$\left. \begin{aligned} & \max \left(\sum_{x \in X} t(x) \right) \\ & t(x) + (1 - \delta) \sum_{k \in K} \tau(k) p_{X|K}(x|k) \leq 0, \quad x \in X, \\ & t(x) + \tau(k) p_{X|K}(x|k) \leq 0, \quad x \in X, \quad k \in K, \\ & \delta \sum_{k \in K} \tau(k) p_{X|K}(x|k) - \sum_{k \in K} q(k) p(x|k) = 0, \quad x \in X, \quad (a) \\ & \sum_{k \in K} \tau(k) = \frac{1}{\delta}, \\ & \tau(k) \geq 0, \quad q(k) \geq 0, \quad k \in K. \end{aligned} \right\} \quad (2.85)$$

Из этого представления видно, что ограничение (2.85a) лишнее. Действительно, если переменные $t(x)$, $x \in X$, $q(k)$, $k \in K$, и $\tau(k)$, $k \in K$, удовлетворяют всем ограничениям, за исключением (2.85a), то изменением одних лишь переменных $q(k)$, $k \in K$, можно добиться, чтобы выполнялись и ограничения (2.85a). Например, переменные $q(k)$ можно сделать равными $\delta \tau(k)$. Таким образом, задачу (2.85) можно выразить как

$$\left. \begin{aligned} & \max \left(\sum_{x \in X} t(x) \right) \\ \alpha(x, 0) \quad & \left. \begin{aligned} & t(x) + (1 - \delta) \sum_{k \in K} \tau(k) p_{X|K}(x|k) \leq 0, \quad x \in X, \\ & t(x) + \tau(k) p_{X|K}(x|k) \leq 0, \quad x \in X, \quad k \in K, \end{aligned} \right\} \\ c \quad & \left. \begin{aligned} & \sum_{k \in K} \tau(k) = \frac{1}{\delta}, \\ & \tau(k) \geq 0. \end{aligned} \right\} \end{aligned} \right\} \quad (2.86)$$

Появились тревожные неожиданности. Меня не удивляет, что переменные $q(k)$, $k \in K$, исчезли в новой задаче. Я же сам потребовал, чтобы порог, с которым сравнивались "апостериорные вероятности" $\gamma(k)$, $k \in K$, не зависели от наблюдения x , а в прежней задаче эта зависимость реализовалась именно через коэффициенты $q(k)$, $k \in K$. Но меня тревожит, что из задачи исчез параметр ε , который определял предельно допустимые значения вероятностей ошибок. Таким образом, уже сейчас ясно, что мой алгоритм не гарантирует вероятность ошибки, меньшую, чем порог ε , так он его просто игнорирует. Значит, мой алгоритм никуда не годится?

Не будем торопиться с выводами и продолжим наш анализ. Задача (2.86)

является двойственной по отношению к задаче

$$\left. \begin{array}{l} \min \left(\frac{c}{\delta} \right) \\ \tau(k) \left| c + \sum_{x \in X} \alpha(x, k) p_{X|K}(x|k) + (1-\delta) \sum_{x \in X} \alpha(x, 0) p_{X|K}(x|k) \geq 0, \quad k \in K, \right. \\ t(x) \left| \alpha(x, 0) + \sum_{k \in K} \alpha(x, k) = 1, \quad x \in X, \right. \\ \alpha(x, 0) \geq 0, \quad \alpha(x, k) \geq 0, \quad x \in X, \quad k \in K. \end{array} \right\} \quad (2.87)$$

Это значит, что предложенный Тобой алгоритм минимизирует число $\max_{k \in K} (\omega(k) + \delta \chi(k))$, где

$$\omega(k) = \sum_{x \in X} \left(\sum_{k' \neq k; k' \neq 0} \alpha(x, k') \right) p_{X|K}(x|k)$$

есть условная вероятность ошибки при условии, что объект находится в состоянии k , и

$$\chi(k) = \sum_{x \in X} \alpha(x, 0) p_{X|K}(x|k)$$

есть условная вероятность решения `not known` при условии, что объект находится в состоянии k . Параметр δ в Твоей задаче может пониматься, как отношение штрафа за отказ к штрафу за ошибку. Таким образом, мы закончили ретроспективный анализ Твоего алгоритма, и теперь Тебе судить, насколько задача, которую этот алгоритм решает, уместна в том или ином приложении.

Ты можешь здесь прийти к разным решениям, но важно то, что вместо сравнения алгоритмов мы предлагаем Тебе сравнивать задачи, которые эти алгоритмы решают. Сравнивать алгоритмы между собой можно только по отношению к одной и той же фиксированной задаче, на решение которой они оба претендуют. Если же задачи, решаемые алгоритмами, разные, то эти алгоритмы невозможно сравнивать. А если для одного из сравниваемых алгоритмов неизвестна задача, которую он решает, то вопрос о его сравнении с другим алгоритмом вообще отпадает. Такого алгоритма как бы и нет.

Сравнивая алгоритмы, Ты склонен был отдать предпочтение своему алгоритму в силу его большей прозрачности, понятности. Прозрачным и понятным должен быть не алгоритм решения задачи, а задача, которую этот алгоритм решает. Хорошо, конечно, когда алгоритм оказывается простым для понимания. Но основное требование к нему состоит в том, что он должен обоснованно выводиться из недвусмысленно сформулированной задачи. А окажется он простым или нет, то как уж получится. При анализе Твоего алгоритма именно на уровне задачи стало видно, что понятность Твоего алгоритма в высшей степени обманчива. Мы имеем в виду коэффициент δ , который появился именно в формулировке задачи, но был тщательно

скрыт в алгоритме. Дело не только в том, известно ли его значение или нет. Существенно то, что само его появление предполагает, что штрафы за ошибку и за отказ измеряются в одной и той же системе единиц. В некоторых приложениях само это предположение бессмысленно. Так, например, в медицинском приложении штраф за отказ от диагностики пациента - это просто стоимость дополнительных, возможно, и дорогих обследований пациента, а штраф за ошибку - это его жизнь, которая измеряется совсем в других единицах.

Уже на этом примере мы видим, что сравнивать задачи сами по себе тоже нельзя. Сравнивать задачи можно только по отношению к некоторому фиксированному приложению. Таким образом, нужно идти от приложения к формулировке задачи, а затем к выводу алгоритма ее решения, не пропуская этапа формулировки задачи, и уж конечно, не двигаясь в обратном направлении. Как говорят прикладные математики в России, прикладной математик должен быть подобен человеку, который стоит перед закрытой дверью и мастерит ключ, чтоб ее открыть, а не выпиливать сначала ключ и только потом искать, которую можно будет этим ключом открыть. А они знают, что говорят.

Не могу сказать, что я решительно согласен с последним высказыванием, хотя и признаю, что оно очень выразительно.

Возможно, и Ты тоже прав.

После этой лекции я увидел, что в категорическом утверждении лекции 1 о детерминированном характере стратегий появилась трещина, хотя и мало заметная. Байесовские стратегии делят пространство вероятностей на выпуклые конусы так, что каждая точка этого пространства принадлежит одному и только одному конусу. Фактически именно это доказывает теорема 1.1 о детерминированном характере байесовских стратегий.

Ситуация с небайесовскими стратегиями несколько иная. Я увидел, что для всех точек внутри конусов решение детерминированное. В то же время ничего не было сказано, какое решение должно приниматься для точек, лежащих точно на границах двух конусов. Для байесовских стратегий ясно, что эти точки можно произвольно присоединить к одному из конусов и таким образом оставаться в рамках детерминированных стратегий. Мне ясно, что такое утверждение для небайесовских стратегий неверно. Более того, мне ясно, что точку на границе нельзя полностью отнести ни к одному из конусов. Ее следует как бы разделить и частично отнести к одному конусу и частично - к другому. Именно такое рандомизированное решение может оказаться лучше, чем любое детерминированное. Я, конечно, не думаю, что такая рандомизация чудесным образом улучшит стратегию по сравнению с детерминированной, потому что речь идет о рандомизации в очень узком слое точек. Для меня сейчас важно, что детерминированный характер стратегий, который в первой лекции прозвучал, как своего рода категорический императив, уже перестал быть таким. В нем появилась

трещина. Можно ожидать, что при дальнейшем расширении класса задач статистических решений эта трещина может расширяться и рандомизированные стратегии станут преобладать над детерминированными, как это имеет место, например, в антагонистических играх.

Такая ситуация, мне кажется, возникнет, когда функция потерь задана не полностью, а с точностью до неслучайного параметра, то есть, вмешательства. В отличие от вмешательства в задачах Линника, где оно влияло на статистическую модель распознаваемого объекта, здесь оно влияет на то, как результаты распознавания оцениваются. Эта ситуация имеет место, когда требуется решить задачу распознавания в ситуации, когда функция штрафов не полностью известна, что-то о ней известно, а что-то - нет. В этой ситуации следует построить стратегию, которая окажется достаточно хорошей при любой функции потерь из заданного класса. Здесь может найтись такая задача, решением которой будет рандомизированная стратегия, принимающая случайное решение в пользу того или иного состояния объекта в соответствии с апостериорными вероятностями этих состояний.

Мы видим, что Ты настойчиво ищешь дверь, которую бы можно было открыть неким определенным ключом. Ты ищешь задачу, которая бы решалась определенным методом, который Тебе почему-то нравится.

Я не могу сказать, что мне это нравится. Но я вижу, что многие этим методом пользуются, не располагая его обоснованием. Следовательно, есть в этом что-то, что понимается на интуитивном уровне, но еще не формализовалось достаточно четко. Не вижу ничего плохого в том, что в данном случае я хотел бы идти от известного метода к задаче, а не наоборот.

Может быть, Ты прав.

Январь 1997.

2.7 Библиографические заметки

Математическим аппаратом данной лекции является теория двойственности в линейном программировании, глубоко исследованная в математической литературе [Kuhn and Tucker, 1950; Zuchovickij and Avdejeva, 1967].

Мы не нашли до сих пор достаточно общего описания класса небайесовских задач. Собственно, это было главной мотивацией для разработки данной лекции. В известной нам математической литературе вопросы рассматриваются с настолько общих позиций, что наши задачи распознавания там просто теряются. Так например, утверждается ([Wald, 1950]), повидимому, совершенно справедливо, что предположение о конечном множестве наблюдений является серьезным ограничением, и поэтому редко исследуется в теории статистических решений.

С другой стороны, в теории статистических решений отчетливо различаются ситуации, когда оцениваемый параметр случайный или неслучай-

ный [Neuman, 1962]. В распознавании образов это понимание еще не укоренилось достаточно прочно.

Практики зачастую решают ту или иную небайесовскую задачу, не осознавая этого. Например, распознающий алгоритм конструируется, как решение определенной байесовской задачи. Затем, уже в процессе опытной эксплуатации этого алгоритма подстраиваются некоторые коэффициенты в алгоритме так, чтобы в каждом состоянии распознаваемого объекта ошибки допускались приблизительно с одинаковой частотой. Настраивая алгоритм таким образом, практик неосознанно решает ту или иную небайесовскую задачу, в приведенном примере - минимаксную. Две статьи [Schlesinger, 1979b; Schlesinger, 1979a] посвящены формализации таких приемов, применяемых на практике. Эти статьи служили исходным материалом для данной лекции, который затем обогащался уже в процессе лекций.

На отдельные, ставшие классическими, небайесовские задачи должны быть сделаны следующие ссылки. Задача Неймана-Пирсона опубликована в [Neuman and Pearson, 1928; Neuman and Pearson, 1933]. Более понятное, хотя и менее строгое обоснование неймановских стратегий содержится в учебнике по статистике [Lehmann, 1959]. Минимаксная задача описана в [Wald, 1950]. Описанная в лекции задача Вальда составляет фрагмент вальдовского последовательного анализа [Wald, 1947; Wald and Wolfowitz, 1948]. Статистические задачи с неслучайными мешающими параметрами, известные также, как задачи проверки сложных статистических гипотез, исследованы Линником [Linnik, 1966].

Глава 3

Две статистические модели распознаваемого объекта

Условное распределение вероятностей $p_{X|K} : X \times K \rightarrow \mathbb{R}$ наблюдений $x \in X$ при условии, что объект находится в состоянии $k \in K$ - это центральное понятие, на котором основаны различные задачи распознавания. Сейчас самое подходящее время, чтобы показать примеры таких распределений, на которых мы будем иллюстрировать как предыдущие, так и последующие теоретические построения. В этой лекции мы прервем на время наше движение по курсу, чтобы описать две простейшие функции $p_{X|K}$, наиболее употребляемые в качестве моделей распознаваемого объекта.

3.1 Условная независимость признаков

Пусть наблюдение x состоит из n измеряемых на объекте признаков x_1, x_2, \dots, x_n . Каждый признак x_i , $i \in I$, принимает значения из конечного множества X_i , в общем случае своего для каждого признака. Множество наблюдений X , таким образом, есть декартово произведение $X_1 \times X_2 \times \dots \times X_n$. Предполагается, что вероятности $p_{X|K}(x | k)$ имеют вид

$$p_{X|K}(x | k) = \prod_{i=1}^n p_{X_i|K}(x_i | k). \quad (3.1)$$

Это значит, что при фиксированном состоянии k признаки становятся взаимно независимыми. Отсюда совсем не следует, что признаки априори независимы. Вообще говоря,

$$p_X(x) \neq \prod_{i=1}^n p_{X_i}(x_i). \quad (3.2)$$

Это значит, что признаки объекта зависят друг от друга, но эта зависимость обусловлена только их зависимостью от меняющегося состояния объекта. Если состояние объекта фиксировано, зависимость между признаками исчезает. Такое отношение между признаками называется *условной*

независимостью признаков. В описываемой модели имеет место простейший пример условной независимости.

Покажем, что в случае, когда каждый признак принимает только два значения $\{0,1\}$ и количество состояний равно 2, стратегия решения любой байесовской или небайесовской задачи реализуется разбиением множества вершин n -мерного куба с помощью гиперплоскостей. Мы уже знаем, что стратегия решения любой такой задачи имеет следующий вид: каждому решению d соответствует интервал значений отношения правдоподобия, и решение d принимается, когда

$$\theta_{\min}^d < \frac{p_{X|K}(x | k = 1)}{p_{X|K}(x | k = 2)} \leq \theta_{\max}^d, \quad (3.3)$$

где θ_{\min}^d и θ_{\max}^d - фиксированные для данной стратегии пороговые значения. Выражения (3.3) эквивалентны отношениям

$$\theta_{\min}^d < \log \frac{p_{X|K}(x | k = 1)}{p_{X|K}(x | k = 2)} \leq \theta_{\max}^d, \quad (3.4)$$

в которых значения порогов уже другие, чем в выражениях (3.3). Когда каждый признак x_i , $i \in I$, принимает только два значения 0 или 1, то справедливо равенство

$$\begin{aligned} \log \frac{p_{X|K}(x | k = 1)}{p_{X|K}(x | k = 2)} &= \sum_{i=1}^n \log \frac{p_{X_i|K}(x_i | k = 1)}{p_{X_i|K}(x_i | k = 2)} \\ &= \sum_{i=1}^n x_i \log \frac{p_{X_i|K}(1 | k = 1) p_{X_i|K}(0 | k = 2)}{p_{X_i|K}(1 | k = 2) p_{X_i|K}(0 | k = 1)} + \sum_{i=1}^n \log \frac{p_{X_i|K}(0 | k = 1)}{p_{X_i|K}(0 | k = 2)}. \end{aligned}$$

Чтобы убедиться в его правильности, достаточно проверить, что i -ое, $i \in I$, слагаемое в левой части равенства равно сумме двух i -ых слагаемых в правой части как при $x_i = 0$, так и при $x_i = 1$. Мы видим, таким образом, что *логарифм отношения правдоподобия есть линейная функция* переменных x_i . В силу этого выражение (3.4) можно переписать в виде

$$\theta_{\min}^d < \sum_{i=1}^n \alpha_i x_i \leq \theta_{\max}^d. \quad (3.5)$$

При фиксированных функциях $p_{X|K}$ стратегии (3.5) решения различных задач отличаются друг от друга только значениями порогов. Если же функции $p_{X|K}$ меняются от задачи к задаче, меняются и коэффициенты α_i . Все эти изменения не меняют того факта, что стратегия состоит в вычислении значения некоторой линейной функции от наблюдения и определении, принадлежит ли это значение тому или иному интервалу.

В частном случае, когда количество решений равно двум, множество X наблюдений должно быть разбито на два подмножества X_1 и X_2 так, что

$$x \in \begin{cases} X_1, & \text{если } \sum_{i=1}^n \alpha_i x_i \leq \theta, \\ X_2, & \text{если } \sum_{i=1}^n \alpha_i x_i > \theta. \end{cases} \quad (3.6)$$

Это значит, что когда объект характеризуется условно независимыми признаками, поиск нужной стратегии сводится к поиску коэффициентов α_i и порогового значения θ . В лекции 5 по линейным дискриминантным функциям будет подробно описано, как следует находить эти коэффициенты и пороги.

3.2 Гауссово распределение вероятностей

Пусть множество X наблюдений есть n -мерное линейное пространство. Отметим, что это предположение выходит за рамки предположений, в которых были изложены две предыдущие лекции. Там мы не раз подчеркивали, что X это конечное множество. Тем не менее, большая часть полученных ранее выводов применима и в данном случае. Достаточно лишь упомянуть, что числа $p_{X|K}(x|k)$ представляют уже не вероятности, а плотности вероятностей.

Мы исходим из того, что функция $p_{X|K}: X \times K \rightarrow \mathbb{R}$ имеет вид

$$p_{X|K}(x|k) = C(A^k) \exp \left(-\frac{1}{2} \sum_{i=1}^n \sum_{j=1}^n a_{ij}^k (x_i - \mu_i^k)(x_j - \mu_j^k) \right), \quad (3.7)$$

где k - это верхний индекс, а не показатель степени. Многомерная случайная величина с распределением вероятностей (3.7) называется многомерной гауссовой (нормальной) случайной величиной. В выражении (3.7) величина x_i - это значение i -ого признака объекта, μ_i^k - это условное математическое ожидание i -ого признака при условии, что объект находится в состоянии k . Символ A^k представляет матрицу, обратную к так называемой ковариационной матрице (B^k), то есть, $A^k = (B^k)^{-1}$. Элемент b_{ij}^k ковариационной матрицы B^k есть ковариация i -ого и j -ого признаков, то есть условное математическое ожидание произведения $(x_i - \mu_i^k)(x_j - \mu_j^k)$ при условии, что объект находится в состоянии k . Наконец, $C(A^k)$ - это коэффициент, обеспечивающий равенство 1 интеграла функции (3.7) по всей области ее определения.

Известно (и это легко показать), что в случае, когда количество состояний равно двум, равно как и количество решений, стратегия распознавания реализуется квадратичной дискриминантной функцией. Это значит, что

$$x \in \begin{cases} X_1, & \text{если } \sum_i \sum_j \alpha_{ij} x_i x_j + \sum_i \beta_i x_i \leq \gamma, \\ X_2, & \text{если } \sum_i \sum_j \alpha_{ij} x_i x_j + \sum_i \beta_i x_i > \gamma. \end{cases} \quad (3.8)$$

Коэффициенты α_{ij} , β_i , $i, j = 1, 2, \dots, n$, и порог γ зависят от статистической модели объекта, то есть матриц A^1, A^2 и векторов μ^1, μ^2 . Они зависят, конечно, и от того, для решения какой именно байесовской или небайесовской задачи эта стратегия предназначена. Даже в двумерном случае разнообразие форм множеств X_1 и X_2 может быть очень большим. Покажем некоторые из них.

1. Границей между множествами X_1 и X_2 может быть прямая линия. Множества X_1 и X_2 лежат по разные стороны от этой прямой.
2. Границу между множествами X_1 и X_2 могут образовать две параллельные линии так, что множество X_1 - это часть плоскости между этими прямыми, а X_2 - остальная часть плоскости.
3. Граница между двумя множествами может образоваться двумя пересекающимися прямыми, которые делят плоскость X на четыре сектора. Два противоположные сектора образуют множество X_1 , а два другие - множество X_2 .
4. Границей может быть эллипс (в частном случае, окружность). Множество X_1 - это множество точек внутри эллипса, а множество X_2 - вне эллипса.
5. Границу может образовать гипербола. Множество X_1 лежит между ветвями гиперболы, а множество X_2 - вне их.

В трехмерном и многомерных случаях разнообразие геометрических форм множеств X_1 и X_2 еще больше. В этой ситуации полезно понимать, что все это разнообразие геометрических форм может быть сведено к рассмотрению одной единственной. А именно, к случаю, когда границей двух множеств является гиперплоскость, а не эллипс, гипербола и т.п. Тогда множество стратегий вида (3.8) становится эквивалентным множеству стратегий вида

$$x \in \begin{cases} X_1, & \text{если } \sum_i \alpha_i x_i \leq \gamma, \\ X_2, & \text{если } \sum_i \alpha_i x_i > \gamma. \end{cases} \quad (3.9)$$

Эта эквивалентность достигается с помощью нехитрого приема, известного в распознавании образов как *спрямление признакового пространства*, хотя здесь одинаково подошло бы и название искривление признакового пространства. Это преобразование деформирует исходное пространство наблюдений так, что множество "кривых" границ между множествами X_1 и X_2 преобразуется в множество гиперплоскостей. В нашем случае для нас представляет интерес класс квадратичных поверхностей вида (3.8). Исходное n -мерное признаковое пространство преобразуется в $(n + \frac{1}{2}n(n + 1))$ -мерное пространство. Вектор $x = (x_1, x_2, \dots, x_i, \dots, x_n)$ преобразуется в

$(n + \frac{1}{2}n(n + 1))$ -мерный вектор

$$y = \begin{pmatrix} x_1, & x_2, & \dots, & x_i, & \dots, & x_{n-1}, & x_n, \\ x_1x_1, & x_1x_2, & \dots, & x_1x_i, & \dots, & x_1x_{n-1}, & x_1x_n, \\ & x_2x_2, & \dots, & x_2x_i, & \dots, & x_2x_{n-1}, & x_2x_n, \\ & & & & & \vdots & \\ & & & x_ix_i, & \dots, & x_ix_{n-1}, & x_ix_n, \\ & & & & & \vdots & \\ & & & & & x_{n-1}x_{n-1}, & x_{n-1}x_n, \\ & & & & & & x_nx_n \end{pmatrix}. \quad (3.10)$$

Если обозначить координаты нового полученного вектора y через y_i , где $i = 1, 2, \dots, n + \frac{1}{2}n(n + 1)$, а через Y_1, Y_2 - множества, в которые превращает множества X_1, X_2 преобразование (3.10), то стратегия (3.8) приобретает вид

$$y \in \begin{cases} Y_1, & \text{если } \sum_i \alpha_i y_i \leq \gamma, \\ Y_2, & \text{если } \sum_i \alpha_i y_i > \gamma. \end{cases} \quad (3.11)$$

Покажем пример спрямления пространства.

Пример 3.1 Спрявление признакового пространства. Пусть x - это одномерная случайная величина в гауссовой модели объекта. Допустим, что множества X_1, X_2 имеют вид

$$x \in \begin{cases} X_1, & \text{если } (x - x_0)^2 \geq \delta. \\ X_2, & \text{если } (x - x_0)^2 < \delta. \end{cases}$$

Эту стратегию можно эквивалентно представить в следующей линейной форме

$$x \in \begin{cases} X_1, & \text{если } \alpha_1 y_1 + \alpha_2 y_2 > \theta, \\ X_2, & \text{если } \alpha_1 y_1 + \alpha_2 y_2 \leq \theta, \end{cases}$$

где $y_1 = x^2, y_2 = x, \alpha_1 = -1, \alpha_2 = 2x_0, \theta = x_0^2 - \delta$. ▲

Вид полученной стратегии (3.11) показывает, что она реализует разбиение определенного линейного пространства с помощью гиперплоскости. Если не выходить за рамки гауссовой модели распознаваемого объекта, то преобразование исходного пространства в спрямляющее является одним и тем же для любой модели такого типа, то есть является стандартным. При решении же конкретной задачи для конкретной гауссовой модели надо только определить коэффициенты α_i и порог γ , которые определяют положение разделяющей гиперплоскости в спрямляющем пространстве. Как осуществляется такая настройка стратегии под конкретную модель, мы покажем в лекции 5, посвященной линейным дискриминантным функциям.

3.3 Обсуждение

Я не впервые читаю о распознавании объектов, которые описываются условно независимыми признаками. При этом всякий раз мне представляется невероятным, что реальный объект можно представить такой простой моделью. Тогда почему ее используют так часто? Единственное объяснение, которое я могу себе дать, что это отсутствие знаний о распознаваемом объекте, достаточных для построения более сложных моделей. Экспериментальные данные, которыми располагает конструктор распознающей системы, могут оказаться достаточными для вывода о зависимости лишь отдельных признаков от состояния объекта. Но этих экспериментальных данных может оказаться недостаточно, чтобы судить о зависимости признаков между собой, и, более того, о зависимости группы признаков от состояния. Однако такое объяснение является не обоснованием, а скорее чисто житейским оправданием типа "Я сделал работу плохо, потому что не хватило времени на то, чтоб выполнить ее хорошо". Вопрос здесь состоит в том, сделана ли работа в каком-то смысле правильно при ограниченных имеющихся ресурсах. Отсутствие знания о взаимной зависимости признаков вовсе не является основанием для принятия гипотезы о их независимости. Незнание зависимости - это что-то одно, а независимость - это совсем другое. Меня интересует вопрос о разумной постановке задачи, в которой в явном виде учитывалась бы неполнота знаний о распознаваемом объекте, а не скрывалась бы взятой с потолка моделью, имеющей лишь то преимущество, что ее легко анализировать. Я постарался разобраться с этими затруднениями и дошел до определенного этапа, но сейчас нуждаюсь в вашей помощи.

Пусть $X = X_1 \times X_2 \times \dots \times X_n$ есть множество наблюдений $x = (x_1, x_2, \dots, x_n)$, где X_i - множество значений признака x_i . Пусть множество состояний есть $K = \{1, 2\}$. Пусть $p_{X|k}(x)$ условная вероятность наблюдения x при условии, что объект находится в состоянии k . Функции $p_{X|k}$, $k \in K$, неизвестны, но известны маргинальные вероятности $p_{X_i|k}(x_i)$, для которых справедливо отношение

$$\left. \begin{aligned} p_{X_1|k}(x_1) &= \sum_{x \in X(1, x_1)} p_{X|k}(x), \\ &\vdots \\ p_{X_i|k}(x_i) &= \sum_{x \in X(i, x_i)} p_{X|k}(x), \\ &\vdots \\ p_{X_n|k}(x_n) &= \sum_{x \in X(n, x_n)} p_{X|k}(x), \end{aligned} \right\} k = 1, 2. \quad (3.12)$$

В этом выражении обозначение $X(i, x_i)$ использовано для множества $X_1 \times X_2 \times \dots \times X_{i-1} \times \{x_i\} \times X_{i+1} \times \dots \times X_n$ тех наблюдений $x = (x_1, x_2, \dots, x_n)$, в которых i -ый признак принимает фиксированное значение x_i .

Если бы признаки не зависели друг от друга при фиксированном состоянии k , была бы известна и функция $p_{X|k}$, так как ее значения вычислялись бы по формуле

$$p_{X|k}(x) = \prod_{i=1}^n p_{X_i|k}(x_i), \quad k = 1, 2. \quad (3.13)$$

Скажем прямо, такая модель слишком проста. Если она имеет место, то конструктору распознающей системы просто повезло: здесь нет каких-либо трудностей, преодоление которых требовало бы привлечения науки о распознавании. Я интересуюсь тем, как следует построить стратегию распознавания, когда известны только маргинальные вероятности $p_{X_i|k}$, $i = 1, \dots, n$, $k = 1, 2$, но предположение (3.13) об условной независимости признаков не обязательно выполняется. Говоря другими словами, как я должен распознать состояние k , когда мне о функциях $p_{X|k}$, $k = 1, 2$, известно лишь, что они удовлетворяют системе уравнений (3.12) с известными левыми частями, и больше ничего.

Меня не удивляет, что поднятые вопросы трудны для меня. Для меня многие другие вопросы тоже трудны. Меня удивляет, что никто не ставит вопрос именно так. Когда я пытался обсудить этот вопрос с более опытными в распознавании коллегами, я получал бойкий ответ, в котором так или иначе было скрыто предположение об условной независимости признаков. Если же я пытался узнать, почему в этом случае следует поступать так, как будто признаки независимы, я получал менее вразумительное объяснение, что стратегия, настроенная на независимые признаки, будет справляться с распознаванием и тогда, когда признаки в действительности окажутся зависимыми. При этом делается невнятная ссылка на общеизвестный факт, что именно при независимых признаках энтропия наблюдения становится максимальной. Этот факт затем достаточно вольно интерпретируется так, что при независимых признаках на вход распознающей системы поступают более разнообразные наблюдения, а при зависимых признаках — лишь некоторое их подмножество. Поэтому система, правильно классифицирующая определенное множество наблюдений, будет правильно классифицировать и их подмножество. При всей размытости этих соображений они уже вызывают возражения. Допустим, что зависимость между признаками действительно в каком-то смысле сужает множество наблюдений, поступающих на вход распознающей системы. Из этого, однако, не следует, что будут исключены именно неправильно распознаваемые наблюдения. Может случиться как раз наоборот.

Однако в этих размытых рассуждениях есть разумное зерно. Оно состоит в предположении, что в классе моделей, удовлетворяющих условию (3.12), имеется некоторая исключительная, в определенном смысле наихудшая модель. Байесовская стратегия, настроенная на эту худшую модель, будучи примененной для других моделей, окажется не хуже, чем будучи примененной для этой наихудшей модели. Кроме того, предполагают, что этой наихудшей моделью окажется именно модель (3.13). Эти предполо-

жения следует точно сформулировать, а затем проверить, действительно ли они выполняются.

Пусть m статистическая модель, которая определяется двумя числами $p_K(k)$, $k = 1, 2$, - априорными вероятностями состояний k и условными вероятностями $p_{X|k}(x)$, $x \in X$, $k = 1, 2$. Таким образом, $m = (p_K(k), p_{X|k}(x), k = 1, 2; x \in X)$. Пусть M - множество моделей. Пусть q обозначает стратегию вида $X \rightarrow \{1, 2\}$, а Q - множество всех возможных стратегий. Обозначим $R(q, m)$ риск, который достигается при применении стратегии q в модели m . Для определенности будем считать, что $R(q, m)$ - это вероятность ошибочного распознавания состояния k объекта. Я буду считать модель $m^* \in M$ худшей моделью на множестве M , если существует стратегия q^* , такая, что:

1. Любая другая стратегия $q \in Q$ удовлетворяет неравенство

$$R(q, m^*) \geq R(q^*, m^*), \quad q \in Q. \quad (3.14)$$

Это значит, что стратегия q^* является байесовской стратегией для модели m^* .

2. Для любой модели $m \in M$ выполняется неравенство

$$R(q^*, m^*) \geq R(q^*, m), \quad m \in M. \quad (3.15)$$

Это значит, что модель m^* является худшей для стратегии q^* .

Пусть множество моделей M состоит из моделей вида $m = (p_K(1), p_K(2), p_{X|1}(x), p_{X|2}(x))$, где функция $p_{X|1}$ удовлетворяет равенства (3.12) при $k = 1$, а функция $p_{X|2}$ удовлетворяет равенства (3.12) при $k = 2$. Для такого множества моделей формулируются следующие два вопроса:

Вопрос 1: Существуют ли модель m^* и стратегия q^* , которые удовлетворяют условиям (3.14) и (3.15)?

Вопрос 2: При положительном ответе на вопрос 1 состоит ли модель m^* именно из таких распределений вероятностей $p_{X|1}(x)$ и $p_{X|2}(x)$, для которых выполняются равенства (3.13)?

Парень, остановись. Не кажется ли Тебе, что Ты переоткрываешь материал второй лекции?

Будьте спокойны, мне это и кажется, и не кажется. Я понимаю, что на самом деле здесь имеет место задача различения сложных гипотез. Однако для ее решения я не могу непосредственно воспользоваться рекомендациями, приведенными в лекции, так как там рассматривается только случай, когда сложная гипотеза состоит из конечного количества простых. Ведь в лекции предполагалось, что мешающий параметр принимает конечное количество значений. Здесь же у меня количество простых гипотез не просто бесконечное. Множество простых гипотез соответствует множеству решений системы уравнений (3.12), а значит, множеству точек некоторого

многомерного пространства. Таким образом, здесь задача сложнее, чем те задачи, которые решались в лекции. Однако я вижу, что в тех случаях, когда класс моделей содержит худшую в моем смысле модель, проверка сложных гипотез существенно упрощается даже в том случае, когда эта сложная модель состоит из бесконечного количества простых. В этом случае стратегия проверки сложных гипотез совпадает со стратегией проверки простых гипотез, а именно, с той стратегией, которая настроена на худшую модель. Это доказывается сравнительно просто следующим образом.

В задаче проверки сложных гипотез (см. подраздел 2.2.5) требуется найти стратегию q^* , которая удовлетворяет неравенство

$$\max_{m \in M} R(q, m) \geq \max_{m \in M} R(q^*, m) \tag{3.16}$$

для любой стратегии q . Если пара (q^*, m^*) удовлетворяет условия (3.14) и (3.15), то q^* есть решение задачи (3.16). Действительно, следующие неравенства и равенство выполняются для любой стратегии $q \in Q$,

$$\max_{m \in M} R(q, m) \geq R(q, m^*) \geq R(q^*, m^*) = \max_{m \in M} R(q^*, m)$$

из которых следует неравенство (3.16). Первое неравенство в приведенной цепочке очевидно. Вторым неравенством является неравенство (3.14), а равенство в этой цепочке есть лишь в иной форме записанное неравенство (3.15).

И после этого Ты будешь утверждать, что Ты плохо понял вторую лекцию? С интересом следим дальше за Твоим рассказом. Будем считать это своего рода Твоей контрольной работой по второй лекции.

К сожалению, я знаю, что наихудшая в моем понимании модель существует не в каждом классе моделей. Я это понял по следующему приеру.

Пусть X - двумерное линейное пространство (плоскость), $K = \{1, 2\}$, а $p_{X|1}(x)$ - двумерное гауссово распределение вероятностей с независимыми компонентами, единичной дисперсией и математическими ожиданиями $\mu_1 = \mu_2 = 0$. Предположим, что распределение вероятностей $p_{X|2}(x)$ не полностью известно, то есть известно лишь, что это одно из двух возможных распределений: либо $p'_{X|2}(x)$ либо $p''_{X|2}(x)$, которые отличаются от $p_{X|1}(x)$ только своими математическими ожиданиями. Это точка $\mu'_1 = 2, \mu'_2 = 0$ в первом случае и точка $\mu''_1 = 0, \mu''_2 = 4$ во втором, см. рис. 3.1. Таким образом я определил множество, состоящее из двух моделей. Первая модель определяется функциями $(p_{X|1}, p'_{X|2})$, а вторая -

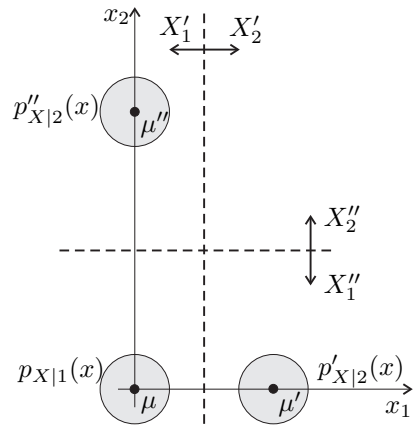


Figure 3.1 Difficulties in seeking for the ‘worst statistical model’ when attempting to decompose the plane into subsets.

функциями $(p_{X|1}, p''_{X|2})$. Для этой пары моделей можно сформулировать задачу различения сложных гипотез и решить ее так, как это было сделано в лекции. Однако ни одна из двух моделей, входящих в указанную пару, не является наилучшей в указанном мною смысле.

Первая модель представляется худшей из двух. Ей соответствует стратегия, которая делит плоскость X вертикальной прямой на подмножества X'_1 и X'_2 , как это показано на рис. 3.1. Легко увидеть, что эта стратегия на второй модели дает худшие результаты, чем на первой. Следовательно, это не та исключительно плохая модель, настройка на которую позволяет надеяться, что полученная стратегия окажется не хуже для второй модели. В то же время видно, что и вторая модель не является таковой. Стратегия, настроенная на нее, делит плоскость X горизонтальной прямой на две полуплоскости X''_1 и X''_2 . Эта стратегия на первой модели дает худшие результаты, чем на второй. Мы видим, таким образом, что стратегия, которая должна справляться как с первой моделью, так и со второй, не совпадает ни со стратегией, настроенной только на первую модель, ни со стратегией, настроенной на вторую модель.

Однако в классе наших моделей, удовлетворяющих равенства (3.12), содержится наилучшая модель, и я это доказываю следующим образом.

Пусть $\mathcal{P}(1)$ - множество функций $p_{X|1}$, которые удовлетворяют равенства (3.12) при $k = 1$, а $\mathcal{P}(2)$ - аналогичное множество для $k = 2$. Решением задачи (3.16), как я знаю из лекции 2, определяется двумя функциями $\alpha_1: X \rightarrow \mathbb{R}$ и $\alpha_2: X \rightarrow \mathbb{R}$, которые являются решением задачи линейного программирования

$$\left. \begin{aligned} \min c \\ c - \sum_{x \in X} \alpha_1(x) p_{X|2}(x) \geq 0, \quad p_{X|2} \in \mathcal{P}(2); \\ c - \sum_{x \in X} \alpha_2(x) p_{X|1}(x) \geq 0, \quad p_{X|1} \in \mathcal{P}(1); \\ \alpha_1(x) + \alpha_2(x) = 1, \quad x \in X; \\ \alpha_1(x) \geq 0, \quad \alpha_2(x) \geq 0, \quad x \in X. \end{aligned} \right\} \quad (3.17)$$

Эта задача имеет бесконечно много ограничений, количество которых равно $|\mathcal{P}(1)| + |\mathcal{P}(2)| + |X|$. Я избавился от этих бесконечностей следующим образом. Множество $\mathcal{P}(1)$ есть множество решений системы линейных уравнений. Это - выпуклое множество и к тому же ограниченное, так как среди решений $p_{X|1}$ системы (3.12) следует отобрать лишь те, которые удовлетворяют естественное ограничение, что $0 \leq p_{X|1}(x) \leq 1$ в любой точке $x \in X$. Поскольку речь идет о конечной системе линейных уравнений, то рассмат-

риваемое множество есть выпуклый многогранник. Количество вершин в этом многограннике может быть очень большим, но это уже конечное количество. Я обозначил эти вершины p_1^j , $j \in J(1)$, где $J(1)$ - конечное множество индексов. Очевидно, что если неравенство $c - \sum_{x \in X} \alpha_2(x) p_{X|1}(x) \geq 0$ выполняется для любой функции $p_{X|1}$ из множества $\mathcal{P}(1)$, то это же неравенство выполняется и для каждой функции p_1^j , $j \in J(1)$, то есть,

$$c - \sum_{x \in X} \alpha_2(x) p_1^j(x) \geq 0, \quad j \in J(1), \quad (3.18)$$

потому что любая вершина p_1^j , $j \in J(1)$, принадлежит множеству $\mathcal{P}(1)$. Обратное утверждение тоже верно: из неравенств (3.18) следуют и неравенства

$$c - \sum_{x \in X} \alpha_2(x) p_{X|1}(x) \geq 0, \quad p_{X|1} \in \mathcal{P}(1). \quad (3.19)$$

Происходит это потому, что любую функцию $p_{X|1}$ из множества $\mathcal{P}(1)$ можно выразить как

$$p_{X|1} = \sum_{j \in J(1)} \gamma_j p_1^j,$$

где γ_j , $j \in J(1)$, неотрицательные коэффициенты, для которых $\sum_{j \in J(1)} \gamma_j = 1$. Таким образом, условия (3.18) и (3.19) эквивалентны. То же самое верно и для условий

$$\begin{aligned} c - \sum_{x \in X} \alpha_1(x) p_{X|2}(x) &\geq 0, \quad p_{X|2} \in \mathcal{P}(2), \\ c - \sum_{x \in X} \alpha_1(x) p_2^j(x) &\geq 0, \quad j \in J(2), \end{aligned}$$

где $\{p_2^j \mid j \in J(2)\}$ - множество вершин многогранника $\mathcal{P}(2)$. Задача (3.17) приобретает вид

$$\left. \begin{array}{l} \min c \\ \tau_{2j} \left| \begin{array}{l} c - \sum_{x \in X_1} \alpha_1(x) p_2^j(x) \geq 0, \quad j \in J(2); \\ c - \sum_{x \in X_1} \alpha_2(x) p_1^j(x) \geq 0, \quad j \in J(1); \end{array} \right. \\ t(x) \left| \begin{array}{l} \alpha_1(x) + \alpha_2(x) = 1, \quad x \in X; \\ \alpha_1(x) \geq 0, \quad \alpha_2(x) \geq 0, \quad x \in X. \end{array} \right. \end{array} \right\} \quad (3.20)$$

Я помню из лекции 2, что решение этой задачи - это стратегия

$$\left. \begin{array}{l} \alpha_1^*(x) = 1, \alpha_2^*(x) = 0, \quad \text{если} \quad \sum_{j \in J(1)} \tau_{1j}^* p_1^j(x) > \sum_{j \in J(2)} \tau_{2j}^* p_2^j(x) \\ \text{и} \quad \alpha_1^*(x) = 0, \alpha_2^*(x) = 1, \quad \text{если} \quad \sum_{j \in J(1)} \tau_{1j}^* p_1^j(x) < \sum_{j \in J(2)} \tau_{2j}^* p_2^j(x), \end{array} \right\} \quad (3.21)$$

где $\tau_{1j}^*, j \in J(1)$, и $\tau_{2j}^*, j \in J(2)$, - коэффициенты Лагранжа, являющиеся решением двойственной задачи

$$\left. \begin{array}{l} \max \sum_{x \in X} t(x), \\ \alpha_1(x) \left| \begin{array}{l} t(x) - \sum_{j \in J(2)} \tau_{2j} p_2^j(x) \leq 0, \quad x \in X; \\ t(x) - \sum_{j \in J(1)} \tau_{1j} p_1^j(x) \leq 0, \quad x \in X; \\ \sum_{j \in J(1)} \tau_{1j} + \sum_{j \in J(2)} \tau_{2j} = 1, \\ \tau_{1j} \geq 0, \quad j \in J(1); \quad \tau_{2j} \geq 0, \quad j \in J(2). \end{array} \right. \end{array} \right\} \quad (3.22)$$

По виду стратегии (3.21), которую обозначим q^* , очевидно, что эта стратегия минимизирует вероятность ошибки на модели, в которой априорные вероятности состояний $k = 1, 2$ равны

$$p_K^*(1) = \sum_{j \in J(1)} \tau_{1j}^*, \quad p_K^*(2) = \sum_{j \in J(2)} \tau_{2j}^*, \quad (3.23)$$

а условные вероятности наблюдений x при условии состояния $k = 1, 2$ равны

$$p_{X|1}^*(x) = \sum_{j \in J(1)} \frac{\tau_{1j}^*}{\sum_{i \in J(1)} \tau_{1i}^*} p_1^j(x), \quad x \in X, \quad (3.24)$$

$$p_{X|2}^*(x) = \sum_{j \in J(2)} \frac{\tau_{2j}^*}{\sum_{i \in J(2)} \tau_{2i}^*} p_2^j(x), \quad x \in X. \quad (3.25)$$

Обозначим m^* модель (3.23), (3.24), (3.25). Очевидно, что эта модель удовлетворяет условия (3.12), так как функции $p_{X|1}^*$ и $p_{X|2}^*$ являются выпуклыми комбинациями функций p_1^j , $j \in J(1)$, и p_2^j , $j \in J(2)$, которые удовлетворяют условия (3.12). Стратегия q^* решает задачу (3.17), которая заключается в минимизации числа $\max_{m \in M} R(q, m)$. Можно записать

$$\min c = \max_{m \in M} R(q^*, m).$$

Коэффициенты τ_{1j}^* , $j \in J(1)$, и τ_{2j}^* , $j \in J(2)$, являются решением двойственной задачи (3.22) и таким образом

$$\max \sum_{x \in X} t(x) = \sum_{x \in X} \min \left(\sum_{j \in J(1)} \tau_{1j}^* p_1^j(x), \sum_{j \in J(2)} \tau_{2j}^* p_2^j(x) \right). \quad (3.26)$$

Выражение в правой части (3.26) есть риск $R(q^*, m^*)$. В силу первой теоремы двойственности 2.1 я имею $\min c = \max \sum_{x \in X} t(x)$ и, следовательно,

$$R(q^*, m^*) = \max_{m \in M} R(q^*, m). \quad (3.27)$$

Таким образом я доказал, что множество моделей, удовлетворяющих условию (3.12), содержит в себе наихудшую модель m^* , так как для нее выполняются неравенства

$$R(q, m^*) \geq R(q^*, m^*) \geq R(q^*, m), \quad q \in Q, \quad m \in M. \quad (3.28)$$

Первое неравенство в выражении (3.28) верно, так как q^* есть байесовская стратегия для модели m^* . Второе неравенство эквивалентно равенству (3.27), только лишь записанному в другой форме.

Теперь можно более определенно сформулировать следующие вопросы: Какая именно модель в классе (3.12) наихудшая? Какая стратегия решения соответствует этой наихудшей модели? Как следует распознавать состояние, если известны только маргинальные вероятности в левой части (3.12) и больше ничего?

Мне очень трудно ответить на эти вопросы. Но после того, как я достаточно сложным путем смог эти вопросы корректно сформулировать, я почти уверен, что на эти вопросы можно ответить.

Мы очень приятно удивлены той большой работой, которую Ты выполнил. Сейчас, пожалуй, трудно будет различить, кто кого здесь учит. Анализ Твоих вопросов потребовал определенного времени, но мы о нем не жалеем, потому что ответ оказался неожиданным для нас. Нам было приятно размышлять над Твоими вопросами, равно, как и отметить, что Ты обнаружил целину в таком поле, которое казалось уже вдоль и поперек перепаханным.

Оказалось, что не очень легко представить в явном виде наихудшую модель m^* , существование которой Ты доказал. Однако можно в явном виде указать байесовскую стратегию, соответствующую этой модели. Для этого достаточно лишь знать, что такая модель существует. Можно доказать, что эта стратегия, обозначим ее, как и прежде, q^* , принимает решение о состоянии k на основании какого-то одного-единственного признака и игнорирует все остальные.

Доказательство этого факта не очень простое и поэтому давай познакомимся на примерах, что этот факт обозначает. Предположим, что Ты имеешь два признака x_1 и x_2 . Ты должен определить, какой из этих двух признаков обеспечивает меньшую вероятность ошибки, а затем принимать решение о состоянии k на основании наблюдения только этого лучшего признака. Второй признак при этом нет нужды даже измерять. Давай посмотрим внимательнее на этот пример и попробуем понять, пока неформально, почему надо поступать именно так.

Пусть признак x_1 принимает только два значения 0, 1 и зависимость этого признака от состояния $k \in \{0, 1\}$ определяется четырьмя условными вероятностями:

$$\begin{aligned} p(x_1 = 1 | k = 1) &= 0.75, & p(x_1 = 0 | k = 1) &= 0.25, \\ p(x_1 = 1 | k = 0) &= 0.25, & p(x_1 = 0 | k = 0) &= 0.75. \end{aligned}$$

$k = 1$			
$p(k = 1) = ?$			
	x_1	0	1
		$p(x_1 k = 1)$	
x_2	$p(x_2 k = 1)$	0.25	0.75
0	0.3	?	?
1	0.7	?	?

$k = 0$			
$p(k = 0) = ?$			
	x_1	0	1
		$p(x_1 k = 0)$	
x_2	$p(x_2 k = 0)$	0.75	0.25
0	0.7	?	?
1	0.3	?	?

Table 3.1 The data of the example determining the probability of the wrong decision. The values in six table entries denoted by the question mark correspond to the unknown parameters of the statistical model.

Можно построить стратегию q^* , которая на основании наблюдения этого признака принимает решение о состоянии k с вероятностью ошибки 0.25. Эта стратегия принимает решение $k = 1$ при $x_1 = 1$ и решение $k = 0$ при $x_1 = 0$. Вероятность ошибки, как легко убедиться, не зависит от априорных вероятностей состояний.

Предположим, что эта вероятность кажется Тебе слишком большой и Ты бы хотел ее уменьшить за счет использования дополнительного признака, и такой признак есть в Твоем распоряжении. В нашем простейшем примере это признак x_2 , который также принимает только два значения. Условные вероятности этих значений при фиксированном состоянии k равны

$$p(x_2 = 1 | k = 1) = 0.7, \quad p(x_2 = 0 | k = 1) = 0.3,$$

$$p(x_2 = 1 | k = 0) = 0.3, \quad p(x_2 = 0 | k = 0) = 0.7.$$

Все имеющиеся в распоряжении данные приведены в таблице 3.1. Кроме имеющихся данных, в таблице зарезервированы места для вероятностей, которые неизвестны. Это - априорные вероятности $p(k = 1)$ и $p(k = 0)$ и условные вероятности $p(x_1, x_2 | k)$. Клетки в таблице 3.1, соответствующие этим данным, отмечены вопросительными знаками. Вопросительные знаки можно заменить любыми числами, лишь бы они удовлетворяли очевидному условию

$$p(k = 1) + p(k = 0) = 1$$

и условиям (3.12). Это значит, что сумма вероятностей $p(x_1 = 1, x_2 = 1 | k = 1)$ и $p(x_1 = 1, x_2 = 0 | k = 1)$ должна быть 0.75 и т.п. То или иное заполнение клеток, отмеченных вопросительными знаками, влияет в конечном итоге на вероятность ошибки, которую допускает та или иная стратегия. Тебе нетрудно убедиться, что для стратегии q^* , которая принимает решение на основании единственного признака x_1 , вероятность ошибки остается равной 0.25 при замене вопросительных знаков любыми числами.

Посмотрим теперь, можно ли уменьшить эту вероятность, используя оба признака x_1 и x_2 , а не только единственный признак x_1 . Естественной, например, представляется стратегия, которая принимает решение $k = 1$ тогда и только тогда, когда $x_1 = 1$ и $x_2 = 1$. Возможна и другая стратегия,

$k = 1$			
$p(k = 1) = 1$			
	x_1	0	1
		$p(x_1 k = 1)$	
x_2	$p(x_2 k = 1)$	0.25	0.75
0	0.3	0	0.3
1	0.7	0.25	0.45

(a)

$k = 0$			
$p(k = 0) = 1$			
	x_1	0	1
		$p(x_1 k = 0)$	
x_2	$p(x_2 k = 0)$	0.75	0.25
0	0.7	0.45	0.25
1	0.3	0.3	0

(b)

Table 3.2 The decision with two features. Two most unfavourable cases with the probability of the wrong decision 0.55 are depicted which correspond to two different strategies.

которая принимает решение $k = 0$ тогда и только тогда, когда $x_1 = 0$ и $x_2 = 0$. Возможны и многие иные стратегии.

Проанализируем сначала первую стратегию. Вероятность ошибки, которую допускает эта стратегия, в отличие от стратегии q^* , уже зависит от того, какими числами заменить вопросительные знаки в таблице 3.1. Эти числа могут быть такими, что вероятность ошибки станет равной 0.55. Это даже хуже, чем если бы решение о состоянии принималось на основании одного лишь худшего признака x_2 . В этом случае вероятность ошибки равнялась бы 0.3. Эти наихудшие для первой стратегии числа представлены в таблице 3.2(a), в которой указаны числа только для случая $k = 1$. Ясно, что при $p(k = 0) = 0$ вероятность ошибки вообще не зависит от чисел $p(x_1, x_2 | k = 0)$.

Числа, наихудшие для второй стратегии, представлены в таблице 3.2(b). При таких числах вторая стратегия допускает ошибку с вероятностью, которая также равна 0.55.

Если хочешь, можешь убедиться, что и любая иная стратегия не окажется лучше, чем ранее определенная стратегия q^* . Одна лишь стратегия q^* принимает ошибочное решение о состоянии k в 25% случаев, и это ее качество не зависит от того, какие числа подставить вместо вопросительных знаков. Для любой другой стратегии на месте вопросительных знаков можно поставить такие числа, что вероятность ошибочного распознавания станет больше, чем 25%.

Перед тем, как перейти от этого простого примера к анализу задачи в общем виде, давай представим себе еще один, пусть не очень конкретный, но выразительный пример, иллюстрирующий свойство, которые мы собираемся доказать.

Представь себе, что через некоторое время Ты станешь директором фирмы, или заведующим кафедрой, или кем-то в этом роде, кто должен принимать решение "да" или "нет". Ты не можешь быть сведущ во всех вопросах и поэтому Ты сформировал консультативное бюро из десяти экспертов. Этим консультантам Ты задаешь свой вопрос и получаешь десять ответов x_1, x_2, \dots, x_{10} "да" или "нет". После некоторого времени сотрудничества Ты узнаешь своих консультантов ближе и можешь каждого из них характеризовать вероятностями $p_i(x_i | k)$, где k - правильный ответ.

В результате Ты все равно должен принять свое собственное решение "да" или "нет" на основании десяти ответов x_1, x_2, \dots, x_{10} . Все было бы просто, если бы Ты был уверен, что эксперты принимают решения независимо друг от друга. Но Ты опытный руководитель и знаешь о сложных взаимоотношениях экспертов, при которых они обязательно зависят друг от друга. Но Ты не знаешь, какова именно эта зависимость.

Из анализа Твоей задачи следует рекомендация, что Ты должен принимать во внимание мнение только одного эксперта, а именно, самого лучшего. Конечно, Ты можешь вежливо выслушать и других экспертов, но Ты должен игнорировать их мнение. Эта рекомендация имеет силу даже в том случае, если в Твоем бюро окажется не один эксперт, который лучше других, а несколько одинаково квалифицированных экспертов. И даже больше. Возможно, Ты и себя считаешь экспертом не хуже других. В этом случае Ты должен либо всегда руководствоваться не своим мнением, а советом того эксперта, которого считаешь не хуже себя, либо не иметь никаких советников, а всегда полагаться только на свое мнение.

Конечно, не следует слишком уж прямолинейно связывать формальное решение Твоей формальной задачи с конкретной жизненной ситуацией, которую мы намеренно упрощенно представили лишь с целью более образного понимания полученного результата. Ведь мы с Тобой знаем, что разнообразие прикладных ситуаций значительно богаче формальной схемы. Однако даже при абстрактном рассмотрении Твоего вопроса полученная рекомендация выглядит парадоксально. Поэтому посмотрим более пристально, как она получается. Сначала мы посмотрим, как выводится эта рекомендация для случая двух признаков, а затем обобщим наши рассуждения на случай произвольного количества признаков.

Пусть x и y - два признака, которые принимают значения из конечных множеств X и Y . Пусть количество состояний равно 2, то есть k может быть равно либо 1, либо 2. Вероятности $p_{XY|k}(x, y)$ неизвестны, но известны вероятности

$$p_{X|k}(x) = \sum_{y \in Y} p_{XY|k}(x, y), \quad x \in X, \quad k = 1, 2, \quad (3.29)$$

$$p_{Y|k}(y) = \sum_{x \in X} p_{XY|k}(x, y), \quad y \in Y, \quad k = 1, 2. \quad (3.30)$$

Пусть M обозначает множество статистических моделей вида $m = (p_K(1), p_K(2), p_{XY|1}, p_{XY|2})$, где $p_{XY|1}$ и $p_{XY|2}$ удовлетворяют условиям (3.29) и (3.30). Ты доказал, что существуют такая стратегия $q^*: X \times Y \rightarrow \{1, 2\}$ и такая модель $m^* \in M$, для которой выполняется

$$R(q, m^*) \geq R(q^*, m^*) \geq R(q^*, m), \quad q \in Q, \quad m \in M. \quad (3.31)$$

Проанализируем стратегию q^* , которая удовлетворяет (3.31).

Обозначим $XY^*(1)$ и $XY^*(2)$ множества, на которые стратегия q^* делит множество $X \times Y$. Обозначим $XY^+(1)$ и $XY^+(2)$ такие два подмножества

$XY^*(1)$ и $XY^*(2)$, что

$$XY^+(1) = \left\{ (x, y) \in X \times Y \mid q^*(x, y) = 1, p_{XY|1}^*(x, y) > 0 \right\},$$

и аналогично,

$$XY^+(2) = \left\{ (x, y) \in X \times Y \mid q^*(x, y) = 2, p_{XY|2}^*(x, y) > 0 \right\}.$$

Подмножество $Z \subset X \times Y$ будем называть *прямоугольником* в множестве $X \times Y$, если существуют такие два подмножества $X' \subset X$ и $Y' \subset Y$, что $Z = X' \times Y'$. Минимальный прямоугольник, содержащий подмножество $Z \subset X \times Y$, будем называть *декартовым замыканием* подмножества Z и обозначать его Z^c .

Множества $XY^+(1)$ и $XY^+(2)$ полностью определяются парой (q^*, m^*) и обладают следующим важным свойством. Если (q^*, m^*) удовлетворяют условиям (3.31), то декартовы замыкания множеств $XY^+(1)$ и $XY^+(2)$ не пересекаются, то есть

$$(XY^+(1))^c \cap (XY^+(2))^c = \emptyset. \quad (3.32)$$

Предположим, что равенство (3.32) не выполняется. Тогда должна существовать точка (x^*, y^*) , которая принадлежит как $(XY^+(1))^c$, так и $(XY^+(2))^c$. Значение $q^*(x^*, y^*)$ в этой точке равно либо 1, либо 2. Примем для определенности, что $q^*(x^*, y^*) = 1$. В силу $(x^*, y^*) \in (XY^+(2))^c$ существуют две точки (x^*, y) и (x, y^*) , для которых

$$q(x^*, y) = 2, \quad p_{XY|2}^*(x^*, y) > 0, \quad q(x, y^*) = 2, \quad p_{XY|2}^*(x, y^*) > 0.$$

Выберем положительное число Δ , которое не превосходит как $p_{XY|2}^*(x^*, y)$, так и $p_{XY|2}^*(x, y^*)$, например,

$$\Delta = \min \left(p_{XY|2}^*(x^*, y), p_{XY|2}^*(x, y^*) \right).$$

Сформируем новую модель $m = (p_K^*(1), p_K^*(2), p_{XY|1}^*, p_{XY|2}^*)$, которая отличается от модели m^* лишь тем, что функция $p_{XY|2}$ в новой модели отличается от функции $p_{XY|2}^*$ в модели m^* . Более того, функция $p_{XY|2}$ отличается от функции $p_{XY|2}^*$ своими значениями только в четырех точках (x^*, y^*) , (x, y^*) , (x^*, y) и (x, y) так, что

$$\left. \begin{aligned} p_{XY|2}(x, y^*) &= p_{XY|2}^*(x, y^*) - \Delta, \\ p_{XY|2}(x^*, y) &= p_{XY|2}^*(x^*, y) - \Delta, \\ p_{XY|2}(x^*, y^*) &= p_{XY|2}^*(x^*, y^*) + \Delta, \\ p_{XY|2}(x, y) &= p_{XY|2}^*(x, y) + \Delta. \end{aligned} \right\} \quad (3.33)$$

При переходе от модели m^* к модели m условная вероятность ошибки при условии, что объект находится в состоянии 2, увеличивается по крайней

мере на Δ . Действительно, стратегия q^* относит точку (x^*, y^*) к первому классу, а точки (x, y^*) и (x^*, y) - ко второму. Вероятность точки (x^*, y^*) во втором состоянии увеличивается на Δ , а вероятности обеих точек (x, y^*) и (x^*, y) уменьшаются на Δ . Следовательно, независимо от того, к какому классу стратегия q^* относит точку (x, y) , суммарная вероятность точек, которые соответствуют второму состоянию, но относятся стратегией q^* к первому классу, увеличивается по крайней мере на Δ .

Очевидно также, что если функция $p_{XY|2}^*$ удовлетворяет условиям (3.29) и (3.30), то новая функция $p_{XY|2}$ после модификации (3.33) также удовлетворяет этим условиям. Таким образом, доказано, что если (3.32) не выполняется, то существует модель $t \in M$, для которой $R(q^*, t) > R(q^*, t^*)$, что противоречит (3.31). Поэтому (3.32) следует из (3.31).

Поскольку прямоугольники $(XY^+(1))^c$ и $(XY^+(2))^c$ не пересекаются, справедливо следующее утверждение.

Утверждение 3.1 *По крайней мере одна из двух следующих ситуаций имеет место:*

1. Существует такое разбиение множества X на два подмножества $X(1)$ и $X(2)$, что

$$(XY^+(1))^c \subset X(1) \times Y, \quad (XY^+(2))^c \subset X(2) \times Y. \quad (3.34)$$

2. Существует такое разбиение множества Y на два подмножества $Y(1)$ и $Y(2)$, что

$$(XY^+(1))^c \subset X \times Y(1), \quad (XY^+(2))^c \subset X \times Y(2). \quad (3.35)$$

▲

Справедливость этого утверждения подкрепляется чисто наглядными представлениями. Если два прямоугольника лежат в одной плоскости, составлены из вертикальных и горизонтальных сторон и не пересекаются, то они могут быть отделены друг от друга либо вертикальной, либо горизонтальной прямой.

Строгое доказательство этого утверждения тоже не очень сложное. Допустим, что не имеет место ни одна из двух ситуаций, указанных в утверждении. Коль скоро не имеет место первая ситуация, существуют такие две точки (x_0, y_1) и (x_0, y_2) , что $(x_0, y_1) \in (XY^+(1))^c$ и $(x_0, y_2) \in (XY^+(2))^c$. Коль скоро не имеет место вторая ситуация, существуют две точки (x_1, y_0) и (x_2, y_0) , такие, что $(x_1, y_0) \in (XY^+(1))^c$ и $(x_2, y_0) \in (XY^+(2))^c$. Из того, что $(x_0, y_1) \in (XY^+(1))^c$ и $(x_1, y_0) \in (XY^+(1))^c$, следует, что $(x_0, y_0) \in (XY^+(1))^c$. Из того, что $(x_0, y_2) \in (XY^+(2))^c$ и $(x_2, y_0) \in (XY^+(2))^c$, следует, что точка (x_0, y_0) принадлежит и множеству $(XY^+(2))^c$. Следовательно, множества $(XY^+(1))^c$ и $(XY^+(2))^c$ пересекаются, а этого не может быть, потому что значения, которые принимает стратегия q^* на этих множествах, различны.

Для дальнейших рассуждений предположим для определенности, что имеет место первая ситуация в утверждении 3.1. Обозначим q' стратегию, которая разбивает множество $X \times Y$ на два класса $X(1) \times Y$ и $X(2) \times Y$. Эта стратегия, формально являющаяся функцией двух переменных x и y , фактически является функцией лишь одной переменной x . Поэтому риск

$R(q', m)$ от модели m не зависит вообще, то есть

$$R(q', m^*) = R(q', m), \quad m \in M. \quad (3.36)$$

Докажем теперь, что стратегия q' для модели m^* является байесовской. В соответствии с условием (3.31) стратегия q^* является байесовской для модели m^* . Пусть для некоторой точки (x, y) выполняется $q'(x, y) \neq q^*(x, y)$. Если такой точки нет, то стратегии q' и q^* совпадают и, следовательно, стратегия q' байесовская. Если $q'(x, y) \neq q^*(x, y)$, то

$$\text{либо } q'(x, y) = 1, \quad q^*(x, y) = 2, \quad \text{либо } q'(x, y) = 2, \quad q^*(x, y) = 1.$$

Пусть $q'(x, y) = 1$ и $q^*(x, y) = 2$. Тогда

$$\begin{aligned} (q'(x, y) = 1) &\Rightarrow ((x, y) \in X(1) \times Y) \Rightarrow ((x, y) \notin XY^+(2))^c \\ &\Rightarrow ((x, y) \notin XY^+(2)) \Rightarrow (p_{XY|2}^*(x, y) = 0). \end{aligned}$$

Это значит, что и вероятность $p_{XY|1}^*(x, y)$ должна равняться нулю, так как в противном случае стратегия q^* , которая относит (x, y) ко второму классу, не была бы байесовской. Подобным образом можно доказать, что для всех точек (x, y) , для которых $q'(x, y) = 2$ и $q^*(x, y) = 1$, тоже выполняется $p_{XY|1}^*(x, y) = p_{XY|2}^*(x, y) = 0$. Таким образом, построенная стратегия q' , которая принимает решение на основании наблюдения лишь признака x , отличается от байесовской стратегии q^* только на множестве точек, суммарная вероятность которых в модели m^* равна нулю. Следовательно, она также является байесовской на модели m^* . Таким образом, в дополнение к полученному ранее равенству (3.36) мы получаем цепочку

$$R(q, m^*) \geq R(q', m^*) \geq R(q', m), \quad q \in Q, \quad m \in M,$$

которая обозначает, что в случае двух признаков искомая стратегия принимает решение на основании наблюдения лишь одного признака.

Почти все шаги в приведенном доказательстве, кроме доказательства отношения (3.32), легко выполнить и для случая, когда количество признаков больше двух. Это отношение утверждает, что декартовы замыкания определенных множеств не пересекаются. Мы ограничимся доказательством только этого факта для многомерного случая.

Пусть x_1, x_2, \dots, x_n - это n признаков, которые принимают значения из множеств X_1, X_2, \dots, X_n соответственно. Обозначим $x = (x_1, x_2, \dots, x_n)$, $X = X_1 \times X_2 \times \dots \times X_n$. Обозначим m^* статистическую модель $(p_K^*(1), p_K^*(2), p_{X|1}^*, p_{X|2}^*)$, где функция $p_{X|1}^*$ удовлетворяет равенствам

$$p_{X_i|1}(x_i) = \sum_{x \in X(i, x_i)} p_{X|1}^*(x), \quad x_i \in X_i, \quad i = 1, \dots, n$$

а функция $p_{X|2}^*$ удовлетворяет равенствам

$$p_{X_i|2}(x_i) = \sum_{x \in X(i, x_i)} p_{X|2}^*(x), \quad x_i \in X_i, \quad i = 1, \dots, n,$$

при известных маргинальных вероятностях в левых частях. Пусть $q^*: X \rightarrow \{1, 2\}$ - байесовская стратегия для такой модели m^* , что

$$R(q^*, m^*) \geq R(q^*, m), \quad m \in M. \quad (3.37)$$

Определим множества $X^+(1)$ и $X^+(2)$ как

$$\begin{aligned} X^+(1) &= \left\{ x \in X \mid q^*(x) = 1, p_{X|1}^*(x|1) > 0 \right\}, \\ X^+(2) &= \left\{ x \in X \mid q^*(x) = 2, p_{X|2}^*(x|2) > 0 \right\}, \end{aligned}$$

и множества $(X^+(1))^c$ и $(X^+(2))^c$ как декартовы замыкания множеств $X^+(1)$ и $X^+(2)$. Мы докажем, что из неравенства (3.37) следует

$$(X^+(1))^c \cap (X^+(2))^c = \emptyset.$$

Предположим, что это отношение неверно, то есть существует точка x^* , принадлежащая как $(X^+(1))^c$, так и $(X^+(2))^c$. Предположим для определенности и простоты дальнейшего изложения, что $q^*(x^*) = 1$, а признаки x_1, x_2, \dots, x_n принимают значение 0 в точке x^* . Таким образом, $x^* = (0, 0, 0, \dots, 0)$. Принадлежность точки x^* множеству $(X^+(2))^c$ обозначает существование совокупности S точек x^1, x^2, \dots, x^t , $t \leq n$, принадлежащих $X^+(2)$, и таких, что для каждого $i = 1, 2, \dots, n$ имеется по крайней мере одна точка x' из совокупности S , такая, что $x'_i = 0$. Напомним, что принадлежность точки $x' \in S$ множеству $X^+(2)$ обозначает, что $q^*(x') = 2$ и $p_{X|2}^*(x') > 0$.

Покажем, что в этом случае можно уменьшить вероятности $p_{X|2}(x)$ точек x^1, x^2, \dots, x^t и увеличить вероятности некоторых других точек, включая точку x^* , не меняя при этом маргинальных вероятностей $p_{X_i|2}(x_i)$. При этом уменьшаются вероятности $p_{X|2}(x)$ только тех точек, которые стратегия q^* относит ко второму классу. Вероятность $p_{X|2}(x)$ ни одной точки из первого класса не уменьшается. Зато вероятности некоторых точек из первого класса увеличиваются. В частности, это точка x^* . В результате такого специально подобранного перераспределения вероятностей вероятность ошибочного распознавания объектов во втором состоянии увеличивается. Поскольку это будет противоречить условию (3.37), то тем самым отношение (3.37) будет доказано.

Покажем, что возможность указанного перераспределения вероятностей следует из отношения $(X^+(1))^c \cap (X^+(2))^c \neq \emptyset$.

Утверждение 3.2 *Если совокупность S состоит только из двух точек x^1 и x^2 , то существует модель m , для которой $R(q^*, m^*) < R(q^*, m)$, $m \in M$. \blacktriangle*

Доказательство. Построим точку x' , координаты которой x'_i , $i = 1, \dots, n$, определим по следующим правилам. Если $x_i^1 = 0$, то $x'_i = x_i^2$, и если $x_i^2 = 0$, то $x'_i = x_i^1$. Говоря иными словами, i -ая координата точки x' равна ненулевой координате x_i^1 либо x_i^2 . Если обе эти координаты равны нулю, то

i -ая координата точки x' также равна нулю. Для таким способом построенной точки x' и точек x^1 , x^2 и x^* выполняется, что, сколько раз какое-то значение i -ой координаты встречается в паре точек x^1 и x^2 , ровно столько раз это значение встречается в паре точек x' и x^* . Напомним, что все координаты точки x^* равны нулю.

Пример 3.2 Пусть $x^1 = (0, 0, 0, 5, 6, 3)$ и $x^2 = (5, -2, 0, 0, 0, 0)$. Как было сказано раньше, $x^* = (0, 0, 0, 0, 0, 0)$. В этом случае $x' = (5, -2, 0, 5, 6, 3)$. ▲

Определим $\Delta = \min(p_{X|2}^*(x^1), p_{X|2}^*(x^2))$ и распределение вероятностей $p_{X|2}$:

$$\left. \begin{aligned} p_{X|2}(x_1) &= p_{X|2}^*(x^1) - \Delta; \\ p_{X|2}(x_2) &= p_{X|2}^*(x^2) - \Delta; \\ p_{X|2}(x^*) &= p_{X|2}^*(x^*) + \Delta; \\ p_{X|2}(x') &= p_{X|2}^*(x') + \Delta; \\ p_{X|2}(x) &= p_{X|2}^*(x), \quad x \notin \{x^1, x^2, x^*, x'\}. \end{aligned} \right\} \quad (3.38)$$

При указанной замене распределения вероятностей $p_{X|2}^*$ на $p_{X|2}$ маргинальные вероятности $p_{X_i|2}$ не изменяются. Сколько слагаемых в правой части равенств (3.12) увеличивается на Δ , ровно столько же и уменьшается. В то же время вероятность ошибочного распознавания объектов во втором состоянии увеличивается по крайней мере на Δ . Действительно, $q^*(x^1) = 2$, $q^*(x^2) = 2$, $q^*(x^*) = 1$. Если при этом $q^*(x') = 1$, то вероятность ошибки увеличивается на 2Δ . Если $q^*(x') = 2$, то на Δ . ■

Утверждение 3.3 Пусть $S = \{x^1, x^2, \dots, x^t\}$, $t > 2$. В этом случае либо существует модель m , для которой $R(q^*, m^*) < R(q^*, m)$, либо существует совокупность S' точек, меньшая, чем S , и такая, что ее декартово замыкание содержит точку x^* . ▲

Доказательство. Обозначим I множество индексов $\{1, 2, \dots, n\}$. Для каждой точки x обозначим $I(x)$ подмножество индексов, соответствующих нулевой координате точки x . Таким образом, $I(x^*) = \bigcup_{x \in S} I(x) = I$. Выберем из совокупности S две точки x^1 и x^2 , для которых $I(x^1) \neq I(x^2)$, и построим две другие точки x' , x'' . Точку x' построим так, чтобы $I(x') = I(x^1) \cup I(x^2)$. Это значит, что i -ая координата точки x' равна нулю, если она равна нулю по крайней мере в одной из точек x^1 или x^2 . Остальные координаты точек x' , x'' выбираем так, чтобы выполнялось свойство, указанное при доказательстве утверждения 3.2. А именно, сколько раз определенное значение i -ой координаты встречается в паре точек x^1 и x^2 , столько же раз это значение встречается в паре точек x' и x'' .

Пример 3.3

$$\begin{aligned} x^1 &= (0, 0, 0, 0, 1, 2, 3, 4, 5), \\ x^2 &= (5, 4, 0, 0, 0, 0, 3, 2, 1). \end{aligned}$$

Пару точек x' и x'' могут образовать, например, точки

$$\begin{aligned}x' &= (0, 0, 0, 0, 0, 3, 4, 1), \\x'' &= (5, 4, 0, 0, 1, 2, 3, 2, 5). \quad \blacktriangle\end{aligned}$$

Величина $\Delta = \min(p_{X|2}(x^1), p_{X|2}(x^2))$ положительна, так как $x^1 \in X^+(2)$ и $x^2 \in X^+(2)$. Новая модель должна строиться точно так же, как и при использовании правил (3.38), только x^* следует заменить на x'' . Пусть стратегия q^* относит только одну из точек x' , x'' к первому классу. В этом случае в результате перераспределения вероятностей увеличится вероятность ошибочного распознавания объектов, находящихся во втором состоянии. Таким образом, выполнится условие $R(q^*, m^*) < R(q^*, m)$. Если $q^*(x') = q^*(x'') = 2$, то точка x' будет принадлежать множеству $X^+(2)$ в новой модели, так как ее вероятность станет положительной. Учитывая, что $I(x') = I(x^1) \cup I(x^2)$, свойство $\cup_{x \in S'} I(x) = I$ будет выполняться и для совокупности $S' = \{x', x^3, \dots, x^k\}$, количество элементов в которой на единицу меньше, чем в совокупности S . Совокупность S' получена так, что из нее исключены точки x^1, x^2 и включена точка x' . ■

Таким образом, мы доказали, что декартовы замыкания определенных множеств не пересекаются и тогда, когда количество признаков больше двух. Все остальные соображения по анализу Твоей задачи те же, что и в случае двух признаков. Так, общими усилиями мы справились с Твоей задачей.

Строго говоря, не до конца. Думаю, что рассмотрение случая, когда количество состояний больше двух, потребует дополнительных усилий.

Ты, конечно, заметил, что выполненное Тобой доказательство без существенных трудностей обобщается на случай произвольного количества состояний. Далее, если взять за основу уже выполненный нами анализ, то нетрудно доказать, что если m^* худшая модель, то соответствующая ей стратегия, как и прежде, делит пространство наблюдений на прямоугольники. Однако, из этого сейчас уже не следует, что решение о состоянии принимается на основании одного-единственного признака. В этом случае решение зависит не более, чем от $|K| - 1$ признаков, где $|K|$ - количество состояний. Например, если количество состояний равно 3, то на основании какого-то одного признака принимается решение, находится ли объект в каком-то определенном состоянии, скажем, в состоянии k_1 . Если это решение отрицательное, то на основании какого-то другого признака, или того же самого, принимается решение, в каком из двух состояний k_2 или k_3 находится объект. Не будем вдаваться в дальнейшие объяснения, и совсем не потому, что они не интересны. Просто есть песенки, которые приятнее напеть самому, чем слушать, как их поют другие. Здесь, в обнаруженном Тобой круге вопросов есть еще много песенок такого рода. Нам хочется похвалить и поблагодарить Тебя за то, что Ты обнаружил этот круг вопросов.

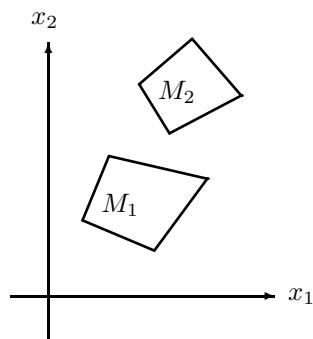


Figure 3.2 Two convex sets corresponding to $K = \{1, 2\}$.

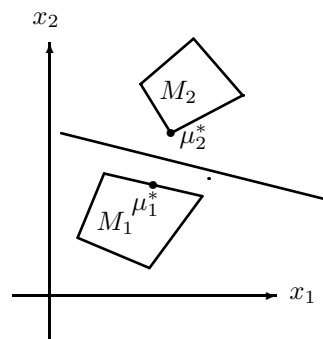


Figure 3.3 The decision as a separation of a plane by a straight line.

В завершение не можем не обратить Твое внимание на исключительную простоту конечного результата - стратегии решения Твоей задачи. Нам кажется, что это вообще характерно для линейных статистических задач с мешающими параметрами (см. подразделы 2.4.6 и 2.4.7). Ведь линейные задачи возникают при неполноте наших сведений о статистическом объекте или в случае, когда статистическая модель в принципе не определима, так как зависит от изменчивых, но неслучайных вмешательств. Стратегия принятия решения в таких не полностью определенных условиях не должна быть слишком чувствительной к статистической модели, она обязана быть в определенном смысле грубой, то есть простой.

Просмотри ради интереса еще одну простую задачу и убедись в том, насколько простым может оказаться точное решение одной задачи с неполным знанием статистической модели.

Пусть X - двумерное пространство (плоскость), $K = \{1, 2\}$. Пусть случайная величина x при условии $k = 1$ есть двумерная гауссова случайная величина с независимыми компонентами, дисперсии которых равны единице. Математическое ожидание μ_1 этой случайной величины неизвестно. Известно однако, что если объект находится в первом состоянии, это математическое ожидание есть точка в ограниченном и замкнутом выпуклом множестве M_1 , как показано на рис. 3.2.

При условии $k = 2$ наблюдение x - это такая же случайная величина, но с другим математическим ожиданием, которое также неизвестно. Известно только, что это - одна из точек в множестве M_2 , которое тоже выпукло, ограничено и не пересекается с M_1 , см. рис. 3.2.

В литературе Тебе могут встретиться две различные рекомендации по распознаванию состояния по сигналу x при такой неопределенности.

1. *Классификатор по ближайшему соседу* для наблюдения $x \in X$ вычисляет числа

$$d_1 = \min_{\mu \in M_1} r(x, \mu) \quad \text{и} \quad d_2 = \min_{\mu \in M_2} r(x, \mu),$$

где $r(x, \mu)$ - евклидово расстояние между точками x и μ . Затем принимается решение в пользу первого или второго состояния в зависимости от того, выполняется ли условие $d_1 \leq d_2$ или $d_1 > d_2$.

2. *Классификация по интегралу вероятностей* основана на предположении, что математические ожидания μ_1 и μ_2 - случайные величины с равномерным распределением вероятностей на множествах M_1 и M_2 соответственно. Вычисляются две величины

$$s_1 = \int_{M_1} f(x, \mu) d\mu \text{ и } s_2 = \int_{M_2} f(x, \mu) d\mu,$$

где $f(x, \mu)$ гауссово распределение вероятностей случайной величины x с математическим ожиданием μ . Решение о состоянии k принимается затем по отношению s_1/s_2 .

Найди стратегию, которая правильно решает линникову задачу в этом примере.

Прежде всего нужно найти пару

$$(\mu_1^*, \mu_2^*) = \operatorname{argmin}_{(\mu_1, \mu_2) \in M_1 \times M_2} r(\mu_1, \mu_2).$$

Затем следует определить байесовскую стратегию для модели m^* , в которой вероятности $p_K^*(1)$ и $p_K^*(2)$ равны друг другу, а $p_{X|1}^*(x) = f(x, \mu_1^*)$ и $p_{X|2}^*(x) = f(x, \mu_2^*)$. Эта стратегия разделяет плоскость X на две полуплоскости прямой линией, как показано на рис. 3.3. Я обозначу эту стратегию q^* , а вероятность ошибки, которую допускает стратегия q^* на модели m^* , обозначу ε^* . Поскольку q^* это байесовская стратегия для m^* , то для любой стратегии q выполняется $R(q, m^*) \geq \varepsilon^*$. Далее, просто по рис. 3.3 очевидно, что для любой модели $m = (p_K(1), p_K(2), f(x, \mu_1), f(x, \mu_2))$, $\mu_1 \in M_1$, $\mu_2 \in M_2$, $p_K(1) + p_K(2) = 1$, $p_K(1) \geq 0$, $p_K(2) \geq 0$, выполняется неравенство $R(q^*, m) \leq \varepsilon^*$. Это значит, что модель m^* является худшей, а стратегия q^* является решением задачи.

Эта стратегия намного проще упомянутых вами стратегий по методу ближайшего соседства и интегралу вероятностей. В первой стратегии нужно для каждого наблюдения x найти точки в двух выпуклых множествах, ближайšie к x . В двумерном случае это еще не очень трудно, но в многомерных случаях тут могут возникнуть проблемы. Во второй стратегии надо вычислить два интеграла гауссовой функции по двум выпуклым множествам, и это следует делать для каждого наблюдения x . Даже если речь идет только о двумерном случае и этими выпуклыми множествами являются многоугольники, я не знаю даже, с чего начать эти вычисления. Простота точного решения линниковой задачи действительно удивляет, если ее сравнивать с известными популярными решениями.

К этому добавь еще, что стратегия q^* выдумана не сама по себе, а получена Тобой как решение однозначно сформулированной задачи, чего

нельзя сказать об упомянутых нами решениях. При любых априорных вероятностях состояний и при любых математических ожиданиях μ_1 и μ_2 в пределах множеств M_1 и M_2 Ты можешь утверждать, что вероятность ошибочного решения не будет превышать ε^* . Значение ε^* здесь является как бы гарантированным уровнем, который обеспечивается вне зависимости от статистической неопределенности распознаваемого объекта. Более того, Ты можешь утверждать, что никакая другая стратегия не обеспечивает лучший гарантированный уровень.

Мы бы хотели предложить Тебе небольшое упражнение по гауссовым моделям распознавания. Оно принадлежит к фольклору программистов, работающих в распознавании.

Мы упомянули в лекции, что в случае двух состояний и гауссовой модели стратегия принятия решения требует вычисления определенной квадратичной функции от наблюдения x . Значение этой функции в наблюдении x следует сравнить с порогом. В случае двух признаков (обозначим их x и y) речь идет о вычислении

$$f(x, y) = ax^2 + bxy + cy^2 + dx + ey + g, \quad (3.39)$$

которое требует восемь умножений и пять сложений для каждого наблюдения x, y . Чтобы сэкономить вычисления непосредственно при распознавании, функцию (3.39) можно заранее протабулировать, скажем, для 1000 целых значений x и y . Допустим, что условия конкретного приложения требуют, чтобы и это предварительное табулирование производилось как можно быстрее. Говорят, что вычисление каждого числа в таблице, соответствующего паре целых чисел (x, y) , $x = 0, 1, 2, \dots, 999$, $y = 0, 1, 2, \dots, 999$, можно выполнить с помощью всего лишь двух сложений и ни одного умножения. Как это сделать?

Я нашел решение на основании равенств

$$f(x, y) = ax^2 + bxy + cy^2 + dx + ey + g, \quad (3.40)$$

$$f(x, y) = f(x - 1, y) + 2ax + by + d + a, \quad (3.40)$$

$$f(0, y) = f(0, y - 1) + 2cy + e - c. \quad (3.41)$$

В программе, которую я покажу, используются константы: $A = 2a$, $B = b$, $C = 2c$, $D = d + a$, $E = e - c$, $G = g$, которые вычисляются заранее. Их вычисление не входит в работу программы. По выражениям (3.40) и (3.41) я получаю

$$f(x, y) = f(x - 1, y) + Ax + By + D, \quad (3.42)$$

$$f(0, y) = f(0, y - 1) + Cy + E, \quad (3.43)$$

$$f(0, 0) = G. \quad (3.44)$$

Используя (3.43) и (3.44), можно написать программу в С, которая табулирует функцию f для значений $x = 0, y = 0, 1, \dots, 999$.

```
fCur = f[0][0] = G; DeltaCur = E;
for (y=1; y < 1000; y++)
    f[0][y] = fCur += DeltaCur += C;          /* L1 */
```

После каждого выполнения команды L1 переменная DeltaCur принимает значение $Cy + E$. Значение переменной fCur равно $f(0, 0)$ до первого выполнения команды L1, а затем после каждого выполнения команды увеличивается на $DeltaCur = Cy + E$ и записывается в $f(0, y)$. Это - правильное значение, так как оно вычислено в соответствии с формулой (3.43).

Следующий фрагмент заполняет элементы массива Delta(x), $x = 1, 2, \dots, 999$ числами $Delta(x) = Ax + D$. Кроме того, заполняются поля таблицы $f(x, y)$ для $x = 1, 2, \dots, 999$ и $y = 0$.

```
fCur = G; DeltaCur = D
for (x=1; x<1000; x++)
    f[x][0] = fCur += Delta[x] = DeltaCur += A;    /* L2 */
```

После каждого выполнения команды L2 переменная DeltaCur принимает значение $Ax + D$. Это значение хранится в соответствующем элементе массива Delta(x) и добавляется к переменной fCur. Ее значение равно $f(x-1, 0)$ до выполнения команды L2, а после выполнения этой команды принимает значение, которое пересылается в $f(x, 0)$. Это значение правильное, так как оно вычислено по формуле (3.42).

Наконец, я покажу фрагмент программы, который вычисляет оставшиеся значения массива $f(x, y)$ для $x = 1, 2, \dots, 999$ и $y = 1, 2, \dots, 999$.

```
for (y=1; y < 1000; y++) {
    fcur = f[0][y];
    for (x=1; x < 1000; x++)
        f[x][y] = fcur += Delta[x] += B;          /* L3 */
}
```

Логика этой программы почти такая же, как и в двух предыдущих фрагментах. Переменная Delta(x) принимает значения $Ax + By + D$ после каждого выполнения команды L3 и, следовательно, значение $f(x, y)$ вычисляется по формуле (3.42).

В приведенных фрагментах программ сложение выполняется только командами, которые имеют метки L1 - L3. Сложение выполняется $2(n_y - 1)$ раз в первом фрагменте, $2(n_x - 1)$ раз - во втором и $2(n_x - 1)(n_y - 1)$ раз в третьем, где n_x и n_y - количества значений переменных x и y соответственно. Общее количество сложений, таким образом, есть $2(n_x n_y - 1)$. Среднее количество вычислений на одно число $f(x, y)$ в таблице меньше, чем 2. Это, конечно, лучше, чем 8 умножений и 5 сложений, которые были бы необходимы при вычислениях непосредственно по формуле (3.39).

Представь теперь, что нужно табулировать квадратичную функцию не двух, а трех переменных x_1, x_2, x_3 ,

$$\begin{aligned}
 f(x_1, x_2, x_3) &= a_{11} x_1^2 + a_{22} x_2^2 + a_{33} x_3^2 \\
 &+ a_{12} x_1 x_2 + a_{23} x_2 x_3 + a_{13} x_1 x_3 \\
 &+ b_1 x_1 + b_2 x_2 + b_3 x_3 \\
 &+ c.
 \end{aligned}$$

В этом случае для вычисления значения в одной точке (x_1, x_2, x_3) требуются 15 умножений и 9 сложений. Посмотри, как увеличится сложность табулирования этой функции по сравнению с табулированием функции двух переменных.

Невероятно, но факт: как и прежде, для табулирования функции трех переменных требуются лишь два сложения на каждое число в таблице. Более того, это свойство сохраняется при табулировании квадратичной функции от произвольного количества переменных. Когда я понял это свойство, я опять, возможно, несколько наивно, сказал себе: как прекрасны эти квадратичные функции!

Ты, конечно, прав, но не думай, что другие функции хуже. Если Ты чуточку подумаешь, Ты увидишь, что кубическую функцию можно протабулировать, выполняя лишь 3 сложения на каждое число в таблице, и это опять не зависит от количества аргументов в этой функции. Вообще, табулирование полиномиальной функции k -ой степени при каком-нибудь количестве аргументов требует лишь k сложений на каждое число в таблице и ни одного умножения.

В лекции вы обратили внимание на то, что при условно независимых признаках, принимающих два значения, стратегия (3.3) состоит в разделении пространства наблюдений с помощью гиперплоскости. Я об этом результате слышал и читал много раз, например, в книге Дуды и Харта [Duda and Hart, 1973]. Это, конечно, приятное свойство, которое облегчает анализ стратегий, делает их более наглядными. Поэтому я удивлен, почему никто не отмечает, что подобное приятное свойство имеет место не только для бинарных признаков, но и для признаков, принимающих значения из какого-либо конечного множества. Я попробую выполнить это обобщение.

Пусть $k(i)$ - количество значений, которые принимает признак x_i . Без потери общности я могу принять, что признак x_i принимает значения из множества $X_i = \{0, 1, 2, \dots, k(i) - 1\}$. Я преобразую совокупность признаков $x = (x_1, x_2, \dots, x_n)$ в совокупность y бинарных признаков по следующему правилу. Я пронумерую новые признаки двумя индексами, $i = 1, 2, \dots, n$ и $j = 0, 1, 2, \dots, k(i) - 1$, и буду считать, что $y_{ij} = 1$, когда признак x_i принимает j -ое значение, и $y_{ij} = 0$, когда признак x_i не принимает j -ое значение. Дискриминантную функцию $\sum_{i=1}^n \log \frac{p_{x_i|1}(x_i)}{p_{x_i|2}(x_i)}$ можно записать в виде $\sum_{i=1}^n \sum_{j=0}^{k(i)-1} y_{ij} \alpha_{ij}$, где коэффициенты α_{ij} равны $\log \frac{p_{x_i|1}(j)}{p_{x_i|2}(j)}$, а стра-

тегия принятия решения приобретает вид

$$x \in \begin{cases} X_1, & \text{если } \sum_{i=1}^n \sum_{j=0}^{k(i)-1} \alpha_{ij} y_{ij} \geq \theta, \\ X_2, & \text{если } \sum_{i=1}^n \sum_{j=0}^{k(i)-1} \alpha_{ij} y_{ij} < \theta. \end{cases} \quad (3.45)$$

Это значит, что любую стратегию вида

$$x \in \begin{cases} X_1, & \text{если } \sum_{i=1}^n \log \frac{p_{X_{i1}}(x_i)}{p_{X_{i2}}(x_i)} \geq \theta, \\ X_2, & \text{если } \sum_{i=1}^n \log \frac{p_{X_{i1}}(x_i)}{p_{X_{i2}}(x_i)} < \theta, \end{cases} \quad (3.46)$$

можно выразить с помощью линейной дискриминантной функции.

Это, однако, еще не обобщает результат, приведенный в лекции. Если все признаки x_i бинарные, то есть $k(i) = 2$, то из равенства (3.45) следует, что стратегию (3.46) можно выразить гиперплоскостью в $2n$ -мерном пространстве. В лекции же доказывается, что в этом случае стратегию (3.46) можно выразить гиперплоскостью в n -мерном пространстве. На самом деле стратегию (3.45) можно представить более экономно, так, что потребуются не $\sum_{i=1}^n k(i)$ бинарных признаков, меньшее количество, то есть $\sum_{i=1}^n k(i) - n$. Я введу новые признаки $y'_{ij} = y_{ij} - y_{i0}$, $i = 1, 2, \dots, n$, $j = 1, 2, \dots, k(i)$, и новое пороговое значение $\theta' = \theta - \sum_{i=1}^n \alpha_{i0}$. Очевидно, что стратегия (3.45) эквивалентна стратегии

$$x \in \begin{cases} X_1, & \text{if } \sum_{i=1}^n \sum_{j=1}^{k(i)-1} \alpha_{ij} y'_{ij} \geq \theta', \\ X_2, & \text{if } \sum_{i=1}^n \sum_{j=1}^{k(i)-1} \alpha_{ij} y'_{ij} < \theta'. \end{cases}$$

Таким образом, результат, показанный в лекции для бинарных признаков, можно сформулировать в следующем, более общем виде.

Любая стратегия вида (3.46) реализуется с помощью гиперплоскости в $(\sum_{i=1}^n k(i) - n)$ -мерном пространстве. В частном случае, когда $k(i) = 2$ для всех i , размерность этого пространства равна n .

Я думаю, что эта более общая формулировка мне еще пригодится.

Полностью с Тобой согласны. Мы даже знаем, когда.

Март 1997

3.4 Библиографические замечания

В этой короткой лекции мы напоминаем читателю достаточно известные вещи, скажем прямо, не новые. Эта лекция введена в данный курс исключительно лишь для полноты изложения. Вопросы, поднятые в дискуссии

Иржи Пехой, обладают значительно большей новизной, чем материал собственно лекции.

Статистические модели, рассмотренные в лекции, значительно более полно исследованы в известных монографиях [Duda and Hart, 1973; Devijver and Kittler, 1982; Fukunaga, 1990]. Из исторических соображений укажем также статью [Chow, 1965], посвященную модели с условно независимыми признаками.

Реализуемость стратегии распознавания по бинарным независимым признакам с помощью линейной дискриминантной функции мы показали в лекции точно так же, как это показано в книгах [Duda and Hart, 1973; Fukunaga, 1990] и, возможно, в других учебниках. Обобщение этого факта на случай небинарных признаков получил Иржи Пеха.

Всестороннее исследование многомерных гауссовых случайных величин, без явной привязки к проблемам распознавания выполнено в учебнике [Anderson, 1958], который мы настойчиво рекомендуем читателю.

Глава 4

Обучение распознаванию образов

4.1 Мифы об обучении в распознавании

Становление любой области науки и техники, которая существенно изменяет человеческую деятельность, проходит как правило через следующие три этапа.

На первом этапе сочиняются сказки о том или ином чудесном устройстве, с помощью которого станет возможным то, что до сих пор было невозможно. Скажем, это мечта о ковре-самолете, который позволит летать по воздуху. На втором этапе конструируются модели, поожие скорее на игрушки, чем на что-то серьезное. При определенной доброжелательности можно считать, что эти модели уже делают сказку былью (то есть как-то уже летают), хотя они и непригодны для практического использования. Только на третьем этапе появляется изделие (к примеру, самолет), которое поначалу мало, а затем все больше и больше становится пригодным для практического использования.

Нет сомнений в важности третьего этапа. Однако очень важно понимать необходимость и первых двух этапов. Именно при сочинении сказок формируется цель исследования и разработки, причем в ее чисто прикладном, потребительском аспекте. При конструировании игрушечных моделей проверяется принципиальная осуществимость замысла, проверяется, не противоречит ли замысел фундаментальным законам природы, например, закону о сохранении энергии. Именно при конструировании игрушек получают основные теоретические результаты, которые лягут в основу будущего изделия. При конструировании изделия уже не до теории, так как на повестку дня выносятся проблемы уже совсем другого порядка. Поэтому не следует торопливо проскакивать через сочинение сказок и конструирование игрушек даже при очень нетерпеливом стремлении к скорому практически ощутимому результату. Это может привести к глубоким и длительным отрицательным последствиям. Не следует пренебрежительно относиться ни к сочинению мифов, ни к конструированию игрушек, ни к

разработке практического изделия. Но очень желательно как можно четче различать эти этапы: не называть прекрасные мечты теоретическим исследованием, а идейно богатую модель - изделием.

Современное распознавание является удивительной смесью мифов, игрушек и изделий, и в наибольшей степени это относится к той части проблемы, которая называется обучением распознаванию образов.

В предыдущих лекциях мы уже видели, что для конструирования стратегии распознавания, то есть функции $q: X \rightarrow K$ требуются знания. Речь идет как о знаниях об объекте распознавания, которые следует вложить в распознающую систему, так и о профессиональных знаниях о распознавании образов. Получение и тех, и других знаний - это длительный, утомительный и зачастую дорогостоящий процесс. Как реакция на эту действительность, возникают мечты о каком-то чудодейственном средстве типа скатерти-самобранки, с помощью которого можно было бы избежать подобных неприятностей.

В распознавании образов эти мечты имеют обычно следующую форму. "Существует система (генетическая, эволюционная, нейронная, или с иным экзотическим названием), которая работает в следующем режиме. Сначала система обучается на основе последовательности x_1, x_2, \dots, x_l примеров наблюдений, подаваемых на ее вход. Каждое наблюдение x_i из обучающего мультимножества сопровождается информацией k_i о реакции распознающей системы, которая считается правильной. После наблюдения l примеров такого правильного поведения обучаемая распознающая система оказывается способной давать правильную реакцию k на любое наблюдение x , в том числе и на те, которые не участвовали при обучении. Поскольку информация о требуемой стратегии вводилась не в явном виде, а только через обучающую последовательность, то обучаемую распознающую систему можно научить решать любую задачу распознавания."

Обычно бесполезно добиваться у автора подобных формулировок более конкретного изложения алгоритма обучения, и тем более, формулировки задачи, которая этим алгоритмом решается. Ожидаемые результаты представляются автору такими замечательными и легко достижимыми, что просто не хочется тратить время на такие мелочи, как однозначно понимаемая формулировка задачи и обоснованный вывод алгоритма для ее решения. Придуманная сказка настолько прекрасна, что любые приземленные вопросы ее только портят.

Реалистический взгляд на проблемы обучения приводит к известным в настоящее время формулировкам задач, которые даны в следующем разделе.

4.2 Три формулировки задач обучения в распознавании

Мы будем обозначать двумя различными способами условную вероятность наблюдения x при условии, что объект находится в состоянии k . Первое обозначение $p_{X|K}(x|k)$ уже использовалось раньше. Здесь функция $p_{X|K}$ рассматривается как функция двух переменных x и k , то есть функция

формата $X \times K \rightarrow \mathbb{R}$ с областью определения на декартовом произведении $X \times K$. Эту же функцию двух переменных можно понимать как семейство функций вида $X \rightarrow \mathbb{R}$ одной переменной x , в котором каждая отдельная функция определяется значением состояния k . Функция из этой совокупности, соответствующая состоянию $k \in K$, будет обозначаться $p_{X|k}$. Условная вероятность наблюдения x при условии состояния k есть, таким образом, значение функции $p_{X|k}$ в точке x , что будет обозначаться $p_{X|k}(x)$.

Необходимость обучения распознаванию образов возникает тогда, когда знаний о распознаваемом объекте недостаточно для построения необучаемой распознающей системы. Чаще всего это недостаточные знания о вероятностях $p_{X|K}(x|k)$, которые указывают, как могут выглядеть объекты, находясь в том или ином состоянии. Недостаточность знаний выражается в том, что известно лишь, что функция $p_{X|K}$ принадлежит определенному известному классу \mathcal{P} функций, но неизвестно, какая именно функция из класса \mathcal{P} имеет место в данной решаемой задаче.

Множество \mathcal{P} функций $p_{X|K}$ можно представить в виде семейства $\mathcal{P}(k)$, $k \in K$, множеств, где $\mathcal{P}(k)$ содержит и неизвестную функцию $p_{X|k}$. Множество \mathcal{P} или, что тоже самое, совокупность множеств $\mathcal{P}(k)$, $k \in K$, достаточно часто (грубо говоря, практически всегда) можно параметризовать заданием функции f двух переменных x и a , которая при каждом фиксированном значении параметра a определяет функцию $f(a): X \rightarrow \mathbb{R}$ одной переменной x . Множество $\mathcal{P}(k)$ есть, таким образом, $\{f(a) \mid a \in A\}$, где A - множество значений параметра a . Ограниченные знания о вероятностях $p_{X|K}(x|k)$, выраженные отношением $p_{X|K} \in \mathcal{P}$, обозначают существование такого значения a^* параметра a , что $p_{X|k} = f(a^*)$.

Сейчас нет необходимости конкретизировать смысл параметра a и преждевременно сужать наше рассмотрение. Ограничимся лишь следующим примером.

Пример 4.1 Параметризация $\mathcal{P}(k)$. Пусть \mathcal{P} - множество, состоящее из распределений вероятностей n -мерной гауссовой случайной величины с единичными дисперсиями независимых компонент. Тогда $\mathcal{P}(k)$ задается в параметрическом виде, как множество $\{f(\mu) \mid \mu \in \mathbb{R}^n\}$ функций $f(\mu): X \rightarrow \mathbb{R}$ вида

$$f(\mu)(x) = \prod_{i=1}^n \frac{1}{\sqrt{2\pi}} \exp\left(\frac{-(x_i - \mu_i)^2}{2}\right).$$

▲

На основании функций $p_{X|k}$, $k \in K$, каждая из которых известна с точностью до значения параметра a_k , можно построить и стратегию $q(x, a_1, a_2, \dots, a_n)$ с точностью до этих неизвестных параметров. Функция $q(x, a_1, a_2, \dots, a_n)$ указывает, как следовало бы распознать наблюдение x , если бы были известны параметры a_k , $k = 1, 2, \dots, n$, определяющие распределения. Говоря другими словами, можно построить параметрическое множество стратегий

$$Q = \{q(a_1, a_2, \dots, a_n) \mid a_1 \in A, a_2 \in A, \dots, a_n \in A\}$$

которое содержит в себе и требуемую стратегию.

Из предыдущих лекций мы уже знаем, что задачи распознавания допускают разумную постановку и в том случае, когда статистическая модель объекта известна не полностью, а известно лишь, что она принадлежит известному множеству моделей. Неизвестные параметры могут пониматься, как неслучайное вмешательство, которое влияет на статистическую модель объекта. Задачи распознавания в этом случае формулируются, как небайесовские задачи проверки сложных гипотез. Однако гарантированный уровень качества распознавания, которое достигается при использовании единственной стратегии для любой модели, может оказаться слишком малым. Это произойдет, когда априори известное множество моделей слишком обширно, как это имеет место в примере 4.1. Тогда окажется чересчур обширным и множество возможных стратегий и его невозможно заменить одной-единственной стратегией без существенных потерь. В этом случае следует как-то сузить либо класс моделей, либо класс стратегий, пользуясь дополнительной информацией, которую будем называть обучающей информацией. Она представлена в виде совокупности $T = ((x_1, k_1), (x_2, k_2), \dots, (x_l, k_l))$, где $x_i \in X$ и $k_i \in K$. Если обработка обучающей информации зависит от того, сколько раз определенная пара (x, k) встретилась в совокупности T , то эта совокупность понимается, как мультимножество, и называется *обучающим мультимножеством*. Если результат обучения зависит только от того, содержится ли определенная пара (x, k) в совокупности T по крайней мере один раз, и не зависит от того, сколько раз, то совокупность T понимается, как *обучающее множество*.

Обучение (с учителем) состоит в том, что на основании обучающей информации тем или иным убедительным способом выбирается одна-единственная стратегия из заранее заданного множества стратегий. Наиболее естественным критерием для выбора стратегии следовало бы считать риск

$$\sum_{x \in X} \sum_{k \in K} p_{XK}(x, k) W(k, q(x)), \quad (4.1)$$

достигаемый при ее использовании. Однако этот критерий невозможно использовать, так как вероятности $p_{XK}(x, k)$ неизвестны. Отсутствие знания о вероятностях $p_{XK}(x, k)$ до некоторой степени восполняется обучающей информацией. Различные формулировки задачи обучения отличаются друг от друга тем, каким образом критерий минимального риска заменяется другими критериями, которые могут быть вычислены на основании обучающей информации. При этом в рамках каждого подхода возникает разрыв между критерием, который надо было бы, но невозможно использовать, и вспомогательными критериями, которые могут быть вычислены. Этот разрыв может оставаться на совести (или интуиции, или опыте) разработчика или оцениваться каким-то более точным способом. Мы опишем наиболее популярные критерии, на которых основаны известные в настоящее время формулировки задач обучения. Затем мы покажем основные

результаты статистической теории обучения, касающиеся именно разрыва, о котором мы говорим.

4.2.1 Обучение как максимально правдоподобное оценивание

Пусть $p_{X|K}(x|k, a_k)$ - условные вероятности наблюдения x при условии состояния k , которые известны с точностью до неизвестного значения параметра a_k . Пусть имеется обучающее мультимножество

$$T = ((x_1, k_1), (x_2, k_2), \dots, (x_l, k_l)), \quad x_i \in X, \quad k_i \in K,$$

которое в данном случае представляет то же самое, что и случайная выборка в статистике.

Наиболее важное предположение об элементах мультимножества T состоит в том, что они считаются взаимно независимыми реализациями случайной величины с распределением вероятностей

$$p_{XK}(x, k) = p_K(k) p_{X|K}(x|k, a_k).$$

В этом случае можно вычислить вероятность обучающего мультимножества T для каждого набора неизвестных значений параметров $a = (a_k, k \in K)$,

$$L(T, a) = \prod_{i=1}^l p_K(k_i) p_{X|K}(x_i | k_i, a_{k_i}). \quad (4.2)$$

При обучении по методу наибольшего правдоподобия отыскиваются такие значения a_k^* , $k \in K$, которые максимизируют вероятность (4.2),

$$a^* = (a_k^*, k \in K) = \operatorname{argmax}_{(a_k, k \in K)} \prod_{i=1}^l p_K(k_i) p_{X|K}(x_i | k_i, a_{k_i}). \quad (4.3)$$

Затем совокупность a^* значений (a_k^* , $k \in K$) используется точно так же, как использовались бы действительные значения параметров, если бы они были известны. Это значит, что найденные значения (a_k^* , $k \in K$) подставляются в общее представление $q(x, a_1, a_2, \dots, a_n)$ стратегии и распознавание осуществляется стратегией $q(x, a_1^*, a_2^*, \dots, a_n^*)$.

Выражение (4.3) можно представить в различных эквивалентных формах, которые будут удобны в дальнейшем. Пусть $\alpha(x, k)$ обозначает, сколько раз пара (x, k) встретилась в обучающем мультимножестве. Для ненулевых вероятностей $p_{X|K}(x|k, a_k)$ можно записать

$$\begin{aligned} a^* &= \operatorname{argmax}_{(a_k, k \in K)} \prod_{x \in X} \prod_{k \in K} (p_K(k) p_{X|K}(x|k, a_k))^{\alpha(x, k)} \\ &= \operatorname{argmax}_{(a_k, k \in K)} \sum_{k \in K} \sum_{x \in X} \alpha(x, k) \log p_K(k) p_{X|K}(x|k, a_k). \end{aligned}$$

Мы видим, что оптимизационная задача (4.3) распадается на $|K|$ независимых оптимизаций в соответствии с требованием

$$\begin{aligned} a_k^* &= \operatorname{argmax}_{a_k} \sum_{x \in X} \alpha(x, k) \log p_K(k) p_{X|K}(x | k, a_k) \\ &= \operatorname{argmax}_{a_k} \sum_{x \in X} \alpha(x, k) \log p_{X|K}(x | k, a_k). \end{aligned} \quad (4.4)$$

По выражению (4.4) видно, что для определения значений a_k^* не нужно знать априорные вероятности $p_K(k)$.

4.2.2 Обучение по неслучайному обучающему множеству

Стратегия, полученная в результате обучения по методу наибольшего правдоподобия (4.3), (4.3), (4.4), зависит от мультимножества T . При этом существенно требование, чтобы обучающее мультимножество было составлено из взаимно независимых примеров. Кроме того, примеры входят в обучающее мультимножество в соответствии с тем же распределением вероятностей, с которым они будут подаваться на распознающую систему в процессе ее эксплуатации. Это очень жесткое требование, которое зачастую трудно выполнить. Иным является подход (в основном применяемый в распознавании изображений), при котором обучение производится не на случайных, а на тщательно подобранных примерах, которые:

1. достаточно хорошо представляют все множество распознаваемых объектов;
2. имеют в определенном неформальном смысле достаточно хорошее качество, они не искажены, и распознающая система должна каждый из этих примеров воспринимать, как весьма вероятный.

Эти неформальные соображения конкретизируются следующим образом. Пусть $X(k)$, $k \in K$, - множества примеров, тщательно отобранных человеком. Алгоритм распознавание нужно настроить так, чтобы условная вероятность каждого из примеров $x \in X(k)$ при условии k -ого состояния была достаточно высокой. Параметр a_k^* , определяющий распределение вероятностей $p_{X|K}$, следует выбрать, как

$$a_k^* = \operatorname{argmax}_{a_k \in A} \min_{x \in X(k)} p_{X|K}(x | k, a_k).$$

В этом случае обучающей информацией является не мультимножество, а множество примеров. Решение задачи зависит только от того, был ли представлен тот или иной пример при обучении или нет, и не зависит от того, сколько раз он был представлен.

Пример 4.2 Сравнение двух задач обучения для многомерных гауссовых распределений. Пусть $\mathcal{P}(k)$ - множество функций вида

$$p(x|k, \mu_k) = \prod_{i=1}^m \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{(x_i - \mu_{ik})^2}{2}\right).$$

При решении задачи обучения по методу максимального правдоподобия математическое ожидание μ_k^* должно оцениваться усреднением $(1/l) \sum_{i=1}^l x_i$ всех наблюдений объектов в k -ом состоянии, включенных в обучающее мультимножество. При обучении по максимумному оцениванию в качестве вектора μ_k^* должен выбираться центр минимальной окружности, включающей все примеры из обучающей выборки, наблюдавшиеся, когда объект находился в k -ом состоянии. ▲

4.2.3 Обучение по минимизации эмпирического риска

Пусть $W(k, d)$ - штрафная функция, а $Q = \{q(a) \mid a \in A\}$ - параметризованное множество стратегий вида $q(a): X \rightarrow D$, определенных с точностью до значения параметра a . Качество стратегии $q(a)$ - это риск R при использовании этой стратегии,

$$R(a) = \sum_{k \in K} \sum_{x \in X} p_{XK}(x, k) W(k, q(a)(x)). \quad (4.5)$$

Риск $R(a)$ следовало бы минимизировать надлежащим выбором значения a . Однако, риск нельзя измерить, так как статистическая модель $p_{XK}(x, k)$ неизвестна. К счастью, располагая обучающим мультимножеством $T = ((x_1, k_1), (x_2, k_2), \dots, (x_l, k_l))$, можно определить эмпирический риск

$$\hat{R}(a) = \frac{1}{l} \sum_{i=1}^l W(k_i, q(a)(x_i)), \quad (4.6)$$

который можно вычислить и считать приемлемой заменой действительного риска (4.5).

Третий подход к обучению в распознавании состоит в том, что на основании частичного знания статистической модели объекта создается параметрическое множество стратегий, из которого выбирается стратегия с минимальным эмпирическим риском (4.6).

Пример 4.3 Обучение, как минимизация эмпирического риска для многомерных гауссовых распределений. Посмотрим, что обозначает третий подход в том же примере, которым иллюстрировалось различие между обучением по наиболее правдоподобному и максимумному оцениванию, см. пример 4.2.

Пусть количество состояний объекта равно 2, равно как и количество принимаемых решений. Пусть также наблюдение при условии фиксированного состояния есть гауссова многомерная случайная величина с единичными дисперсиями независимых компонент. В этом случае множество стратегий состоит из стратегий, которые делят пространство наблюдений на два подмножества с помощью гиперплоскости. Третий подход к обучению распознаванию состоит в отыскании гиперплоскости, которая обеспечивает минимальный эмпирический риск (в частности, минимальное количество ошибок) на обучающем мультимножестве. ▲

Разнообразие подходов к обучению распознаванию (а мы упомянули далеко не все подходы) вовсе не значит, что следует один из них предпочесть другому. Мы думаем, что только в будущем, при учете каких-то дополнительных требований, которые сейчас еще недостаточно ясны, сфера применимости каждого подхода очертится более точно. Такая ясность сейчас еще не достигнута и с определенным преувеличением можно сказать, что в области обучения распознаванию мы еще не извлекли все возможное, что следовало бы получить на стадии сказок и игрушек. Тем не менее уже сейчас, пусть даже на основе недостаточно исследованных моделей, можно утверждать, что прекрасная сказка об обучаемых распознающих системах претерпела существенные изменения. Обучение полностью утратило черты палочки-выручалочки или скатерти-самобранки для сачков, которые надеются построить распознающую систему, минуя утомительный процесс тщательного изучения объектов, которые надо распознавать. Повидимому, надо распрощаться с мечтами о каком-то универсальном алгоритме, который как-то сам по себе построится.

Приведенные постановки задач обучения позволяют утверждать, что различие между обучаемыми распознающими алгоритмами и алгоритмами распознавания без обучения не так уж велико. Это различие состоит лишь в том, что необучаемая распознающая система реализует одну-единственную стратегию, а обучаемая предназначена для определенного, однозначно очерченного класса стратегий. Процесс же обучения есть нечто иное, как своеобразное распознавание, при котором на основании заданного наблюдения (обучающей информации) должно быть принято решение, о какой именно задаче распознавания из заданного класса идет речь. Для конструирования такого обучения конструктор должен сначала сам решить все задачи этого класса, то есть решить их в общем виде, а затем представить полученное общее решение в виде параметрического множества стратегий. Только после выполнения всей этой работы полученное общее решение встраивается в обучаемый распознающий алгоритм. Таким образом деформированная сказка об обучении, конечно же, потеряла большую часть своей привлекательности, но зато приобрела прозаичную прочность, потому что перестала быть чудом.

4.3 Основные понятия статистической теории обучения

При всем разнообразии подходов к обучению распознающих систем (а мы упомянули далеко не все) существует группа вопросов, которые возникают в рамках каждого подхода. Это вопросы о том, как связано качество стратегии при ее нормальной эксплуатации с качеством, которое эта стратегия показывала в процессе обучения. Точная формулировка этих вопросов и ответ на них составляет содержание статистической теории обучения. Основные положения этой теории были сформулированы Вапником и Червоненкисом. Перед изложением этих положений покажем основные проблемы обучения на неформальном уровне.

4.3.1 Неформальное описание проблем обучения распознаванию

Представим себе, что некто, кого мы будем называть заказчиком, пришел к кому-то другому, который называется разработчиком. Содержанием заказа является распознающая система. После, возможно, длительного диалога заказчик и разработчик пришли к выводу, что они одинаково понимают, что должна делать распознающая система. В качестве приложения к договору заказчик представляет экспериментальный материал - совокупность изображений x_1, x_2, \dots, x_l и соответствующих требуемых ответов k_1, k_2, \dots, k_l . Эти ответы считаются правильными в том смысле, что распознающая система должна дать ответ k_i при подаче на ее вход изображения x_i . Договорено, что распознающая система будет испытываться именно на этом экспериментальном материале, который таким образом становится и тестовым материалом. Пусть размер этого тестового материала достаточно большой, скажем, $l = 10000$. Предметом поставки является стратегия распознавания, и заказчика не интересует, каким образом она будет получена и будет ли она получена с помощью обучения. Его интересует лишь качество полученной стратегии. Примем для определенности, что этим качеством является вероятность ошибочного распознавания. Поскольку эту вероятность невозможно непосредственно измерить, заказчик и разработчик договорились использовать не вероятность ошибки, а количество ошибок, допущенных на прилагаемом экспериментальном материале. Обе стороны согласились, что такая замена оправдана (**ВНИМАНИЕ! СЛЕДУЕТ ОШИБКА!**) в силу закона больших чисел, который, грубо говоря, утверждает, что частота события при большом количестве испытаний мало отличается от вероятности.

Возникающие же здесь вопросы значительно сложнее и глубже, чтобы от них можно было отмахнуться легкомысленной и непродуманной ссылкой на закон больших чисел. Рассмотрим более пристально, в чем здесь состоит основная трудность.

Пусть Q - множество стратегий, а q обозначает одну из стратегий, $q \in Q$. Пусть T - обучающее мультимножество $(x_1, k_1), (x_2, k_2), \dots, (x_l, k_l)$, а T^* - множество всех возможных обучающих мультимножеств. Пусть $\hat{R}(T, q)$ - частота ошибочных решений, которые допускает стратегия q на мультимножестве T , а $R(q)$ - вероятность ошибочных решений, которые допускает стратегия q . Наконец, пусть $V: T^* \rightarrow Q$ - алгоритм обучения, который для каждого мультимножества $T \in T^*$ определяет стратегию $V(T) \in Q$. Следовательно, число $\hat{R}(T, V(T))$ - это качество распознавания, которое достигается на обучающем мультимножестве T при использовании стратегии, созданной на основании этого же мультимножества T . В соответствии с законом больших чисел, применяемом все еще в несколько вульгаризованном виде, для любой стратегии q случайное число $\hat{R}(T, q)$ при бесконечном увеличении объема мультимножества T стремится к вероятности $R(q)$. Это, несколько неточное, но в принципе правильное утверждение не гово-

рит равным счетом ничего о соотношении случайных чисел $\hat{R}(T, V(T))$ и $R(V(T))$. И когда со ссылкой на закон больших чисел утверждается, что при увеличении объема мультимножества T эти два числа в каком-то смысле становятся близкими, это значит, что слова "закон больших чисел" используются просто как заклинание без ясного понимания, что же этот закон утверждает.

На самом деле он ничего не утверждает о соотношении между этими числами. Числа $\hat{R}(T, V(T))$ и $R(V(T))$ совсем необязательно стремятся к одному и тому же пределу. В одних случаях такая сходимость имеет место, а в других - нет. Укажем пример именно второй ситуации.

Пример 4.4 Оценка риска и действительного риска могут отличаться и при сколь угодно большом объеме обучающего мультимножества.

Пусть множество X наблюдений - это одномерный континуум, например, интервал вещественных чисел. Пусть $p_{X|1}(x)$ и $p_{X|2}(x)$ - это два распределения плотности вероятности случайного значения x на множестве X при условии, что объект находится в первом или втором состоянии. Пусть известно, что плотности $p_{X|1}(x)$ и $p_{X|2}(x)$ ограничены сверху. Это значит, что вероятность каждого значения x равна нулю, равно, как и вероятность любого конечного подмножества значений. Пусть $V(T)$ - следующая стратегия, зависящая от $V(T)$: если $x \in X$ входит в обучающее мультимножество T , то есть равно некоторому x_i , то принимается решение k_i ; в противном случае принимается решение $k = 1$.

Для такого алгоритма обучения V справедливы два утверждения. Вероятность $R(V(T))$ ошибочного решения равна априорной вероятности $p_K(2)$ второго состояния. Действительно, стратегия $V(T)$ принимает решение $k = 1$ почти для всех наблюдений, за исключением разве что наблюдений, вошедших в обучающее мультимножество T . Этот факт не зависит от объема мультимножества T . Поскольку вероятность любого конечного мультимножества равна нулю, вероятность решения $k = 2$ также равна нулю.

С другой стороны, число $\hat{R}(T, V(T))$ равно нулю с вероятностью 1. Действительно, $\hat{R}(T, V(T)) = 0$ для любого мультимножества T , в котором любая пара (x, k) встречается не более одного раза. Вероятность же мультимножеств, в которых это условие не соблюдается, равна 0.

Мы имеем, таким образом, два случайных числа. Первое из них равно $p_K(2)$ с вероятностью 1, а второе равно нулю с вероятностью 1, и этот факт выполняется для обучающих мультимножеств любого объема. Числа $\hat{R}(T, V(T))$ и $R(V(T))$, таким образом, не становятся близкими ни в каком смысле. Это никак не противоречит закону больших чисел, потому что этот закон по поводу таких случайных чисел не говорит равным счетом ничего. ▲

Представленный в примере алгоритм обучения, конечно же, чистый обман, так как он основан на простом запоминании всего обучающего мультимножества и правильном распознавании только запомненных наблюдений.

ний. Обман или самообман стал здесь возможен в результате непонимания, что эмпирический риск $\hat{R}(T, V(T))$ не всегда служит состоятельной оценкой действительного риска $R(V(T))$. Понятие обучения распознаванию должно быть разумно сужено так, чтобы допускались не любые алгоритмы $V: T^* \rightarrow Q$ обучения. В первую очередь следует исключить из рассмотрения алгоритмы, при которых возможен приведенный в примере явный обман, а также все те алгоритмы, где такой обман возможен, но в более скрытом виде. Это те алгоритмы, в которых эмпирический риск не является состоятельной оценкой действительного риска. Множество всех оставшихся алгоритмов не однородно. Там есть худшие и лучшие в зависимости от того, насколько точной оценкой риска является эмпирический риск. Мы рассмотрим это различие, но сначала приведем точную формулировку знаменитого закона больших чисел, чтобы понять, как следует его обобщить, чтобы в сферу его действия попали и вопросы, возникающие при обучении распознаванию.

Закон больших чисел известен в нескольких различных формулировках. При использовании нашей терминологии одна из этих формулировок имеет следующий вид. Для любой стратегии q выполняется равенство

$$P \{ |\nu_l(q) - p(q)| > \varepsilon \} \leq \exp(-2\varepsilon^2 l), \quad (4.7)$$

где l - объем обучающего мультимножества, $\nu_l(q)$ - относительная частота ошибок, которые стратегия q допускает на мультимножестве объема l , $p(q)$ - вероятность ошибок, допускаемых стратегией q , $P\{\}$ - вероятность события, записанного в скобках. Неравенство (4.7) показывает, что эксперимент по оценке качества $p(q)$ стратегии q характеризуется тремя параметрами. Первый параметр - это длительность l эксперимента. Вторым параметром - это точность ε оценки качества $p(q)$ стратегии. Этот параметр присутствует в утверждении "вероятность $p(q)$ не превышает $\nu_l(q) + \varepsilon$ и не меньше $\nu_l(q) - \varepsilon$ " или, как это пишется в практических руководствах,

$$p(q) = \nu_l(q) \pm \varepsilon. \quad (4.8)$$

И наконец, третий параметр - это надежность η утверждения (4.8), которым подчеркивается тот факт, что утверждение (4.8) может быть ошибочным. Надежность - это вероятность того, что утверждение (4.8) неверно. Закон больших чисел (4.7) утверждает, что все эти три параметра противоречивы. Например, с помощью короткого эксперимента невозможно обеспечить и большую точность, и большую надежность. В то же время закон (4.7) показывает, что никакая пара параметров из указанных двух не является противоречивой. Так, для любого объема l можно дать сколь угодно надежную формулировку вида (4.8), то есть обеспечить сколь угодно малую вероятность $\eta > 0$. Однако это будет достигнуто лишь ценой потери точности, то есть ценой большого значения ε . Для нас важно, что любую точность, то есть малую погрешность $\varepsilon > 0$, и любую надежность, то есть малую вероятность $\eta > 0$ можно достигнуть при достаточно боль-

шом объеме эксперимента. Это объем

$$l \geq \frac{-\ln \eta}{2\varepsilon^2}. \quad (4.9)$$

Пример 4.5 Точность, надежность о объем эксперимента. *Вооруженный неравенством (4.9), наш воображаемый заказчик может сейчас более точно сформулировать, какой результат экспериментальной проверки алгоритма он будет считать положительным. Он задается величинами $\varepsilon = 2\%$ и $\eta = 0.1\%$ и с использованием формулы (4.9) получает, что*

$$l \geq \frac{-\ln 0.001}{2(0.02)^2} = \frac{-(-6.9077)}{0.0008} \doteq 8635.$$

Теперь он может по крайней мере для себя сформулировать правило приемки алгоритма распознавания. Если алгоритм безошибочно распознал 8635 случайно выбранных изображений, то можно считать, что при последующем его использовании он вряд ли будет допускать ошибки чаще, чем в 2% случаев. Слова "вряд ли" в приведенном утверждении следует понимать, как "с вероятностью, не большей, чем 0.001".

▲

На основании уже приобретенного опыта и знаний заказчик запасается тестовым материалом достаточного объема и отправляется уже не к разработчику, а на рынок программного обеспечения с намерением приобрести именно ту программу, которая на его тестовом материале не допустит ни одной ошибки. Так он надеется оградить себя от прямого обмана: ведь программа, которую он приобретет, была разработана без знания тестового материала и поэтому не могла быть приспособлена именно к этому тестовому материалу. И здесь заказчик, превратившийся уже в покупателя, снова ошибается.

Программа, оказавшаяся на прилавке, действительно была разработана без учета тестового материала и, следовательно, не могла быть приспособленной именно к этому тестовому материалу. Но программа, оказавшаяся у покупателя, уже зависит от тестового материала. Это та программа из множества лежащих на прилавке, которая на данном тестовом материале безошибочно сработала. Таким образом, покупатель сам себя подвергает тому же обману, которому раньше его подверг разработчик. Точно так же, как раньше разработчик, покупатель из имеющегося множества алгоритмов распознавания выбирает тот, который на имеющемся тестовом материале допускает минимальное количество ошибок.

Допустим, что являются плохими все имеющиеся в наличии алгоритмы, из которых производится выбор. Однако эти алгоритмы различны: одни допускают ошибки на одних изображениях, другие - на других. Если же этих плохих алгоритмов достаточно много, то какой-то из них, будучи тоже плохим, может дать положительные результаты именно на данном тестовом материале. Поэтому при выборе алгоритма из определенного множества алгоритмов объем тестового материала должен быть таким, чтобы

ни один плохой алгоритм не прошел этот тест, так как заказчик, или покупатель, или разработчик произведут плохой выбор, когда хотя бы один плохой алгоритм успешно пройдет через тест. Таким образом, при подготовке тестового материала следует учитывать, насколько обширен класс алгоритмов, из которых производится выбор. Объем тестового материала должен выбираться так, чтобы оказалось маловероятным, что хотя бы один плохой алгоритм будет казаться хорошим.

Для надежного выбора стратегии распознавания из заданного множества стратегий недостаточно, чтобы объем тестового материала удовлетворял условию

$$P \{ |\nu_l(q) - p(q)| > \varepsilon \} < \eta, \quad (4.10)$$

которое слишком слабое. Он должен удовлетворять условию

$$P \left\{ \sup_{q \in Q} |\nu_l(q) - p(q)| > \varepsilon \right\} < \eta,$$

которое: (а) более сильное, чем условие (4.10); (б) зависит от множества Q алгоритмов, из которых производится выбор.

Таким образом наш воображаемый заказчик-покупатель-разработчик приблизился к пониманию тех вопросов, которые составляют содержание статистической теории распознавания. Точная формулировка этих результатов приводится в следующем подразделе.

4.3.2 Основы статистической теории обучения распознаванию по Вапнику и Червоненкису

Пусть Q - множество стратегий вида $q: X \rightarrow K$, $p_{XK}: X \times K \rightarrow \mathbb{R}$ - статистическая модель распознаваемого объекта, которая неизвестна. Вероятность $p(q)$ ошибочного распознавания при использовании стратегии q определяется по формуле

$$p(q) = \sum_{x \in X} \sum_{k \in K} p_{XK}(x, k) W(k, q(x)),$$

в которой W - это функция потерь,

$$W(k, k^*) = \begin{cases} 1, & \text{если } k \neq k^*, \\ 0, & \text{если } k = k^*, \end{cases} \quad (4.11)$$

k^* - это решение, принимаемое стратегией q о состоянии объекта, а k - действительное состояние.

Пусть $\nu_l(q)$ - это случайная величина, а именно, частота ошибочных решений при применении стратегии q к случайному обучающему множеству $T = ((x_1, k_1), (x_2, k_2), \dots, (x_l, k_l))$ с длиной l ,

$$\nu_l(q) = \frac{1}{l} \sum_{i=1}^l W(k_i, q(x_i)).$$

Из закона больших чисел известно, что взаимосвязь вероятности $p(q)$ и случайной величины $\nu_l(q)$ выражается следующим неравенством, которое справедливо для любого $\varepsilon > 0$,

$$P \{ |\nu_l(q) - p(q)| > \varepsilon \} \leq \exp(-2\varepsilon^2 l). \quad (4.12)$$

Мы будем говорить о двух характеристиках стратегий. Стратегии могут быть хорошими или плохими и прошедшими тест или не прошедшими. Стратегия q считается хорошей, если $p(q) \leq p^*$, и плохой, если $p(q) > p^*$. Стратегия считается прошедшей тест, когда $\nu_l(q) \leq p^* - \varepsilon$, и не прошедшей тест в противном случае. На основе экспериментальной проверки стратегии невозможно непосредственно ответить, является ли она хорошей или плохой. Однако на основе неравенства (4.12) можно утверждать, что если стратегия прошла тест, то есть $\nu_l(q) \leq p^* - \varepsilon$, то она скорее всего хорошая. Можно оценить надежность теста, то есть вероятность, что плохая стратегия пройдет тест. В соответствии с (4.12) эта вероятность не превышает $\exp(-2\varepsilon^2 l)$.

При обучении распознающей системы тестируется не одна-единственная стратегия q , а множество Q стратегий. Тем или иным способом выбирается стратегия $q^* \in Q$, для нее затем проверяется, превосходит ли число $\nu_l(q^*)$ допустимое значение $p^* - \varepsilon$, после чего стратегия окончательно принимается или отвергается. Эта процедура приводит к ложному результату, когда в множестве Q содержится хотя бы одна плохая стратегия, которая успешно проходит тест. Вероятность этой ситуации мала, если мала вероятность

$$P \{ \exists q \in Q (|\nu_l(q) - p(q)| > \varepsilon) \}$$

или, что то же самое, если мала вероятность

$$P \left\{ \max_{q \in Q} |\nu_l(q) - p(q)| > \varepsilon \right\}. \quad (4.13)$$

Надежность процедуры обучения в целом определяется вероятностью (4.13), а не вероятностью (4.12). Вероятности (4.12) и (4.13) существенно различны. Вероятность (4.12) можно сделать сколь угодно малой при любом, сколь угодно малом значении $\varepsilon > 0$. Для этого нужно только выбрать достаточно большой объем l обучающего мультимножества. Именно в силу этого замечательного свойства неравенство (4.12) составляет фундамент классической математической статистики.

К сожалению, аналогичное свойство для вероятности (4.13) не гарантируется. Вероятность (4.13) не обязательно стремится к нулю при неограниченном росте объема l . Вероятность (4.13) стремится или не стремится к нулю при $l \rightarrow \infty$ в зависимости от множества Q стратегий. Этот факт составляет сердцевину научной проблематики обучения в распознавании, которую невозможно лихо отщелкнуть ссылкой на закон больших чисел.

Покажем наиболее важные свойства вероятности (4.13) и начнем с простейшего случая, когда множество Q состоит из конечного количества N

стратегий. В этом случае

$$P \left\{ \max_{q \in Q} |\nu_l(q) - p(q)| > \varepsilon \right\} \leq \sum_{q \in Q} P \{ |\nu_l(q) - p(q)| > \varepsilon \} \\ \leq N \exp(-2\varepsilon^2 l). \quad (4.14)$$

В принятых нами терминах это неравенство имеет следующий содержательный смысл.

1. Q - это множество, состоящее из N стратегий вида $X \rightarrow K$;
2. T - это случайное мультимножество $(x_1, k_1), (x_2, k_2), \dots, (x_l, k_l)$ объема l , имеющее вероятность $\prod_{i=1}^l p_{XK}(x_i, k_i)$; $p_{XK}(x, k)$ - это совместная вероятность наблюдения $x \in X$ и состояния $k \in K$;
3. Пусть неким способом из множества Q была извлечена некоторая стратегия q , и эта стратегия прошла тест.
4. В силу неравенства (4.14) вероятность того, что стратегия q окажется плохой, не превышает $N \exp(-2\varepsilon^2 l)$.

Приведенное утверждение обладает огромной общностью. Его справедливость не зависит ни от множества Q стратегий, ни от статистической модели p_{XK} , и даже не зависит от алгоритма обучения. Ведь мы же ничего не сказали о том, как выбирается стратегия q из множества Q , по методу ли максимально правдоподобного оценивания, или минимизацией эмпирического риска, или каким-то еще иным методом.

Это утверждение выражено Вапником и Червоненкисом в виде следующей, правда, не очень аккуратно сформулированной теоремы, которую мы приводим почти дословно.

Теорема 4.1 Вапник и Червоненкис. Оценка объема обучающего мультимножества. *Если из N стратегий выбирается стратегия с минимальной частотой ν ошибок на обучающем мультимножестве объема l , то с вероятностью $1 - \eta$ можно утверждать, что вероятность ошибочного решения будет меньше, чем $\nu + \varepsilon$, если*

$$l = \frac{\ln N - \ln \eta}{2\varepsilon^2}. \quad (4.15) \quad \blacktriangle$$

Неаккуратность в приведенной формулировке мы усматриваем в словосочетании "с вероятностью $1 - \eta$ можно утверждать". Несмотря на эту нашу придирку, эта легко доказываемая теорема выражает самое фундаментальное свойство обучения в распознавании: чем обширнее класс априори возможных стратегий, то есть чем меньше исследован объект распознавания, тем длительней должно быть обучение, если только мы хотим, чтобы результат обучения был надежным. А он должен быть надежным всегда.

Если посмотреть на неравенство (4.15) с практической точки зрения, то длительность обучения оценивается с большим запасом, то есть слишком грубо. При решении практических задач обучения требуется, как правило, выбрать стратегию из бесконечного параметрически заданного множества стратегий. В этом случае рекомендация (4.15) не дает ничего, так как

утверждает только, что достаточная надежность распознавания достигается при бесконечном объеме выборки. Вапник и Червоненкис существенно улучшают оценку (4.15) и показывают, что длительность обучения определяется не такой грубой характеристикой множества стратегий, как количество стратегий, а более тонкими характеристиками. Этими характеристиками являются энтропия множества стратегий, функция роста и емкость множества. Определим эти понятия.

Пусть Q - множество стратегий, а x_1, x_2, \dots, x_l мультимножество наблюдений. Стратегии $q_1 \in Q$ и $q_2 \in Q$ называются *эквивалентными* на мультимножестве x_1, x_2, \dots, x_m , если $q_1(x_i) = q_2(x_i)$ для любого i . Таким образом каждое мультимножество наблюдений определяет эквивалентность на множестве стратегий Q . Пусть $\Delta(Q, x_1, x_2, \dots, x_l)$ - количество классов эквивалентности. Говоря иными словами, $\Delta(Q, x_1, x_2, \dots, x_l)$ - это количество различных классификаций мультимножества x_1, x_2, \dots, x_l , которые могут быть осуществлены стратегиями из множества Q .

Пример 4.6 Сравнение вещественных чисел с порогом. Пусть x_1, x_2, \dots, x_l вещественные числа, а q - стратегия, принимающая одно из двух решений и имеющая следующий вид. Она характеризуется порогом θ и по наблюдению x принимает первое решение при $x < \theta$, и второе - при $x \geq \theta$. Очевидно, число $\Delta(Q, x_1, x_2, \dots, x_l)$ на единицу больше, чем количество различных элементов в мультимножестве x_1, x_2, \dots, x_l . Понятно, таким образом, что $\Delta(Q, x_1, x_2, \dots, x_l) = l + 1$ почти всегда, так как вероятность одинаковых элементов в случайной выборке равна нулю. ▲

Поскольку мультимножество наблюдений случайно, случайным является и число $\Delta(Q, x_1, x_2, \dots, x_l)$. Математическое ожидание логарифма этого числа,

$$\sum_{x_1 \in X} \dots \sum_{x_l \in X} \sum_{k_1 \in K} \dots \sum_{k_l \in K} \prod_{i=1}^l p_{XK}(x_i, k_i) \log \Delta(Q, x_1, x_2, \dots, x_l), \quad (4.16)$$

будет обозначаться $H_l(Q)$ и называться *энтропией* множества Q стратегий на мультимножествах объема l .

Нас интересует главный вопрос о том, какой должна быть длительность l обучения, чтобы результаты обучения были одновременно и надежными, и точными. Это значит, что вероятность

$$P \left\{ \max_{q \in Q} |\nu_l(q) - p(q)| > \varepsilon \right\}$$

должна быть достаточно малой при достаточно малом ε . Однако прежде, чем ответить на этот вопрос, мы должны исключить из рассмотрения все те ситуации, при которых эта вероятность не стремится к нулю вообще. В этом случае обучение вообще теряет смысл, так как частота ошибок, достигнутая на мультимножествах сколь угодно большой длины, не имеет ничего общего с вероятностью ошибок. Исчерпывающее описание таких безнадежных ситуаций дает следующая теорема.

Теорема 4.2 Вапник и Червоненкис. Необходимые и достаточные условия равномерной сходимости эмпирического риска к действительному риску. *Вероятность*

$$P \left\{ \max_{q \in Q} |\nu_l(q) - p(q)| > \varepsilon \right\} \quad (4.17)$$

стремится к нулю при $l \rightarrow \infty$ и при любом $\varepsilon > 0$ тогда и только тогда, когда удельная энтропия $H_l(Q)/l$ стремится к нулю при $l \rightarrow \infty$. \blacktriangle

Доказательство. Доказательство теоремы 4.2 очень длинное и непростое [Vapnik and Chervonenkis, 1974]. \blacksquare

Будучи исчерпывающим ответом на очень трудный вопрос, теорема 4.2, как и любые другие теоремы такой общности, с трудом применима для ответа на вопрос в каждом частном случае. Теорема сводит вопрос о сходимости к нулю вероятности

$$P \left\{ \max_{q \in Q} |\nu_l(q) - p(q)| > \varepsilon \right\}$$

к вопросу о сходимости к нулю удельной энтропии $H_l(Q)/l$. Этот вопрос не менее трудный. Достаточно взглянуть на формулу (4.16) и заметить, что для вычисления энтропии требуется знание вероятностей $p_{XK}(x, k)$, которые неизвестны. Поэтому важными являются следующие два шага, которые заглубляют теорему 4.2, но приводят к конструктивным рекомендациям.

Первый шаг связан с понятием функции роста. Пусть $\Delta_l(Q, x_1, x_2, \dots, x_l)$ - количество возможных классификаций мультимножества x_1, x_2, \dots, x_l стратегиями из множества Q . Определим число $m_l(Q)$, как

$$m_l(Q) = \max_{x_1, \dots, x_l} \Delta_l(Q, x_1, x_2, \dots, x_l).$$

Последовательность чисел $m_l(Q)$, $l = 1, 2, \dots, \infty$, будем называть *функцией роста*. Число $\log m_l(Q)$ связано с энтропией $H_l(Q)$ очевидным неравенством $\log m_l(Q) \geq H_l(Q)$. Следовательно, если

$$\lim_{l \rightarrow \infty} \frac{\log m_l(Q)}{l} = 0, \quad (4.18)$$

то $\lim_{l \rightarrow \infty} \frac{H_l(Q)}{l} = 0$ и условие (4.18) становится достаточным (но не необходимым) условием сходимости вероятности (4.17) к нулю. Проверка условия (4.18) менее проблематична, так как она не требует знания вероятностей $p_{XK}(x, k)$. На основе функции роста можно не только проверять сходимость вероятности (4.17) к нулю, но и оценивать уклонение эмпирического риска от действительного.

Теорема 4.3 Уклонение эмпирического риска от действительного.

$$P \left\{ \max_{q \in Q} |\nu(\bar{q}) - p(q)| > \varepsilon \right\} < 3 m_{2l}(Q) e^{-\frac{1}{4} \varepsilon^2 (l-1)}. \quad (4.19)$$



Неравенство (4.19) очень похоже на неравенство (4.14), полученное для случая конечных множеств Q . Функция роста в неравенстве (4.19) играет ту же самую роль, что и количество стратегий в неравенстве (4.14). Это значит, что функция роста характеризует сложность бесконечных множеств Q подобно тому, как количество стратегий характеризует конечные множества. Если же речь идет только о конечных множествах стратегий, то функция роста является более точной характеристикой, чем просто количество стратегий в множестве. Она учитывает то, что можно было назвать структурой множества стратегий, учитывает их разнообразие, в то время, как количество стратегий это разнообразие просто игнорирует.

Следующий шаг упрощения основан на понятии емкости множества стратегий. В западной литературе это понятие известно также как *VC-размерность*, названная по первым буквам в фамилиях Вапник и Червоненкис. Мы предпочитаем более выразительное название емкость, данное авторами в их первоначальных публикациях. Неформально говоря, *емкость множества стратегий* - это минимальное количество наблюдений, которые невозможно классифицировать всеми возможными способами, используя стратегии только из данного множества.

Проиллюстрируем это понятие на примере ранее рассмотренного множества стратегий, а потом определим его точно.

Пример 4.7 Емкость множества стратегий. Пусть X - множество вещественных чисел, а Q , как и в предыдущем примере, - это множество стратегий, принимающих одно из двух решений. Каждая стратегия из Q характеризуется порогом θ и по наблюдению $x \in X$ принимает первое решение при $x < \theta$ и второе решение при $x \geq \theta$. Для любого наблюдения $x \in X$ существует такая стратегия q' в множестве Q , которая по этому наблюдению x принимает первое решение, и какая-то другая стратегия q'' , которая по этому же наблюдению принимает второе решение.

Пусть x_1 и x_2 - две различные точки на вещественной оси X , такие, что $x_1 < x_2$. Для этой пары точек существует классификация, которую не реализует ни одна стратегия из множества Q . Действительно, в рассматриваемом множестве Q стратегий нет стратегии, которая по наблюдению x_2 принимала бы первое решение, а по наблюдению x_1 - второе, так как $x_2 > x_1$. Таким образом, используя стратегии из данного множества Q , любую точку можно классифицировать обеими, то есть всеми возможными способами, но никакую пару точек уже невозможно классифицировать всеми возможными способами. В этом случае говорят, что емкость рассматриваемого множества Q стратегий равна 2. ▲

Пример 4.8 Емкость более обширного множества стратегий. Расширим множество Q стратегий из предыдущего примера и посмотрим, как это изменит емкость множества. Пусть расширенное множество Q включает стратегии, которые характеризуются двумя параметрами α ,

θ . По наблюдению x принимается первое решение при $ax < \theta$ и второе - при $ax \geq \theta$. Пусть x_1, x_2 - две различные точки на вещественной оси X , такие, что $x_1 \neq x_2$. Существуют 2^2 классификации этой пары точек на два класса, и каждая из них реализуется какой-то стратегией из введенного множества Q .

Рассмотрим теперь некоторую тройку точек $x_1 < x_2 < x_3$. Существуют 2^3 классификации этой тройки на два класса, но не все они реализуются стратегиями из введенного множества Q . В множестве Q отсутствует, например, стратегия, которая принимает первое решение по наблюдениям x_1 и x_3 и второе - по наблюдению x_2 . Данное множество стратегий таково, что существуют пары точек, которые можно расклассифицировать на два класса всеми возможными способами, используя стратегии из Q , но никакую тройку точек уже невозможно расклассифицировать всеми возможными способами. Про такое множество стратегий говорят, что его емкость равна 3. ▲

Понятие емкости множества Q стратегий вида $q: X \rightarrow \{1, 2\}$ определяется следующим образом. Пусть x_1, x_2, \dots, x_l - мультимножество наблюдений, $C_l: \{1, 2, \dots, l\} \rightarrow \{1, 2\}$ - разбиение (классификация) этого мультимножества на два класса, C_l^* - множество всех возможных классификаций вида $\{1, 2, \dots, l\} \rightarrow \{1, 2\}$. Это множество, очевидно, состоит из 2^l классификаций.

Число r есть емкость множества Q стратегий вида $q: X \rightarrow \{1, 2\}$, если

1. Существует такое мультимножество x_1, x_2, \dots, x_{r-1} объемом $r - 1$, что для любой классификации $C_{r-1} \in C_{r-1}^*$ существует такая стратегия $q \in Q$, что $q(x_i) = C_{r-1}(i)$, $i = 1, 2, \dots, r - 1$;
2. Для любого мультимножества x_1, x_2, \dots, x_r длины r существует такая классификация $C_r \in C_r^*$, что не существует стратегии $q \in Q$, для которой бы выполнялось $q(x_i) = C_r(i)$, $i = 1, 2, \dots, r - 1$.

Емкость множества Q стратегий можно эквивалентным образом определить на основе функции роста. Пусть

$$m_1(Q), m_2(Q), \dots, m_{r-1}(Q), m_r(Q), \dots, m_l(Q), \dots \quad (4.20)$$

- функция роста для множества стратегий Q . В этой последовательности число $m_1(Q)$, естественно, не превышает 2^1 , $m_2(Q)$ не превышает 2^2 , и вообще, l -ый элемент $m_l(Q)$ не превышает 2^l . Если какой-то элемент, скажем, l -ый элемент $m_l(Q)$ равен 2^l , то предыдущий, $(l - 1)$ -ый элемент равен 2^{l-1} . Действительно, если какое-то мультимножество можно расклассифицировать любым возможным способом, то это же можно сделать и для любой его части. Это значит, что последовательность (4.20) состоит из двух связанных подпоследовательностей. Первая из них (возможно пустая) состоит из чисел, равных 2^l , где l - порядковый номер числа в последовательности (4.20). Числа во второй подпоследовательности меньше, чем 2^l .

Мощность множества Q стратегий - это номер первого числа во второй подпоследовательности, то есть минимальное число l , для которого выполняется $m_l(Q) < 2^l$.

Уже был указан тот очевидный факт, что если r - емкость множества Q , то $m_l(Q) = 2^l$ для любого $l < r$. Намного менее очевидно, что и числа $m_l(Q)$, $l \geq r$, не могут быть любыми. Они также зависят от емкости, как об этом говорит следующая теорема.

Теорема 4.4 Ограничение сверху для функции роста. *Если r емкость множества Q , то неравенство*

$$m_l(Q) \leq \frac{1,5 l^{r-1}}{(r-1)!} \quad (4.21)$$

справедливо для любого $l \geq r$. ▲

Доказательство. Мы ссылаемся на [Vapnik and Chervonenkis, 1974]. ■

В силу теоремы 4.4 из неравенства (4.19) следует неравенство

$$\eta < 4,5 \frac{(2l)^{r-1}}{(r-1)!} e^{-\frac{1}{4}\varepsilon^2(l-1)}, \quad (4.22)$$

которое более слабое, чем (4.19), но зато кратко и в явном виде представляет взаимосвязь всех трех параметров, характеризующих обучение: точность ε , надежность η и длительность l . Множество Q стратегий представлено в формуле (4.22) своей единственной характеристикой - емкостью r .

Напомним основные опорные моменты статистической теории обучения, которые приводят к этому результату.

1. Для обоснования алгоритмов обучения распознаванию недостаточно знать, как ведет себя вероятность

$$P\{|\nu_l(q) - p(q)| > \varepsilon\} \quad (4.23)$$

при $l \rightarrow \infty$. Необходимо знать поведение более сложной вероятности

$$P\{\max_{q \in Q} |\nu_l(q) - p(q)| > \varepsilon\}. \quad (4.24)$$

2. Фундаментальное различие вероятностей (4.23) и (4.24) состоит в том, что при $l \rightarrow \infty$ вероятность (4.23) стремится к нулю при любой стратегии q и любом $\varepsilon > 0$, в то время как вероятность (4.24) иногда стремится к нулю, а иногда - нет, в зависимости от сложности множества Q стратегий.
3. Исчерпывающей характеристикой множества стратегий Q , при которых обеспечивается сходимость вероятности (4.24) к нулю, является поведение энтропии множества Q (см. теорему 4.2). Хотя энтропия множества стратегий однозначно определена, ее вычисление невозможно при известной статистической модели распознаваемого объекта.
4. Функцию роста множества Q можно вычислить конструктивно, так как она не зависит от статистической модели объекта. Если функция роста

- известна, то вероятность (4.24) можно оценить сверху и определить скорость, с которой эта вероятность стремится к нулю, см. формулу (4.19).
5. Простейшей характеристикой множества Q является емкость, которая в явном виде входит в неравенство, связывающее точность, надежность и длительность обучения.

4.4 Критический взгляд на статистическую теорию обучения

В силу своей математической безупречности изложенные основы статистической теории обучения пользуются всеобщим признанием и не нуждаются в каком-либо снисхождении. Они могут выдержать любое критическое рассмотрение, в частности, предложенное в данном разделе.

Результаты статистической теории обучения, как и любой теории, имеют вид "Если А, то В". Давайте придирчиво оценим практическую приемлемость рекомендаций статистической теории обучения, например, элегантной формулы (4.22). Естественно, мы здесь не преуспеем, если будем исходить из тех же предпосылок, из которых формула (4.22) была выведена. Если мы хотим подвергнуть сомнению практическую приемлемость этой формулы, мы должны прежде всего спросить, насколько естественны, прозрачны те условия (утверждение А в нашей импликации), при которых эта формула верна. Если условие А окажется практически приемлемым, то возникнет вопрос о содержательном смысле следствия В, обозначает ли оно именно то, что требуется.

Основная предпосылка статистической теории обучения состоит в случайности обучающего мультимножества $(x_1, k_1), (x_2, k_2), \dots, (x_l, k_l)$. Предполагается, что элементы этого мультимножества взаимно независимы, и их вероятности $p_{X,K}(x, k)$ те же, что и при последующей эксплуатации обученного алгоритма. Наши придирки состоят вовсе не в том, что это предположение иногда не выполняется или выполняется слишком редко. Это может происходить с любым предположением. Проблема состоит в том, что никакими экспериментальными средствами не удастся проверить, выполняется ли оно в конкретной практической ситуации или нет. Проблема соответствия теоретических моделей и реальности вообще сложна. Но она особенно сложна, когда ставится вопрос о соответствии статистической модели и реальности. Эта досадная трудность хорошо (пусть даже несколько преувеличенно) выражена в советском rozpoзнавальческом фольклоре.

Пример 4.9 Геолог как пользователь обучаемой rozpoзнающей системы. [Zagorujko, 1999] *Однажды некий геолог пришел к специалисту по rozpoзнаванию образцов со своей задачей. Она состоит в том, что по признакам образца породы следует определить, является ли она ценной по содержанию в ней железа или нет. Заказчик гарантирует, что стратегия rozpoзнавания в его задаче состоит в вычислении линейной функции от измеряемых признаков и сравнении значения этой функции с порогом. Заказчику, таким образом, известен класс стратегий, который содержит требуемую стратегию. Его просьба к специалисту по rozpoзнаванию со-*

стоит лишь в том, чтобы профессионально были найдены параметры линейной функции и значение порога. Как видим, это идеальный заказчик, о котором только может мечтать специалист в распознавании.

Естественно, исполнитель предполагаемого контракта просит у заказчика обучающее мультимножество, и оно тоже у заказчика имеется. Он вытаскивает из рюкзака две глыбы, содержащую и не содержащую железо. Такой объем мультимножества представляется исполнителю слишком малым. Он тут же пользуется известными формулами, возможно, приведенными в лекции, и говорит заказчику, что для решения задачи ему нужно не менее, чем 200 образцов каждого класса пород. Это требование не приводит заказчика в замешательство. Он достает из рюкзака геологический молоток и каждую глыбу разбивает на 200 кусков. Исполнитель возмущен этой выходкой, да и заказчик догадывается, что он сделал совсем не то, что нужно. Но он готов выслушать и исполнить любые рекомендации специалиста по распознаванию о том, как следует это обучающее множество подготовить. И тут оказывается, что специалист не может дать никаких вразумительных указаний, как именно должны отбираться примеры в обучающее мультимножество, чтобы обучение на этих образцах давало правильные результаты. ▲

Первым серьезным препятствием для практического применения рекомендаций статистической теории обучения является то, что они применимы в условиях, которые невозможно на практике проверить, выполняются ли они или нет. Но это не единственное препятствие. Давайте посмотрим, опять преувеличенно критически, что же обозначает В в утверждении "Если А, то В".

Необходимо видеть существенное различие между двумя утверждениями. Первым является утверждение "Вероятность ошибочного решения не превышает ε ". Второе утверждение - это "Вероятность того, что вероятность ошибочного решения превышает ε , не превышает η ". Первое утверждение характеризует одну конкретную стратегию. Каждую стратегию, в том числе и полученную в результате обучения, можно исследовать и в конечном итоге с большей или меньшей точностью подтвердить или опровергнуть это утверждение. Второе же утверждение характеризует не какую-то фиксированную стратегию, а определенную совокупность стратегий, на которой задано распределение вероятностей. Мы не поднимаем здесь вопрос о том, насколько трудно подтвердить или опровергнуть это утверждение для той или иной совокупности стратегий. Существенно то, что, располагая одной-единственной стратегией из некоторой совокупности, невозможно ни подтвердить, ни опровергнуть утверждение, касающееся всей совокупности.

Мы хотим обратить сейчас внимание на серьезный недостаток статистической теории обучения. Задачи обучения формулируются так, что решением задачи является та или иная стратегия, а статистическая теория обучения указывает свойства некоторой совокупности стратегий. Таким образом, в результате обучения получается что-то одно, а теория описыва-

ет свойства чего-то совсем другого. Констатация таких свойств не может служить гарантией для достаточно требовательного потребителя.

Здесь имеется (хотя и в очень скрытом виде) существенное противоречие того же характера, которое часто возникает между потребителем и разработчиком. Представьте себе, что Вы приобрели некий продукт и обнаружили его неисправность. Вы заявляете об этой неисправности разработчику, который в ответ на Ваши претензии говорит Вам нечто, возможно и правду, но то, что Вас совершенно не интересует. А именно, что такие ситуации, которые произошли с Вашим изделием, у него происходят не чаще, чем одна на тысячу и т.п. Худшее, что Вы можете в этой ситуации сделать, это вступить с ним в дискуссию по навязанному вопросу. На основании имеющегося у Вас единственного изделия Вы не сможете опровергнуть утверждения разработчика, а он не сможет их подтвердить. Надо его вежливо прервать, сказав, что он говорит о чем-то, что Вас не интересует. Вас не интересует исправность изделий у остальных 999 пользователей. Вас интересует исправность того единственного изделия, которое есть у Вас.

Приведенные критические соображения о статистической теории обучения свидетельствуют об определенной зрелости этой теории. Такая критика возможна только после очень пристального внимания к ней и многократного использования ее рекомендаций. Конечно, эти критические замечания не приводились бы, если бы не было хотя бы эскизного подхода к обучению, при котором такие недостатки менее заметны.

4.5 набросок детерминированного обучения

Чтобы избавиться от несовершенства в современной теории и практике обучения, необходимо подвергнуть сомнению или даже отказать от некоторых положений, которые сейчас принимаются, как само собою разумеющиеся.

Общепринято, что целью обучения в распознавании образов является стратегия распознавания, и это уже содержит в себе некорректность. Исходные данные задачи обучения недостаточны для однозначного определения стратегии. Если намеренно симплифицировать ситуацию с обучением, то общепринятая формулировка цели обучения не отличается существенно от такой бессмыслицы, как "Известно, что число q удовлетворяет неравенству $3 \leq q \leq 4$; требуется определить, чему равно q ".

Исходная информация об искомой стратегии состоит из двух частей. Первая часть - это заданное априори множество стратегий, в котором содержится искомая стратегия. Вторая часть - это обучающее множество или мультимножество. Обе эти части необязательно определяют искомую стратегию однозначно. Обучающее множество всего лишь сужает множество возможных стратегий, исключая из априори заданного множества те стратегии, которые ведут себя неправильно на примерах из обучающего множества. В результате такого сужения оставшееся множество, как правило, содержит более одной стратегии, и задача определить ее становится

подобной указанной выше бессмыслице.

Чтобы не сталкиваться с этой бессмыслицей, надо окончательно распрощаться с представлением, что целью обучения является отыскание стратегии. Результатом обучения не может быть стратегия $q^*: X \rightarrow K$, потому что исходных данных при обучении недостаточно для ее определения. Цель обучения можно формулировать только как определение значения $q^*(x)$, которое принимает действительная, но неизвестная стратегия q^* на конкретном наблюдении x , которое подлежит распознаванию. Конструкция, которую мы назовем *обучаемым распознаванием*, основана на следующей базовой идее: *даже если стратегия q^* не определяется однозначно на основании исходной информации, значение ее для некоторых наблюдений $x \in X$ можно определить однозначно*. Конечно, такая однозначность достигается не для всех наблюдений $x \in X$. Когда для некоторого наблюдения $x \in X$ такая однозначность не имеет места, распознающая система не должна принимать никакого решения, а дать ответ, что исходной информации недостаточно для принятия решения. Покажем более точно, как может происходить такое обучаемое распознавание.

Пусть X - это множество наблюдений, а $q^*: X \rightarrow D$ - стратегия, которую будем называть *правильной*. Стратегия q^* неизвестна, но известно множество Q , которое ее содержит.

Приллюстрируем эту ситуацию на примере.

Пример 4.10 Разделение плоскости на две части с помощью прямой.

Пусть X - двумерное пространство (плоскость), Q - множество стратегий, каждая из которых делит плоскость X на две части с помощью прямой. ▲

Пусть XD обучающее множество вида

$$XD = ((x_1, d_1), (x_2, d_2), \dots, (x_l, d_l)), \quad (4.25)$$

где $d_i = q^*(x_i)$ это решение, $i = 1, 2, \dots, l$. Множество (4.25) отличается существенно от обучающего мультимножества T , которое имеет место при статистическом обучении и которое имеет вид

$$T = ((x_1, k_1), (x_2, k_2), \dots, (x_l, k_l)), \quad (4.26)$$

где k_i - состояние, в котором находился объект, когда наблюдалось x_i .

Для получения обучающего мультимножества T требуется перевести объект распознавания в особый режим, при котором становится наблюдаемым состояние k , которое в обычном режиме ненаблюдаемо. Для некоторых объектов такой особый режим просто невозможен. Очень часто (особенно при распознавании изображений) значительно легче получить обучающее множество XD , чем обучающее мультимножество T .

Есть еще одно важное отличие между множеством XD и мультимножеством T . При статистическом обучении требуется, чтобы мультимножество T обладало определенными статистическими свойствами, которые невозможно проверить, выполняются ли они или нет. Конструкция, которую мы

сейчас описываем, свободна от таких требований. Для получения множества XD необходимо, чтобы имелось устройство, возможно, сложное и дорогое, например, человек, назовем его *учитель*, который способен выдавать правильный ответ для любого наблюдения. Устройство, которое обучается, должно в конечном итоге заменить учителя и распознавать самостоятельно новые наблюдения. Это устройство назовем обучаемым распознавателем.

Обучаемое распознавание основано на знании, что правильная стратегия удовлетворяет условиям

$$q^* \in Q, \quad q^*(x_i) = d_i, \quad i = 1, \dots, l. \quad (4.27)$$

Множество Q стратегий в этом выражении, наблюдения x_i и решения d_i , $i = 1, \dots, l$, известны, а стратегия q^* неизвестна. Пусть $x \in X$ - наблюдение, предъявленное для распознавания. Обучаемое распознавание состоит в том, чтобы для любого решения $d \in D$ определить, следует ли $q^*(x) = d$ из исходной информации (4.27) или нет. Доказательство этой импликации может выполняться следующим образом. Обозначим $Q(XD)$ множество стратегий, удовлетворяющих (4.27). Если множество $Q(XD)$ непустое, то для данного наблюдения $x \in X$ проверяется

$$\exists d \in D [\forall q \in Q(XD) (q(x) = d)]. \quad (4.28)$$

Эта формула говорит, что все стратегии, которые удовлетворяют условиям (4.27) (а следовательно, и правильная стратегия q^*) принимают одно и то же решение d по данному наблюдению x . Если утверждение (4.28) верно, значение $q^*(x)$ определяется однозначно. В противном случае единственным возможным является ответ *not known*, так как исходной информации, полученной от учителя, недостаточно для вывода о значении $q^*(x)$.

Пример 4.11 Разделена плоскости прямой линией.

(Продолжение примера 4.10) Обучающее множество показано на рис. 4.1 белыми и черными кружочками. Для белых кружочков $q^*(x) = 1$, а для черных - $q^*(x) = 2$. Множество $Q(XD)$ не показано на рисунке. Это множество всех тех прямых, которые правильно разделяют обучающее множество. На рисунке показаны также два выпуклых множества \tilde{X}_1 , \tilde{X}_2 , ограниченных ломаными линиями. Множество $\tilde{X}_1 \cup \tilde{X}_2$ состоит из тех точек $x \in X$, для которых верно утверждение (4.28). Для каждой точки $x \in \tilde{X}_1$ выполняется $q^*(x) = 1$, а для $x \in \tilde{X}_2$ выполняется $q^*(x) = 2$. ▲

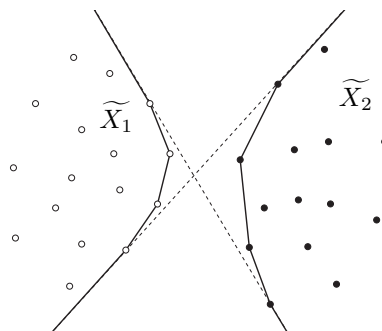


Figure 4.1 Training set consisting of white and black circles. \tilde{X}_1 and \tilde{X}_2 denote regions where unambiguous decision is possible.

Для проверки правильности утверждения (4.28) нет нужды в явном виде представлять множество $Q(XD)$ всех возможных стратегий, удовлетво-

ряющих (4.27). Само исходное отношение (4.27) уже достаточно хорошо приспособлено для верификации утверждения (4.28). Покажем, как это должно делаться.

Для каждого решения $d \in D$ и предложенного для распознавания наблюдения $x \in X$ напишем систему условий, подобную (4.27),

$$\left. \begin{aligned} q^* &\in Q, \\ q^*(x_i) &= d_i, \quad i = 1, 2, \dots, l, \\ q^*(x) &= d, \end{aligned} \right\} d \in D. \quad (4.29)$$

Выражение (4.29) представляет не одну систему условий, а совокупность, состоящую из $|D|$ систем, каждая из которых соответствует одному решению $d \in D$. Для каждой системы из совокупности (4.29), то есть для каждого значения d , надо проверить его противоречивость. Утверждение (4.28) эквивалентно тому, что система условий (4.29) непротиворечива только при каком-то одном значении d .

Завершим рассмотрение нашего примера и покажем, как эквивалентность этих двух утверждений используется для распознавания некоторого данного наблюдения.

Пример 4.12 Распознавание фиксированного наблюдения на основе обучающего множества. На основе обучающего множества XD построим два вспомогательные обучающие множества $XD_1 = (XD \cup \{(x, 1)\})$ и $XD_2 = (XD \cup \{(x, 2)\})$. Для каждого из них должно быть проверено, существует ли прямая, которая его правильно разделяет. Эта проверка может иметь следующие четыре исхода.

1. Если обучающее множество XD_1 можно правильно разделить прямой линией, а множество XD_2 - нет, то гарантируется равенство $q^*(x) = 1$.
2. Если обучающее множество XD_1 нельзя разделить прямой линией, а множество XD_2 можно, то гарантируется равенство $q^*(x) = 2$.
3. Если оба обучающие множества можно классифицировать с помощью прямой, то обучаемый распознаватель еще не получил достаточно информации, чтобы распознать данное конкретное наблюдение. В этом случае должен быть дан ответ **not known**. Здесь важно то, что обучаемый распознаватель сам определяет степень своей обученности в каждом конкретном случае. Для распознавания одних наблюдений он достаточно обучен, а для других нет. В случае, когда обучаемый распознаватель обнаруживает свою недостаточную компетентность, он обращается с вопросом к учителю, включает его ответ в обучающее множество и последующие наблюдения распознает более компетентно.
4. Если ни одно из обучающих множеств нельзя классифицировать с помощью прямой, это значит, что исходная для обучения информация противоречива. В этом случае обучаемый распознаватель должен выдать ответ "Не знаешь", как бы обнаруживая недостаточную компетентность учителя. Достаточно продвинутый обучаемый

распознаватель может предъявить учителю минимальную противоречивую часть обучающей информации. В этом случае учитель должен скорректировать обучающую информацию, например, расширить априорное множество Q стратегий, исключить или отредактировать часть обучающего множества, или сделать что-нибудь третье, чтобы обнаруженное противоречие устранить. В этом случае, пожалуй, уже трудно определить, кто кого здесь учит, то ли учитель распознавателя, то ли распознаватель учителя. ▲

В следующей лекции, полностью посвященной линейным дискриминантным функциям, мы покажем, как обнаруживать противоречивость обучающих множеств.

Анализ противоречий в совокупности (4.29) условий дает, вообще говоря, больше полезной информации, чем только что мы показали. Пусть, например, не единственная система в совокупности условий (4.29) оказывается непротиворечивой и текущее наблюдение нельзя распознать однозначно. Пусть $D(x) \subset D$ - это множество тех решений $d \in D$, включение которых в условия (4.29) не приводит к противоречию. В этом случае, хотя и $|D(x)| \neq 1$, но $D(x) \neq \emptyset$, $D(x) \neq D$, и можно исключить по крайней мере некоторые решения, как заведомо невозможные, и таким образом сузить круг возможных решений. Практически полезны не только те ситуации, когда можно однозначно определить правильное решение $q^*(x)$, но и те ситуации, когда можно гарантировать, что какое-то решение d не является правильным.

В приведенном наброске обучаемого распознавателя уже мало что осталось от ранее описанной схемы обучения. Что осталось, так это зависимость результатов распознавания от обучающего множества. Существенным же является то, что полностью исчезло разбиение технологии обучения на два этапа: обучение и распознавание. От этого, как нам кажется, обучаемый распознаватель не потерял черты интеллектуального поведения, а наоборот, приобрел их. Действительно, процесс, при котором обучение предшествует распознаванию, а затем происходит распознавание без какого-либо обучения, очень далек от взаимно плодотворного отношения между учителем и обучаемым, которое только и можно назвать обучением. Современные же технологии обучения скорее напоминают натаскивание на экзамен, чем обучение. Отсюда и возникают все те трудности о необходимой длительности обучения, которые так блестяще преодолевает статистическая теория обучения Вапника и Червоненкиса. Действительно, очень трудно определить, достаточно ли информация, сообщаемая обучаемому, для принятия правильного решения во всех будущих ситуациях, когда обучение уже будет невозможно.

В приведенном наброске обучаемого распознавателя все происходит иначе. Обучаемый распознаватель готов к распознаванию на каждом этапе своего обучения, и при этом не возникает вопрос, достаточно ли уже обучение или нет. Обучающая информация достаточно для распознавания одних

наблюдений и недостаточна для других. По одним наблюдениям принимается решение, и это решение правильное. По другим наблюдениям распознаватель воздерживается от принятия решения, обращается к учителю и, таким образом, готов к обучению на каждом этапе распознавания. Более того, обучаемый распознаватель может и не ожидать, когда поступит наблюдение, в котором он некомпетентен. Он может сам сгенерировать такое наблюдение и предъявить его учителю с вопросом, как следует с ним справиться. Действуя таким образом, обучаемый распознаватель активно участвует в формировании обучающего множества, ускоряя процесс обучения, направленно извлекая информацию от учителя.

Только сейчас мы видим, как мало мы еще знаем об обучении распознаванию образов и как опрометчиво и самоуверенно мы проскочили этапы сочинения сказок и конструирования игрушек, о которых мы говорили в начале лекции. Что касается современной теории обучения, то это очень интеллектуальное исследование информационных технологий, которые еще очень далеки от того, что можно было бы назвать интеллектуальностью.

4.6 Обсуждение

У меня возникло много вопросов, и первый из них, боюсь, не очень конкретный. Я исполнен уважения перед изящными математическими построениями при анализе основных асимптотических свойств процедур обучения. Я могу только догадываться, какими головокружительными должны быть доказательства теорем, приведенных в лекции. Мне очень жаль, что после такого восторженного признания обычно следует проклятое "но". Я прихожу к довольно прочному убеждению, что основная ценность приведенной теории находится внутри самой теории и сравнительно меньше рекомендаций, адресованных во внешний мир. При изучении лекции мне навязчиво вспоминались разного рода восточные сувениры типа модели парусника, собранной внутри бутылки. Я восхищаюсь такими изделиями, как проявлением мастерства, которое когда-нибудь окажется полезным и для чего-то другого. Но не могу отделаться и от неприятного мне самому вопроса, зачем нужен парусник в бутылке. Я не хочу конкретизировать свой первый вопрос о лекции, чтобы не показаться невежливым.

Вопрос, вообще говоря, значительно шире, чем проблематика нашей лекции. Прежде всего согласись с нами, что научные знания, существенно меняющие наши представления о мире, очень редко имели форму рекомендаций, а значительно чаще формулировались в виде запрета. Они отвечают не на вопрос "как надо делать", а на вопрос "что невозможно сделать". Вспомни закон сохранения энергии, знание которого вряд ли окажется Тебе полезным в практической ситуации, когда Ты захочешь поднять рояль на десятый этаж. Но он окажет Тебе огромную пользу, если к Тебе придет некто с проектом устройства, поднимающего рояль на десятый этаж без затраты энергии. Не тратя времени на анализ этого проекта (а это может требовать очень много времени), Ты можешь быть уверен, что автор

проекта либо шарлатан, либо малообразован. А ведь еще совсем недавно, каких-то триста лет назад человечество не располагало таким мощным орудием, как закон сохранения энергии. Поэтому тратились огромные интеллектуальные усилия для создания вечного двигателя, а затем еще большие усилия для понимания, почему же он все-таки не работает. Ведь было же совершенно очевидно, что он должен работать, а главное, как было бы замечательно, если бы он работал.

Научное и практическое значение теории Вапника и Червоненкиса в том, что они ставят полный запрет на некоторые проекты обучаемых распознающих систем. Давай посмотрим на фундаментальную теорему 4.2 о необходимых и достаточных условиях обучаемости распознающих систем. Это условие говорит, что удельная энтропия $H(l)/l$ должна стремиться к нулю при неограниченном увеличении длины l обучающего мультимножества. Хотя энтропию $H(l)$ почти никогда нельзя вычислить и само условие нельзя как-то конструктивно проверить, эта теорема имеет огромную запрещающую силу. Энтропия $H(l)$ все-таки легко вычисляется по крайней мере для универсального множества стратегий, то есть множества, содержащего любую стратегию. В этом случае энтропия $H(l)$ равна (или пропорциональна) объему l обучающего мультимножества, а удельная энтропия $H(l)/l$ есть постоянная величина, не зависящая от l . Следовательно, не выполняются необходимые условия обучаемости, и бессмысленно конструировать процедуры обучения таких универсальных распознавателей.

Понимание этого запрета поможет Тебе сэкономить много времени и усилий. Ты наверняка много раз слушал доклады или читал статьи, где в первом разделе приводится теорема об универсальности предлагаемого распознавателя, а во втором разделе приводится алгоритм его обучения. Обычно очень трудно убедительно доказать неуниверсальность какого-то распознающего устройства. Ведь для этого нужно указать конкретную стратегию, которая этим устройством не может быть реализована. Не менее трудно доказать ограниченность какого-то алгоритма обучения, то есть наличие стратегии, которая не может быть достигнута ни при какой длительности обучения. Не погружаясь в дебри этого анализа, Ты можешь быть уверен, что по крайней мере в одном из разделов автор допустил грубую ошибку, и требовать уже от автора, чтобы он объяснил, как соотносятся его результаты с теорией Вапника и Червоненкиса. Если он о них не знает, то без колебаний прекрати с ним дискуссию, потому что он в распознавании находится на том же уровне, на котором находилась механика 300 лет назад. Он не знает то, что должен знать каждый.

Я не ожидал именно такого ответа на мой вопрос. Но в этом я сам виноват: мой вопрос был недопустимо расплывчатый. Я думал, что вы подчеркнете конструктивность и практическое значение таких понятий, как функция роста и емкость. Я ожидаю, что у меня возникнут некоторые трудности, когда мне придется подсчитать емкость того или конкретного класса стратегий. Я здесь не все понимаю так, как мне бы хотелось. Существуют ли

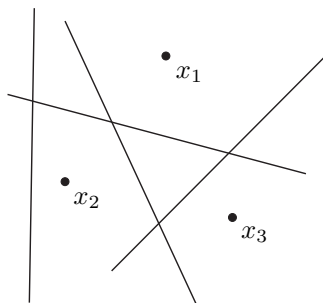


Figure 4.2 Four possible decompositions of three points in the plane with the help of straight lines.

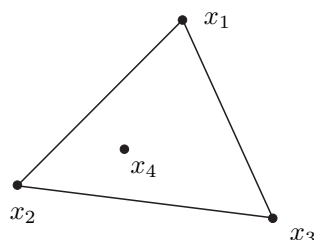


Figure 4.3 The convex hull of four points can be a triangle.

какие-нибудь рекомендации по практическому определению емкости класса стратегий?

К сожалению, мы вряд ли можем что-то добавить к тому, что было сказано в лекции. Но нам кажется, что сказанного в лекции вполне достаточно при Твоей все сокрушающей сообразительности. Исследуй различные частные случаи стратегий, и через некоторое время увидишь, что в них нет ничего страшного. Начни, пожалуй, со следующих двух классов.

Пусть X - двумерное линейное пространство, а первый класс стратегий включает в себя те стратегии, которые разбивают пространство X на две части с помощью прямой линии. Второй класс стратегий похож на первый, но речь идет о трехмерном пространстве и классе стратегий, которые разбивают пространство на две части с помощью плоскости.

Я нашел, что в первом из приведенных примеров емкость класса равна $CA_P = 4$. Доказывается это так. Существует такая тройка точек на плоскости, что любое разбиение их на две группы можно реализовать с помощью той или иной прямой линии так, что точки принадлежащие разным группам, будут лежать по разные стороны от прямой. Например, это могут быть точки x_1 , x_2 и x_3 , показанные на рис. 4.2, где показаны все возможные разбиения трех точек на две группы и как эти разбиения реализуются с помощью разделяющей прямой линии.

Легко понять, что любую четверку x_1, x_2, x_3, x_4 точек на плоскости можно разбить на две группы так, что одну группу точек уже невозможно будет отделить от второй группы с помощью точек. Очевидно, что если три из четырех точек лежат на одной прямой, то с помощью никакой прямой нельзя отделить среднюю точку от двух других. Если же никакие три точки не лежат на одной прямой, то выпуклая оболочка точек x_1, x_2, x_3, x_4 образует либо треугольник, как на рис. 4.3, либо четырехугольник, как на рис. 4.4.

В первом случае невозможно отделить с помощью прямой точку внутри треугольника от его вершин. Во втором случае невозможно пару проти-

воположных вершин четырехугольника отделить от двух других вершин. Следовательно, мощность рассматриваемого класса стратегий равна 4.

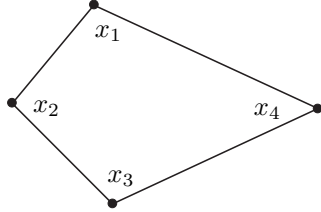


Figure 4.4 The convex hull of four points can be a quadrilateral.

Оказывается, подобные геометрические рассуждения для трехмерного случая уже довольно утомительны, и поэтому я сразу же рассмотрел общий k -мерный случай.

Пусть X - k -мерное пространство, а x_i , $i = 0, 1, \dots, k+1$, - совокупность из $k+2$ точек в этом пространстве. Векторы $x_i - x_0$, $i = 1, \dots, k+1$, естественно, линейно зависимы. Это значит, что существуют такие коэффициенты α_i , $i = 1, \dots, k+1$, не все равные нулю, что выполняется равенство

$$\sum_{i=1}^{k+1} \alpha_i (x_i - x_0) = 0,$$

и эквивалентное равенство

$$x_0 \sum_{i=1}^{k+1} \alpha_i = \sum_{i=1}^{k+1} \alpha_i x_i.$$

Обозначим сумму $\sum_{i=1}^{k+1} \alpha_i$ как $-\alpha_0$ и получим равенство $\sum_{i=0}^{k+1} \alpha_i x_i = 0$, в котором сумма $\sum_{i=0}^{k+1} \alpha_i$ тоже равна нулю. Поскольку не все коэффициенты α_i равны нулю, некоторые из них положительны, а некоторые - отрицательны. Обозначим I^+ множество индексов положительных коэффициентов, а I^- - множество индексов отрицательных коэффициентов, то есть,

$$I^+ = \{i \mid \alpha_i > 0\}, \quad I^- = \{i \mid \alpha_i < 0\}.$$

Поскольку $\sum_{i=0}^{k+1} \alpha_i = 0$, справедливо равенство

$$\sum_{i \in I^-} \alpha_i = - \sum_{i \in I^+} \alpha_i \neq 0.$$

Введем новые переменные β_i , $i = 0, 1, \dots, k+1$, такие, что

$$\beta_i = \begin{cases} -\frac{\alpha_i}{\sum_{i \in I^-} \alpha_i}, & \text{если } i \in I^-, \\ \frac{\alpha_i}{\sum_{i \in I^+} \alpha_i}, & \text{если } i \in I^+. \end{cases}$$

Из равенства $\sum_{i=0}^{k+1} \alpha_i x_i = 0$ следует, что

$$\left. \begin{aligned} \sum_{i \in I^-} \alpha_i x_i + \sum_{i \in I^+} \alpha_i x_i &= 0 \\ \Rightarrow \left(\sum_{i \in I^-} \alpha_i x_i = - \sum_{i \in I^+} \alpha_i x_i \right) \\ \Rightarrow \left(\sum_{i \in I^-} \beta_i x_i = \sum_{i \in I^+} \beta_i x_i \right), \end{aligned} \right\} \quad (4.30)$$

где

$$\sum_{i \in I^-} \beta_i = \sum_{i \in I^+} \beta_i = 1. \quad (4.31)$$

Я утверждаю, что не существует гиперплоскости, которая отделяет точки $x_i, i \in I^-$, от точек $x_i, i \in I^+$. Действительно, если бы такая гиперплоскость существовала, существовал бы вектор $\alpha \in X$ и число θ , которые удовлетворяли бы систему неравенств

$$\left. \begin{aligned} \langle \alpha, x_i \rangle &\geq \theta, \quad i \in I^+, \\ \langle \alpha, x_i \rangle &< \theta, \quad i \in I^-, \end{aligned} \right\} \quad (4.32)$$

где $\langle \alpha, x_i \rangle$ обозначает скалярное произведение векторов α и x_i . Поскольку сумма всех коэффициентов $\beta_i, i \in I^+$, равна 1 и все они положительны, из первой группы неравенств в (4.32) следует, что

$$\sum_{i \in I^+} \beta_i \langle \alpha, x_i \rangle \geq \theta,$$

а из второй группы неравенств следует, что

$$\sum_{i \in I^-} \beta_i \langle \alpha, x_i \rangle < \theta.$$

Это значит, что справедливо неравенство

$$\sum_{i \in I^+} \beta_i \langle \alpha, x_i \rangle > \sum_{i \in I^-} \beta_i \langle \alpha, x_i \rangle. \quad (4.33)$$

Из цепочки (4.30) следует равенство

$$\sum_{i \in I^-} \beta_i \langle \alpha, x_i \rangle = \sum_{i \in I^+} \beta_i \langle \alpha, x_i \rangle$$

которое противоречит неравенству (4.33). Таким образом я доказал, что любую совокупность из $k+2$ точек в k -мерном пространстве можно разбить на две группы так, что точки одной группы нельзя отделить с помощью гиперплоскости от точек второй группы.

Нетрудно показать, что существует такая совокупность из $k + 1$ точек, что для любого разбиения этой совокупности на две группы существует гиперплоскость, отделяющая точки одной группы от точек второй группы. Это может быть, например, такая совокупность: точка x_0 есть начало координат, а в точке x_i , $i = 1, \dots, k$, все координаты нулевые, кроме i -ой координаты, которая ненулевая. Таким образом, я доказал следующее утверждение.

Пусть X - k -мерное пространство, а Q - множество стратегий вида $X \rightarrow \{1, 2\}$, разделяющих пространство X на две части с помощью гиперплоскости. Емкость множества Q равна $k + 2$.

С этими емкостями, действительно, все не так ужасно, как кажется на первый взгляд.

Рады это слышать. Но для большей уверенности давай взглянем еще на пару примеров. Определи емкость множества стратегий, которые делят множество точек двумерной плоскости с помощью окружности на две части: множество точек внутри окружности и вне окружности.

Я быстро справился с задачей, воспользовавшись спрямлением пространства.

Прежде всего, существует четверка точек на плоскости, которую можно произвольным образом разбить на две группы с помощью окружности. Например, это могут быть четыре точки, представленные на рис. 4.5, где представлены и все возможные (их всего 8) разбиения четырех точек на две группы. Следовательно, емкость данного класса стратегий не меньше, чем 5.

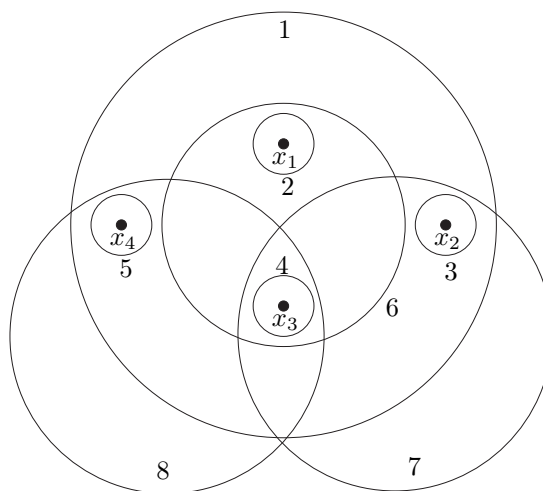


Figure 4.5 Eight possible decompositions of four points in the plane by means of eight circles.

Я докажу теперь, что любую пятерку точек можно разделить на две группы так, что с помощью окружности уже нельзя будет отделить одну группу точек от другой. Таким образом я докажу, что емкость данного класса стратегий равна 5.

Пусть x и y - координаты точки на плоскости. Исследуемый класс стратегий содержит стратегии q вида

$$q(x, y) = \begin{cases} 1, & \text{если } (x - x_0)^2 + (y - y_0)^2 \leq r^2, \\ 2, & \text{если } (x - x_0)^2 + (y - y_0)^2 > r^2, \end{cases} \quad (4.34)$$

или

$$q(x, y) = \begin{cases} 1, & \text{если } (x - x_0)^2 + (y - y_0)^2 > r^2, \\ 2, & \text{если } (x - x_0)^2 + (y - y_0)^2 \leq r^2. \end{cases} \quad (4.35)$$

Любую стратегию вида (4.34) или (4.35) можно выразить как

$$q(x, y, z) = \begin{cases} 1, & \text{если } \alpha x + \beta y + \gamma z \geq \theta, \\ 2, & \text{если } \alpha x + \beta y + \gamma z < \theta, \end{cases} \quad (4.36)$$

где $z = x^2 + y^2$. Обратное также справедливо. Любую стратегию вида (4.36) на множестве точек, удовлетворяющих ограничению $z = x^2 + y^2$, можно выразить в виде (4.34) или (4.35). Прямого утверждения уже достаточно, чтобы определить емкость класса (4.34), (4.35). Она не может быть больше емкости класса (4.36). Класс (4.36) - это множество стратегий, разделяющих трехмерное пространство на две части с помощью плоскости. Раньше я уже доказал, что емкость класса (4.36) равна 5. Таким образом я доказал, что емкость класса (4.34), (4.35) тоже равна 5.

Теперь я вижу, что можно точно определить емкость класса стратегий с квадратичной дискриминантной функцией. В третьей лекции вы показали, что оптимальные для гауссовой модели стратегии принадлежат именно этому классу. Эти стратегии имеют вид

$$\left. \begin{aligned} q(x) &= 1, & \text{если } f(x) &\geq \theta, \\ q(x) &= 2, & \text{если } f(x) &< \theta, \end{aligned} \right\} \quad (4.37)$$

при некотором пороговом значении θ и квадратичной функции вида

$$f(x) = \sum_{i=1}^n \alpha_i x_i^2 + \sum_{i=1}^n \sum_{j=i+1}^n \beta_{ij} x_i x_j + \sum_{i=1}^n \gamma_i x_i. \quad (4.38)$$

Переменная x_i в выражении (4.38) обозначает i -ую координату точки x . Я покажу, что существует такое множество X^* , состоящее из $2n + \frac{1}{2}n(n-1) + 1$ точек, что любое разбиение его на две группы можно реализовать с помощью квадратичной дискриминантной функции. Из этого немедленно получится вывод, что емкость SAP рассматриваемого класса стратегий удовлетворяет неравенству

$$SAP > 2n + \frac{1}{2}n(n-1) + 1. \quad (4.39)$$

Совокупность точек X^* строится следующим образом. В нее войдут множества X_1^- и X_1^+ , определяемые далее, состоящие из n точек, затем множество X_2 , состоящее из $\frac{1}{2}n(n-1)$ точек, и наконец, множество X^0 , состоящее из одной точки. Пронумеруем точки множества X_1^- индексами $i = 1, 2, \dots, n$. Точку с номером i обозначим $(x^i)^-$ и определим, как точку со всеми нулевыми координатами, кроме i -ой, которая равна -1 . Подобным образом i -ую точку множества X_1^+ обозначим $(x^i)^+$. У нее все координаты нулевые, кроме i -ой, которая равна $+1$. Точки множества X_2 пронумеруем двумя индексами $i = 1, 2, \dots, n$ и $j = i+1, i+2, \dots, n$. Обозначим (ij) -ую точку x^{ij} и определим, что все ее координаты равны нулю, кроме i -ой и j -ой, которые равны 1 . Единственную точку множества X_0 обозначим x^0 и определим, что это начало координат.

Разделим множество $X^* = X_1^- \cup X_1^+ \cup X_2 \cup X_0$ на две группы X_1^* и X_2^* и зафиксируем это разбиение для дальнейшего рассмотрения. Докажем, что для данного разбиения существуют такие коэффициенты $\alpha_i, \beta_{ij}, \gamma_i$ и порог θ , что стратегия (4.37), (4.38) удовлетворяет неравенствам

$$\left. \begin{aligned} f(x) &\geq \theta, & x \in X_1^*, \\ f(x) &< \theta, & x \in X_2^*. \end{aligned} \right\} \quad (4.40)$$

Поскольку $f(x^0) = 0$, порог θ не может быть положительным, если $x^0 \in X_1^*$. С другой стороны, порог θ должен быть положительным, если $x^0 \in X_2^*$. Я выберу $\theta = \frac{1}{2}$, если $x^0 \in X_2^*$, и $\theta = -\frac{1}{2}$, если $x^0 \in X_1^*$. Я рассмотрю только первый случай, так как второй случай рассматривается почти так же. Я покажу, как можно выбрать коэффициенты $\alpha_i, \beta_{ij}, \gamma_i$, чтобы выполнялись требования

$$\left. \begin{aligned} f(x) &= 1, & x \in X_1^*, \\ f(x) &= 0, & x \in X_2^*, \end{aligned} \right\} \quad (4.41)$$

а следовательно, и требования (4.40). Введем вспомогательные числа $(k^i)^-, (k^i)^+, k^{ij}$, $i = 1, 2, \dots, n$, $j = i+1, i+2, \dots, n$, такие, что

$$\begin{aligned} (k^i)^- &= 0, & \text{если } (x^i)^- \in X_2^*, \\ (k^i)^- &= 1, & \text{если } (x^i)^- \in X_1^*, \\ (k^i)^+ &= 0, & \text{если } (x^i)^+ \in X_2^*, \\ (k^i)^+ &= 1, & \text{если } (x^i)^+ \in X_1^*, \\ k^{ij} &= 0, & \text{если } x^{ij} \in X_2^*, \\ k^{ij} &= 1, & \text{если } x^{ij} \in X_1^*. \end{aligned}$$

С использованием этих чисел система равенств (4.41) приобретает вид

$$\left. \begin{aligned} f((x^i)^-) &= (k^i)^-, \\ f((x^i)^+) &= (k^i)^+, \\ f(x^{ij}) &= k^{ij}, \end{aligned} \right\} \begin{aligned} &i = 1, 2, \dots, n, \\ &j = i+1, i+2, \dots, n, \end{aligned} \quad (4.42)$$

или, что то же самое,

$$\left. \begin{aligned} \alpha_i - \gamma_i &= (k^i)^-, \\ \alpha_i + \gamma_i &= (k^i)^+, \\ \alpha_i + \alpha_j + \gamma_i + \gamma_j + \beta_{ij} &= k^{ij}, \end{aligned} \right\} \begin{aligned} i &= 1, 2, \dots, n, \\ j &= i + 1, i + 2, \dots, n. \end{aligned} \quad (4.43)$$

Система уравнений (4.43) имеет очевидное решение

$$\alpha_i = \frac{(k^i)^- + (k^i)^+}{2}, \quad \gamma_i = \frac{(k^i)^+ - (k^i)^-}{2}, \quad \beta_{ij} = k^{ij} - (k^i)^+ - (k^j)^+,$$

что и доказывает неравенство (4.39).

Далее сравнительно нетрудно доказывается, что емкость рассматриваемого класса стратегий не может быть больше, чем $2n + \frac{1}{2}n(n-1) + 2$. В лекции было показано, что n -мерное линейное пространство X можно так отобразить в $2n + \frac{1}{2}n(n-1)$ -мерное пространство Y , что любой стратегии вида (4.37) и (4.38) в пространстве X соответствует разбиение пространства Y с помощью гиперплоскости. Ранее я доказал, что емкость множества гиперплоскостей в m -мерном пространстве равна $m+2$. Из этого следует, что емкость множества гиперплоскостей в пространстве Y равна $2n + \frac{1}{2}n(n-1) + 2$. Емкость множества стратегий вида (4.37) и (4.38) не может быть больше и, таким образом,

$$CAP \leq 2n + \frac{1}{2}n(n-1) + 2.$$

Отсюда, с учетом полученного ранее неравенства (4.39) следует

$$CAP = 2n + \frac{1}{2}n(n-1) + 2.$$

Таким образом, сейчас я в некотором смысле завершил анализ гауссовой модели, начатый в третьей лекции. А именно, я доказал, что гауссова модель является обучаемой. И более того, по качеству распознавания, достигнутому на обучающем множестве, можно эффективно предсказывать качество распознавания при последующей эксплуатации алгоритма, используя рекомендации уже данной лекции.

Молодец! Но для полноты неплохо бы подобным образом завершить рассмотрение и второй модели, начатое в третьей лекции, а именно, модели с независимыми признаками. Речь идет о емкости класса стратегий, заданных в виде

$$x \in \begin{cases} X_1, & \text{если } \sum_{i=1}^n \log \frac{p_{X_i|1}(x_i)}{p_{X_i|2}(x_i)} \geq \theta, \\ X_2, & \text{если } \sum_{i=1}^n \log \frac{p_{X_i|1}(x_i)}{p_{X_i|2}(x_i)} < \theta. \end{cases} \quad (4.44)$$

Эти стратегии являются байесовскими для модели объекта, при которой наблюдение x при фиксированном состоянии есть случайная величина $x = (x_1, x_2, \dots, x_n)$ с независимыми компонентами.

Я как в воду глядел, когда после лекции 3 доказал, что любую стратегию вида (4.44) можно реализовать с помощью линейной дискриминантной функции в пространстве с размерностью

$$\sum_{i=1}^n k(i) - n,$$

где $k(i)$ - количество значений признака x_i . Этот результат является прямым обобщением результата, приведенного в лекции 3 для случая, когда все признаки принимают два значения. Из полученного мною результата немедленно следует, что

$$CAP \leq \sum_{i=1}^n k(i) - n + 2, \quad (4.45)$$

потому что сегодня я доказал, что емкость множества гиперплоскостей в m -мерном линейном пространстве равна $m + 2$.

Теперь мне осталось доказать, что в неравенстве (4.45) знак \leq можно заменить на равенство. Мне нужно доказать, что в пространстве наблюдений X существует такая совокупность, состоящая из $\sum_{i=1}^n k(i) - n + 1$ точек, что любое их разбиение на две группы можно реализовать стратегией вида (4.44). Эти точки можно задать, например, следующим способом. Выберем произвольную точку $x^0 = (x_1^0, x_2^0, \dots, x_n^0)$. Затем будем считать выбранной любую точку, отличающуюся от x^0 значением только одного признака. Количество точек, выбранных таким способом, равно как раз $\sum_{i=1}^n k(i) - n + 1$. Должен ли я доказать, что любое разбиение такой совокупности точек на две группы можно реализовать стратегией вида (4.44)?

Думаем, что нам троим это ясно. Благодарим Тебя за сотрудничество, и на сегодня, пожалуй, хватит.

Апрель 1997.

4.7 Библиографические заметки

В лекции приведены три формулировки задачи обучения распознаванию. Первая из них является прямым переносом наиболее правдоподобного оценивания, широко известного в математической статистике, в тематику распознавания образов [Nilsson, 1965]. Из классических источников укажем работы Гаусса и Фишера [Fisher, 1936]. В качестве учебника по методам наиболее правдоподобного оценивания заслуженно пользуется наибольшей популярностью учебник Ван дер Вардена [Waerden, 1957]. Теоретический анализ алгоритмов обучения в рамках различных моделей выполнил Ш.Раудис с коллегами, см. например, [Raudys and Pikelis, 1980].

Вторая, максиминная постановка задачи обучения по неслучайному обучающему множеству опубликована в [Schlesinger, 1989], как теоретическое

осмысление приемов, оказавшихся плодотворными при решении прикладных задач распознавания буквенных символов [Schlesinger and Svjatogor, 1967]. Теоретический анализ этого направления в обучении мы продолжим в лекции 8.

Третья формулировка является обобщением задач по безошибочному распознаванию обучающего множества [Rosenblatt, 1962; Ajzerman et al., 1970]. Приведенные работы вызвали целую лавину работ этого направления. Приведенная в лекции формулировка о минимизации эмпирического риска опубликована в работах Вапника и Червоненкиса [Vapnik and Chervonenkis, 1974; Vapnik, 1995], [Vapnik, 1998]. Статистическая теория обучения излагается в лекции по работе [Vapnik and Chervonenkis, 1974]

Из сравнительно недавних работ по статистической теории обучения укажем [Vidyasagar, 1996].

Прекрасную побасенку о геологе нам рассказал Загоруйко. Она настолько выразительна, что мы не могли отказаться от того, чтобы ее не пересказать. Единственное, что нас смущало до самого последнего времени, это то, что мы не знали, как на нее сослаться. Сейчас мы знаем, что она приведена в недавно вышедшей монографии Загоруйко [Zagorujko, 1999] и с удовольствием на нее ссылаемся. Ее мы настойчиво рекомендуем всем, кто читает по русски. В книге имеется обширный обзор методов кластеризации и хороший послужной список решенных практических задач. В наших лекциях этот круг вопросов почти не рассматривается, за исключением краткого рассмотрения в шестой лекции.

Глава 5

Линейные дискриминантные функции

5.1 Вводные замечания о линейных дискриминантных функциях

В предыдущих лекциях мы неоднократно обращали внимание на то, что линейные дискриминантные функции заслуживают особого внимания. Во первых, известно, что для некоторых статистических моделей решение байесовских или небайесовских задач достигается именно с помощью линейных дискриминантных функций.

Во вторых, некоторые нелинейные дискриминантные функции можно выразить как линейные в так называемом спрямляющем пространстве, как это было показано в разделе 3.2. Это возможно, когда нелинейная дискриминантная функция $f: X \rightarrow \mathbb{R}$ имеет вид

$$f(x) = \sum_{j \in J} \alpha_j f_j(x)$$

при известных функциях $f_j: X \rightarrow \mathbb{R}$, $j \in J$, и неизвестных коэффициентах α_j . В этом случае поиск дискриминантной функции, которая нелинейна в исходном пространстве X , сводится к поиску линейной дискриминантной функции в $|J|$ -мерном спрямляющем пространстве Y . Пространство X отображается в пространство Y так, что точке $x \in X$ ставится в соответствие точка $y \in Y$, j -ая координата которой равна $f_j(x)$.

В третьих, важный теоретический результат, указанный в первой лекции, состоит в том, что в пространстве вероятностей любая байесовская стратегия реализуется линейными дискриминантными функциями, и это справедливо для любой статистической модели распознаваемого объекта.

В четвертых, известно, что емкость класса стратегий, реализуемых линейными дискриминантными функциями в n -мерном пространстве, равна $n + 2$ и поэтому задача обучения корректна. Это значит, что стратегия, построенная по конечному обучающему множеству достаточно большой длины, не слишком отличается от стратегии, найденной по статистиче-

ской модели. Более того, в этом случае взаимосвязь точности, надежности и длительности обучения выражается простой и удобной для практического применения формулой.

Естественно, все эти преимущества не имели бы большого значения при отсутствии хорошо разработанных методов построения линейных дискриминантных функций, которые составляют содержание данной лекции. Мы увидим, что задачи построения линейных дискриминантных функций, кажущиеся различными на первый взгляд, имеют на самом деле много общего. Эти задачи как-бы взаимодействуют друг с другом, действуют сообща так, что понимание свойств и алгоритмов решения одной задачи позволяет обнаружить неожиданные свойства и ранее неизвестные алгоритмы решения другой задачи.

5.2 Путеводитель по лекции

Данная лекция довольно обширная. Поэтому приведем обзор ее основных составных частей, затем более подробно рассмотрим каждую часть.

Пусть X - это n -мерное линейное пространство, а результат наблюдения объекта - это точка x в этом пространстве. Пусть k - это состояние объекта, недоступное для непосредственного наблюдения, и пусть состояние принимает только два значения $\{1, 2\}$. Пусть известно, что распределение условных вероятностей $p_{X|K}(x|k)$, $x \in X$, $k \in K$, есть n -мерное гауссово распределение. Математические ожидания μ_k и ковариационные матрицы σ_k , $k = 1, 2$, этих распределений неизвестны. Известно, однако, что значения (μ_1, σ_1) принадлежат заданному конечному множеству $\{(\mu^j, \sigma^j) \mid j \in J_1\}$ значений. Подобным образом, (μ_2, σ_2) - это неизвестные значения, о которых известно, что они принадлежат другому конечному множеству $\{(\mu^j, \sigma^j) \mid j \in J_2\}$. Во введенных обозначениях мы использовали как верхние, так и нижние индексы. Так, μ_1 и σ_1 - это действительные, но неизвестные статистические параметры объектов в первом состоянии, а (μ^j, σ^j) при некотором верхнем индексе j - это один из возможных вариантов значений действительных параметров.

Введенные понятия иллюстрируются на рис. 5.1. На рисунке пять эллипсов представляют пять двумерных случайных величин. Не следует сейчас обращать внимание на прямую линию q на рисунке. Пять эллипсов на рисунке представляют ситуацию, когда объект в первом состоянии характеризуется случайным вектором, который имеет первое, или второе, или третье распределение вероятностей, но неизвестно, какое именно. Подобным образом, объектам во втором состоянии соответствуют четвертое или пятое распределение вероятностей.

При указанной неполноте сведений о статистических свойствах объекта за-

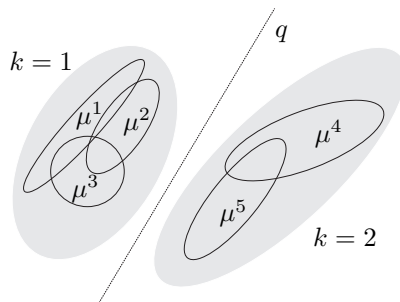


Figure 5.1 Generalised Anderson task in a two-dimensional space.

дача распознавания состояния k по наблюдению x может формулироваться, как задача статистических решений при неслучайных вмешательствах. Эти задачи в общем виде были рассмотрены в подразделе 2.2.5. Применительно к нашему конкретному случаю в этих задачах требуется найти стратегию $q: X \rightarrow \{1, 2\}$, которая минимизирует число

$$\max_{j \in J_1 \cup J_2} \varepsilon(j, \mu^j, \sigma^j, q), \quad (5.1)$$

где $\varepsilon(j, \mu^j, \sigma^j, q)$ -это вероятность того, что гауссовский случайный вектор x с математическим ожиданием μ^j и ковариационной матрицей σ^j удовлетворит равенству $q(x) = 1$ при $j \in J_2$ и равенству $q(x) = 2$ при $j \in J_1$.

Говоря иными словами, следует найти минимальное значение ε и такую стратегию q^* , для которых справедливы следующие два утверждения:

1. Независимо от математического ожидания μ_1 и ковариационной матрицы σ_1 вероятность ошибочного распознавания объектов в первом состоянии не превышает ε , но только если $(\mu_1, \sigma_1) \in \{(\mu^j, \sigma^j), j \in J_1\}$.
2. Независимо от математического ожидания μ_2 и ковариационной матрицы σ_2 вероятность ошибочного распознавания объектов во втором состоянии не превышает ε , но только если $(\mu_2, \sigma_2) \in \{(\mu^j, \sigma^j), j \in J_2\}$.

В подразделе 2.2.5 было доказано, что решение сформулированной задачи сводится к отысканию определенной смеси распределений из множества J_1 и отысканию определенной смеси распределений из множества J_2 и стратегии, которая оптимально отделяет одну смесь от другой.

Нас интересует решение задачи (5.1) при ограничении на стратегию q . Мы требуем, чтобы стратегия реализовалась линейной дискриминантной функцией, то есть имела вид

$$q(x) = \begin{cases} 1, & \text{если } \langle \alpha, x \rangle > \theta, \\ 2, & \text{если } \langle \alpha, x \rangle < \theta, \end{cases} \quad (5.2)$$

при некотором векторе $\alpha \in X$ и некотором пороговом значении θ . Обозначение $\langle \alpha, x \rangle$ мы применяем для скалярного произведения векторов α и x . Для двумерного случая, рассмотренного ранее и представленного на рис. 5.1, эта стратегия состоит в разделении плоскости с помощью прямой на две полуплоскости. Одна полуплоскость должна содержать большинство случайных наблюдений, имеющих первое, второе и третье распределение вероятностей, а вторая полуплоскость - большинство наблюдений, имеющих четвертое и пятое распределение. Это разделение плоскости представлено на рис. 5.1 прямой q .

Задача (5.1) при ограничении (5.2) есть обобщение известной задачи Андерсона и Бахадура [Anderson and Bahadur, 1962], которая является частным случаем сформулированной задачи при $|J_1| = |J_2| = 1$. Сформулированную нами задачу будем называть *обобщенной задачей Андерсона-Бахадура*, а иногда более коротко - задачей Андерсона.

Помимо первоначальной задачи Андерсона приведенная формулировка включает в себя и другой частный случай, заслуживающий внимания. Это случай, когда все ковариационные матрицы σ^j , $j \in J_1 \cup J_2$, единичные. В этом случае задача Андерсона трансформируется в отыскание такой гиперплоскости, что заданные два конечные множества точек оказываются расположенными по разные стороны от нее. При этом следует отыскать не любую гиперплоскость, а оптимальную, то есть наиболее удаленную от множеств точек, которые она отделяет друг от друга. К отысканию такой гиперплоскости сводятся некоторые задачи обучения. Мы будем называть такую задачу *оптимальным разделением конечных множеств точек*. Более точно говоря, задача заключается в том, чтобы для заданных двух конечных множеств \tilde{X}_1 и \tilde{X}_2 найти такой вектор α и такое пороговое значение θ , что:

1. для любого $x \in \tilde{X}_1$ удовлетворяется неравенство

$$\langle \alpha, x \rangle > \theta ; \quad (5.3)$$

2. для любого $x \in \tilde{X}_2$ удовлетворяется неравенство

$$\langle \alpha, x \rangle < \theta ; \quad (5.4)$$

3. при выполнении условий (5.3) и (5.4) максимизируется число

$$\min \left(\min_{x \in \tilde{X}_1} \frac{\langle \alpha, x \rangle - \theta}{|\alpha|}, \min_{x \in \tilde{X}_2} \frac{\theta - \langle \alpha, x \rangle}{|\alpha|} \right). \quad (5.5)$$

Дальнейшее упрощение задачи состоит в снятии требования оптимальности. Эту задачу мы будем называть *задачей простого разделения конечных множеств точек*. Заключается она в том, что при заданных множествах \tilde{X}_1 и \tilde{X}_2 требуется найти какое-нибудь решение системы неравенств (5.3), (5.4), без учета требования (5.5).

Мы начнем лекцию с тщательного анализа задачи Андерсона. Самым важным результатом этого анализа является тот факт, что функция, максимум которой требуется найти, оказывается одноэкстремальной. Это значит, что максимизацию этой функции можно осуществить методами локальной оптимизации, не опасаясь, что эти методы приведут к застреванию на ложных локальных максимумах. Таких ложных локальных максимумов просто нет. Но наряду с этим положительным свойством оптимизируемая функция не является ни вогнутой, ни дифференцируемой. Следовательно, локальная оптимизация этой функции не может выполняться движением по градиенту, так как в некоторых точках градиент не существует. Более того, в точке максимума градиент как правило не существует, и поэтому невозможно воспользоваться таким известным условием максимума, как нулевое значение градиента. Для обобщенной задачи Андерсона надо сформулировать такие условия максимума, в которых не используется понятие градиента. Эти необходимые и достаточные условия максимума будут сформулированы и доказаны в лекции, и будет описан метод решения

задачи, основанный на локальной оптимизации, но не являющийся методом движения по градиенту.

Для частного случая задачи Андерсона, каким является оптимальное разделение двух конечных множеств, будет показано, что эта задача сводится к минимизации квадратичной функции на выпуклом многограннике. Для решения такой задачи применимы многие хорошо разработанные методы квадратичной оптимизации.

Что касается задачи простого разделения конечных множеств точек, то мы напомним читателю хорошо известные алгоритмы персептрона и менее известные на Западе алгоритмы русского математика Козинца.

В конце лекции мы покажем, что все эти задачи могут решаться единым алгоритмом. Само по себе это не удивительно. Естественно, все эти задачи, являясь частными случаями задачи Андерсона, могут решаться с помощью алгоритма решения этой общей задачи. Неожиданным является, что обобщенная задача Андерсона, самая трудная из рассматриваемых задач может решаться после незначительной модификации алгоритмов персептрона и Козинца, то есть алгоритмов, первоначально предназначенных для решения предельно упрощенных частных случаев обобщенной задачи Андерсона.

5.3 Задачи Андерсона

5.3.1 Эквивалентные формулировки задач Андерсона

Мы будем иметь дело с обобщением известной задачи Андерсона на случай, когда количества элементов в множествах J_1 и J_2 не обязательно равны 1. Без потери общности мы можем вместо стратегий, сравнивающих значение линейной функции $\langle \alpha, x \rangle$ с порогом θ , ограничиться рассмотрением стратегий, принимающих решение на основании знака линейной функции $\langle \alpha, x \rangle$. Такая замена осуществляется следующим стандартным приемом.

Отобразим исходное n -мерное пространство X в $(n+1)$ -мерное пространство X' так, что точке x в исходном пространстве X соответствует точка x' в пространстве X' . Первые n координат точки x' те же, что и n координат точки x , а $(n+1)$ -ая координата равна $+1$. Обозначим порог θ как $-\alpha_{n+1}$ и увидим, что неравенство $\langle \alpha, x \rangle > \theta$ эквивалентно неравенству $\langle \alpha', x' \rangle > 0$. При этом первые n компонент вектора α' должны совпадать с компонентами вектора α , а $(n+1)$ -ая компонента должна равняться $-\theta$. Сформулируем задачу Андерсона с учетом этой модификации.

Пусть X - многомерное линейное пространство, а J - множество номеров определенной совокупности гауссовых случайных величин, принимающих значения на этом пространстве. Для j -ой, $j \in J$, случайной величины задано ее математическое ожидание μ^j и ковариационная матрица σ^j . Множество J разделено на два подмножества J_1 и J_2 . Пусть при заданном векторе α стратегия имеет следующий вид. При $\langle \alpha, x \rangle > 0$ принимается решение, что наблюдение x есть реализация случайной величины из первого подмножества, а при $\langle \alpha, x \rangle < 0$ считается, что x есть реализация случайной величины из второго подмножества. Обозначим $\varepsilon^j(\alpha)$ вероятность того,

что реализация j -ой случайной величины будет ошибочно распознана. Для $j \in J_1$ число $\varepsilon^j(\alpha)$ есть вероятность неравенства $\langle \alpha, x \rangle \leq 0$ для случайного гауссова вектора x с математическим ожиданием μ^j и ковариационной матрицей σ^j . Для $j \in J_2$, аналогично, $\varepsilon^j(\alpha)$ есть вероятность неравенства $\langle \alpha, x \rangle \geq 0$ для случайного гауссова вектора x с математическим ожиданием μ^j и ковариационной матрицей σ^j .

В задаче Андерсона требуется найти вектор $\alpha \neq 0$, который минимизирует число $\max_{j \in J} \varepsilon^j(\alpha)$, то есть

$$\alpha = \operatorname{argmin}_{\alpha} \max_{j \in J} \varepsilon^j(\alpha). \quad (5.6)$$

Для удобства дальнейшего рассмотрения выполним еще одно эквивалентное преобразование задачи. Определим векторы μ'^j так, что

$$\mu'^j = \begin{cases} \mu^j & \text{для } j \in J_1, \\ -\mu^j & \text{для } j \in J_2. \end{cases}$$

На рис. 5.2 представлено это преобразование для случая, когда $J_1 = \{1, 2\}$ и $J_2 = \{3, 4\}$. Для любого вектора α справедливо, что вероятность неравенства $\langle \alpha, x \rangle \geq 0$ для случайного гауссова вектора x с математическим ожиданием μ^j и ковариационной матрицей σ^j равна вероятности неравенства $\langle \alpha, x \rangle \leq 0$ для случайного вектора x с математическим ожиданием $-\mu^j$ и ковариационной матрицей σ^j . В силу этого задачу Андерсона (5.6) можно выразить и в следующей эквивалентной формулировке.

Для заданной совокупности $((\mu^j, \sigma^j), j \in J)$, следует найти ненулевой вектор α , минимизирующий число $\max_j \varepsilon^j(\alpha)$,

$$\alpha = \operatorname{argmin}_{\alpha} \max_j \varepsilon^j(\alpha), \quad (5.7)$$

где $\varepsilon^j(\alpha)$ - это вероятность того, что случайный гауссов вектор x с математическим ожиданием μ^j и ковариационной матрицей σ^j удовлетворит неравенство $\langle \alpha, x \rangle \leq 0$.

Для иллюстрации вернемся к геометрической интерпретации сказанного. В исходной задаче требовалось отделить одну совокупность гауссовых случайных величин от другой с помощью гиперплоскости. Сейчас имеется только одна совокупность гауссовых случайных величин и требуется добиться, чтобы все они находились в каком-то одном полупространстве, то есть по одну сторону от какой-то гиперплоскости, проходящей через начало

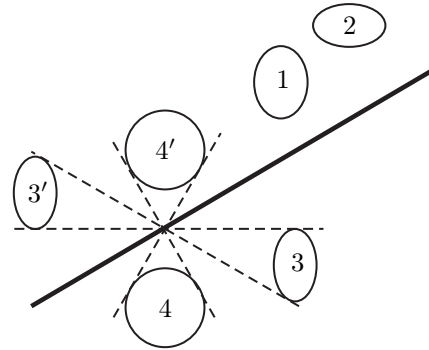


Figure 5.2 The straight line separating the ellipses 1, 2 from the ellipses 3, 4 is equivalent to the straight line leaving the ellipses 1, 2 and 3', 4' along one side.

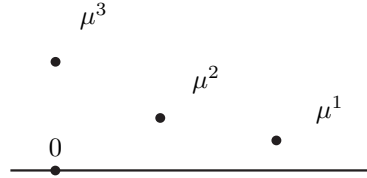


Figure 5.3 Straight line passing through the origin, leaving the points μ_1, μ_2 and μ_3 along one side.

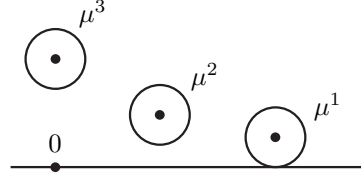


Figure 5.4 Contact of one ellipse with the straight line.

координат. Эта формулировка окажется удобной для дальнейшего анализа, так как не потребуются всякий раз для того или иного j упоминать, взято ли это значение из множества J_1 или J_2 . Для обоих этих случаев будут записываться одни и те же формулы.

Перед тем, как начать формальный анализ сформулированной задачи (5.7), рассмотрим ее неформально.

5.3.2 Неформальный анализ задачи Андерсона

Исходными данными задачи Андерсона является совокупность пар $((\mu^j, \sigma^j), j \in J)$. Эта совокупность характеризует определенную группу многомерных гауссовых случайных переменных, принимающих значения на линейном пространстве X . Для заданного вектора μ , положительно определенной матрицы σ и числа r определим множество точек x , удовлетворяющих условию

$$\langle (x - \mu), \sigma^{-1} \cdot (x - \mu) \rangle \leq r^2, \quad (5.8)$$

где \cdot обозначает произведение матрицы на вектор. Введенное множество точек обозначим $E(r, \mu, \sigma)$ и назовем эллипсом, хотя на самом деле это многомерное тело, которое следовало бы назвать эллипсоидом.

Перефразируем задачу Андерсона, пока что без какого-либо обоснования, а на основании одних лишь разумных соображений. Для заданной совокупности пар $((\mu^j, \sigma^j), j \in J)$ следует найти такой вектор α , чтобы полупространство $\{x \in X \mid \langle \alpha, x \rangle \geq 0\}$ содержало объединение $\bigcup_{j \in J} E(r, \mu^j, \sigma^j)$ эллипсов с как можно большим размером r .

Если приведенная формулировка действительно эквивалентна требованию (5.7) (а мы покажем позже, что это так), то гиперплоскость, которая решает задачу (5.7), можно было бы найти с помощью процедуры, которую мы опишем для простейшего, двумерного случая.

Пусть μ^1, μ^2 и μ^3 - математические ожидания трех случайных переменных, как показано на рис. 5.3. Прежде всего следует найти такую прямую, проходящую через начало координат, чтобы все три точки μ^1, μ^2 и μ^3 оказались по одну сторону от нее. Если такой прямой нет, то для любой прямой по крайней мере одна из трех случайных величин будет распознаваться с вероятностью, худшей, чем 0.5. Вряд ли в этом случае целесообразно находить оптимальную стратегию, так как даже оптимальная стратегия

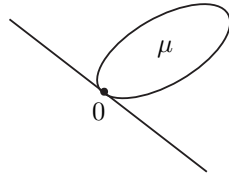


Figure 5.5 A particular case in which the straight line is closely contacted with one ellipse.

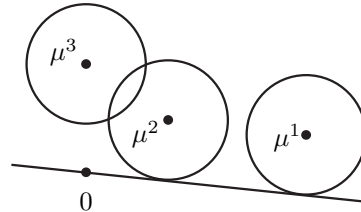


Figure 5.6 The straight line contacts another ellipse.

окажется плохой. На примере рис. 5.3 искомая прямая есть. Это горизонтальная прямая.

Теперь представим себе, что вокруг точек μ^1 , μ^2 и μ^3 начинают расти эллипсы. В любой момент времени размеры трех эллипсов равны между собой, а их ориентация зависит от матриц σ^1 , σ^2 и σ^3 . Одновременно с ростом эллипсов изменяется положение прямой так, что все три эллипса все время остаются по одну сторону от нее. Рост эллипсов продолжается до тех пор, пока некоторые эллипсы (или один из них), достигнув определенного размера, не заклинят прямую в единственно возможном положении. Если будет достигнуто такое положение, то дальнейший рост эллипсов становится невозможным. С ростом одних эллипсов прямую надо повернуть в одну сторону, с ростом других - в другую, и невозможно повернуть прямую так, чтобы при росте всех эллипсов они все еще оказались по одну сторону от прямой.

Посмотрим, как выглядит этот рост эллипсов на примере, представленном на рис. 5.3. В самом начале эллипсы растут, не заставляя поворачиваться прямую линию. Этот рост продолжается, пока один из эллипсов не коснется прямой. В нашем случае (предполагая, что матрицы σ_1 , σ_2 , σ_3 одинаковы) это произойдет с первым эллипсом, см.рис. 5.4. Если бы точка касания эллипса попала в точности в начало координат, то дальнейший рост этого эллипса был бы невозможен. Такой случай, когда искомая прямая определяется единственным эллипсом, показан на рис. 5.5. В нашем примере точка касания не совпадает с началом координат и рост эллипсов может продолжаться, заставляя прямую линию поворачиваться по часовой стрелке, пока она не коснется какого-то другого эллипса. В нашем случае это второй эллипс на рис. 5.6. Если бы точки касания первого и второго эллипсов лежали по разные стороны от начала координат, дальнейший рост эллипсов стал бы невозможен. В нашем примере этого не произошло и эллипсы продолжают расти, поворачивая прямую. При этом первый эллипс перестает касаться прямой, и положение прямой зависит только от размера второго эллипса, см рис. 5.7.

Рост эллипсов продолжается либо до тех пор, пока точка касания не достигнет начала координат, либо до тех пор, пока прямая не коснется еще какого-то эллипса. В нашем случае это третий эллипс на рис. 5.8. Сейчас точки касания уже лежат на противоположных полупрямых по отноше-

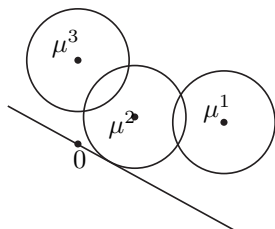


Figure 5.7 Turning the straight line clockwise further.

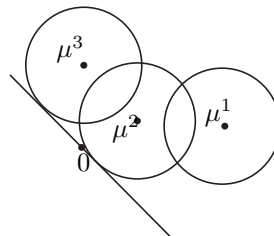


Figure 5.8 The straight line has contacted another ellipse. The growth of the ellipses ends.

нию к началу координат. Рост второго эллипса заставляет поворачиваться прямую в одну сторону, а рост третьего - в противоположную. Поэтому одновременный рост обоих эллипсов невозможен и найденное положение прямой есть решение задачи.

При определенном пространственном воображении можно представить себе, как этот рост эллипсоидов выглядит в трехмерном пространстве. В этом случае эллипсоиды растут до тех пор, пока точка касания одного из эллипсоидов не совпадет с началом координат, либо пока точки касания двух каких-то эллипсоидов не окажутся на одной прямой с началом координат и по разные стороны от начала координат, либо пока начало координат не окажется внутри треугольника, вершинами которого являются точки касания трех каких-то эллипсоидов.

На основании такого неформального понимания задачи можно предположить следующее необходимое и достаточное условие оптимального положения плоскости в задаче (5.7).

Пусть H - гиперплоскость, проходящая через начало координат, а μ^j , σ^j , $j \in J$, параметры $|J|$ случайных гауссовых векторов. Пусть r^j , $j \in J$, - положительные числа, а x_0^j - точки, в которых эллипсы

$$\langle (x - \mu^j), (\sigma^j)^{-1} \cdot (x - \mu^j) \rangle = r^j$$

касаются гиперплоскости H . Пусть J^0 - подмножество индексов $j \in J$, для которых

$$r^j = \min_{j' \in J} r^{j'}.$$

Гиперплоскость H является решением задачи (5.7) тогда и только тогда, когда многогранник с вершинами в точках касания x_0^j , $j \in J^0$, содержит начало координат.

Это утверждение будет сформулировано более точно и доказано. Для этого определим более точно понятие эллипсов и точек касания, которыми мы пользовались при интуитивном рассмотрении задачи.

5.3.3 Определение вспомогательных понятий для задачи Андерсона

Пусть X - n -мерное линейное пространство, $\mu \in X$ - это n -мерный вектор, а σ - симметричная положительно определенная матрица размера $(n \times n)$. Далее, пусть вектор $\alpha \in X$ и число θ делят пространство X на три подмножества:

$$\begin{aligned} \text{положительное полупространство } X^+ &= \{x \in X \mid \langle \alpha, x \rangle > \theta\}, \\ \text{отрицательное полупространство } X^- &= \{x \in X \mid \langle \alpha, x \rangle < \theta\}, \\ \text{и гиперплоскость } X^0 &= \{x \in X \mid \langle \alpha, x \rangle = \theta\}. \end{aligned}$$

Примем, что $\mu \in X^+$. Обозначим F квадратичную функцию

$$F(x) = \langle (x - \mu), \sigma^{-1} \cdot (x - \mu) \rangle \quad (5.9)$$

и для определенного неотрицательного числа r определим множество $E(r, \mu, \sigma) = \{x \in X \mid F(x) \leq r^2\}$, которое назовем эллипсом размера r . Наибольший размер r , при котором эллипс $E(r, \mu, \sigma)$ остается подмножеством множества $X^+ \cup X^0$, обозначим r^* и будем называть этот размер *расстоянием пары (μ, σ) от гиперплоскости X^0* . Очевидно, что существует единственная точка, принадлежащая как гиперплоскости X^0 , так и эллипсу $E(r^*, \mu, \sigma)$. Обозначим ее x_0 и будем называть ее *точкой касания*. Очевидно также, что в точке касания достигается минимум функции F на гиперплоскости X^0 , а значение функции F в точке касания равно $(r^*)^2$, то есть, квадрату расстояния (μ, σ) от гиперплоскости X^0 . Выведем явные формулы для вычисления расстояния и точки касания.

Точка касания есть решение оптимизационной задачи, которой соответствует функция Лагранжа

$$\Phi(x, \lambda) = \langle (x - \mu), \sigma^{-1} \cdot (x - \mu) \rangle + \lambda \cdot \langle \alpha, x \rangle,$$

а искомая точка x_0 есть решение системы уравнений

$$\left. \begin{aligned} \text{grad } \Phi(x, \lambda) &= 0, \\ \langle \alpha, x \rangle &= \theta \end{aligned} \right\} \quad (5.10)$$

относительно переменных x и λ . Первое уравнение в системе (5.10) в более развернутом виде есть

$$2 \sigma^{-1} \cdot (x - \mu) + \lambda \cdot \alpha = 0,$$

из чего следует, что

$$x_0 = \mu - \frac{\lambda}{2} \sigma \cdot \alpha. \quad (5.11)$$

Значение коэффициента λ должно удовлетворять системе (5.10), то есть уравнению

$$\left\langle \alpha, \left(\mu - \frac{\lambda}{2} \sigma \cdot \alpha \right) \right\rangle = \theta.$$

Его решением λ есть число

$$\lambda = 2 \frac{\langle \alpha, \mu \rangle - \theta}{\langle \alpha, \sigma \cdot \alpha \rangle}. \quad (5.12)$$

Подставим полученное значение (5.12) коэффициента λ в выражение (5.11) и получим явную формулу для точки касания x_0 ,

$$x_0 = \mu - \frac{\langle \alpha, \mu \rangle - \theta}{\langle \alpha, \sigma \cdot \alpha \rangle} \sigma \cdot \alpha. \quad (5.13)$$

Подставим выражение для x_0 в (5.9) и получим явное выражение для размера $(r^*)^2$ эллипса $\langle (x_0 - \mu), \sigma^{-1} \cdot (x_0 - \mu) \rangle = (r^*)^2$, который касается гиперплоскости $\langle \alpha, x \rangle = \theta$,

$$\begin{aligned} (r^*)^2 &= \langle (x_0 - \mu), \sigma^{-1} \cdot (x_0 - \mu) \rangle \\ &= \left\langle \left(\mu - \frac{\lambda}{2} \sigma \cdot \alpha - \mu \right), \sigma^{-1} \cdot \left(\mu - \frac{\lambda}{2} \sigma \cdot \alpha - \mu \right) \right\rangle \\ &= \left(\frac{\lambda}{2} \right)^2 \langle (\sigma \cdot \alpha), \sigma^{-1} \cdot (\sigma \cdot \alpha) \rangle = \left(\frac{\lambda}{2} \right)^2 \langle (\sigma \cdot \alpha), \alpha \rangle \\ &= \left(\frac{\lambda}{2} \right)^2 \langle \alpha, \sigma \cdot \alpha \rangle = \left(\frac{\langle \alpha, \mu \rangle - \theta}{\langle \alpha, \sigma \cdot \alpha \rangle} \right)^2 \langle \alpha, \sigma \cdot \alpha \rangle = \left(\frac{\langle \alpha, \mu \rangle - \theta}{\sqrt{\langle \alpha, \sigma \cdot \alpha \rangle}} \right)^2. \end{aligned}$$

Это значит, что размер r^* эллипса, касающегося гиперплоскости $\langle \alpha, x \rangle = \theta$, есть

$$r^* = \left| \frac{\langle \alpha, \mu \rangle - \theta}{\sqrt{\langle \alpha, \sigma \cdot \alpha \rangle}} \right|. \quad (5.14)$$

Поскольку вектор μ принадлежит X^+ , можно опустить знак абсолютной величины,

$$r^* = \frac{\langle \alpha, \mu \rangle - \theta}{\sqrt{\langle \alpha, \sigma \cdot \alpha \rangle}}. \quad (5.15)$$

В случае, когда вектор μ принадлежит X^- , соответствующее выражение имеет вид

$$r^* = \frac{\theta - \langle \alpha, \mu \rangle}{\sqrt{\langle \alpha, \sigma \cdot \alpha \rangle}}. \quad (5.16)$$

Посмотрим внимательно на полученные формулы (5.15) и (5.16). Числитель в (5.15) есть математическое ожидание случайной величины $\langle \alpha, x \rangle - \theta$ для случайного вектора x с математическим ожиданием μ . Знаменатель - это среднее квадратичное отклонение случайной величины $\langle \alpha, x \rangle - \theta$ для случайного вектора x с ковариационной матрицей σ . Из этого немедленно следует, что размер эллипса, который касается гиперплоскости, строго монотонно зависит от вероятности того, что случайная величина x попадет в полупространство, противоположное тому, в котором находится μ . Более точно это утверждение формулирует следующая лемма.

Лемма 5.1 Пусть x - многомерная случайная гауссова переменная с математическим ожиданием μ и ковариационной матрицей σ , принимающая значения на линейном пространстве X . Пусть вектор $\alpha \in X$ и число θ определяют разбиение пространства X на три подмножества:

$$\begin{aligned} X^+ &= \{x \in X \mid \langle \alpha, x \rangle > \theta\}, \\ X^- &= \{x \in X \mid \langle \alpha, x \rangle < \theta\}, \\ X^0 &= \{x \in X \mid \langle \alpha, x \rangle = \theta\}. \end{aligned}$$

1. Если $\mu \in X^+$, то вероятность события $x \in X^-$ строго убывает с ростом расстояния пары (μ, σ) от гиперплоскости X^0 ;
2. Если $\mu \in X^-$, то вероятность события $x \in X^+$ строго убывает с ростом расстояния пары (μ, σ) от гиперплоскости X^0 . \blacktriangle

Доказательство. Лемма 5.1 есть очевидное следствие равенств (5.15) и (5.16). \blacksquare

5.3.4 Решение исходной задачи Андерсона

Рассмотрим исходную задачу Андерсона в ее первоначальной, а не обобщенной формулировке, но с использованием введенных нами понятий. Пусть μ_1 и μ_2 - два вектора, а σ_1 и σ_2 две положительно определенные матрицы. Следует найти вектор α и число θ , которые разделяют пространство X на полупространства X^+ и X^- и гиперплоскость X^0 так, чтобы $\mu_1 \in X^+$, $\mu_2 \in X^-$, а расстояние пар (μ_1, σ_1) и (μ_2, σ_2) от гиперплоскости X^0 стало как можно большим.

Основываясь на неформальных соображениях, мы утверждаем, что гиперплоскость $X^0 = \{x \in X \mid \langle \alpha, x \rangle = \theta\}$ есть (а) общая касательная к двум эллипсам одинакового размера; (б) эллипсы касаются гиперплоскости в одной и той же точке, как показано на рис. 5.9.

Используем явные выражения (5.15), (5.16) для максимального размера эллипса и (5.11) для точек касания и запишем приведенное выше утверждение в виде системы уравнений

$$\left. \begin{aligned} \mu_1 - \frac{1}{2}\lambda_1 \sigma_1 \cdot \alpha &= \mu_2 - \frac{1}{2}\lambda_2 \sigma_2 \cdot \alpha, \\ \left| \frac{\langle \alpha, \mu_1 \rangle - \theta}{\sqrt{\langle \alpha, \sigma_1 \cdot \alpha \rangle}} \right| &= \left| \frac{\langle \alpha, \mu_2 \rangle - \theta}{\sqrt{\langle \alpha, \sigma_2 \cdot \alpha \rangle}} \right|, \\ \lambda_1 &= 2 \frac{\langle \alpha, \mu_1 \rangle - \theta}{\langle \alpha, \sigma_1 \cdot \alpha \rangle}, \\ \lambda_2 &= 2 \frac{\langle \alpha, \mu_2 \rangle - \theta}{\langle \alpha, \sigma_2 \cdot \alpha \rangle}, \end{aligned} \right\} \quad (5.17)$$

которая при замене переменных $\lambda'_1 = \frac{1}{2}\lambda_1$, $\lambda'_2 = -\frac{1}{2}\lambda_2$ приобретает вид

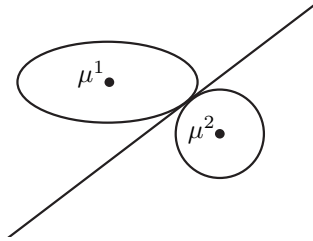


Figure 5.9 Both ellipses touch the hyperplane in the same point.

$$\left. \begin{aligned} \alpha &= (\lambda_1 \cdot \sigma_1 + \lambda_2 \cdot \sigma_2)^{-1} \cdot (\mu_1 - \mu_2), \\ \frac{\langle \alpha, \mu_1 \rangle - \theta}{\sqrt{\langle \alpha, \sigma_1 \cdot \alpha \rangle}} &= -\frac{\langle \alpha, \mu_2 \rangle - \theta}{\sqrt{\langle \alpha, \sigma_2 \cdot \alpha \rangle}}, \\ \lambda_1 &= \frac{\alpha \cdot \mu_1 - \theta}{\alpha \cdot \sigma_1 \cdot \alpha}, \\ \lambda_2 &= -\frac{\langle \alpha, \mu_2 \rangle - \theta}{\langle \alpha, \sigma_2 \cdot \alpha \rangle}. \end{aligned} \right\} \quad (5.18)$$

Второе уравнение в системе (5.17) было записано в другом виде в системе (5.18) в связи с требованием $\mu_1 \in X^+$, $\mu_2 \in X^-$. Коэффициенты λ_1 и λ_2 положительны, следовательно, положительна и их сумма $\lambda_1 + \lambda_2$. Отметим, что вектор α необязательно определять однозначно, а только с точностью до постоянного множителя. Решение задачи, следовательно, зависит не от точного значения коэффициентов λ_1 и λ_2 , а только от их отношения. На их значения можно наложить ограничение, например, что

$$\lambda_1 + \lambda_2 = 1, \quad (5.19)$$

так как любое отношение чисел λ_1 и λ_2 можно достигнуть и при ограничении (5.19). Их отношение равно

$$\begin{aligned} \frac{\lambda_1}{\lambda_2} &= \frac{\langle \alpha, \mu_1 \rangle - \theta}{\langle \alpha, \sigma_1 \cdot \alpha \rangle} \cdot \left(-\frac{\langle \alpha, \sigma_2 \cdot \alpha \rangle}{\langle \alpha, \mu_2 \rangle - \theta} \right) \\ &= \frac{\langle \alpha, \mu_1 \rangle - \theta}{\sqrt{\langle \alpha, \sigma_1 \cdot \alpha \rangle}} \cdot \left(-\frac{\sqrt{\langle \alpha, \sigma_2 \cdot \alpha \rangle}}{\langle \alpha, \mu_2 \rangle - \theta} \right) \cdot \sqrt{\frac{\langle \alpha, \sigma_2 \cdot \alpha \rangle}{\langle \alpha, \sigma_1 \cdot \alpha \rangle}}. \end{aligned} \quad (5.20)$$

В силу второго уравнения в системе (5.18) произведение первых двух сомножителей в правой части (5.20) равно единице, а само выражение (5.20) приобретает следующий простой вид:

$$\frac{\lambda_1}{\lambda_2} = \sqrt{\frac{\langle \alpha, \sigma_2 \cdot \alpha \rangle}{\langle \alpha, \sigma_1 \cdot \alpha \rangle}}.$$

Таким образом мы пришли к изящному решению задачи Андерсона в ее первоначальной формулировке, которое получено самим Андерсоном. Используя (5.19), запишем первое равенство в системе (5.18) в виде

$$\alpha = ((1 - \lambda) \cdot \sigma_1 + \lambda \cdot \sigma_2)^{-1} \cdot (\mu_1 - \mu_2), \quad (5.21)$$

который с точностью до коэффициента λ определяет искомый вектор α по исходным данным μ_1 , μ_2 , σ_1 и σ_2 . Коэффициент λ следует выбрать так, чтобы выполнялось условие

$$\frac{1 - \lambda}{\lambda} = \sqrt{\frac{\langle \alpha, \sigma_2 \cdot \alpha \rangle}{\langle \alpha, \sigma_1 \cdot \alpha \rangle}}.$$

Полученное решение изящно, так как сводит непростую минимаксную задачу с $n + 1$ переменными (а n может быть очень большим) к поиску всего лишь единственного коэффициента λ . Если только число λ тем или иным образом найдено, сразу же определяются и компоненты вектора α по явной формуле (5.21). Значение коэффициента λ можно вычислить итеративно. Для произвольного начального значения, скажем, $\lambda = 0.5$, по формуле (5.21) вычисляется вектор α и отношение

$$\gamma = \sqrt{\frac{\langle \alpha, \sigma_2 \cdot \alpha \rangle}{\langle \alpha, \sigma_1 \cdot \alpha \rangle}}.$$

Если это отношение равно $(1 - \lambda)/\lambda$, то задача решена. В противном случае выбирается новое значения λ' , которое удовлетворяет условию $(1 - \lambda')\lambda' = \gamma$, то есть $\lambda' = 1/(1 + \gamma)$, и итерации продолжаются.

Еще раз подчеркнув элегантность этого решения, мы должны все же отметить, что она достигается тем, что все трудности упрятаны в операцию обращения матриц в выражении (5.21). Мы хотим построить алгоритм, в котором бы отсутствовала операция обращения матриц. Более того, нашей целью является исследование более общей задачи, чем та, которую только что рассмотрели. Мы хотим найти линейную стратегию, которая отличает не одну-единственную гауссову случайную величину от другой, а отделяет целую совокупность гауссовых случайных величин от другой совокупности.

5.3.5 Формальный анализ обобщенной задачи Андерсона

Напомним основные понятия обобщенной задачи Андерсона, которые были введены в подразделе 5.3.1. Пусть X - это n -мерное линейное пространство, а J - конечное множество индексов. Для каждого индекса j из множества J заданы n - мерный вектор μ^j и симметричная положительно определенная матрица σ^j размером $(n \times n)$. Пусть α - это n -мерный вектор. Для каждой тройки α, μ, σ определена вероятность $\varepsilon(\alpha, \mu, \sigma)$ того, что случайный гауссов вектор x с математическим ожиданием μ и ковариационной матрицей σ удовлетворит неравенство $\langle \alpha, x \rangle \leq 0$.

Задача состоит в нахождении вектора α , который минимизирует число $\max_{j \in J} \varepsilon(\alpha, \mu^j, \sigma^j)$ для заданных совокупности $((\mu^j, \sigma^j) | j \in J)$,

$$\alpha = \operatorname{argmin}_{\alpha} \max_{j \in J} \varepsilon(\alpha, \mu^j, \sigma^j).$$

Обозначим $f(\alpha)$ функцию $\max_{j \in J} \varepsilon(\alpha, \mu^j, \sigma^j)$, которую следует минимизировать. Функция $f(\alpha)$ одноэкстремальна в том смысле, как об этом говорит следующая теорема.

Теорема 5.1 Выпуклость множества векторов α . Для любого числа $b < 0.5$ множество векторов α , удовлетворяющих неравенству $f(\alpha) \leq b$, выпукло. ▲

Доказательство. Лемма (5.1) утверждает, что вероятность $\varepsilon(\alpha, \mu^j, \sigma^j)$ строго убывает с ростом числа

$$\frac{\langle \alpha, \mu^j \rangle}{\sqrt{\langle \alpha, \sigma^j \cdot \alpha \rangle}}.$$

Это значит, что для каждого числа b существует такое число c , что если $\varepsilon(\alpha, \mu^j, \sigma^j) \leq b$, то справедливо также

$$\frac{\langle \alpha, \mu^j \rangle}{\sqrt{\langle \alpha, \sigma^j \cdot \alpha \rangle}} \geq c.$$

Если $\varepsilon(\alpha, \mu^j, \sigma^j) = 0.5$, то

$$\frac{\langle \alpha, \mu^j \rangle}{\sqrt{\langle \alpha, \sigma^j \cdot \alpha \rangle}} = 0.$$

Условие $f(\alpha) \leq b$ эквивалентно следующей системе неравенств

$$\varepsilon(\alpha, \mu^j, \sigma^j) \leq b, \quad j \in J.$$

В силу монотонности, доказанной в лемме (5.1), эта система эквивалентна системе неравенств

$$\frac{\langle \alpha, \mu^j \rangle}{\sqrt{\langle \alpha, \sigma^j \cdot \alpha \rangle}} \geq c, \quad j \in J, \quad (5.22)$$

где число c определенным образом зависит от b . Систему (5.22) можно записать в виде

$$\langle \alpha, \mu^j \rangle - c \cdot \sqrt{\langle \alpha, \sigma^j \cdot \alpha \rangle} \geq 0, \quad j \in J. \quad (5.23)$$

Левые части неравенств (5.23) состоят из двух слагаемых. Первое из них - это линейная функция от вектора α и, следовательно, вогнутая функция. Функция $\sqrt{\langle \alpha, \sigma^j \cdot \alpha \rangle}$ - это выпуклая функция от α , а следовательно, функция $-c\sqrt{\langle \alpha, \sigma^j \cdot \alpha \rangle}$ вогнута, так как число c положительно в силу условия $b < 0.5$. Левые части неравенств (5.23) - это суммы двух вогнутых функций, следовательно, это вогнутые функции. Поэтому множество векторов, удовлетворяющих j -ое неравенство, выпукло для каждого j . Множество векторов, удовлетворяющих систему неравенств (5.23) в целом, есть пересечение выпуклых множеств, и следовательно, выпукло. ■

Из теоремы немедленно следует, что в области, где $f(\alpha) < 0.5$, не может быть никакого строгого локального минимума, который не совпадал бы с глобальным минимумом. Под строгим локальным минимумом мы понимаем такую точку α_0 , что для любой точки $\alpha \neq \alpha_0$ из некоторой окрестности точки α_0 выполняется строгое неравенство $f(\alpha_0) < f(\alpha)$. Предположим обратное, и пусть α' и α'' - два различных локальных минимума. Примем для определенности, что $f(\alpha'') \leq f(\alpha') = c < 0.5$. Соединим точки α'' и

α' прямолинейным отрезком. Поскольку точка α' - локальный минимум, на этом отрезке находится точка α^0 (возможно, очень близкая к α'), для которой $f(\alpha^0) > f(\alpha') = c$. Отсюда следует, что множество векторов α , для которых $f(\alpha) \leq c < 0.5$, не является выпуклым.

Из теоремы 5.1 следует также, что если достигнута какая-то точка α' , для которой $f(\alpha') \leq 0.5$, то из этой точки можно достигнуть глобальный минимум α^* , двигаясь по прямолинейному отрезку, соединяющему точку α' с точкой α^* . При движении точки α вдоль этого отрезка значение $f(\alpha)$ не будет нигде возрастать, а будет только убывать.

Эти свойства функции f можно использовать для организации процедуры ее минимизации. Но эта процедура не может быть основана на движении вдоль градиента с целью достигнуть точку с нулевым градиентом, как это обычно делается в процедурах локальной оптимизации, так как функция f не является ни выпуклой, ни дифференцируемой. Оптимизация таких функций исследуется в разделе вычислительной математики, известного как теория негладкой оптимизации. Мы сформулируем необходимые и достаточные условия минимума, не привлекая понятие градиента, но и не привлекая понятий из теории негладкой оптимизации.

Сначала мы сформулируем лемму о необходимых и достаточных условиях минимума числа $\varepsilon(\alpha, \mu, \sigma)$. В формулировке леммы используется понятие точки касания эллипса $\{x \in X \mid \langle x - \mu, \sigma \cdot (x - \mu) \rangle \leq r^2\}$ и гиперплоскости $\{x \in X \mid \langle \alpha, x \rangle = 0\}$. Обозначим точку касания $x_0(\alpha, \mu, \sigma)$. На основании (5.13) и учитывая, что $\theta = 0$, для точки касания $x_0(\alpha, \mu, \sigma)$ справедливо

$$x_0(\alpha, \mu, \sigma) = \mu - \frac{\langle \alpha, \mu \rangle}{\langle \alpha, \sigma \cdot \alpha \rangle} \sigma \cdot \alpha. \quad (5.24)$$

При доказательстве леммы используется понятие расстояния от пары (μ, σ) до гиперплоскости $\{x \in X \mid \langle \alpha, x \rangle = 0\}$. Обозначим это расстояние $r^*(\alpha, \mu, \sigma)$. На основании (5.15) и учитывая, что $\theta = 0$, значение $r^*(\alpha, \mu, \sigma)$ равно

$$r^*(\alpha, \mu, \sigma) = \frac{\langle \alpha, \mu \rangle}{\sqrt{\langle \alpha, \sigma \cdot \alpha \rangle}}. \quad (5.25)$$

Лемма 5.2 **Необходимые и достаточные условия оптимальности α для одного распределения.** Пусть тройка (α, μ, σ) такова, что $\langle \alpha, \mu \rangle > 0$. Пусть $x_0(\alpha, \mu, \sigma)$ - точка касания, а $\Delta\alpha$ - любой вектор, не коллинеарный с α . В таком случае:

1. Если выполняется условие

$$\langle \Delta\alpha, x_0(\alpha, \mu, \sigma) \rangle > 0, \quad (5.26)$$

то существует такое положительное число T , что для любого t , $0 < t \leq T$, выполняется неравенство

$$\varepsilon(\alpha + t \cdot \Delta\alpha, \mu, \sigma) < \varepsilon(\alpha, \mu, \sigma). \quad (5.27)$$

2. Если выполняется условие

$$\langle \Delta\alpha, x_0(\alpha, \mu, \sigma) \rangle \leq 0, \quad (5.28)$$

то

$$\varepsilon(\alpha + \Delta\alpha, \mu, \sigma) > \varepsilon(\alpha, \mu, \sigma). \quad (5.29)$$

▲

Примечание 5.1 Лемма 5.2 фактически доказывает, что необходимым и достаточным условием оптимальности вектора α является то, что точка касания $x_0(\alpha, \mu, \sigma)$ совпадает с началом координат. Это утверждалось без доказательства раньше, при неформальном рассмотрении задачи Андерсона.

Действительно, если $x_0(\alpha, \mu, \sigma) = 0$, то условие (5.28) выполняется для любого вектора $\Delta\alpha$, и следовательно, для любого вектора, неколлинеарного с α . Неравенство (5.29) утверждает, что неравенство $\varepsilon(\alpha, \mu, \sigma) < \varepsilon(\alpha', \mu, \sigma)$ выполняется для любого вектора α' , неколлинеарного с α , что обозначает, что вектор α обеспечивает минимально возможное значение вероятности $\varepsilon(\alpha, \mu, \sigma)$.

С другой стороны, если $x_0(\alpha, \mu, \sigma) \neq 0$, то существует вектор $\Delta\alpha$, для которого выполняется неравенство (5.26). Это может быть, например, вектор $\Delta\alpha = x_0(\alpha, \mu, \sigma)$. Следовательно, существует точка $\alpha' = \alpha + t \cdot \Delta\alpha$ со значением $\varepsilon(\alpha', \mu, \sigma)$, меньшим, чем $\varepsilon(\alpha, \mu, \sigma)$.

Лемма 5.2 формулирует то же самое в менее прозрачной форме, так как именно в этой форме она будет нужна при доказательстве последующей теоремы. Нашей целью ведь является не минимизация вероятности $\varepsilon(\alpha, \mu, \sigma)$ для некоторой единственной пары (μ, σ) , а минимизация числа $\max_{j \in J} \varepsilon(\alpha, \mu^j, \sigma^j)$ для определенной совокупности таких пар. Лемма 5.2 есть лишь вспомогательный результат для нашей конечной цели. ▲

Доказательство. (Лемма 5.2) Рассмотрим случай, когда выполняется условие (5.26), то есть $\langle \Delta\alpha, x_0(\alpha, \mu, \sigma) \rangle > 0$. Рассмотрим функцию $r^*(\alpha + t \cdot \Delta\alpha, \mu, \sigma)$ от переменной t . В соответствии с (5.25) - это функция

$$\frac{\langle \alpha + t \cdot \Delta\alpha, \mu \rangle}{\sqrt{\langle \alpha + t \cdot \Delta\alpha, \sigma \cdot (\alpha + t \cdot \Delta\alpha) \rangle}}.$$

Ее производная в точке $t = 0$ равна

$$\left. \frac{dr^*(\alpha + t \cdot \Delta\alpha, \mu, \sigma)}{dt} \right|_{t=0} = \frac{\left\langle \left(\mu - \frac{\langle \alpha, \mu \rangle}{\langle \alpha, \sigma \cdot \alpha \rangle} (\sigma \cdot \alpha) \right), \Delta\alpha \right\rangle}{\sqrt{\langle \alpha, \sigma \cdot \alpha \rangle}}. \quad (5.30)$$

По выражению (5.24) для $x_0(\alpha, \mu, \sigma)$ ясно, что числитель в дроби в правой части (5.30) - это скалярное произведение $\langle x_0(\alpha, \mu, \sigma), \Delta\alpha \rangle$, которое по предположению положительно. Следовательно, исследуемая производная также положительна. Это значит, что существует такое положительное

число T , что для каждого t , $0 < t \leq T$, выполняется следующее неравенство,

$$r^*(\alpha + t \cdot \Delta\alpha, \mu, \sigma) > r^*(\alpha, \mu, \sigma),$$

из которого в силу леммы 5.1 следует неравенство (5.27), которое и следовало доказать. Таким образом, первое утверждение леммы 5.2 доказано.

Докажем теперь второе утверждение леммы 5.2, в котором предполагается, что выполнено условие

$$\langle \Delta\alpha, x_0(\alpha, \mu, \sigma) \rangle \leq 0.$$

Здесь надо исследовать поведение функции $r^*(\alpha, \mu, \sigma)$ не только в окрестности точки α , как в предыдущем утверждении, а в глобальном смысле. Этого нельзя сделать на основании знания тех или иных производных в точке α , а требуются дополнительные, тоже не очень сложные соображения.

Для краткости обозначим $\alpha + \Delta\alpha$ как α' и $x_0(\alpha, \mu, \sigma)$ как x_0 . Для вектора α' могут иметь место три ситуации:

$$\langle \alpha', \mu \rangle \leq 0; \quad (5.31)$$

$$\langle \alpha', \mu \rangle > 0, \quad \langle \alpha', x_0 \rangle < 0; \quad (5.32)$$

$$\langle \alpha', \mu \rangle > 0, \quad \langle \alpha', x_0 \rangle = 0. \quad (5.33)$$

Случай, когда $\langle \alpha', x_0 \rangle > 0$, можно исключить, так как $\langle \alpha', x_0 \rangle = \langle (\alpha + \Delta\alpha), x_0 \rangle = \langle \alpha, x_0 \rangle + \langle \Delta\alpha, x_0 \rangle \leq 0$. Действительно, слагаемое $\langle \alpha, x_0 \rangle$ равно нулю, так как точка касания x_0 принадлежит гиперплоскости $X^0(\alpha) = \{x \in X \mid \langle \alpha, x \rangle = 0\}$, а слагаемое $\langle \Delta\alpha, x_0 \rangle$ не положительно по предположению (5.28).

Если выполняется условие (5.31), то утверждение леммы 5.2 очевидно справедливо, так как α удовлетворяет неравенство $\langle \alpha, \mu \rangle > 0$, и, следовательно, $\varepsilon(\alpha, \mu, \sigma) < 0.5$. Неравенство же (5.31) обозначает, что $\varepsilon(\alpha', \mu, \sigma) \geq 0.5$.

Рассмотрим ситуации (5.32) и (5.33). Обозначим $F(x)$ квадратичную функцию $\langle (x - \mu), \sigma^{-1} \cdot (x - \mu) \rangle$ и докажем, что в обеих ситуациях (5.32) и (5.33) существует такая точка x^* на гиперплоскости $X^0(\alpha') = \{x \in X \mid \langle \alpha', x \rangle = 0\}$, что

$$F(x^*) < F(x_0). \quad (5.34)$$

Из этого будет следовать второго утверждения леммы 5.2. Действительно, если неравенство (5.34) справедливо (а мы это докажем) то выполняется

$$(r^*(\alpha', \mu, \sigma))^2 = \min_{x \in X^0(\alpha')} F(x) \leq F(x^*) < F(x_0) = (r^*(\alpha, \mu, \sigma))^2. \quad (5.35)$$

В силу леммы 5.1 из этого следует неравенство (5.29), которое следует доказать.

Докажем вначале, что при условии (5.32) справедливо утверждение (5.34). Скалярное произведение $\langle \alpha', x \rangle$ есть непрерывная функция вектора x и поэтому из неравенств (5.32) следует существование такого числа $0 < k < 1$, что точка $x^* = x_0 \cdot (1-k) + k \cdot \mu$ лежит на гиперплоскости $X^0(\alpha')$. Значение функции F в точке x^* равно $F(x^*) = F(x_0 \cdot (1-k) + k \cdot \mu) = (1-k)^2 \cdot F(x_0) < F(x_0)$, что доказывает (5.34).

Докажем теперь, что утверждение (5.34) следует и из условия (5.33). Поскольку векторы α и α' не коллинеарны, гиперплоскости $X^0(\alpha)$ и $X^0(\alpha')$ не совпадают. Следовательно, существует точка x' , которая принадлежит $X^0(\alpha')$ и не принадлежит $X^0(\alpha)$. Эта точка не может совпадать с x_0 , так как x_0 лежит в гиперплоскости $X^0(\alpha)$. Точка x_0 принадлежит и гиперплоскости $X^0(\alpha')$, так как по условию (5.33) выполняется равенство $\langle \alpha', x_0 \rangle = 0$. Проведем прямую линию через точки x_0 и x' , которые обе лежат в гиперплоскости $X^0(\alpha')$, и исследуем поведение функции F на этой прямой.

$$\begin{aligned} & F(x_0 + k \cdot (x' - x_0)) & (5.36) \\ &= \langle (\mu - x_0 - k \cdot (x' - x_0)), \sigma^{-1} \cdot (\mu - x_0 - k \cdot (x' - x_0)) \rangle \\ &= F(x_0) - 2k \cdot \langle (x' - x_0), \sigma^{-1}(\mu - x_0) \rangle + k^2 \cdot \langle (x' - x_0), \sigma^{-1} \cdot (x' - x_0) \rangle. \end{aligned}$$

Выберем значение k , равное

$$k = \frac{\langle (x' - x_0), \sigma^{-1} \cdot (\mu - x_0) \rangle}{\langle (x' - x_0), \sigma^{-1} \cdot (x' - x_0) \rangle},$$

и продолжим преобразование выражения (5.36)

$$\begin{aligned} & F(x_0 + k \cdot (x' - x_0)) \\ &= F(x_0) - 2k \cdot \langle (x' - x_0), \sigma^{-1} \cdot (\mu - x_0) \rangle \\ &\quad + k \cdot \frac{\langle (x' - x_0), \sigma^{-1} \cdot (\mu - x_0) \rangle}{\langle (x' - x_0), \sigma^{-1} \cdot (x' - x_0) \rangle} \cdot \langle (x' - x_0), \sigma^{-1} \cdot (x' - x_0) \rangle \\ &= F(x_0) - k \cdot \langle (x' - x_0), \sigma^{-1} \cdot (\mu - x_0) \rangle \\ &= F(x_0) - \frac{(\langle (x' - x_0), \sigma^{-1} \cdot (\mu - x_0) \rangle)^2}{\langle (x' - x_0), \sigma^{-1} \cdot (x' - x_0) \rangle}. \end{aligned} \quad (5.37)$$

Используя выражение (5.11) для x_0 , получаем $\sigma^{-1} \cdot (\mu - x_0) = -\frac{1}{2}\lambda\alpha$ и упрощаем формулу (5.37) так, что

$$F(x_0 + k \cdot (x' - x_0)) = F(x_0) - \frac{(\frac{1}{2}\lambda\langle \alpha, (x' - x_0) \rangle)^2}{\langle (x' - x_0), \sigma^{-1} \cdot (x' - x_0) \rangle}. \quad (5.38)$$

В силу $x' \notin X^0(\alpha)$ и $x_0 \in X^0(\alpha)$ выполняется неравенство $\langle \alpha, x' \rangle \neq 0$ и равенство $\langle \alpha, x_0 \rangle = 0$. Следовательно, скалярное произведение $\langle \alpha, (x' - x_0) \rangle$ не равно нулю. В соответствии с (5.12) число λ не равно нулю, равно как и число $\frac{1}{2}\lambda\langle \alpha, (x' - x_0) \rangle$ в правой части (5.38). Поэтому

$$F(x_0 + k \cdot (x' - x_0)) < F(x_0).$$

Это значит, что на прямой линии, проходящей через точки x_0 и x' , имеется точка $x^* \in X^0(\alpha')$, для которой справедливо (5.34). ■

Теперь мы обладаем достаточными знаниями, чтобы сформулировать и доказать необходимые и достаточные условия того, что вектор α минимизирует число $\max_{j \in J} \varepsilon(\alpha, \mu^j, \sigma^j)$. В теореме говорится о векторе α , который удовлетворяет условию $\langle \alpha, \mu^j \rangle > 0$, $j \in J$, а также о зависящих от этого вектора числах

$$(r^j)^2 = \min_{x \in H(\alpha)} \langle x - \mu^j, \sigma^j \cdot (x - \mu^j) \rangle, \quad j \in J,$$

подмножестве J^0 индексов $j \in J$, для которых $r^j = \min_{j \in J} r^j$, и точках касания x_0^j , $j \in J^0$.

Теорема 5.2 Необходимые и достаточные условия решения обобщенной задачи Андерсона. *Если выпуклое замыкание множества точек касания x_0^j , $j \in J^0$, включает начало координат, то для любого вектора α' , неколинеарного с вектором α справедливо, что*

$$\max_{j \in J} \varepsilon(\alpha', \mu^j, \sigma^j) > \max_{j \in J} \varepsilon(\alpha, \mu^j, \sigma^j).$$

Если это выпуклое замыкание не содержит начало координат, то существуют такие вектор $\Delta\alpha$ и положительное число T , что для любого t , $0 < t \leq T$, выполняется неравенство

$$\max_{j \in J} \varepsilon(\alpha + t \cdot \Delta\alpha, \mu^j, \sigma^j) < \max_{j \in J} \varepsilon(\alpha, \mu^j, \sigma^j). \quad \blacktriangle$$

Доказательство. Докажем вначале первое утверждение теоремы 5.2. Здесь предполагается существование таких чисел γ^j , $j \in J^0$, которые удовлетворяют условиям

$$\gamma^j \geq 0, \quad j \in J^0, \quad \sum_{j \in J^0} \gamma^j = 1, \quad \sum_{j \in J^0} \gamma^j \cdot x_0^j = 0.$$

Отсюда следует, что для любого вектора α' справедливо равенство

$$\sum_{j \in J^0} \gamma^j \cdot \langle \alpha', x_0^j \rangle = 0,$$

в том числе для некоторого ненулевого вектора α' , не колинеарного с вектором α . Приведенная сумма может равняться нулю лишь в том случае, если по крайней мере для одного $j^* \in J^0$ выполняется неравенство

$$\langle \alpha', x_0^{j^*} \rangle \leq 0.$$

Равенство $\langle \alpha, x_0^j \rangle = 0$ выполняется для всех $j \in J$, и следовательно, и для j^* . Это значит, что для вектора $\Delta\alpha = \alpha' - \alpha$ выполняется неравенство $\langle \Delta\alpha, x_0^{j^*} \rangle \leq 0$. Отсюда в силу леммы 5.2 следует, что $\varepsilon(\alpha', \mu^{j^*}, \sigma^{j^*}) > \varepsilon(\alpha, \mu^{j^*}, \sigma^{j^*})$. Очевидно, что число $\max_{j \in J} \varepsilon(\alpha, \mu^j, \sigma^j)$ равно $\varepsilon(\alpha, \mu^{j^*}, \sigma^{j^*})$, так как $j^* \in J^0$, а число $\max_{j \in J} \varepsilon(\alpha', \mu^j, \sigma^j)$ не превосходит $\varepsilon(\alpha', \mu^{j^*}, \sigma^{j^*})$. Таким образом,

$\max_{j \in J} \varepsilon(\alpha', \mu^j, \sigma^j) > \max_{j \in J} \varepsilon(\alpha, \mu^j, \sigma^j)$, и первое утверждение теоремы 5.2 доказано.

Обозначим X_0 выпуклое замыкание точек касания x^j , $j \in J^0$, и докажем второе утверждение теоремы 5.2, в котором предполагается, что $0 \notin X_0$. В этом случае существует вектор $\Delta\alpha$, который удовлетворяет неравенство $\langle \Delta\alpha, x_0^j \rangle > 0$ при любом $j \in J^0$. Это может быть, например, точка $\arg \min_{x \in X_0} |x|$. Отсюда в силу первого утверждения леммы 5.2 следует, что существуют такие положительные числа T^j , $j \in J^0$, что

$$\forall (j \in J^0) \forall (t \mid 0 \leq T^j): \varepsilon(\alpha + \Delta\alpha \cdot t, \mu^j, \sigma^j) < \varepsilon(\alpha, \mu^j, \sigma^j).$$

Это утверждение остается верным, если все числа T^j заменить числом $T' = \min_{j \in J^0} T^j$, и после этого поменять местами кванторы. Таким образом получим

$$\forall (t \mid 0 < t \leq T') \forall (j \in J^0): \varepsilon(\alpha + \Delta\alpha \cdot t, \mu^j, \sigma^j) < \varepsilon(\alpha, \mu^j, \sigma^j). \quad (5.39)$$

По определению множества J^0 каждое значение $\varepsilon(\alpha, \mu^j, \sigma^j)$, $j \in J^0$, равно $\max_{j \in J} \varepsilon(\alpha, \mu^j, \sigma^j)$ и поэтому выражение (5.39) можно модифицировать следующим образом:

$$\forall (t \mid 0 < t \leq T') \forall (j \in J^0): \varepsilon(\alpha + \Delta\alpha \cdot t, \mu^j, \sigma^j) < \max_{j \in J} \varepsilon(\alpha, \mu^j, \sigma^j),$$

$$\forall (t \mid 0 < t \leq T'): \max_{j \in J^0} \varepsilon(\alpha + \Delta\alpha \cdot t, \mu^j, \sigma^j) < \max_{j \in J} \varepsilon(\alpha, \mu^j, \sigma^j). \quad (5.40)$$

Величина $\varepsilon(\alpha, \mu^j, \sigma^j)$ есть непрерывная функция вектора α . Поэтому, если для некоторого индекса j' выполняется неравенство

$$\varepsilon(\alpha, \mu^{j'}, \sigma^{j'}) \neq \max_{j \in J} \varepsilon(\alpha, \mu^j, \sigma^j),$$

это неравенство остается в силе по крайней мере для достаточно малых значений t ,

$$\varepsilon(\alpha + \Delta\alpha \cdot t, \mu^{j'}, \sigma^{j'}) \neq \max_{j \in J} \varepsilon(\alpha + \Delta\alpha \cdot t, \mu^j, \sigma^j).$$

Следовательно, существует положительное число T (оно может быть меньше, чем T'), для которого неравенство

$$\max_{j \in J^0} \varepsilon(\alpha + \Delta\alpha \cdot t, \mu^j, \sigma^j) = \max_{j \in J} \varepsilon(\alpha + \Delta\alpha \cdot t, \mu^j, \sigma^j)$$

остается в силе для любого t , $0 < t \leq T$. На этом основании можно переписать утверждение (5.40) в виде

$$\forall (t \mid 0 < t \leq T): \max_{j \in J} \varepsilon(\alpha + \Delta\alpha \cdot t, \mu^j, \sigma^j) < \max_{j \in J} \varepsilon(\alpha, \mu^j, \sigma^j),$$

что доказывает второе утверждение теоремы 5.2. ■

Доказанная теорема 5.2 указывает путь к построению процедуры минимизации величины $\max_{j \in J} \varepsilon(\alpha, \mu^j, \sigma^j)$. Алгоритм минимизации должен прийти в такое состояние, при котором многогранник, вершинами которого являются точки касания $x_0(\alpha, \mu^j, \sigma^j)$, $j \in J_0$, содержит начало координат. В этом состоянии достигается решение задачи Андерсона. Если текущее состояние алгоритма не является таким, то гарантированно существует такое направление $\Delta\alpha$, движение вдоль которого уменьшает величину $\max_{j \in J} \varepsilon(\alpha, \mu^j, \sigma^j)$. В доказательстве теоремы 5.2 указывается один из возможных способов нахождения направления $\Delta\alpha$. Это направление есть вектор, скалярное произведение которого с каждой точкой касания положительно. Отыскание такого вектора $\Delta\alpha$ совпадает с нахождением простого разделения конечных множеств точек. Таким образом, для решения задачи Андерсона требуется многократно решить задачу простого разделения конечных множеств точек. Этой задаче мы посвятим следующий раздел, так как эта задача представляет самостоятельный интерес, а не только как вспомогательное средство для решения задачи Андерсона. Мы рассмотрим эту задачу с особой тщательностью, так как в конце лекции мы увидим, что при определенной разумной постановке задача Андерсона во всей ее общности может формулироваться как задача простого разделения бесконечного множества точек. Хорошо поняв алгоритмы разделения конечных множеств точек, мы сможем их несколько видоизменить так, что они станут пригодны и для разделения бесконечных множеств определенного вида, к которому сводятся задачи Андерсона.

5.4 Линейное разделение конечных множеств точек

5.4.1 Формулировка задачи и ее анализ

Пусть J - конечное множество индексов, которое разделено на два подмножества J_1 и J_2 , а $X = \{x^j, j \in J\}$ - конечное множество точек в линейном пространстве. Требуется найти вектор α , который является решением системы неравенств

$$\left. \begin{aligned} \langle \alpha, x^j \rangle &> 0, & j \in J_1, \\ \langle \alpha, x^j \rangle &< 0, & j \in J_2. \end{aligned} \right\} \quad (5.41)$$

Решение этой системы неравенств назовем *простым разделением конечных множеств точек*.

Если система (5.41) имеет решение, то имеет решение и система

$$\left. \begin{aligned} \left\langle \frac{\alpha}{|\alpha|}, x^j \right\rangle &> 0, & j \in J_1, \\ \left\langle \frac{\alpha}{|\alpha|}, x^j \right\rangle &< 0, & j \in J_2. \end{aligned} \right\} \quad (5.42)$$

Если система (5.42) имеет по крайней мере одно решение, то она имеет и бесконечно много решений. Поэтому ужесточим требование (5.42), поставив задачу поиска вектора α и как можно большего числа r , при котором система

$$\left. \begin{aligned} \left\langle \frac{\alpha}{|\alpha|}, x^j \right\rangle &\geq r, & j \in J_1, \\ \left\langle \frac{\alpha}{|\alpha|}, x^j \right\rangle &\leq -r, & j \in J_2, \end{aligned} \right\} \quad (5.43)$$

еще не противоречива. Иными словами говоря, следует найти вектор

$$\alpha = \operatorname{argmax}_{\alpha} \min \left(\min_{j \in J_1} \frac{\langle \alpha, x^j \rangle}{|\alpha|}, \min_{j \in J_2} \frac{-\langle \alpha, x^j \rangle}{|\alpha|} \right). \quad (5.44)$$

Решение задачи (5.44) назовем *оптимальным разделением конечных множеств точек*. Эта задача является частным случаем обобщенной задачи Андерсона, когда все матрицы σ^j , $j \in J$, единичные.

Этот частный случай имеет наглядную *геометрическую интерпретацию*, на которую мы будем несколько раз опираться как при неформальных, та и при формальных рассуждениях. В задаче (5.42) требуется найти гиперплоскость, которая отделяет множество точек $\{x^j, j \in J_1\}$ от множества точек $\{x^j, j \in J_2\}$. Выражения в левой части неравенств (5.42) — это расстояния от точек до гиперплоскости. В задачах (5.43) и (5.44) требуется среди гиперплоскостей, удовлетворяющих условиям (5.42), найти гиперплоскость, наиболее удаленную от заданных точек.

Задачи (5.43), (5.44) могут иметь и другую интерпретацию, более близкую к задачам Андерсона. Любая гиперплоскость, решающая задачу (5.42), отделяет не только точки x^j , $j \in J_1$, от точек x^j , $j \in J_2$, но и определенные r -окрестности этих точек. Размер этих окрестностей, то есть величина r , зависит от гиперплоскости. В задачах (5.43), (5.44) требуется найти гиперплоскость, которая, отделяя одно множество точек от другого, отделяет и как можно большие их окрестности.

Введем обозначение x'^j , $j \in J$, так, что $x'^j = x^j$ при $j \in J_1$ и $x'^j = -x^j$ при $j \in J_2$. Тогда система неравенств (5.42) приобретает вид

$$\langle \alpha, x'^j \rangle > 0, \quad j \in J, \quad (5.45)$$

а задача (5.44) приобретает вид

$$\alpha = \operatorname{argmax}_{\alpha} \min_{j \in J} \left\langle \frac{\alpha}{|\alpha|}, x'^j \right\rangle. \quad (5.46)$$

Мы будем исследовать задачи в обеих формулировках. Первая формулировка (5.45) соответствует задаче простого разделения двух заданных множеств. Как видим, она сведена к задаче отыскания полупространства, которое содержит в себе полностью некоторое заданное множество. Вторая формулировка (5.46) соответствует задаче оптимального разделения двух

множеств. Как видим, она свелась к нахождению такого полупространства, которое содержит в себе полностью некоторое заданное множество, причем так, что это множество оказывается как можно больше удаленным от границы полупространства.

Необходимые и достаточные условия оптимальности гиперплоскости, которые для задачи Андерсона сформулированы в теореме 5.2, в данном частном случае можно выразить в следующей, почти геометрической форме.

Теорема 5.3 **Необходимые и достаточные условия оптимального разделения конечных множеств точек.** Пусть \overline{X} - выпуклое замыкание множества точек $\{x^j, j \in J\}$, а α^* - точка в \overline{X} , ближайшая к началу координат,

$$\alpha^* = \operatorname{argmin}_{x \in \overline{X}} |x|. \quad (5.47)$$

Если $\alpha^* \neq 0$, то α^* есть решение задачи (5.46). \blacktriangle

Доказательство. Рассмотрим для каждого $j \in J$ треугольник, вершинами которого являются начало координат, точка α^* и точка x^j . Обозначим \overline{X}^j сторону треугольника, которая соединяет вершины α^* и x^j . Имеет место отношение $\overline{X}^j \subset \overline{X}$, так как $\alpha^* \in \overline{X}$ и $x^j \in \overline{X}$, а множество \overline{X} выпуклое. Отсюда в силу условия (5.47) следует, что

$$\alpha^* = \operatorname{argmin}_{x \in \overline{X}^j} |x|,$$

то есть, что α^* есть ближайшая к началу координат точка на отрезке \overline{X}^j . Следовательно, угол при вершине α^* в рассматриваемом треугольнике не может быть острым, а скалярное произведение векторов $-\alpha^*$ и $x^j - \alpha^*$ не может быть положительным,

$$\langle -\alpha^*, x^j - \alpha^* \rangle \leq 0,$$

или, что тоже самое,

$$|\alpha^*|^2 \leq \langle \alpha^*, x^j \rangle. \quad (5.48)$$

Вектор α^* принадлежит \overline{X} . Следовательно,

$$\alpha^* = \sum_{j \in J} \gamma^j \cdot x^j \quad (5.49)$$

при некоторых неотрицательных коэффициентах γ^j , сумма которых равна 1. Из сказанного следует равенство $\langle \alpha^*, \alpha^* \rangle = \sum_{j \in J} \gamma^j \cdot \langle \alpha^*, x^j \rangle$ или, в несколько ином виде,

$$\sum_{j \in J} \gamma^j \cdot (\langle \alpha^*, x^j \rangle - |\alpha^*|^2) = 0.$$

Мы видим, что сумма неотрицательных чисел (см. (5.48)) равна нулю. Это возможно лишь тогда, когда все слагаемые равны нулю,

$$\gamma^j \cdot (\langle \alpha^*, x^j \rangle - |\alpha^*|^2) = 0, \quad j \in J.$$

Некоторые коэффициенты γ^j , $j \in J$, обязаны быть ненулевыми, так как сумма всех коэффициентов равна 1. Обозначим J^0 множество индексов j , для которых $\gamma^j \neq 0$. Для каждого из таких $j \in J^0$ должно выполняться $\langle \alpha^*, x^j \rangle = |\alpha^*|^2$. Отсюда в силу неравенства (5.48) следует, что для каждого $j \in J^0$ и $j' \in J$ выполняется

$$\langle \alpha^*, x^j \rangle \leq \langle \alpha^*, x^{j'} \rangle$$

или, что тоже самое,

$$\left\langle \frac{\alpha^*}{|\alpha^*|}, x^j \right\rangle = \min_{j' \in J} \left(\left\langle \frac{\alpha^*}{|\alpha^*|}, x^{j'} \right\rangle \right), \quad j \in J^0.$$

Мы доказали, что в выражении (5.49) ненулевыми являются коэффициенты γ^j только при тех векторах x^j , которые наименее удалены от гиперплоскости $\langle \alpha, x \rangle = 0$. Поэтому выражение (5.49) приобретает вид

$$\alpha^* = \sum_{j \in J^0} \gamma^j \cdot x^j.$$

Исходя из этого последнего выражения мы докажем, что выпуклое замыкание точек касания содержит начало координат, и как следствие теоремы 5.2 получим, что α^* есть решение задачи. Действительно, используя формулу (5.15) для точки касания и учитывая, что $\sigma^j = 1$, запишем

$$x_0^j = x^j - \alpha^* \cdot \frac{\langle \alpha^*, x^j \rangle}{|\alpha^*|^2}$$

и

$$\begin{aligned} \sum_{j \in J_0} \gamma^j \cdot x_0^j &= \sum_{j \in J_0} \gamma^j \cdot x^j - \alpha^* \cdot \frac{\langle \alpha^*, \sum_{j \in J_0} \gamma^j \cdot x^j \rangle}{|\alpha^*|^2} \\ &= \alpha^* - \alpha^* \cdot \frac{\langle \alpha^*, \alpha^* \rangle}{|\alpha^*|^2} = \alpha^* - \alpha^* = 0. \end{aligned}$$

Таким образом, выпуклое замыкание точек касания действительно содержит начало координат. ■

Доказанная теорема сводит задачу оптимального разделения двух конечных множеств точек к минимизации квадратичной функции на выпуклом многограннике. Полученный результат является фактически решением задачи, так как проблема минимизации квадратичных функций достаточно хорошо изучена и имеется ряд методов для ее решения. Однако специфика

исследуемой задачи позволяет решить ее более наглядными и более простыми алгоритмами, которые будут описаны далее.

Доказанную теорему и идею ее доказательства можно выразить в следующих неравенствах, которые нам еще не раз пригодятся: Для любого вектора α и любого вектора $x \in \bar{X}$ выполняется

$$\min_{j \in J} \left\langle \frac{\alpha}{|\alpha|}, x^j \right\rangle \leq \min_{j \in J} \left\langle \frac{\alpha^*}{|\alpha^*|}, x^j \right\rangle = |\alpha^*| \leq |x|. \quad (5.50)$$

5.4.2 Алгоритмы линейного разделения конечных множеств точек

Оптимальное разделение конечных множеств точек может показаться на первый взгляд более важным, чем простое разделение. Естественно, алгоритм решения задачи (5.46) можно применить и для решения задачи (5.45). При всей несомненности этого утверждения мы подчеркиваем, что задача простого разделения имеет свое самостоятельное значение и не только потому, что она проще. Наиболее важно то, что при достаточно хорошем понимании алгоритмов простого разделения они поддаются разнообразным несложным модификациям. При некоторых модификациях получаются алгоритмы решения не только задачи оптимального разделения, но и задач Андерсона в их полной общности.

Алгоритм Козинца для линейного разделения конечного множества точек

Пусть $\{x^j, j \in J\}$ - конечное множество точек, для которого существует вектор $\bar{\alpha}$ (хотя и неизвестный), который удовлетворяет систему неравенств

$$\langle \bar{\alpha}, x^j \rangle > 0, \quad j \in J. \quad (5.51)$$

Мы покажем процедуру, известную, как алгоритм Козинца, которая находит вектор α , удовлетворяющий системе (5.51), хотя, вообще говоря, отличающийся от $\bar{\alpha}$.

Построим последовательность векторов $\alpha_1, \alpha_2, \dots, \alpha_t, \alpha_{t+1}, \dots$ по следующему алгоритму.

Algorithm 5.1 Алгоритм КОзинца для простого разделения множеств

Вектор α_1 - это произвольный вектор из \bar{X} то есть, из выпуклого замыкания множества $\{x^j \mid j \in J\}$. Например, это может быть один из векторов $x^j, j \in J$. Допустим, что вычислен вектор α_t . Вектор α_{t+1} следует построить по следующим правилам:

1. Найти вектор $x^j, j \in J$, который удовлетворяет условию

$$\langle \alpha_t, x^j \rangle \leq 0. \quad (5.52)$$

2. Если такого вектора x^j нет, то задача уже решена и α_t является искомым вектором.

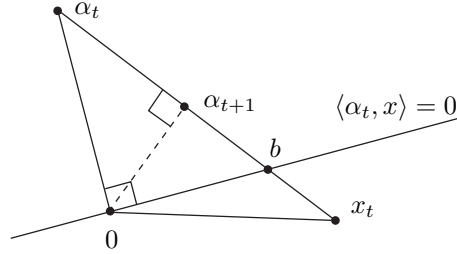


Figure 5.10 Geometrical interpretation of properties of the points α_t , α_{t+1} and x_t .

3. Пусть такой вектор x^j найден и обозначен x_t . Вектор α_{t+1} определяется как точка на прямолинейном отрезке, соединяющем точки α_t и x_t , ближайшая к началу координат. Это значит, что

$$\alpha_{t+1} = (1 - k) \cdot \alpha_t + k \cdot x_t, \quad k \in \mathbb{R}, \quad (5.53)$$

где

$$k = \operatorname{argmin}_k |(1 - k) \cdot \alpha_t + k \cdot x_t|. \quad (5.54)$$

Приведенный алгоритм 5.1 гарантирует, что вектор α_t на некотором шаге t будет удовлетворять требованиям (5.51), как это утверждает следующая теорема.

Теорема 5.4 **О конечно-шаговой сходимости алгоритма Козинца.** Если условия (5.51) не противоречивы, а последовательность $\alpha_1, \alpha_2, \dots, \alpha_t, \alpha_{t+1}, \dots$ построена алгоритмом Козинца, то существует число t^* , при котором

$$\langle \alpha_{t^*}, x^j \rangle > 0, \quad j \in J.$$

▲

Доказательство. Доказательство выполним, прибегая к геометрической интерпретации свойств точек α_t , α_{t+1} и x_t , которые определяет алгоритм. Вектор α_{t+1} - это основание перпендикуляра, опущенного из начала координат на прямую, проходящую через точки α_t и x_t , как показано на рис. 5.10. Вектор α_{t+1} , очевидно, есть выпуклая линейная комбинация векторов α_t и x_t .

Так как α_1 принадлежит выпуклому множеству \overline{X} , а α_{t+1} есть выпуклая линейная комбинация точек α_t и x_t , $x_t \in \overline{X}$, то вся последовательность векторов $\alpha_2, \alpha_3, \dots, \alpha_t, \dots$ также принадлежит \overline{X} . Для каждого из этих векторов, следовательно, справедливо неравенство

$$|\alpha_t| \geq \varepsilon, \quad \text{где } \varepsilon = \min_{x \in \overline{X}} |x| \quad (5.55)$$

Поскольку неравенства в системе (5.51) строгие, а сама система не противоречива, множество \overline{X} не содержит начало координат. Следовательно, $\varepsilon > 0$.

На основе чисто геометрических соображений оценим отношение $|\alpha_{t+1}|/|\alpha_t|$ для $\alpha_{t+1} \neq \alpha_t$. Обозначим b точку пересечения гиперплоскости $\langle \alpha_t, x \rangle = 0$ и прямой линии, проходящей через точки α_t и x_t . Обозначим g расстояние от точки α_t до точки b . Мы видим по рис. 5.10, что

$$\frac{|\alpha_{t+1}|}{|\alpha_t|} = \frac{g}{\sqrt{|\alpha_t|^2 + g^2}} = \frac{1}{\sqrt{1 + (|\alpha_t|^2/g^2)}}.$$

Обозначим $D = \max_{j \in J} |x_j|$. В силу $b \in \bar{X}$ справедливо неравенство $g \leq D$. Используя (5.55), можно написать

$$\frac{|\alpha_t|^2}{|g|^2} \geq \frac{\varepsilon^2}{D^2}, \quad \sqrt{1 + \frac{|\alpha_t|^2}{|g|^2}} \geq \sqrt{1 + \frac{\varepsilon^2}{D^2}}$$

и

$$\frac{|\alpha_{t+1}|}{|\alpha_t|} \leq \frac{1}{\sqrt{1 + \varepsilon^2/D^2}} < 1.$$

Таким образом, последовательность длин $|\alpha_1|, \dots, |\alpha_t|, \dots$ убывает быстрее, чем некоторая убывающая геометрическая прогрессия. Если бы последовательность $\alpha_1, \dots, \alpha_t, \dots$ была неограниченно длинной, длина $|\alpha_t|$ могла бы стать меньше любого положительного числа. Но в силу (5.55) число $|\alpha_t|$ не может стать меньше, чем ε . Следовательно, при некотором t^* вектор α_{t^*} обязан перестать меняться. Теорема доказана. ■

Для полноты укажем, что величину t^* можно оценить с помощью неравенства

$$\begin{aligned} \frac{|\alpha_{t+1}|}{|\alpha_1|} &\leq \left(\frac{1}{\sqrt{1 + \varepsilon^2/D^2}} \right)^{t^*}, & \frac{\varepsilon}{D} &\leq \left(\frac{1}{\sqrt{1 + \varepsilon^2/D^2}} \right)^{t^*}, \\ -t^* \frac{1}{2} \ln \left(1 + \frac{\varepsilon^2}{D^2} \right) &\geq \ln \frac{\varepsilon}{D}; & t^* &\leq \frac{\ln(D^2/\varepsilon^2)}{\ln(1 + \varepsilon^2/D^2)}. \end{aligned}$$

При достаточно малом значении ε^2/D^2 становится справедливым приближительное равенство $\ln(1+x) \approx x$ и более удобная оценка

$$t^* \approx \frac{D^2}{\varepsilon^2} \ln \frac{D^2}{\varepsilon^2}.$$

Персептрон и теорема Новикова

Алгоритм Козинца настолько элегантен и прост, что, казалось бы, трудно придумать что-нибудь еще более простое. Оказывается, это возможно, и таким алгоритмом является знаменитый алгоритм персептрона. Мы сформулируем алгоритм персептрона и теорему Новикова о том, что этот алгоритм находит решение задачи простого разделения конечного множества точек, если только эта задача вообще имеет решение.

Пусть $X = \{x^j \mid j \in J\}$ - конечное множество векторов, \bar{X} - выпуклое замыкание этого множества, и пусть

$$\varepsilon = \min_{x \in \bar{X}} |x| > 0, \quad D = \max_{j \in J} |x^j|.$$

Пусть последовательность $\alpha_1, \alpha_2, \dots, \alpha_t, \alpha_{t+1}, \dots$ векторов получена следующим образом.

Algorithm 5.2 Разделение конечных множеств точек с помощью персептрона

1. Вектор α_1 равен нулю. Если вектор α_t , $t = 1, 2, \dots$ известен, то вектор α_{t+1} определяется по правилам:
2. Если для каждого $j \in J$ выполняется неравенство $\langle \alpha_t, x^j \rangle > 0$, алгоритм заканчивает свою работу.
3. Если x_t один из векторов x^j , $j \in J$, для которого выполняется неравенство $\langle \alpha_t, x^j \rangle \leq 0$, то

$$\alpha_{t+1} = \alpha_t + x_t.$$

Американский математик Новиков доказал знаменитую теорему.

Теорема 5.5 **Теорема Новикова о сходимости персептрона.** *Существует число $t^* \leq D^2/\varepsilon^2$, такое, что вектор α_{t^*} удовлетворяет неравенство*

$$\langle \alpha_{t^*}, x^j \rangle > 0$$

для каждого $j \in J$. ▲

Доказательство. Если на некотором шаге t не выполнены условия $\langle \alpha_t, x^j \rangle > 0$, $j \in J$, и, следовательно, $\alpha_{t+1} \neq \alpha_t$, то на всех предыдущих шагах $t' \leq t$ тоже имело место неравенство $\alpha_{t'+1} \neq \alpha_{t'}$. Кроме того, на всех предыдущих шагах $t' \leq t$ было найдено такое $x_{t'}$, что $\langle \alpha_{t'}, x_{t'} \rangle \leq 0$. Следовательно на шаге t , равно как и на всех предыдущих шагах, имеет место следующая цепочка равенств и неравенств,

$$|\alpha_{t+1}|^2 = |\alpha_t + x_t|^2 = |\alpha_t|^2 + 2 \cdot \langle \alpha_t, x_t \rangle + |x_t|^2 \leq |\alpha_t|^2 + |x_t|^2 \leq |\alpha_t|^2 + D^2,$$

из которой следует, что

$$|\alpha_{t+1}|^2 \leq t \cdot D^2, \quad (5.56)$$

так как $\alpha_1 = 0$. Введем обозначение $\alpha^* = \operatorname{argmin}_{x \in \bar{X}} |x|$. Число $|\alpha^*|$ равно ε . В соответствии с (5.50) получим

$$\left\langle \frac{\alpha^*}{|\alpha^*|}, x^j \right\rangle \geq |\alpha^*| = \varepsilon.$$

Для скалярного произведения $\langle \alpha^*/|\alpha^*|, \alpha_{t+1} \rangle$ справедлива цепочка

$$\left\langle \frac{\alpha^*}{|\alpha^*|}, \alpha_{t+1} \right\rangle = \left\langle \frac{\alpha^*}{|\alpha^*|}, (\alpha_t + x_t) \right\rangle = \left\langle \frac{\alpha^*}{|\alpha^*|}, \alpha_t \right\rangle + \left\langle \frac{\alpha^*}{|\alpha^*|}, x_t \right\rangle \geq \left\langle \frac{\alpha^*}{|\alpha^*|}, \alpha_t \right\rangle + \varepsilon,$$

из которой немедленно следует

$$\left\langle \frac{\alpha^*}{|\alpha^*|}, \alpha_{t+1} \right\rangle \geq t \cdot \varepsilon,$$

так как $\alpha_1 = 0$. Скалярное произведение векторов не превосходит произведение длин векторов. Поэтому

$$|\alpha_{t+1}| \geq |\alpha_{t+1}| \cdot \left| \frac{\alpha^*}{|\alpha^*|} \right| \geq \left\langle \frac{\alpha^*}{|\alpha^*|}, \alpha_{t+1} \right\rangle \geq t \cdot \varepsilon,$$

то есть

$$|\alpha_{t+1}| \geq t \cdot \varepsilon.$$

Разделим неравенство (5.56) на неравенство $|\alpha_{t+1}|^2 \geq t^2 \cdot \varepsilon^2$ и получим $t \leq D^2/\varepsilon^2$. Это значит, что алгоритм персептрона может изменить вектор α не более, чем D^2/ε^2 раз. Следовательно, не позже, чем на $(D^2/\varepsilon^2 + 1)$ -ом шаге будут выполнены все неравенства $\langle \alpha_t, x^j \rangle > 0, j \in J$. ■

Если сравнивать персептронные алгоритмы с алгоритмами Козинца, то на первый взгляд может показаться, что первые лучше, так как количество итераций в алгоритме Козинца оценивается числом

$$\frac{D^2}{\varepsilon^2} \ln \frac{D^2}{\varepsilon^2},$$

а в персептроне - числом D^2/ε^2 . Такой вывод, однако, был бы чересчур поспешным, так как обе оценки слишком грубые. Мы многократно использовали как алгоритм Козинца, так и алгоритм персептрона, и при решении наших прикладных задач алгоритмы Козинца оказывались заметно более быстродействующими, чем алгоритмы персептрона. Однако значение этого опыта тоже нельзя преувеличивать и утверждать, что алгоритмы персептрона хуже. Речь идет всего лишь об эмпирическом наблюдении. Вопрос о вычислительной трудоемкости обоих описанных алгоритмов, конечно же, заслуживает большего внимания, чем до сих пор ему уделялось.

Пожалуй, не так уж важно, какой из этих алгоритмов и в каком отношении лучше. Важно, что достоинством их обоих является исключительная простота и наглядность. Кроме того, они богаты идейно, если позволительно так выразиться, они легко модифицируются, в результате чего становятся пригодными для решения и задач, далеких от тех, для которых они первоначально предназначались. В одной из последних лекций мы рассмотрим задачи настройки одного класса алгоритмов структурного распознавания, решением которых совершенно неожиданно для нас оказались несколько видоизмененные алгоритмы Козинца и персептронные алгоритмы. Другие модификации этих алгоритмов описаны далее в этой лекции.

5.4.3 Алгоритм ε -оптимального разделения конечных множеств точек с помощью гиперплоскости

Пусть $\{x^j, j \in J\}$ - конечное множество точек, таких, что их выпуклое замыкание \bar{X} не содержит начало координат. Пусть также

$$r^* = \min_{x \in \bar{X}} |x| > 0, \quad D = \max_{x \in \bar{X}} |x| = \max_{j \in J} |x^j|, \quad \alpha^* = \operatorname{argmin}_{x \in \bar{X}} |x|.$$

Мы уже доказали раньше, что вектор α^* максимизирует число

$$\min_{j \in J} \left\langle \frac{\alpha}{|\alpha|}, x^j \right\rangle,$$

то есть является решением задачи оптимального разделения конечного множества точек.

Вектор α (возможно, и не равный α^*) определяется как ε -оптимальное решение задачи разделения конечных множеств точек, если выполняется неравенство

$$\min_{j \in J} \left\langle \frac{\alpha^*}{|\alpha^*|}, x^j \right\rangle - \min_{j \in J} \left\langle \frac{\alpha}{|\alpha|}, x^j \right\rangle \leq \varepsilon.$$

Отыскание ε -оптимального решения задачи выполняется алгоритмом, который является незначительной модификацией алгоритма Козинца. Алгоритм строит последовательность векторов $\alpha_1, \alpha_2, \dots, \alpha_t, \alpha_{t+1}, \dots$ по следующим правилам.

Algorithm 5.3 ε -оптимальное разделение конечных множеств точек

1. Вектор α_1 может быть любым вектором из множества \overline{X} , например, любым вектором из $x^j, j \in J$. Допустим, что найден вектор α_t . Вектор α_{t+1} строится следующим образом:
2. Проверяется условие

$$|\alpha_t| - \min_{j \in J} \left\langle \frac{\alpha_t}{|\alpha_t|}, x^j \right\rangle \leq \varepsilon, \quad j \in J. \quad (5.57)$$

3. Если предыдущее условие выполняется, алгоритм заканчивает свою работу.
 4. Если условие (5.57) не выполняется, то находится точка x_t из множества $\{x^j, j \in J\}$ с наименьшим значением скалярного произведения с вектором α_t .
 5. Находится вектор α_{t+1} , как точка на прямолинейном отрезке, соединяющем точки α_t и x_t , наименее удаленная от начала координат.
-

Как видно, этот алгоритм почти не отличается от алгоритма простого разделения множеств, определенного правилами (5.52)–(5.53). Эти два алгоритма отличаются лишь правилом остановки. В алгоритме простого разделения условие (5.52) прекращает работу алгоритма, когда все скалярные произведения $\langle \alpha_t/|\alpha_t|, x^j \rangle$ положительны. В алгоритме ε -оптимального решения используется другое условие (5.57), которое более жесткое при малых ε . Алгоритм останавливается по этому условию, когда все скалярные произведения не меньше, чем $|\alpha_t| - \varepsilon$.

Для такого модифицированного алгоритма тоже можно доказать, что на некотором шаге гарантированно выполнится условие (5.57), так как длины $|\alpha_t|$ убывают быстрее, чем некоторая убывающая геометрическая прогрессия. При неограниченно длительной работе алгоритма длина $|\alpha_t|$ могла

бы стать меньше любой положительной величины. Однако это невозможно, так как вектор α_t на каждом шаге t остается в пределах выпуклого множества \bar{X} . Поэтому его длина не может стремиться к нулю.

Из условия (5.57) и из неравенства

$$\min_{j \in J} \left\langle \frac{\alpha^*}{|\alpha^*|}, x^j \right\rangle \leq |\alpha_t|,$$

(см. 5.50) получаем, что

$$\min_{j \in J} \left\langle \frac{\alpha^*}{|\alpha^*|}, x^j \right\rangle - \min_{j \in J} \left\langle \frac{\alpha_t}{|\alpha_t|}, x^j \right\rangle \leq \varepsilon.$$

Это доказывает, что вектор α_t , полученный в момент остановки алгоритма, есть решение задачи ε -оптимального разделения.

Ра уж модифицированный алгоритм так прост, возникает естественный вопрос, как этот алгоритм будет себя вести при $\varepsilon = 0$, то есть, когда его захотят использовать не для поиска ε -оптимального решения, а для оптимального. В этом случае алгоритм будет работать бесконечно долго (естественно, при условии, что он реализуется на вычислителе со сколь угодно точным представлением вещественных чисел). В этом случае алгоритм перестает быть, строго говоря, алгоритмом. Несмотря на это, для такого "алгоритма" можно доказать, что последовательность $\alpha_1, \alpha_2, \dots, \alpha_t, \dots$ стремится к предельному вектору α^* , который является оптимальным решением задачи. Однако в этом случае уже нельзя утверждать, что длины последовательно получаемых векторов ограничены сверху убывающей геометрической прогрессией. Здесь требуются более тонкие рассуждения. Не будем входить в этот интересный, но не очень простой круг вопросов, так как это не мешает практически использовать предложенный "алгоритм" и при $\varepsilon = 0$. Пользователь запускает алгоритм и на каждом шаге наблюдает величину $|\alpha_t|$ и число

$$\max_{t' \leq t} \min_{j \in J} \left\langle \frac{\alpha_{t'}}{|\alpha_{t'}|}, x^j \right\rangle = f_t.$$

Это число показывает, что среди уже полученных ранее векторов имеется вектор α с качеством f_t . Число $|\alpha_t|$ показывает, что даже при бесконечном продолжении работы "алгоритма" не будет достигнуто качество, лучшее, чем $|\alpha_t|$. Наблюдая, как меняются эти два числа в процессе работы "алгоритма", пользователь, как правило, не затрудняется в принятии одного из двух решений: прекратить ли работу "алгоритма" или продолжить ее в надежде, что качество решения еще улучшится.

5.4.4 Построение фишеровских классификаторов модифицированными алгоритмами Козинца и персептронными алгоритмами

Мы увидели, что алгоритмы, описанные в предыдущих подразделах и разработанные для решения некоторых задач распознавания, являются ничем

иным, как специфичными алгоритмами решения системы строгих линейных неравенств. Особенностью этих алгоритмов, определяющих их сферу применимости, является только то, что априори предполагается непротиворечивость решаемой системы неравенств. При таком взгляде на эти алгоритмы становится ясным, что они применимы не только для тех целей, для которых они первоначально были придуманы. Их сферой применимости являются любые задачи, которые сводятся к решению непротиворечивой системы строгих линейных неравенств. Чтобы привести примеры таких задач, нет нужды выходить за пределы распознавания образов: в рамках распознавания образов вполне достаточно таких задач. Одной из них является синтез *фишеровских классификаторов* [Fisher, 1936].

Пусть X , как и прежде, n -мерное линейное пространство, K - целое число и $\alpha_k, k = 1, \dots, K$, - это K векторов. Эти векторы определяют разделение пространства X на K выпуклых конусов $X_k, k = 1, \dots, K$, так, что точка $x \in X$ принадлежит множеству X_k , когда

$$\langle \alpha_k, x \rangle > \langle \alpha_j, x \rangle, \quad j = 1, 2, \dots, K, \quad j \neq k.$$

Указанное разделение пространства X на выпуклые конусы называется *фишеровским классификатором*.

Пусть \tilde{X} - конечное множество точек в пространстве X , которое разделено на K подмножеств $\tilde{X}_1, \tilde{X}_2, \dots, \tilde{X}_K$. В задаче, которую мы назовем *фишеровской задачей*, надо найти такие векторы $\alpha_1, \alpha_2, \dots, \alpha_K$, которые удовлетворяют неравенство

$$\langle \alpha_k, x \rangle > \langle \alpha_j, x \rangle \quad (5.58)$$

при любой тройке (x, k, j) , такой, что $x \in \tilde{X}_k, j \neq k$. Система (5.58) состоит из конечного количества неравенств. Это количество равно $|\tilde{X}|(K - 1)$. Наша цель состоит в решении системы (5.58), когда априори известно, что это решение существует. Очевидно, что задача простого разделения двух конечных множеств - это частный случай фишеровской задачи при $K = 2$. В этом случае простое разделение достигается вектором $\alpha_2 - \alpha_1$. Менее очевидно, но все же верно, что любая задача Фишера сводится к этому частному случаю. Покажем, как это происходит.

Пусть Y - линейное пространство размерности nK . Отобразим в это пространство множество \tilde{X} и множество совокупностей векторов вида $\alpha_1, \alpha_2, \dots, \alpha_K$. Разделим множество координат пространства Y на K подмножеств, каждое из которых состоит из n координат. Таким образом, мы можем говорить "первая n -ка координат", "вторая n -ка координат", " n -ка координат с номером k " и тому подобно.

Совокупность векторов $\alpha_1, \alpha_2, \dots, \alpha_K$ представим в (nK) -мерном пространстве Y вектором α , в котором k -ая n -ка координат равна координатам вектора α_k . Проще говоря, последовательность (nK) координат вектора α построена так, что вектора $\alpha_1, \alpha_2, \dots, \alpha_K$ записаны один за другим в одну последовательность.

Для каждого $x \in \tilde{X}$ построим множество $\tilde{Y}(x) \subset Y$, состоящее из $K - 1$ векторов. Это будет сделано следующим образом. Пусть k - порядковый

номер подмножества $\tilde{X}(k)$, которому принадлежит x . Пронумеруем векторы из множества $\tilde{Y}(x)$ номерами $j = 1, 2, \dots, K, j \neq k$. Символ $y(j, x)$ будет обозначать j -ый вектор из множества $\tilde{Y}(x)$. Он будет построен так, что его j -ая n -ка координат равна $-x$, k -ая n -ка равна x , а все остальные координаты равны нулю. Определим множество \tilde{Y} как

$$\tilde{Y} = \bigcup_{x \in \tilde{X}} \tilde{Y}(x).$$

Пусть k и j - два различных числа, а x - точка из множества \tilde{X}_k . В следствие того, каким образом были построены векторы α и множество \tilde{Y} , справедливо равенство $\langle \alpha_k, x \rangle - \langle \alpha_j, x \rangle = \langle \alpha, y(j, x) \rangle$, а следовательно, неравенство $\langle \alpha_k, x \rangle > \langle \alpha_j, x \rangle$ эквивалентно неравенству $\langle \alpha, y(j, x) \rangle > 0$. Система неравенств (5.58) становится эквивалентной системе

$$\langle \alpha, y \rangle > 0, \quad y \in \tilde{Y}. \quad (5.59)$$

Решение системы (5.59) можно найти, пользуясь перцептронным алгоритмом или алгоритмом Козинца. Решение будет получено в виде (nK) -мерного вектора α , который удовлетворяет условию (5.59), и представляет совокупность n -мерных векторов $\alpha_1, \alpha_2, \dots, \alpha_K$, которые удовлетворяют условиям (5.58).

Описанная процедура является объяснительной процедурой, а не указанием на непосредственное написание программы. При написании конкретной программы нет необходимости представлять в программе в явном виде множество \tilde{Y} , состоящее из (nK) -мерных векторов. Это множество должно учитываться программой, которая реализует модифицированный алгоритм решения системы (5.59). В этой модификации учитывается, что система (5.59) является вторичной, а получена она из исходной системы (5.58). Мы покажем, как проста модификация, скажем, алгоритма перцептрона для решения фишеровской задачи.

Пусть на t -ом шаге алгоритма получены векторы $\alpha_k^t, k = 1, \dots, K$. Для этих векторов следует проверить, существует ли точка x в множестве \tilde{X} , которая неправильно распознается с помощью этих векторов. Векторы $x \in \tilde{X}$ проверяются один за другим, и для каждого вычисляется число

$$b = \max_k \langle \alpha_k^t, x \rangle.$$

Затем проверяется, выполняется ли равенство $\langle \alpha_j^t, x \rangle = b$ для некоторого $j \neq k$. Напомним, что k - это номер подмножества \tilde{X}_k , которому принадлежит точка x . Если такая точка x найдена, это значит, что она неправильно распознана текущими векторами $\alpha_k^t, k = 1, \dots, K$. В этом случае векторы α_j и α_k должны измениться по правилам

$$\alpha_k^{t+1} = \alpha_k^t + x, \quad \alpha_j^{t+1} = \alpha_j^t - x.$$

Если такая точка не найдена, то текущие значения векторов являются решением фишеровской задачи.

Вряд ли можно придумать более простой алгоритм. Модификация алгоритма Козинца для решения фишеровской задачи также очень проста, хотя несколько сложнее.

5.4.5 Другие модификации алгоритмов Козинца

Пусть \tilde{X}_1 и \tilde{X}_2 - два конечных множества точек в линейном пространстве X , а \bar{X}_1 и \bar{X}_2 - их выпуклые оболочки, которые не пересекаются. Пусть вектор α и число θ определяют разделение пространства X на три подмножества $X^+(\alpha, \theta)$, $X^-(\alpha, \theta)$ и $X^0(\alpha, \theta)$ так, что

$$\begin{aligned} X^+(\alpha, \theta) &= \{x \in X \mid \langle \alpha, x \rangle > \theta\}, \\ X^-(\alpha, \theta) &= \{x \in X \mid \langle \alpha, x \rangle < \theta\}, \\ X^0(\alpha, \theta) &= \{x \in X \mid \langle \alpha, x \rangle = \theta\}. \end{aligned}$$

Задача простого разделения множеств \tilde{X}_1 и \tilde{X}_2 состоит в нахождении такого вектора α и такого числа θ , что $\tilde{X}_1 \subset X^+(\alpha, \theta)$ и $\tilde{X}_2 \subset X^-(\alpha, \theta)$, или, что то же самое, в решении системы строгих неравенств

$$\left. \begin{aligned} \langle \alpha, x \rangle &> \theta, & x \in \tilde{X}_1, \\ \langle \alpha, x \rangle &< \theta, & x \in \tilde{X}_2, \end{aligned} \right\} \quad (5.60)$$

относительно вектора α и числа θ при заданных множествах \tilde{X}_1 и \tilde{X}_2 .

Мы показывали раньше, как эту задачу можно свести к задаче

$$\begin{aligned} \langle \alpha, x \rangle &> 0, & x \in \tilde{X}_1, \\ \langle \alpha, x \rangle &< 0, & x \in \tilde{X}_2, \end{aligned}$$

а затем к задаче

$$\langle \alpha, x \rangle > 0, \quad x \in \tilde{X}.$$

Теперь мы рассмотрим задачи оптимального и ε -оптимального разделения в их исходной постановке и покажем, как алгоритмы Козинца решают эти задачи без их эквивалентного преобразования. Мы увидим, что такое прямое решение обладает определенными преимуществами.

Пусть α и θ - решение системы (5.60). Расстояние от точки $x \in \tilde{X}_1$ до гиперплоскости $X^0(\alpha, \theta)$ равно $(\langle \alpha, x \rangle - \theta)/|\alpha|$, а расстояние от точки $x \in \tilde{X}_2$ до гиперплоскости $X^0(\alpha, \theta)$ равно $(\theta - \langle \alpha, x \rangle)/|\alpha|$. Задача оптимального разделения множеств \tilde{X}_1 и \tilde{X}_2 определяется как нахождение такого решения системы (5.60), которое максимизирует число

$$f(\alpha, \theta) = \min \left(\min_{x \in \tilde{X}_1} \frac{\langle \alpha, x \rangle - \theta}{|\alpha|}, \min_{x \in \tilde{X}_2} \frac{\theta - \langle \alpha, x \rangle}{|\alpha|} \right).$$

Для $r^* = \max_{\alpha, \theta} f(\alpha, \theta)$ задача ε -оптимального разделения множеств \tilde{X}_1 и \tilde{X}_2 состоит в отыскании вектора α и числа θ , для которых

$$r^* - f(\alpha, \theta) \leq \varepsilon.$$

Давайте бегло рассмотрим основные соображения, которые приведут нас к решению этих задач. Ключевая идея состоит в том, что оптимальное разделение множеств \tilde{X}_1 и \tilde{X}_2 достигается гиперплоскостью, которая является серединным перпендикуляром к отрезку, соединяющему ближайшие друг к другу точки α_1^* и α_2^* из выпуклых оболочек \overline{X}_1 и \overline{X}_2 . Алгоритмы оптимального и ε -оптимального разделения основаны на последовательном сближении точек α_1 и α_2 при соблюдении условий $\alpha_1 \in \overline{X}_1$ и $\alpha_2 \in \overline{X}_2$. Сформулируем этот алгоритм, а затем кратко опишем основную идею его обоснования. Это обоснование несущественно отличается от обоснования алгоритма Козинца, приведенного в подразделе 5.4.3. Желаяющим узнать это обоснование мы рекомендуем получить его самостоятельно. После сказанного в этой лекции это уже не более, чем математическое упражнение.

Мы создадим последовательность точек $\alpha_1^1, \alpha_1^2, \dots, \alpha_1^t, \alpha_1^{t+1}, \dots$ и последовательность точек $\alpha_2^1, \alpha_2^2, \dots, \alpha_2^t, \alpha_2^{t+1}, \dots$ по следующим правилам.

Algorithm 5.4 Модификация алгоритма Козинца

1. Точка α_1^1 - это произвольно выбранная точка из \tilde{X}_1 , а точка α_2^1 - это произвольно выбранная точка из \tilde{X}_2 .
2. Предположим, что пара α_1^t и α_2^t уже построена. Для этой пары должна быть найдена либо точка $x^t \in \tilde{X}_1$, удовлетворяющая условиям

$$\left\langle x^t - \alpha_2^t, \frac{\alpha_1^t - \alpha_2^t}{|\alpha_1^t - \alpha_2^t|} \right\rangle \leq |\alpha_1^t - \alpha_2^t| - \frac{\varepsilon}{2}, \quad (5.61)$$

либо точка $x^t \in \tilde{X}_2$, удовлетворяющая условиям

$$\left\langle x^t - \alpha_1^t, \frac{\alpha_2^t - \alpha_1^t}{|\alpha_2^t - \alpha_1^t|} \right\rangle \leq |\alpha_2^t - \alpha_1^t| - \frac{\varepsilon}{2}. \quad (5.62)$$

3. Если таких точек нет, алгоритм выходит на останов с векторами α_1^t и α_2^t в качестве результата.
4. Если найден вектор $x^t \in \tilde{X}_1$, удовлетворяющий (5.61), вектор α_2 не меняется, то есть, $\alpha_2^{t+1} = \alpha_2^t$, а вектор α_1 меняется по правилу

$$\left. \begin{aligned} \alpha_1^{t+1} &= \alpha_1^t(1-k) + x^t \cdot k, \\ \text{где } k &= \min \left(1, \frac{\langle \alpha_1^t - \alpha_2^t, \alpha_1^t - x^t \rangle}{|\alpha_1^t - x^t|^2} \right) \end{aligned} \right\} \quad (5.63)$$

Это правило обозначает, что вектор α_1^{t+1} устанавливается в точку, ближайшую к точке α_2^t , и находящуюся на отрезке, соединяющем точки α_1^t и x^t .

5. Если найден вектор $x^t \in \tilde{X}_2$, удовлетворяющий (5.62), то $\alpha_1^{t+1} = \alpha_1^t$ и

$$\left. \begin{aligned} \alpha_2^{t+1} &= \alpha_2^t(1-k) + x^t \cdot k, \\ \text{где } k &= \min \left(1, \frac{\langle \alpha_2^t - \alpha_1^t, \alpha_2^t - x^t \rangle}{|\alpha_2^t - x^t|^2} \right) \end{aligned} \right\} \quad (5.64)$$

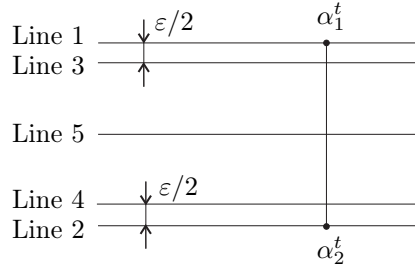


Figure 5.11 Geometrical interpretation of conditions as a relation of two points and five possible straight lines.

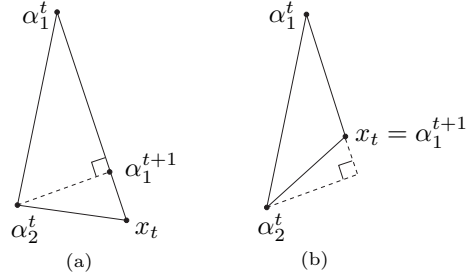


Figure 5.12 Two possible geometrical interpretations of the change of point α_1 positions.

В соответствии с этим правилом α_2^{t+1} есть точка на отрезке, соединяющем точки α_2^t и x^t , ближайшая к точке α_1^t .

В зависимости от величины ε в выражениях (5.61) и (5.62) приведенные правила (5.61)–(5.64) определяют три различных алгоритма.

1. Когда ε - положительная константа, это алгоритм ε -оптимального разделения множеств.
2. Когда ε - переменная, равная $\frac{1}{2} |\alpha_1^t - \alpha_2^t|$, это алгоритм простого разделения множеств.
3. Когда $\varepsilon = 0$, это алгоритм построения бесконечной последовательности пар векторов α_1^t и α_2^t , которая стремится к паре

$$(\alpha_1^*, \alpha_2^*) = \underset{(\alpha_1, \alpha_2) \in \bar{X}_1 \times \bar{X}_2}{\operatorname{argmin}} |\alpha_1 - \alpha_2| .$$

Приведенный алгоритм допускает следующую *геометрическую интерпретацию*. Условия (5.61) и (5.62) можно изобразить с помощью рис. 5.11, на котором показаны две точки α_1^t, α_2^t и пять прямых линий, перпендикулярных отрезку, который соединяет точки α_1^t и α_2^t . Линия 1 проходит через точку α_1^t . Линия 2 проходит через точку α_2^t . Линии 3 и 4 лежат между линиями 1 и 2 так, что линия 3 удалена от линии 1 на $\frac{1}{2}\varepsilon$, а линия 4 удалена от линии 2 на $\frac{1}{2}\varepsilon$. Наконец, линия 5 равноудалена от точек α_1^t и α_2^t .

Условия (5.61) и (5.62) имеют следующий геометрический смысл. Когда $\varepsilon = |\alpha_1^t - \alpha_2^t|$, условия (5.61) или (5.62) обозначают либо, что точка из X_1 оказалась ниже линии 5, либо, что точка из X_2 оказалась выше этой линии. В обоих этих случаях это значит, что какая-то точка распознана неправильно. В этом случае одна из точек α_1 или α_2 меняет свое положение. Если такой точки нет, это значит, что линия 5 разделяет множества \tilde{X}_1 и \tilde{X}_2 . Когда ε - положительная константа, условие (5.61) обозначает, что точка из \tilde{X}_1 оказалась ниже линии 3, а условие (5.62) обозначает, что точка из \tilde{X}_2 оказалась выше линии 4. В этом случае либо α_1 , либо α_2 меняет свое положение, после чего может достигаться правильное распознавание точек из множества $\tilde{X}_1 \cup \tilde{X}_2$. Если такой точки нет, то линия 5 ε -оптимально распознает точки из множества $\tilde{X}_1 \cup \tilde{X}_2$. И наконец, если никакая точка из

\tilde{X}_1 не оказывается ниже линии 1 и никакая точка из \tilde{X}_2 не оказывается выше линии 2, то линия 5 оптимально разделяет множества \tilde{X}_1 и \tilde{X}_2 .

Алгоритм пересчета точек α_1 и α_2 по правилам (5.63) и (5.64) также допускает наглядную геометрическую интерпретацию. На рис. 5.12 представлен случай, когда меняется вектор α_1 , а вектор α_2 не меняется. Точка α_1^{t+1} - это точка на отрезке, соединяющем точки α_1^t и x_t , которая наименее удалена от точки α_2^t . Это либо основание перпендикуляра, опущенного с точки α_2^t на прямую, проходящую через точки α_1^t и x_t (рис. 5.12a), либо точка x_t (рис. 5.12b). Первый случай имеет место, когда основание перпендикуляра оказывается внутри отрезка, а второй случай - когда вне отрезка.

Таким образом, в результате этих неформальных, но вполне понятных соображений мы убедились, что неподвижная точка алгоритма является решением одной из следующих трех задач в зависимости от величины ε .

1. При переменном значении ε , равном

$$\varepsilon = \frac{1}{2} |\alpha_1^t - \alpha_2^t| ,$$

неподвижная точка алгоритма есть решение простой задачи разделения множеств.

2. При постоянном значении ε неподвижная точка алгоритма есть ε -оптимальное решение задачи.
3. При $\varepsilon = 0$ неподвижная точка алгоритма есть решение задачи оптимального разделения точек.

Осталось рассмотреть вопрос, сходится ли описанный алгоритм к неподвижной точке. Доказать такую сходимость при $\varepsilon \neq 0$ можно подобно тому, как это доказывалось для алгоритма Козинца в подразделе 5.4.3. Основная идея доказательства состоит в том, что последовательность длин $|\alpha_1^t - \alpha_2^t|$ монотонно убывает, причем быстрее, чем геометрическая прогрессия с показателем

$$\frac{1}{\sqrt{1 + \varepsilon^2/D^2}} < 1 .$$

Величина D здесь уже не

$$\max_{x \in \tilde{X}} |x| ,$$

как это было в прежнем алгоритме из подраздела 5.4.3. Сейчас это величина

$$\max \left(\max_{x, y \in \tilde{X}_1} |x - y| , \max_{x, y \in \tilde{X}_2} |x - y| \right) ,$$

которая меньше, а иногда значительно меньше. Поэтому алгоритм Козинца в приведенной модификации сходится, как правило, быстрее.

5.5 Решение обобщенной задачи Андерсона

Понимание задачи линейного разделения конечных множеств точек позволяет нам вернуться опять к задачам Андерсона, которые представляются самыми трудными из рассмотренных в данной лекции. В заключительном

аккорде лекции мы покажем, что после незначительного видоизменения андерсоновых задач они сводятся к самой легкой из рассмотренных задач, а именно, к задаче простого разделения множеств. Однако в этом случае эти множества уже не окажутся конечными. Тем не менее задачи их разделения могут быть конструктивно решены.

5.5.1 ε -решение задач Андерсона

Напомним обобщенную задачу Андерсона, которую мы сформулировали в подразделе 5.3.1. Пусть $\{\mu^j, j \in J\}$ - конечное множество n -мерных векторов, а $\{\sigma^j, j \in J\}$ - конечное множество положительно определенных симметричных матриц размерностью $n \times n$. Множество J разделено на два подмножества J_1 и J_2 . Для индекса $j \in J_1$, вектора α и числа θ определено число $\text{er}(j, \alpha, \theta)$ - вероятность, что гауссов случайный вектор x с математическим ожиданием μ^j и ковариационной матрицей σ^j удовлетворит неравенство $\langle \alpha, x \rangle \leq \theta$. Обозначение 'er' вводится вместо прежнего обозначения ε , так как сейчас символ ε будет обозначать некоторую фиксированную малую вероятность. Для индекса $j \in J_2$ число $\text{er}(j, \alpha, \theta)$ есть вероятность неравенства $\langle \alpha, x \rangle \geq \theta$. Сформулированная ранее задача (см. (5.6)) состоит в отыскании

$$(\alpha^*, \theta^*) = \underset{\alpha, \theta}{\text{argmin}} \max_j \text{er}(j, \alpha, \theta). \quad (5.65)$$

Отстранимся сейчас от исключительно интересного математического содержания андерсоновых задач и посмотрим на них с чисто практической точки зрения. В подавляющем большинстве приложений основной интерес разработчика состоит в получении не оптимального, а хорошего решения. Если оптимальное решение обеспечивает вероятность ошибки, скажем, 30%, то задача совсем не решена с практической точки зрения, и оптимальность ее никаким образом не спасает. И наоборот, если найден алгоритм, дающий правильный ответ в 99.9% случаев, трудно представить, что его отвергнут только потому, что он неоптимален. Проще говоря, оптимальность и пригодность алгоритма - это совершенно разные понятия.

Эти неформальные, но разумные соображения заставляют нас заменить задачу (5.65) на поиск такого вектора α и числа θ , которые удовлетворяют неравенство

$$\max_j \text{er}(j, \alpha, \theta) < \varepsilon, \quad (5.66)$$

где ε - предельная вероятность ошибочного распознавания, допустимая в данном приложении. Дальнейший формальный анализ задачи (5.66) мы выполним при обычном предположении, что ее решение существует.

Пусть $E(r, \mu, \sigma)$ - эллипс, то есть замкнутое множество точек x , удовлетворяющих неравенство

$$\langle (\mu - x), \sigma^{-1} \cdot (\mu - x) \rangle \leq r^2.$$

Определим два множества

$$X_1(r) = \bigcup_{j \in J_1} E(r, \mu^j, \sigma^j) \quad \text{и} \quad X_2(r) = \bigcup_{j \in J_2} E(r, \mu^j, \sigma^j).$$

Задачу (5.66) можно свести к задаче простого разделения бесконечных множеств $X_1(r)$ и $X_2(r)$ при определенном значении r . Это доказывает следующая теорема.

Теорема 5.6 Об ε -решении обобщенной задачи Андерсона. Пусть для данного положительного числа $\varepsilon < 0.5$ число r есть решение уравнения

$$\varepsilon = \int_r^{\infty} \frac{1}{\sqrt{2\pi}} e^{-\frac{1}{2}x^2} dx .$$

Вектор α и число θ удовлетворяют требованию

$$\max_j er(j, \alpha, \theta) < \varepsilon \quad (5.67)$$

тогда и только тогда, когда α и θ удовлетворяют следующей бесконечной системе линейных неравенств

$$\left. \begin{aligned} \langle \alpha, x \rangle &> \theta, & x \in X_1(r), \\ \langle \alpha, x \rangle &< \theta, & x \in X_2(r). \end{aligned} \right\} \quad (5.68) \quad \blacktriangle$$

Доказательство. Для доказательства теоремы нам потребуются явные выражения для минимального и максимального значения скалярного произведения

$$f(x) = \langle \alpha, x \rangle \quad (5.69)$$

при условии

$$F(x) = \langle \mu - x, \sigma^{-1} \cdot (\mu - x) \rangle \leq r^2 . \quad (5.70)$$

Искомые точки экстремумов являются решением уравнений

$$\text{grad} (f(x) + \lambda \cdot F(x)) = 0 \quad (5.71)$$

при некотором значении коэффициентов λ . На основании (5.71) можно написать

$$\alpha + \lambda \cdot \sigma^{-1} \cdot (\mu - x) = 0 ,$$

из чего следует

$$x = \mu + \frac{1}{\lambda} \sigma \cdot \alpha .$$

Очевидно, что экстремум линейной функции на выпуклом множестве (5.70) достигается на его границе, то есть, когда $F(x) = r^2$. Подставим полученное значение для x в уравнение $F(x) = r^2$ и получим

$$\begin{aligned} &\left\langle \left(\mu - \mu - \frac{1}{\lambda} \sigma \cdot \alpha \right), \sigma^{-1} \cdot \left(\mu - \mu - \frac{1}{\lambda} \sigma \cdot \alpha \right) \right\rangle \\ &= \frac{1}{\lambda^2} \langle (\sigma \cdot \alpha), \sigma^{-1} \cdot (\sigma \cdot \alpha) \rangle \\ &= \frac{1}{\lambda^2} \langle (\sigma \cdot \alpha), (\sigma^{-1} \cdot \sigma) \cdot \alpha \rangle = \frac{1}{\lambda^2} \langle \alpha, \sigma \cdot \alpha \rangle = r^2 . \end{aligned}$$

Отсюда следует, что в случае, когда ищется минимум,

$$\lambda = \frac{\sqrt{\langle \alpha, \sigma \cdot \alpha \rangle}}{r},$$

так как число r положительно при $\varepsilon < 0.5$. Для случая, когда ищется максимум,

$$\lambda = -\frac{\sqrt{\langle \alpha, \sigma \cdot \alpha \rangle}}{r}.$$

Искомые точки экстремума - это

$$\left. \begin{aligned} x_{\min} &= \mu - \frac{r}{\sqrt{\langle \alpha, \sigma \cdot \alpha \rangle}} \cdot \sigma \cdot \alpha, \\ x_{\max} &= \mu + \frac{r}{\sqrt{\langle \alpha, \sigma \cdot \alpha \rangle}} \cdot \sigma \cdot \alpha. \end{aligned} \right\} \quad (5.72)$$

Экстремальные значения функции (5.69) при условии (5.70) равны

$$\left. \begin{aligned} \min f(x) &= \langle \alpha, \mu \rangle - r \cdot \sqrt{\langle \alpha, \sigma \cdot \alpha \rangle}, \\ \max f(x) &= \langle \alpha, \mu \rangle + r \cdot \sqrt{\langle \alpha, \sigma \cdot \alpha \rangle}. \end{aligned} \right\}$$

Докажем, что любое решение системы (5.68) удовлетворяет условию (5.67). Поскольку выполняется условие (5.68), выполняется и неравенство $\langle \alpha, x \rangle > \theta$ для любой точки эллипса $E(r, \mu^j, \sigma^j)$, $j \in J_1$. Поскольку эллипс - это замкнутое множество, множество неравенств

$$\langle \alpha, x \rangle > \theta, \quad x \in E(r, \mu^j, \sigma^j), \quad j \in J_1,$$

можно записать в другом виде,

$$\min_{x \in E(r, \mu^j, \sigma^j)} \langle \alpha, x \rangle > \theta, \quad j \in J_1,$$

или, с учетом уже выведенного выражения (5.72),

$$\langle \alpha, \mu^j \rangle - r \cdot \sqrt{\langle \alpha, \sigma^j \cdot \alpha \rangle} > \theta, \quad j \in J_1,$$

и

$$\frac{\langle \alpha, \mu^j \rangle - \theta}{\sqrt{\langle \alpha, \sigma^j \cdot \alpha \rangle}} > r, \quad j \in J_1. \quad (5.73)$$

Подобным образом, в силу второго неравенства в (5.68) имеем

$$\langle \alpha, x \rangle < \theta, \quad x \in E(r, \mu^j, \sigma^j), \quad j \in J_2,$$

$$\max_{x \in E(r, \mu^j, \sigma^j)} \langle \alpha, x \rangle < \theta, \quad j \in J_2,$$

$$\langle \alpha, \mu^j \rangle + r \cdot \sqrt{\langle \alpha, \sigma^j \cdot \alpha \rangle} < \theta, \quad j \in J_2,$$

$$\frac{\theta - \langle \alpha, \mu^j \rangle}{\sqrt{\langle \alpha, \sigma^j \cdot \alpha \rangle}} > r, \quad j \in J_2. \quad (5.74)$$

Числитель в левой части (5.73) есть математическое ожидание случайного числа $\langle \alpha, x \rangle - \theta$, а знаменатель - среднеквадратичный разброс этого же числа для случайного вектора x с математическим ожиданием μ^j и ковариационной матрицей σ^j . Неравенство (5.73) утверждает, что вероятность отрицательного значения числа $\langle \alpha, x \rangle - \theta$ не превосходит

$$\int_r^\infty \frac{1}{\sqrt{2\pi r}} e^{-\frac{1}{2}x^2} dx.$$

Это значит, что $\text{er}(j, \alpha, \theta) < \varepsilon$ для всех $j \in J_1$. Аналогично, из неравенства (5.74) следует, что $\text{er}(j, \alpha, \theta) < \varepsilon$ для всех $j \in J_2$. Таким образом, мы доказали, что любое решение системы (5.68) удовлетворяет и условию (5.67).

Докажем сейчас, что если пара (α, θ) не удовлетворяет системе (5.68), то она не удовлетворяет и неравенство (5.67). Предположим, что не выполняется первое неравенство в (5.68). Тогда существуют такие $j \in J_1$ и $x \in E(r, \mu^j, \sigma^j)$, что $\langle \alpha, x \rangle \leq \theta$. Отсюда немедленно следует, что

$$\min_{x \in E(j, \mu^j, \sigma^j)} \langle \alpha, x \rangle \leq \theta, \quad \langle \alpha, \mu^j \rangle - r \sqrt{\langle \alpha, \sigma^j \cdot \alpha \rangle} \leq \theta, \quad \frac{\langle \alpha, \mu^j \rangle - \theta}{\sqrt{\langle \alpha, \sigma^j \cdot \alpha \rangle}} \leq r$$

и выполняется $\text{er}(j, \alpha, \theta) \geq \varepsilon$. Аналогично, если не выполняется второе неравенство в системе (5.68), то для некоторого $j \in J_2$ выполняется $\text{er}(j, \alpha, \theta) \geq \varepsilon$. ■

В силу теоремы 5.6 ε -решение андерсоновой задачи свелось к поиску числа θ и вектора α , которые удовлетворяют бесконечной системе неравенств

$$\left. \begin{aligned} \langle \alpha, x \rangle > \theta, \quad x \in \{x \mid \langle \mu^j - x, (\sigma^j)^{-1} \cdot (\mu^j - x) \rangle \leq r^2\}, \quad j \in J_1, \\ \langle \alpha, x \rangle < \theta, \quad x \in \{x \mid \langle \mu^j - x, (\sigma^j)^{-1} \cdot (\mu^j - x) \rangle \leq r^2\}, \quad j \in J_2, \end{aligned} \right\} \quad (5.75)$$

при определенном значении r , зависящем от ε .

Такое родство задач Андерсона и уже исследованных задач простого разделения множеств очень благоприятно, так как позволяет увидеть в задачах Андерсона простоту, которая раньше была скрыта. Однако с этим упрощением возникают и некоторые опасения. Во первых, система (5.75) состоит из бесконечного количества неравенств, а алгоритмы персептрона и Козинца сформулированы и исследованы нами только для систем, состоящих из конечного количества неравенств. Во вторых, в выражениях для системы (5.75) присутствуют обратные ковариационные матрицы, а именно этой операции мы с самого начала хотели избежать. Вопреки этим опасениям мы продолжим анализ задачи (5.75) и увидим, что алгоритмы Козинца или персептрона можно применять и для решения бесконечной системы (5.75) неравенств и при этом избежать операции обращения матриц.

5.5.2 Линейное разделение бесконечных множеств точек

Для того, чтобы применять известные нам алгоритмы простого линейного разделения множеств \tilde{X}_1 и \tilde{X}_2 , нет необходимости проверять все точки множества $\tilde{X}_1 \cup \tilde{X}_2$, правильно ли они классифицируются текущими значениями α^t и θ^t , а достаточно найти хотя бы одну точку $x \in \tilde{X}_1 \cup \tilde{X}_2$, которая классифицируется неправильно. При определенном способе задания множеств X_1 и X_2 такую ошибочно распознаваемую точку можно найти и в случае, когда эти множества бесконечные, то есть в случае, когда требуется проверить бесконечную систему неравенств. В нашем случае это система (5.75).

При обосновании алгоритма Козинца (для определенности будем говорить об алгоритме из подраздела 5.4.4) никаким образом не использовался тот факт, что множества \tilde{X}_1 и \tilde{X}_2 конечные. Требовалось только, чтобы "диаметры" этих множеств

$$\max_{x,y \in \tilde{X}_1} |x - y| \quad \text{и} \quad \max_{x,y \in \tilde{X}_2} |x - y|$$

были конечные, а их выпуклые оболочки не пересекались. Для наших множеств \tilde{X}_1 и \tilde{X}_2 эти требования выполняются.

Для множеств \tilde{X}_1 и \tilde{X}_2 выполняется

$$\begin{aligned} \tilde{X}_1 &= \bigcup_{j \in J_1} \{x \mid \langle \mu^j - x, (\sigma^j)^{-1} \cdot (\mu^j - x) \rangle \leq r^2\}, \\ \tilde{X}_2 &= \bigcup_{j \in J_2} \{x \mid \langle \mu^j - x, (\sigma^j)^{-1} \cdot (\mu^j - x) \rangle \leq r^2\}. \end{aligned}$$

Поскольку матрицы σ в выражениях (5.75) положительно определенные, множества \tilde{X}_1 и \tilde{X}_2 ограничены. Эти множества не пересекаются по крайней мере при некоторых значениях параметра ε .

Рассмотрим, как следует находить неравенство в системе (5.73), которое не выполняется для текущих значений α и θ . Это значит, что следует найти индекс $j \in J$ и вектор x , для которого выполняются неравенства $\langle x - \mu^j, (\sigma^j)^{-1} \cdot (x - \mu^j) \rangle \leq r^2$ и $\langle \alpha, x \rangle \leq \theta$ при $j \in J_1$, или вектор x , для которого выполняются неравенства $\langle x - \mu^j, (\sigma^j)^{-1} \cdot (x - \mu^j) \rangle \leq r^2$ и $\langle \alpha, x \rangle \geq \theta$ при $j \in J_2$. На основании рассуждений, выполненных при доказательстве теоремы 5.6, мы утверждаем, что первое условие эквивалентно утверждению, что для некоторого $j \in J_1$ минимальное значение скалярного произведения $\langle \alpha, x \rangle$ на множестве $\{x \mid \langle (x - \mu^j), (\sigma^j)^{-1} \cdot (x - \mu^j) \rangle \leq r^2\}$ не превосходит θ . Второе условие эквивалентно утверждению, что для некоторого $j \in J_2$ максимальное значение скалярного произведения $\langle \alpha, x \rangle$ на соответствующем множестве не меньше, чем θ . Таким образом, текущие значения α и θ не удовлетворяют бесконечной системе (5.75) неравенств тогда и только тогда, когда

$$\left(\exists j \in J_1 \mid \min_{x \in E(j, \mu^j, \sigma^j)} \langle \alpha, x \rangle \leq \theta \right) \vee \left(\exists j \in J_2 \mid \max_{x \in E(j, \mu^j, \sigma^j)} \langle \alpha, x \rangle \geq \theta \right),$$

где символ \vee обозначает дизъюнкцию двух утверждений.

На основании (5.73) и (5.74) последнее утверждение можно представить в виде: пара (α, θ) не удовлетворяет бесконечной системе неравенств (5.75) тогда и только тогда, когда она удовлетворяет некоторому из неравенств

$$\begin{aligned} \frac{\langle \alpha, \mu^j \rangle - \theta}{\sqrt{\langle \alpha, \sigma^j \cdot \alpha \rangle}} &\leq r, \quad j \in J_1, \\ \frac{\theta - \langle \alpha, \mu^j \rangle}{\sqrt{\langle \alpha, \sigma^j \cdot \alpha \rangle}} &\leq r, \quad j \in J_2. \end{aligned} \quad (5.76)$$

Хотя эти неравенства уже нелинейны, но количество их конечно, и каждое из них легко может быть проверено, так для этого не требуется инвертировать матрицу σ^j .

Предположим, что выполняется одно из неравенств в системе (5.76). Пусть это будет для определенности неравенство из первой строчки в (5.76). Тогда можно найти и то линейное неравенство в бесконечной системе (5.75), которое не выполняется, и указать точку x , соответствующую этому неравенству,

$$x = \mu^j - \frac{r}{\sqrt{\langle \alpha, \sigma^j \cdot \alpha \rangle}} \cdot \sigma^j \cdot \alpha. \quad (5.77)$$

Если пара (α, θ) удовлетворяет некоторое неравенство из второй строчки системы (5.76) для $j \in J_2$, то неравенство из системы (5.75), которое не выполняется, соответствует точке

$$x = \mu^j + \frac{r}{\sqrt{\langle \alpha, \sigma^j \cdot \alpha \rangle}} \cdot \sigma^j \cdot \alpha. \quad (5.78)$$

Выражения (5.77) и (5.78) легко вычисляются, так как для этого не требуется обращать матрицу σ^j . Мы видим, что справедливость бесконечной системы (5.75) неравенств легко проверяется, и в случае, когда эта система не выполняется, легко найти именно то неравенство, которое не выполняется. Это позволяет нам написать следующий алгоритм поиска вектора α и порога θ , которые являются решением системы (5.75). Как и прежде, приведенный ниже алгоритм находит решение этой системы только в случае, если система непротиворечива.

Algorithm 5.5 ε -решение обобщенной задачи Андерсона

1. Для данного числа ε следует найти число r - решение уравнения

$$\varepsilon = \int_r^\infty \frac{1}{\sqrt{2\pi}} e^{-\frac{1}{2}x^2} dx.$$

2. Алгоритм создает две последовательности векторов $\alpha_1^1, \alpha_1^2, \dots, \alpha_1^t, \dots$ и $\alpha_2^1, \alpha_2^2, \dots, \alpha_2^t, \dots$. Этим векторам соответствует вектор α^t и число θ^t в соответствии с формулами

$$\alpha^t = \alpha_1^t - \alpha_2^t, \quad \theta^t = \frac{1}{2} (|\alpha_1^t|^2 - |\alpha_2^t|^2).$$

Вектор α_1^1 - это произвольно выбранный вектор из множества \tilde{X}_1 , например, один из векторов μ^j , $j \in J_1$. Вектор α_2^1 - это произвольно выбранный вектор из множества \tilde{X}_2 , например, один из векторов μ^j , $j \in J_2$.

3. Предположим, что уже построены векторы α_1^t и α_2^t . По ним строится вектор α^t и число θ^t и проверяются неравенства

$$\frac{\langle \alpha^t, \mu^j \rangle - \theta^t}{\sqrt{\langle \alpha^t, \sigma^j \cdot \alpha^t \rangle}} > r, \quad j \in J_1, \quad (5.79)$$

равно как и неравенства

$$\frac{\theta^t - \langle \alpha^t, \mu^j \rangle}{\sqrt{\langle \alpha^t, \sigma^j \cdot \alpha^t \rangle}} > r, \quad j \in J_2. \quad (5.80)$$

4. Если все эти неравенства выполняются, то вектор α^t и число θ^t являются ε -решением задачи.
 5. Если для некоторого $j \in J_1$ не выполняется неравенство (5.79), то вычисляется вектор

$$x^t = \mu^j - \frac{r}{\sqrt{\langle \alpha^t, \sigma^j \cdot \alpha^t \rangle}} \cdot \sigma^j \cdot \alpha^t$$

и новый вектор α_1^{t+1} по формуле

$$\alpha_1^{t+1} = \alpha_1^t \cdot (1 - k) + x^t \cdot k,$$

где

$$k = \min \left(1, \frac{\langle \alpha_1^t - \alpha_2^t, \alpha_1^t - x^t \rangle}{|\alpha_1^t - x^t|^2} \right).$$

Вектор α_2 в этом случае не меняется, то есть $\alpha_2^{t+1} = \alpha_2^t$.

6. Если для некоторого $j \in J_2$ не выполняется неравенство (5.80), то вычисляется вектор

$$x^t = \mu^j + \frac{r}{\sqrt{\langle \alpha^t, \sigma^j \cdot \alpha^t \rangle}} \cdot \sigma^j \cdot \alpha^t$$

и новый вектор α_2^{t+1} по формуле

$$\alpha_2^{t+1} = \alpha_2^t \cdot (1 - k) + x^t \cdot k,$$

где

$$k = \min \left(1, \frac{\langle \alpha_2^t - \alpha_1^t, \alpha_2^t - x^t \rangle}{|\alpha_2^t - x^t|^2} \right).$$

Вектор α_1 в этом случае не меняется, то есть $\alpha_1^{t+1} = \alpha_1^t$.

Если задача Андерсона имеет ε -решение, алгоритм гарантированно приходит в состояние, при котором выполняются как условия (5.79), так и условия (5.80), и заканчивает свою работу.

5.6 Обсуждение

Я заметил определенное несоответствие в лекции, которое разрушает значительную ее часть. С одной стороны, вы неоднократно указываете, что ковариационные матрицы σ^j положительно определенные. Это предположение активно используется при некоторых доказательствах. С другой стороны, с самого начала вводится трюк, с помощью которого отыскание гиперплоскости общего вида сводится к отысканию гиперплоскости, проходящей через начало координат. Это достигается введением дополнительной координаты, принимающей постоянное значение и, следовательно, имеющей нулевую дисперсию. Уже по одной этой причине невозможно говорить о положительной определенности ковариационной матрицы, а можно говорить только о ее неотрицательной определенности.

Я ожидал, что это противоречие как-то будет развязано в лекции, но так и не дождался. Сейчас я уверен, что это промах. Возможно, это какой-то педагогический прием с вашей стороны, имеющий целью проверить, насколько внимательно я изучаю ваши лекции. В таком случае вы делаете это напрасно. Я изучаю лекции достаточно тщательно, хотя и не всегда это дается легко.

Время от времени мы применяем подобные педагогические трюки, но не сейчас. Несответствие, о котором Ты говоришь, появилось не в результате недосмотра, но и не намеренно. Это действительно несоответствие, но оно не разрушает конечные результаты. Взгляни еще раз на все места, где анализируются андерсоновы задачи, и Ты увидишь, что везде, где предполагается положительная определенность матриц, можно было бы обойтись и без нее. Нужно было бы только везде оговориться, как следует вводить то или иное понятие для случая вырожденных ковариационных матриц. Это бы не было трудно, но это бы прерывало связность в изложении, делало бы его менее стремительным. Мы благодарим Тебя за это замечание и просмотрим сейчас вместе все те понятия, при определении которых мы упоминали о невырожденности ковариационных матриц.

С самого начала это нужно при вычислении значений

$$f^j(\alpha) = \frac{\langle \mu^j, \alpha \rangle}{\sqrt{\langle \alpha, \sigma^j \cdot \alpha \rangle}}, \quad j \in J, \quad (5.81)$$

затем при вычислении точек касания

$$x_0^j(\alpha) = \mu^j - \frac{\langle \mu^j, \alpha \rangle}{\langle \alpha, \sigma^j \cdot \alpha \rangle} \cdot (\sigma^j \cdot \alpha), \quad (5.82)$$

которые должны вычисляться только для тех j , для которых

$$f^j(\alpha) = \min_{j \in J} f^j(\alpha).$$

В алгоритмах ϵ -решения андерсоновых задач ковариационные матрицы опять используются при вычислении точек

$$x^j = \mu^j - \frac{r}{\sqrt{\langle \alpha, \sigma^j \cdot \alpha \rangle}} \cdot (\sigma^j \cdot \alpha), \quad (5.83)$$

в которых достигается экстремум скалярного произведения $\langle \alpha, x \rangle$ на множестве точек x , удовлетворяющих неравенство

$$\langle (x - \mu^j), (\sigma^j)^{-1} \cdot (x - \mu^j) \rangle \leq r^2. \quad (5.84)$$

В алгоритмах используются только формулы (5.81), (5.82) и (5.83). По формуле (5.84) не производятся никакие вычисления. Она служит лишь для обоснования равенства (5.83).

Формально говоря, формулы (5.81), (5.82) и (5.83) определены только для положительно определенных матриц σ^j , $j \in J$. Матрицы определяют квадратичные функции $\langle \alpha, \sigma^j \cdot \alpha \rangle$, значения которых присутствуют в знаменателях, и поэтому числа $\langle \alpha, \sigma^j \cdot \alpha \rangle$ должны быть строго положительными при любых значениях $\alpha \neq 0$. Однако, судя по тому, как числа $f^j(\alpha)$ используются в алгоритме (см. формулу (5.81)), и судя по смыслу векторов x^j (см. формулу (5.83)), алгоритм можно доопределить и для случая нулевых значений квадратичной функции $\langle \alpha, \sigma^j \cdot \alpha \rangle$. Значения $f^j(\alpha)$ вычисляются только для того, чтобы затем найти самое меньшее из них. Значение $f^j(\alpha)$ для $\langle \alpha, \sigma^j \cdot \alpha \rangle = 0$ можно понимать, как достаточно большое число, которое больше любых других чисел $f^j(\alpha)$, вычисленных для тех j , для которых $\langle \alpha, \sigma^j \cdot \alpha \rangle \neq 0$. Точки касания x_0^j , таким образом, должны вычисляться только для тех индексов j , для которых $\langle \alpha, \sigma^j \cdot \alpha \rangle \neq 0$. Если таких точек нет, то это значит, что достигнута нулевая вероятность ошибочного распознавания j -ой гауссовой случайной величины при любом j . В этом случае алгоритм может прекратить работу, так как улучшить текущее решение уже невозможно. Мы видим, что при таком доопределении алгоритма в его область определения входят и ситуации с вырожденными матрицами σ^j .

Посмотрим теперь, как следует вычислять точки касания x^j , когда $\langle \alpha, \sigma^j \cdot \alpha \rangle = 0$, и формула (5.83) неприменима. Формально говоря, формула (5.84) определяет эллипс только для положительно определенных матриц σ^j , то есть матриц с ненулевыми собственными числами. Только тогда матрицы σ^j можно инвертировать. Однако само понятие эллипса можно определить и для случая, когда некоторые собственные числа равны нулю. В этом случае эллипс следует определять не с помощью (5.84), а как-то по другому. Для любого размера r это будет множество в гиперплоскости, размерность которой будет равна количеству ненулевых собственных чисел. Во всех случаях, независимо от того, можно ли инвертировать матрицу σ^j или нет, точка x^j должна вычисляться по формуле (5.83). Исключением является ситуация, когда число $\langle \alpha, \sigma^j \cdot \alpha \rangle$ равно нулю. Тогда скалярное произведение с вектором α одно и то же для любой точки эллипса, потому что эллипс целиком лежит в гиперплоскости, параллельной плоскости $\langle \alpha, x \rangle = 0$. В этом случае в качестве точки x^j можно взять любую точку эллипса. Проще всего взять точку μ^j .

Как видишь, алгоритм можно доопределить и обосновать и для случая неотрицательно определенных матриц. Сделать это не очень трудно, но трудоемко. Если хочешь, можешь это сделать, но здесь вряд ли можно

найти что-то существенное с точки зрения теории распознавания. В лучшем случае Ты поупражняешься в линейной алгебре и теории матриц.

В лекции вы свели обобщенную задачу Андерсона к задаче негладкой оптимизации и довольно бегло упомянули, что в современной вычислительной математике уже достаточно глубоко исследованы методы оптимизации недифференцируемых функций. Я полностью согласен, что эти методы не являются общеизвестными в среде распознавания. Я бы хотел хотя бы немного узнать об основных понятиях теории негладкой оптимизации. Более конкретно, как бы выглядела эта лекция, если бы она базировалась на этих понятиях?

Как и во всех хорошо разработанных теориях, основные идеи негладкой оптимизации можно изложить довольно лаконично. Но эти идеи открывают поле для очень глубоких размышлений.

Пусть X - конечномерное линейное пространство, на котором задана вогнутая функция $f: X \rightarrow \mathbb{R}$. Пусть x_0 - некоторая точка этого пространства, а $g(x_0)$ - такой вектор, что для любой точки $x \in X$ число

$$f(x_0) + \langle g(x_0), (x - x_0) \rangle$$

не меньше, чем $f(x)$,

$$f(x_0) + \langle g(x_0), (x - x_0) \rangle \geq f(x) \quad (5.85)$$

или, что то же самое,

$$\langle g(x_0), (x - x_0) \rangle \geq f(x) - f(x_0). \quad (5.86)$$

Вектор $g(x_0)$ называется *обобщенный градиент* функции f в точке x_0 .

Понятие обобщенного градиента важно по следующим причинам:

1. Если функция f в точке x_0 дифференцируема, градиент функции f удовлетворяет условиям (5.85) и (5.86). Следовательно, градиент, понимаемый в обычном смысле этого слова, есть обобщенный градиент.
2. Определение обобщенного градиента никаким образом не основано на понятиях частных производных, и следовательно, имеет смысл и для недифференцируемых функций.
3. Даже если это и неочевидно на первый взгляд, позже на неформальном уровне мы поймем, что для вогнутой функции обобщенный градиент существует (возможно, не единственный) во всех точках ее области определения.
4. И наконец, в нашем контексте самое главное. Многие известные градиентные методы оптимизации, основанные на движении по градиенту, могут быть обоснованы и в том случае, если движение осуществляется по обобщенному градиенту. Это утверждает следующая базовая теорема негладкой оптимизации.

Теорема 5.7 Оптимизация по обобщенным градиентам. Пусть $f: X \rightarrow \mathbb{R}$ - вогнутая функция, а $\gamma_i, i = 1, 2, \dots, \infty$, - расходящийся ряд положительных чисел, стремящихся к нулю; пусть f^* - это

$$\max_{x \in X} f(x),$$

а X_0 - множество точек $x \in X$, для которых $f(x) = f^*$; пусть x_0 - любая точка в пространстве X , и

$$x_i = x_{i-1} + \gamma_i \cdot \frac{g(x_i)}{|g(x_i)|},$$

где $g(x_i)$ - обобщенный градиент функции f в точке x_i .
В таком случае

$$\lim_{i \rightarrow \infty} f(x_i) = f^*,$$

$$\lim_{i \rightarrow \infty} \min_{x \in X_0} |x_i - x| = 0. \quad \blacktriangle$$

Доказательство.

Доказательство этой базовой теоремы содержится в работе Н.З.Шора [Shor, 1979; Shor, 1998]. ■

Попробуем теперь на неформальном уровне понять свойства обобщенного градиента. Представим себе вогнутую функцию $y = f(x)$, определенную на n -мерном пространстве X и представим ее график в $(n + 1)$ -мерном пространстве. В этом $(n + 1)$ -мерном пространстве n координат совпадают с координатами исходного пространства X , а $(n + 1)$ -ая координата соответствует значению y . Функция f изображается в этом пространстве, как подмножество, состоящее из всех тех точек (x, y) , для которых $y = f(x)$. Назовем это подмножество графиком функции f и обозначим его $D, D = \{(x, y) \mid y = f(x)\}$. Рассмотрим множество точек (y, x) , которые лежат "под графиком" функции, назовем его телом функции f и обозначим его T . Более точно, $T = \{(x, y) \mid y \leq f(x)\}$. Поскольку функция f вогнута, ее тело T выпукло.

Подобно тому, как функция f изображается графиком D , любая линейная функция изображается гиперплоскостью. И наоборот, любая гиперплоскость, не ортогональная подпространству X , является графиком некоторой линейной функции. Представим себе гиперплоскость, проходящую через точку $(x_0, f(x_0))$ на графике D так, что тело T полностью находится под гиперплоскостью. Так как тело T выпукло, такую плоскость можно провести для каждой точки на графике D , то есть на поверхности тела T . Точная формулировка этого свойства является содержанием знаменитой леммы Фаркаша. Построенную таким образом гиперплоскость назовем касательной к телу T в точке x_0 , которая лежит на графике D .

Из приведенных определений следует, что обобщенный градиент функции f в точке x_0 - это градиент линейной функции, графиком которой служит касательная гиперплоскость к телу функции в точке $(x_0, f(x_0))$. В некоторых точках графика возможна единственная касательная. Тогда обобщенный градиент совпадает с градиентом в обычном смысле этого слова. В других точках графика таких касательных может быть несколько.

Неформально говоря, такая неоднозначность касательной имеет место в точках, где график функции "изломан", "не гладкий". Если в какой-то точке x_0 возможны две различные касательные гиперплоскости, то и любая промежуточная гиперплоскость тоже касательная. Более точно это выражает следующая теорема.

Теорема 5.8 О промежуточных касательных. *Множество векторов, которые являются обобщенными градиентами функции f в точке x_0 , выпукло.* ▲

Доказательство. Пусть g_1 и g_2 - обобщенные градиенты в точке x_0 . Следовательно, для любой точки $x \in X$ выполняется

$$\langle g_1, x - x_0 \rangle \geq f(x) - f(x_0),$$

$$\langle g_2, x - x_0 \rangle \geq f(x) - f(x_0).$$

Из этих неравенств следует, что для любых положительных чисел α_1 и $\alpha_2 = 1 - \alpha_1$ выполняется

$$\langle \alpha_1 \cdot g_1 + \alpha_2 \cdot g_2, x - x_0 \rangle \geq f(x) - f(x_0).$$

Это значит, что выпуклая линейная комбинация $\alpha_1 \cdot g_1 + \alpha_2 \cdot g_2$ тоже является обобщенным градиентом функции f в точке x_0 . ■

При неформальном понимании обобщенного градиента, как векторного параметра касательной гиперплоскости, становится почти очевидной следующая базовая теорема о необходимых и достаточных условиях максимума вогнутой функции.

Теорема 5.9 Необходимые и достаточные условия максимума. *В точке x_0 достигается максимум вогнутой функции тогда и только тогда, когда нулевой вектор является обобщенным градиентом функции в точке x_0 .* ▲

Доказательство. Вместо доказательства теоремы дадим ее неформальную интерпретацию. Теорема утверждает, что точка (y, x_0) является самой "высокой" точкой на теле T функции тогда и только тогда, когда "горизонтальная" гиперплоскость является касательной в этой точке. Точное доказательство этой очевидной теоремы тоже очень краткое. ■

Используем теперь сформулированные понятия для анализа нашей задачи.

Теорема 5.10 Об обобщенном градиенте для минимальной функции.

Пусть $\{f^j(x), j \in J\}$ - совокупность вогнутых дифференцируемых функций, x_0 - некоторая точка, а $g^j(x_0), j \in J$, - градиенты функций $f^j(x)$ в точке x_0 .

Пусть J_0 - множество индексов j , для которых

$$f^j(x_0) = \min_{i \in J} f^i(x_0), \tag{5.87}$$

а $\gamma^j, j \in J_0$, - неотрицательные коэффициенты, сумма которых равна 1. Тогда выпуклая комбинация $g_0 = \sum_{j \in J_0} \gamma^j \cdot g^j(x_0)$ является градиентом функции

$$f(x) = \min_{j \in J} f^j(x)$$

в точке x_0 . И наоборот, если вектор g_0 является обобщенным градиентом функции f в точке x_0 , то вектор g_0 есть выпуклая линейная комбинация векторов $g^j(x_0), j \in J_0$. \blacktriangle

Доказательство. Докажем первое утверждение теоремы. Поскольку векторы $g^j(x_0), j \in J_0$, являются градиентами функций f^j , справедливы неравенства

$$\langle g^j(x_0), x - x_0 \rangle \geq f^j(x) - f^j(x_0), \quad j \in J_0,$$

или, что то же самое,

$$\langle g^j(x_0), x - x_0 \rangle \geq f^j(x) - \min_{j \in J} f^j(x_0), \quad j \in J_0.$$

Отсюда следует

$$\sum_{j \in J_0} \gamma^j \cdot \langle g^j(x_0), x - x_0 \rangle \geq \sum_{j \in J_0} \gamma^j \cdot f^j(x) - \min_{j \in J} f^j(x_0). \quad (5.88)$$

Из неравенства

$$\sum_{j \in J_0} \gamma^j \cdot f^j(x) \geq \min_{j \in J_0} f^j(x) = \min_{j \in J} f^j(x)$$

с учетом (5.88) следует

$$\begin{aligned} \left\langle \sum_{j \in J_0} \gamma^j \cdot g^j(x_0), x - x_0 \right\rangle &= \sum_{j \in J_0} \gamma^j \cdot \langle g^j(x_0), x - x_0 \rangle \\ &\geq \min_{j \in J} f^j(x) - \min_{j \in J} f^j(x_0) = f(x) - f(x_0) \end{aligned}$$

и

$$\left\langle \sum_{j \in J_0} \gamma^j \cdot g^j(x_0), x - x_0 \right\rangle \geq f(x) - f(x_0),$$

что и доказывает первое утверждение теоремы 5.10.

Докажем теперь, что если вектор g_0 не является выпуклой линейной комбинацией векторов $g^j(x_0), j \in J_0$, то вектор g_0 не является обобщенным градиентом. Поскольку вектор g_0 не принадлежит выпуклому многограннику, вершины которого образуют вектора $g^j(x_0), j \in J_0$, существует гиперплоскость, которая отделяет вектор g_0 от векторов $g^j(x_0), j \in J_0$. Более точно, существует вектор x' и порог θ , удовлетворяющие неравенствам

$$\langle g_0, x' \rangle < \theta, \quad (5.89)$$

$$\langle g^j(x_0), x' \rangle > \theta, \quad j \in J_0. \quad (5.90)$$

Вычитая неравенство (5.89) из каждого неравенства системы (5.90), получим

$$\langle g^j(x_0), x' \rangle - \langle g_0, x' \rangle > 0, \quad j \in J_0. \quad (5.91)$$

Левая часть j -ого неравенства - это производная непрерывной дифференцируемой функции $f^j(x_0 + tx') - \langle g_0, x_0 + tx' \rangle$ по переменной t в точке $t = 0$. Так как согласно неравенству (5.91) эти производные положительны, то при достаточно малых положительных t выполняется неравенство

$$f^j(x_0 + tx') - \langle g_0, x_0 + tx' \rangle > f^j(x_0) - \langle g_0, x_0 \rangle,$$

из которого следует

$$\min_{j \in J_0} (f^j(x_0 + tx') - f^j(x_0)) > \langle g_0, x_0 + tx' \rangle - \langle g_0, x_0 \rangle.$$

По определению (5.87), все числа $f^j(x_0)$, $j \in J_0$, равны $f(x_0)$ и в силу этого не зависят от j . Поэтому

$$\min_{j \in J_0} f^j(x_0 + tx') - f(x_0) > \langle g_0, x_0 + tx' \rangle - \langle g_0, x_0 \rangle. \quad (5.92)$$

Все функции $f^j(x)$, $j \in J$, непрерывны, и поэтому, если для некоторого индекса $j \in J$ не выполняется $f^j(x_0) = \min_{j \in J} f^j(x_0)$, то по крайней мере при достаточно малых значениях t не выполнится и равенство $f^j(x_0 + tx') = \min_{j \in J} f^j(x_0 + tx')$. Поэтому

$$\min_{j \in J_0} f^j(x_0 + tx') = \min_{j \in J} f^j(x_0 + tx'),$$

неравенство (5.92) принимает вид

$$f(x_0 + tx') - f(x_0) > \langle g_0, x_0 + tx' \rangle - \langle g_0, x_0 \rangle,$$

а g_0 не является обобщенным градиентом функции f . Этим доказано второе утверждение теоремы 5.10. ■

Закроем пока что глаза на то, что сформулированная теорема выражает свойства вогнутых функций, а функции, максимум которых должен отыскиваться в андерсоновых задачах, не являются вогнутыми, хотя и являются унимодальными. Тогда мы увидим, что теорема 5.2 о необходимых и достаточных условиях максимума в андерсоновых задачах довольно очевидно следует из только что приведенной теоремы. Действительно, теорема 5.10 утверждает, что любой обобщенный градиент функции

$$\min_{j \in J} f^j(x) \quad \text{имеет вид} \quad \sum_{j \in J_0} \gamma_j \cdot g^j(x).$$

Теорема 5.9 утверждает, что максимум вогнутой функции достигается в тех и только тех точках, в которых один из обобщенных градиентов равен

нулю. Это значит, что существуют такие положительные числа γ^j , $j \in J_0$, что

$$\sum_{j \in J_0} \gamma^j \cdot g^j = 0.$$

В лекции же было доказано, что градиенты g^j коллинеарны векторам, направленным на точки касания. Следовательно, утверждение, что выпуклое замыкание градиентов g^j , $j \in J_0$, содержит начало координат, эквивалентно утверждению, доказанному в лекции, что начало координат содержится в выпуклом замыкании точек касания. Доказанное в лекции необходимое и достаточное условие решения андерсоновых задач можно было бы получить, как простое следствие теоремы 5.7, известной из теории негладкой оптимизации. Однако, это было бы сделано с определенной натяжкой, так как теорема 5.7 доказана для вогнутых функции, а в задачах Андерсона они не являются таковыми.

Однако давай используем для наших задач и другие выводы теории негладкой оптимизации, помня при этом, что это делается с натяжкой. Для нашей задачи могут быть применимы не только необходимые и достаточные условия максимума, но и алгоритмы оптимизации, общий вид которых указан в теореме 5.7. Из анализа задачи, представленного в лекции, следует, что на каждом шаге оптимизации следует найти направление, в котором возрастает функция $f(x) = \min_j f^j(x)$, то есть возрастает каждая из функций $f^j(x)$, $j \in J_0$. Сейчас мы уже знаем, что это направление есть один из обобщенных градиентов. Однако из теоремы 5.7 следует, что такое требование к выбору направления слишком жесткое. В качестве направления движения можно выбрать любой обобщенный градиент, а не обязательно именно то направление, при котором возрастают все функции $f^j(x)$, $j \in J_0$. Проще говоря, в качестве направления движения можно выбрать любой из градиентов g^j , $j \in J_0$. Алгоритм решения обобщенной задачи Андерсона приобретет следующую невероятно простую форму.

Алгоритм строит последовательность векторов $\alpha_1, \alpha_2, \dots, \alpha_t, \dots$. Если вектор α_t уже построен, то выбирается ЛЮБОЙ (!!!) индекс j , для которого

$$f^j(\alpha_t) = \min_{l \in J} f^l(\alpha_t).$$

Затем вычисляется точка касания x_0^j и новое значение α_{t+1} ,

$$\alpha_{t+1} = \alpha_t + \gamma_t \cdot \frac{x_0^j}{|x_0^j|},$$

где γ_t , $t = 1, 2, \dots, \infty$, заранее заданная последовательность коэффициентов, удовлетворяющих условиям

$$\sum_{t=0}^{\infty} \gamma_t = \infty \quad \text{и} \quad \lim_{t \rightarrow \infty} \gamma_t = 0.$$

Мне кажется, что я перестал что-то понимать. До тех пор, пока формулировались понятия и теоремы общей теории негладкой оптимизации, все представлялось мне понятным и естественным. Но когда оказалось, какой невероятно простой алгоритм решения задачи Андерсона следует из этой теории, у меня появились сомнения. Я вижу, что я недостаточно ясно воспринимаю именно базовую теорему 5.7 негладкой оптимизации, из которой следует невероятно простой алгоритм решения андерсоновых задач.

Я покажу, почему трудно принять на веру этот алгоритм. Допустим, что мне нужно максимизировать функцию

$$f(x) = \min_{j \in J} f^j(x)$$

и алгоритм достиг точку, в которой две функции, скажем f^1 и f^2 , принимают одно и то же значение, которое меньше, чем значения остальных функций в текущей точке. Здесь следует изменить точку x так, чтобы функция $f(x)$ увеличилась, следовательно, чтобы увеличилась как функция f^1 , так и функция f^2 . Однако алгоритм, вытекающий из общей теории, ведет себя так, словно он просто игнорирует одну из этих двух функций, а увеличивает значение только одной из них. В этом случае, конечно же, не исключено, что значение одной из двух функций уменьшится, а следовательно, уменьшится и функция f в целом. Такая ситуация не только в принципе возможна. Она гарантированно случится, так как по мере приближения к максимуму увеличивается количество функций, которые следует принимать во внимание, а алгоритм все время принимает во внимание только одну из них.

Ты полностью прав, но это не значит, что теорема 5.7 неверна. Ты всего лишь утверждаешь, что алгоритм не обеспечивает монотонного роста оптимизируемой функции, но теорема 5.7 это и не утверждает. Она только утверждает, что алгоритм стремится к множеству точек, в которых достигается максимум, а это совсем не обязательно предполагает монотонность. Давай рассмотрим, хотя и не очень строго, ситуацию, которую Ты выдвинул как контрпример. Если уж так случилось, что с ростом одной функции, скажем, f_1 , значение функции f_2 уменьшилось (а это вполне может произойти), то уже на следующем шаге функция f_1 не будет приниматься во внимание, и именно функция f_2 может оказаться худшей и именно ее будет стремиться улучшить алгоритм.

Доказательство теоремы 5.7 должно быть чертовски интересным!

Да, и не очень простым!

Ну хорошо, если уж мы верим в правильность теоремы 5.7, я осмелюсь задать дерзкий вопрос. Почему же именно эта теорема не стала основой лекции? Почему лекция построена так, как будто этой теоремы нет?

Напомним Тебе, что только что Ты сказал, что Ты не можешь принять на веру теорему 5.7. Сейчас Ты уже можешь ее принять, но это ничего не меняет. Мы не хотим строить лекцию на основе теоремы, которую мы не понимаем досконально, хотя она где-то и безукоризненно доказана. Основой лекции должна быть не вера в где-то доказанную теорему, а знания, изложенные в этой же лекции.

Но значительно важнее то, что применение этой теоремы к нашей задаче все же происходило бы с определенной натяжкой, о которой мы уже упоминали. Прежде, чем ее применять, надо было бы ее разумно обобщить, чтобы не ограничивать сферу ее действия только лишь вогнутыми функциями, а включить в нее и функции, которые возникают в нашей задаче. Но это была бы уже совсем другая лекция.

И все же мне трудно представить, что вы не применяли этот простой алгоритм для каких-то своих целей. Неужели при решении каких-то практических задач вы отказались от его применения только потому, что он не полностью обоснован. Ведь та натяжка, о которой вы говорите, представляется мне очень незначительной.

Конечно же, мы применяли его многократно и получали вполне приличные результаты при решении задач, которые нужно было решать, несмотря на неполную теоретическую ясность. Однако так мы делали до тех пор, пока не получили алгоритмы ε -решения задач Андерсона, описанные в лекции. После приобретения ясности в ε -решениях именно они оказались более предпочтительными как в теоретическом, так и во многих чисто прагматических отношениях.

Мне бы хотелось узнать ваше мнение об отношении распознавания образов и окружающих его математических дисциплин, откуда распознавание черпает те или иные результаты. Я часто вижу в статьях и докладах, как при решении той или иной задачи распознавания вводятся определения и доказываются утверждения, которые с точностью до терминологии повторяют результаты, известные в других областях чистой или прикладной математики. Как мне к этому относиться? Как к повторному изобретению велосипеда? Или считать, что наша работа только в том и состоит, чтобы грамотно использовать знания из других областей для решения наших прикладных задач? И вообще, что же такое распознавание образов? Наука, или искусство, или набор маленьких хитростей, или совокупность результатов, почерпнутых из других областей? Я понимаю, что эти вопросы не по теме лекции, но рано или поздно я должен их задать, если я не исключаю свое пребывание в распознавании в будущем.

Рассмотрим определенный момент в вашей лекции, который я беру только в качестве примера своих затруднений. Он относится к отысканию гиперплоскости, проходящей между двумя заданными множествами. Ключевым моментом здесь является то, что этой плоскостью является срединный перпендикуляр к отрезку, соединяющему две ближайшие точки в

выпуклых замыканиях исходных множеств. Это известный результат из выпуклого анализа и выпуклого программирования. Я сам его не раз использовал при решении задач вычислительной геометрии. Какие же новые знания я получаю, когда еще раз вижу этот результат в лекции по распознаванию?

Очень много вопросов. Время от времени они возникают не только у Тебя. Начнем с самого трудного вопроса, что такое распознавание, искусство или наука или еще что-то. Мы можем лишь отмахнуться от этого вопроса, ответив, что это то, чем мы сейчас втроем занимаемся, когда работаем над данными лекциями. Мы понимаем, что это не очень умный ответ, но уверяем Тебя, что другие известные нам ответы ненамного умнее. Не стоит даже пытаться ответить на этот вопрос в двух-трех предложениях.

Теперь Твой второй вопрос, как относиться к тому, что наука о распознавании включает в себя знания, давно известные в других областях теоретической и прикладной информатики. Этому следует только радоваться и сожалеть лишь о том, что это происходит значительно реже, чем хотелось бы. При этом, конечно же, время от времени возникает и такая ситуация, когда при решении какой-то задачи необходимые математические средства разрабатываются внутри самого распознавания, хотя эти средства уже давно известны за пределами распознавания. Такая ситуация огорчительна разве что для автора такого повторного открытия, но не для науки о распознавании. В повторно изобретенном велосипеде, конечно же, больше пользы, чем в отсутствии велосипеда вообще.

Рассмотрим, наконец, пример, который Ты предложил для более пристального рассмотрения. Опишем еще раз этот пример. Пусть X_1 и X_2 - два конечных множества точек линейного пространства. Требуется найти такую гиперплоскость, чтобы множества X_1 и X_2 оказались по разные стороны от нее и были от нее как можно больше удалены. Для поиска такой гиперплоскости следует найти ближайшие друг к другу точки x_1 и x_2 , принадлежащие выпуклым замыканиям множеств X_1 и X_2 . Искомая гиперплоскость ортогональна к отрезку, соединяющему точки x_1 и x_2 , и проходит через его середину.

Это действительно известный результат из выпуклого анализа, известный как теорема отделимости. Если Ты хорошо ее знаешь, то вряд ли Ты стал ее понимать еще лучше оттого лишь, что еще раз ее увидел в лекции по распознаванию. Но здесь важно другое, а именно, что будучи использованной именно в распознавании, эта теорема дополнилась новыми результатами, которые, может быть, и не появились бы вне распознавания. Будучи перенесенной в распознавание, эта теорема позволяет ответить на вопросы, которые возникли при распознавании и нигде больше, и следовательно, до поры до времени оставались без ответа. Например, это вопросы о выборе так называемых эталонов в следующем классе распознающих алгоритмов.

Пусть X_1 и X_2 - два выпуклых подмножества линейного пространства X с евклидовой метрикой $d: X \rightarrow \mathbb{R}$. Пусть стратегия принятия решения имеет следующий вид. Имеются два эталона α_1 и α_2 , которыми являются

точки в пространстве X . Для наблюдения x принимается первое решение, если расстояние от x до α_1 меньше, чем до α_2 . В противном случае принимается второе решение. Вопрос состоит в том, как должны выглядеть эталоны α_1 и α_2 , чтобы для всех точек из X_1 принималось первое решение, а для всех точек их X_2 - второе.

По этому вопросу известны различные рекомендации. На одних рекомендациях явно чувствуется влияние того, что точки α_1 и α_2 названы эталонами. Поэтому предполагается, что это должны быть какие-то "наименее поврежденные", "идеальные", самые "хорошие" представители множеств X_1 и X_2 . В случае, когда речь идет об изображениях, предлагается в качестве эталонов выбирать искусственно построенные изображения, пусть даже не из множеств X_1 и X_2 . Другие рекомендации исходят из того, что, наоборот, эталоны должны быть близкими к реальным изображениям и поэтому должны выбираться в "центрах тяжести" X_1 и X_2 . Бытует и много других рекомендаций такой же степени обоснованности и убедительности.

Убедительный ответ получается после применения теоремы отделимости. Согласись, что ответ довольно неожиданный. Эталонами не должны быть ни наилучшие, ни средние представители. Если, например, надо выбрать эталоны для распознавания двух букв O и C , то в качестве эталона буквы O Ты должен выбрать такое изображение буквы O , которое наиболее похоже на букву C . И наоборот, в качестве эталона буквы C следует выбрать такое изображение буквы C , которое наиболее похоже на букву O . Это значит, что эталонами должны быть наихудшие в определенном смысле представители.

Эти новые знания получены только после применения теоремы отделимости в рамках распознавания. В рамках выпуклого анализа, где теорема родилась, эти выводы не могли быть получены хотя бы потому, что там просто не возникают такие вопросы, как выбор эталонов. Новые знания возникли только после перенесения теоремы отделимости из одной области знаний в другую.

В то же время результаты выпуклого анализа, побывав в области распознавания, существенно обогащаются. Теорема отделимости, взятая сама по себе, утверждает лишь, что задача линейной отделимости одного конечного множества точек от другого сводится к решению определенной задачи квадратичного программирования. Последняя вроде бы уже хорошо исследована и может решаться разнообразными методами, например, методом сопряженных градиентов. Однако именно в рамках распознавания обнаруживается тот факт, что эта, казавшаяся бы, простейшая задача выпуклой оптимизации исследована недостаточно. Количество переменных в квадратичной функции, которую требуется минимизировать, равно количеству точек в конечных множествах X_1 и X_2 , которые следует отделить друг от друга. Если это количество равно 10000, то применение стандартных методов квадратичной оптимизации проблематично. Одно только задание функции, которую следует минимизировать, требует памяти, которая растет квадратично с ростом количества точек в множествах X_1 и X_2 . И тут вступают в силу алгоритмы Козинца, персептронные алгоритмы, получен-

ные уже внутри распознавания и которые такими недостатками не обладают. Это новые алгоритмы решения систем строгих неравенств или новые алгоритмы вычисления расстояния между выпуклыми множествами. Таким путем распознавание образов, используя результаты из соседних разделов информатики, возвращает их туда с некоторыми процентами.

Укажем еще пример такого обогащения заимствованных результатов после их пребывания в распознавании. Ты упомянул, что неоднократно использовал теорему отделимости для решения некоторых задач вычислительной геометрии. Повидимому, это были задачи об отыскании прямой на плоскости, при которой две заданные на плоскости совокупности точек оказываются по разные стороны от нее. Теперь Ты узнал, как теорема отделимости работает в распознавании и как она соприкасалась с другими приемами, применяемыми в распознавании. Мы имеем в виду такой нехитрый прием, как спрямление пространства. Поэтому сейчас у Тебя не возникнут затруднения, если придется отделить друг от друга две совокупности точек на плоскости не с помощью прямой, а с помощью окружности, или эллипса, или с помощью двух ветвей гиперболы. Не приведет Тебя в смущение и ситуация, если подобные задачи возникнут в трехмерном пространстве, так как Ты знаешь, что трудоемкость решения задачи зависит не от размерности пространства, а от чего-то совсем другого. Таким путем теорема отделимости в результате ее использования в распознавании обростаёт новыми результатами и окрепнувшая возвращается в среду, из которой она пришла в распознавание.

Мы все еще не осмеливаемся сказать, является ли распознавание ремеслом, или наукой, или искусством. Но мы имеем смелость сказать, что вряд ли в какой-то другой области различные разделы прикладной математики приходят в столь тесное соприкосновение, как в распознавании. Поэтому распознавание образов может служить испытательным полигоном не только для тех, кто как Ты, хотел бы профессионально работать в этой области, но для всех, кто на собственном опыте и быстро хотел бы узнать все "прелести" работы в прикладной информатике.

Анализ задач Андерсона очень трудный. Само по себе это не является недостатком. И все же жаль, что такой сложный анализ потребовался для сравнительно частной задачи распознавания. Особенно жестким мне видится ограничение, что определенные случайные величины должны быть многомерными гауссовыми случайными величинами. Ни один пользователь не может быть уверен, что наблюдения в его приложении действительно являются таковыми. Я вообще сомневаюсь, что это предположение можно как-то экспериментально проверить. Меня очень интересует такая постановка задачи, в которой не предполагается гауссовость определенных многомерных наблюдений, а исходят из существенно более слабого предположения, что случайная величина имеет известное математическое ожидание μ и ковариационную матрицу σ . Это предположение уже можно экспериментально проверить. Постараюсь сформулировать такую задачу более точно.

Пусть X , как и в андерсоновой задаче, многомерное пространство, $\{\mu^j, j \in J\}$ - это совокупность векторов, $\{\sigma^j, j \in J\}$ - совокупность симметричных положительноопределенных матриц. Пусть \mathcal{P}^j - множество функций $p: X \rightarrow \mathbb{R}$, для которых выполняется

$$\begin{aligned} \sum_X p(x) &= 1, \\ \sum_X p(x) \cdot x &= \mu^j, \\ \sum_X p(x) \cdot x \cdot x^T &= \sigma^j, \\ p(x) &\geq 0, \quad x \in X. \end{aligned}$$

Обозначение x^T служит для вектора (строки), полученного после транспонирования вектора (столбца) x . Мне понятно, что во всех этих формулах следовало бы писать интегралы, а не конечные суммы. Но сейчас не в этом дело.

Пусть вектор α и число θ определяют множество $X(\alpha, \theta) = \{x \mid \langle \alpha, x \rangle \leq \theta\}$. Далее, пусть для каждой функции $p: X \rightarrow \mathbb{R}$ определено число

$$\varepsilon(\alpha, \theta, p) = \sum_{x \in X(\alpha, \theta)} p(x),$$

которое обозначает вероятность неравенства $\langle \alpha, x \rangle \leq \theta$ для случайного вектора x с распределением вероятностей $p(x)$.

В задаче, которую я имею в виду, я бы хотел избежать предположения о гауссовом характере случайного вектора x . Поэтому я формулирую задачу, как отыскание вектора α и числа θ , которые минимизируют значение

$$\max_{j \in J} \max_{p \in \mathcal{P}^j} \varepsilon(\alpha, \theta, p).$$

Искомыми в задаче параметрами являются, таким образом,

$$(\alpha, \theta) = \operatorname{argmin}_{\alpha, \theta} \max_{j \in J} \max_{p \in \mathcal{P}^j} \varepsilon(\alpha, \theta, p). \quad (5.93)$$

Мне кажется, что сформулированная задача по своему характеру очень близка к задаче, которая меня заинтересовала после лекции 3. Тогда я старался найти разумную стратегию, которая не исходит из общепринятого предположения о независимости признаков. С вашей помощью я увидел, что и полученная стратегия отличается от общепринятой. Сейчас меня интересует подобный вопрос: совпадает ли стратегия решения задачи (5.93) со стратегией, полученной при более жестком предположении, что наблюдение x является гауссовым случайным вектором? Или, говоря другими словами: можно ли использовать стратегию решения задачи Андерсона в ситуации, когда я не уверен, что случайный вектор x является случайным гауссовым вектором, и более того, когда мне ничего неизвестно об этом

векторе, кроме его математического ожидания и ковариационной матрицы?

Ну парень, за что мы Тебя уважаем, так это за Твое умение время от времени ставить такие глубоко продуманные и хорошо сформулированные вопросы. Мы нашли убедительный и точный ответ на Твой вопрос. Однако нам с трудом верится, что Ты не знаешь ответ на этот вопрос, раз Ты его уже так грамотно сформулировал. Тебе следует исследовать функцию $\max_{p \in \mathcal{P}^j} \varepsilon(\alpha, \theta, p)$. Если Ты увидишь, что она монотонно убывает с ростом отношения

$$\frac{\langle \alpha, \mu^j \rangle - \theta}{\sqrt{\langle \alpha, \sigma^j \cdot \alpha \rangle}}$$

то решение задачи Андерсона можно применять не только в случае гауссовых случайных векторов. В противном случае придется дополнительно исследовать Твою задачу.

Да, это я знаю, но я не знаю, как двигаться дальше. Все представляется мне слишком сложным. Ведь существуем несчетное множество многомерных случайных величин с одинаковыми математическим ожиданием и ковариационной матрицей.

Не спеши и начни анализ со следующей, например ситуации. Пусть математическое ожидание n -мерного случайного вектора $x = (x_1, x_2, \dots, x_n)$ равно нулю, то есть,

$$\sum_{x \in X} p(x) \cdot x = 0. \quad (5.94)$$

Предыдущее выражение есть краткая запись следующих n уравнений

$$\sum_{(x_1, \dots, x_n) \in X} p(x_1, x_2, \dots, x_n) \cdot x_i = 0, \quad i = 1, 2, \dots, n.$$

Пусть ковариационная матрица этой же случайной величины равна σ , то есть,

$$\sum_{x \in X} p(x) \cdot x \cdot x^T = \sigma, \quad (5.95)$$

что в сокращенном виде обозначает следующие $n \times n$ уравнений

$$\sum_{x \in X} p(x) \cdot x_i \cdot x_j = \sigma_{ij}, \quad i = 1, 2, \dots, n, \quad j = 1, 2, \dots, n.$$

Пусть α - это вектор, $\theta < 0$ - это число, а $X^-(\alpha, \theta)$, $X^0(\alpha, \theta)$ и $X^+(\alpha, \theta)$ - множества

$$\begin{aligned} X^-(\alpha, \theta) &= \{x \in X \mid \langle \alpha, x \rangle \leq \theta\}, \\ X^0(\alpha, \theta) &= \{x \in X \mid \langle \alpha, x \rangle = \theta\}, \\ X^+(\alpha, \theta) &= \{x \in X \mid \langle \alpha, x \rangle > \theta\}. \end{aligned}$$

Вероятность $\varepsilon(\alpha, \theta, p)$ в этом случае равна

$$\varepsilon(\alpha, \theta, p) = \sum_{x \in X^-(\alpha, \theta)} p(x). \quad (5.96)$$

Ты бы хотел узнать, какими должны быть числа $p(x)$, $x \in X$, которые удовлетворяют условиям (5.94), (5.95), условию

$$\sum_{x \in X} p(x) = 1, \quad p(x) \geq 0, \quad x \in X,$$

и максимизируют число $\varepsilon(\alpha, \theta, p)$, определенное равенством (5.96). Чтобы немного привыкнуть к задаче, взглянь на одномерный частный случай задачи. Какими должны быть числа $p(x)$, которые максимизируют число

$$\sum_{x \leq \theta} p(x) \quad (5.97)$$

при соблюдении условий

$$\begin{aligned} \sum_{x \in X} p(x) &= 1, \\ \sum_{x \in X} p(x) \cdot x &= 0, \\ \sum_{x \in X} p(x) \cdot x^2 &= \sigma, \\ p(x) &\geq 0, \quad x \in X. \end{aligned}$$

Как бы там ни было, но это же всего навсего задача линейного программирования. Правда, здесь бесконечное количество переменных. Но нам же не нужно решать эту задачу на компьютере. Нам следует ее анализировать на бумаге. Задача напоминает известную задачу Чебышева, которая отличается от нашего случая лишь тем, что вместо максимизации (5.97) максимизируется сумма

$$\sum_{|x| \geq \theta} p(x). \quad (5.98)$$

Решение задачи Чебышева так прекрасно, что мы не можем удержаться, чтоб его не привести, хотя оно и общеизвестно.

$$\begin{aligned} \sum_{|x| \geq \theta} p(x) &= \theta^2 \cdot \frac{1}{\theta^2} \sum_{|x| \geq \theta} p(x) = \frac{1}{\theta^2} \sum_{|x| \geq \theta} p(x) \cdot \theta^2 \\ &\leq \frac{1}{\theta^2} \sum_{|x| \geq \theta} p(x) \cdot x^2 \leq \frac{1}{\theta^2} \sum_{x \in X} p(x) \cdot x^2 = \frac{\sigma}{\theta^2}. \end{aligned}$$

Следовательно,

$$\sum_{|x| \geq \theta} p(x) \leq \frac{\sigma}{\theta^2}. \quad (5.99)$$

Пусть $\sigma/\theta^2 \leq 1$. Рассмотрим следующую функцию $p^*(x)$,

$$p^*(x) = \begin{cases} \frac{1}{2} \cdot \frac{\sigma}{\theta^2} & \text{при } |x| = \theta, \\ 1 - \frac{\sigma}{\theta^2}, & \text{при } x = 0 \\ 0 & \text{для всех прочих } x. \end{cases}$$

Для приведенной функции $p^*(x)$ сумма (5.98) в точности равна σ/θ^2 , а для всех других функций в силу (5.99) не больше, чем σ/θ^2 . Следовательно, максимальное значение суммы (5.98) при ограничениях (5.99) равно σ/θ^2 . Ты видишь, что максимизация суммы (5.98) не обязательно сложная, даже если эта сумма состоит из бесконечно большого количества слагаемых. Согласись, что Чебышевское решение прекрасно. Сделай что-нибудь подобное, сначала потренируйся на одномерном случае, а потом реши дело с максимизацией (5.97) при условиях (5.99) в общем многомерном случае.

Я это сделал!!!! Не буду вас затруднять своими одномерными заготовками и опишу сразу общий случай. Мне следует решить следующую задачу линейного программирования,

$$\left. \begin{array}{l} \max \sum_{x \in X(\alpha, \theta)} p(x) \\ \lambda_0 \left| \begin{array}{l} \sum_{x \in X} p(x) = 1, \\ \sum_{x \in X} p(x) \cdot x = 0, \\ \sum_{x \in X} p(x) \cdot x \cdot x^T = \sigma, \\ p(x) \geq 0, \quad x \in X. \end{array} \right. \end{array} \right\} \quad (5.100)$$

В первой строчке (5.100) записана целевая функция, которую следует максимизировать. Обозначение $X(\alpha, \theta)$ в целевой функции - это $X^-(\alpha, \theta) \cup X^0(\alpha, \theta)$. Во второй строчке записано ограничение, которому соответствует двойственная переменная λ_0 . В третьей строчке записаны в краткой форме n ограничений, которым соответствуют n двойственных переменных. Обозначим их для краткости, как вектор λ_1 . В четвертой строчке кратко записаны $n \times n$ ограничений, которым соответствует совокупность двойственных переменных, для краткости обозначенных матрицей λ_2 .

Переменными в прямой задаче (5.100) являются числа $p(x)$, $x \in X$. Каждому из них соответствует ограничение на двойственные переменные: число λ_0 , вектор λ_1 и матрицу λ_2 . Переменной $p(x)$, $x \in X(\alpha, \theta)$, соответствует ограничение

$$\langle x, \lambda_2 \cdot x \rangle + \langle \lambda_1, x \rangle + \lambda_0 \geq 1, \quad x \in X(\alpha, \theta), \quad (5.101)$$

а переменной $p(x)$, $x \in X^+(\alpha, \theta)$, - ограничение

$$\langle x, \lambda_2 \cdot x \rangle + \langle \lambda_1, x \rangle + \lambda_0 \geq 0, \quad x \in X^+(\alpha, \theta). \quad (5.102)$$

Ограничения (5.101) и (5.102) следует понимать так, что совокупность двойственных переменных порождает на пространстве X квадратичную функцию F ,

$$F(x) = \langle x, \lambda_2 \cdot x \rangle + \langle \lambda_1, x \rangle + \lambda_0,$$

которая должна быть не меньше, чем 1, на множестве $X(\alpha, \theta)$ и быть неотрицательной на множестве $X^+(\alpha, \theta)$. Функция $F(x)$ в целом должна быть неотрицательно определенной.

Я рассмотрю задачу (5.100) только для случая, когда математическое ожидание случайной величины x принадлежит множеству $X^+(\alpha, \theta)$. Это значит, что $\theta < 0$.

Пусть p^* - решение задачи. Тогда по крайней мере для некоторой точки $x \in X^+(\alpha, \theta)$ должно выполняться $p^*(x) \neq 0$, так как в противном случае

$$\begin{aligned} \left\langle \alpha, \sum_{x \in X} p^*(x) \cdot x \right\rangle &= \left\langle \alpha, \sum_{x \in X(\alpha, \theta)} p^*(x) \cdot x \right\rangle \\ &= \sum_{x \in X(\alpha, \theta)} p^*(x) \cdot \langle \alpha, x \rangle \\ &\leq \sum_{x \in X(\alpha, \theta)} p^*(x) \cdot \theta = \theta < 0, \end{aligned} \quad (5.103)$$

и следовательно,

$$\left\langle \alpha, \sum_{x \in X} p^*(dx) \cdot x \right\rangle < 0$$

что противоречит ограничению в третьей строчке (5.100). Обозначим x_0 точку $x \in X^+(\alpha, \theta)$, для которой $p^*(x) \neq 0$, а скалярное произведение $\langle \alpha, x_0 \rangle$ обозначим Δ .

Сейчас я докажу, что для любой точки $x_1 \in X^+(\alpha, \theta)$, для которой $p^*(x_1) \neq 0$, скалярное произведение $\langle \alpha, x_1 \rangle$ также равно Δ . Говоря другими словами, все точки $x_1 \in X^+(\alpha, \theta)$, для которых $p^*(x) \neq 0$, лежат на гиперплоскости $\langle \alpha, x \rangle = \theta$. Обозначим ее $X^0(\alpha, \theta)$. Допустим, что я не прав, и рассмотрим пару точек, одной из которых является точка x_0 , а другой - точка x_1 , такая, что

$$\left. \begin{aligned} p^*(x_0) &\neq 0, \\ p^*(x_1) &\neq 0, \\ \langle \alpha, x_0 \rangle &\neq \langle \alpha, x_1 \rangle. \end{aligned} \right\} \quad (5.104)$$

Посмотрим, как ведет себя функция $F(x)$ на прямой линии, проходящей через точки x_0 и x_1 . Эта прямая не параллельна гиперплоскости $X^0(\alpha, \theta)$. Следовательно, гарантированно существует точка x_2 , которая лежит на этой прямой и одновременно на гиперплоскости $X^0(\alpha, \theta)$. В силу ограничения (5.101) значение функции F в этой точке удовлетворяет неравенству

$$F(x_2) \geq 1.$$

По второй теореме двойственности из первого и второго неравенств в (5.104) следует соответственно

$$F(x_0) = 0, \quad F(x_1) = 0.$$

Любая выпуклая линейная комбинация точек x_0 и x_1 принадлежит множеству $X^+(\alpha, \theta)$, и следовательно, в силу (5.102), для каждой точки на отрезке, соединяющем эти точки, выполняется

$$F(x^*) \geq 0.$$

Таким образом, функция F на прямой, проходящей через точки x_0 и x_1 , ведет себя следующим образом. На отрезке между точками x_0 и x_1 эта функция неотрицательна, в точках x_0 и x_1 она принимает нулевое значения, а в некоторой точке x_2 вне этого отрезка - не меньше, чем 1. Посекольку такой квадратичной функции не бывает, предположение (5.104) неверно. Таким образом я доказал, что во всех точках $x \in X^+(\alpha, \theta)$, для которых $p^*(x) \neq 0$, скалярное произведение $\langle \alpha, x \rangle$ принимает одно и то же значение, которое я обозначил Δ ,

$$[(x \in X^+(\alpha, \theta)) \wedge (p^*(x) \neq 0)] \Rightarrow [\langle \alpha, x \rangle = \Delta].$$

Я докажу теперь утверждение

$$[(x \in X(\alpha, \theta))] \wedge (p^*(x) \neq 0) \Rightarrow [\langle \alpha, x \rangle = \theta], \quad (5.105)$$

которое эквивалентно утверждению

$$(\langle \alpha, x \rangle < \theta) \Rightarrow (p^*(x) = 0). \quad (5.106)$$

Предположим, что утверждение (5.106) ложно. Пусть для какой-то точки x_1 выполняется

$$p^*(x_1) \neq 0, \quad \langle \alpha, x_1 \rangle < \theta. \quad (5.107)$$

Посмотрим, как ведет себя функция F на прямой, проходящей через точки x_1 и x_0 . Для точки x_0 , рассмотренной ранее, выполняется

$$p^*(x_0) \neq 0, \quad \langle \alpha, x_0 \rangle > \theta \quad (5.108)$$

Прямая, проходящая через точки x_1 и x_0 , обязательно пересекает гиперплоскость $X^0(\alpha, \theta)$. Обозначим точку пересечения x^* . Из (5.101) следует, что

$$F(x^*) \geq 1.$$

В силу второй теоремы двойственности (Теорема 2.2) из второго неравенства в (5.107) следует, что

$$F(x_1) = 1,$$

а из первого неравенства в (5.108) следует

$$F(x_0) = 0.$$

Функция F ведет себя на прямой линии, проходящей через точки x_1 и x_0 , следующим образом. В точке x_1 функция F принимает значение 1, в точке x_0 - значение 0, а в промежуточной точке x^* принимает значение, не меньшее, чем 1. Такой неотрицательно определенной квадратичной функции не бывает и поэтому предположение (5.107) неверно. Таким образом я доказал (5.105).

Множество точек x , для которых $\langle \alpha, x \rangle = \Delta$, это гиперплоскость $X^0(\alpha, \Delta)$. Я обозначу

$$p_\theta^* = \sum_{x \in X^0(\alpha, \theta)} p^*(x),$$

$$p_\Delta^* = \sum_{x \in X^0(\alpha, \Delta)} p^*(x),$$

где p^* , как и прежде, решение задачи (5.100). Функция p^* , следовательно, удовлетворяет условиям этой задачи и поэтому

$$\left. \begin{aligned} p_\theta^* + p_\Delta^* &= 1, \\ p_\theta^* \cdot \theta + p_\Delta^* \cdot \Delta &= 0, \\ p_\theta^* \cdot \theta^2 + p_\Delta^* \cdot \Delta^2 &= \langle \alpha, \sigma \cdot \alpha \rangle. \end{aligned} \right\} \quad (5.109)$$

Я покажу, как (5.109) следует из условий (5.100). Первое равенство вполне очевидно следует из условия (5.109), потому что любая точка x , такая, что $p(x) \neq 0$, принадлежит либо $X^0(\alpha, \Delta)$, либо $X^0(\alpha, \theta)$.

Равенство $\sum_{x \in X} p(x) \cdot x = 0$ в (5.100) есть краткая запись n равенств

$$\sum_{x \in X} p^*(x) \cdot x_i = 0, \quad i = 1, 2, \dots, n.$$

Из этих n равенств следует, что

$$\alpha_i \cdot \sum_{x \in X} p^*(x) \cdot x_i = 0, \quad i = 1, 2, \dots, n,$$

и далее

$$\begin{aligned} 0 &= \sum_i \alpha_i \cdot \sum_{x \in X} p^*(x) \cdot x_i = \sum_{x \in X} p^*(x) \cdot \left(\sum_i \alpha_i \cdot x_i \right) \\ &= \sum_{x \in X} p^*(x) \cdot \langle \alpha, x \rangle = \sum_{x \in X^0(\alpha, \theta)} p^*(x) \cdot \theta + \sum_{x \in X^0(\alpha, \Delta)} p^*(x) \cdot \Delta \\ &= p_\theta^* \cdot \theta + p_\Delta^* \cdot \Delta. \end{aligned}$$

Равенство $\sum_{x \in X} p^*(x) \cdot x \cdot x^T = \sigma$ в (5.100) - это просто краткая запись $n \times n$ равенств

$$\sum_{x \in X} p^*(x) \cdot x_i \cdot x_j = \sigma_{ij}, \quad i = 1, \dots, n, \quad j = 1, 2, \dots, n,$$

из которых следует

$$\alpha_i \cdot \alpha_j \cdot \sum_{x \in X} p^*(x) \cdot x_i \cdot x_j = \alpha_i \cdot \sigma_{ij} \cdot \alpha_j, \quad i = 1, \dots, n, \quad j = 1, \dots, n,$$

и далее

$$\begin{aligned} \langle \alpha, \sigma \cdot \alpha \rangle &= \sum_{i,j} (\alpha_i \cdot \sigma_{ij} \cdot \alpha_j) = \sum_{i,j} \alpha_i \cdot \alpha_j \cdot \sum_{x \in X} p^*(x) \cdot x_i \cdot x_j \\ &= \sum_{x \in X} p^*(x) \cdot \sum_{i,j} \alpha_i \cdot \alpha_j \cdot x_i \cdot x_j \\ &= \sum_{x \in X} p^*(x) \cdot \left(\sum_i \alpha_i \cdot x_i \right) \cdot \left(\sum_j \alpha_j \cdot x_j \right) \\ &= \sum_{x \in X} p^*(x) \cdot \left(\sum_i \alpha_i \cdot x_i \right)^2 \\ &= \sum_{x \in X^0(\alpha, \theta)} p^*(x) \cdot \theta^2 + \sum_{x \in X^0(\alpha, \Delta)} p^*(x) \cdot \Delta^2 \\ &= p_\theta^* \cdot \theta^2 + p_\Delta^* \cdot \Delta^2. \end{aligned}$$

Система (5.109) состоит всего из трех уравнений с тремя неизвестными p_θ^* , p_Δ^* и Δ . Система имеет единственное решение, для которого выполняется

$$p_\theta^* = \frac{\langle \alpha, \sigma \cdot \alpha \rangle}{\theta^2 + \langle \alpha, \sigma \cdot \alpha \rangle}.$$

Число p_θ^* - это как раз значение суммы $\sum_{x \in X(\alpha, \theta)} p(x)$ при подстановке в нее чисел $p^*(x)$, которые являются решением задачи. Искомая максимальная вероятность в задаче (5.100) выражается в явном виде величиной

$$\frac{\langle \alpha, \sigma \cdot \alpha \rangle}{\theta^2 + \langle \alpha, \sigma \cdot \alpha \rangle}, \quad (5.110)$$

которая монотонно убывает с ростом числа

$$\frac{\theta}{\sqrt{\langle \alpha, \sigma \cdot \alpha \rangle}}.$$

Так что, улучшил я неравенство Чебышева?

Ни в коем случае, хотя Ты и замечательно справился со своей задачей. Чебышевское неравенство невозможно улучшить, потому что, как мы видели, это неравенство указывает точную границу сверху для определенной вероятности. Это значит, что существует такое распределение вероятностей, для которого неравенство Чебышева обращается в равенство. Но неравенство Чебышева оценивает вероятность двухстороннего неравенства

$|x - \mu| \geq \theta$, в то время, как Твое неравенство указывает вероятность одностороннего неравенства $x - \mu \leq -\theta$. Так как Ты доказал, что для некоторых случайных чисел неравенство (5.110) обращается в равенство, Твоя оценка также не улучшаема.

Однако в контексте нашей лекции наиболее важно то, что полученный Тобой результат позволяет существенно расширить сферу применимости разработанных в лекции алгоритмов. Сейчас ты можешь применять эти алгоритмы не только для гауссовых многомерных случайных векторов, но и в том случае, когда их распределения вероятностей неизвестны, а известны лишь математические ожидания этих векторов и ковариационные матрицы.

Из-за этого практические рекомендации, вытекающие из лекции, стали, пожалуй, менее определенными. Если у меня имеются два конечных множества X_1 и X_2 , которые мне нужно отделить друг от друга с помощью гиперплоскости, то сейчас для этого имеются уже по крайней мере две возможности. Я могу найти разделяющую гиперплоскость с помощью алгоритмов Козинца, например. Но если количество точек в множествах X_1 и X_2 очень большое, я могу решать свою задачу и по иному. Я могу подсчитать векторы μ^1, μ^2 и матрицы σ^1, σ^2 . После этого я могу воспользоваться алгоритмами решения задачи Андерсона для построения разделяющей гиперплоскости. Я могу это сделать, так как я доказал, что применение этих алгоритмов правомерно не только для случая гауссовых распределений. В этом случае мне придется решать простейшую задачу Андерсона, когда количество гауссовых величин равно всего лишь 2. Так какому из этих двух путей я должен отдать предпочтение?

Повидимому, обоим. Извини нас за не очень серьезный ответ, но такие вещи научись решать сам в каждом отдельном приложении. Это уже приятные заботы. Трудности при выборе одного из нескольких известных Тебе методов значительно меньшие, чем в случае, когда Ты не знаешь ни одного метода. Однако было бы наиболее опасно, если бы Ты знал только один метод и применял его, где надо и не надо.

Я могу действовать и менее прямолинейно. Нет необходимости представлять все множество X_1 одной-единственной случайной величиной, а смесью нескольких. Правда, я не знаю, как это сделать.

Не забегай вперед. Это уже предмет следующей лекции. Мы уверены, что после ознакомления с ней Ты нам опять скажешь что-то интересное.

Июль 1997.

5.7 Связь с toolbox'ом

Toolbox по статистическому распознаванию образов написал В.Франц во время выполнения своей дипломной работы весной 2000 года. Toolbox до-

ступен по адресу http://cmp.felk.cvut.cz/cmp/cmp_software.html. Он реализован в Matlab'е версии 5.3 и выше. Доступны исходные тексты алгоритмов. Toolbox постоянно совершенствуется и пополняется.

Та часть Toolbox'a, которая имеет отношение к данной лекции, реализует линейное разделение конечных множеств точек с помощью перцептронных алгоритмов и алгоритмов Козинца, ε -оптимальное решение с помощью алгоритмов Козинца, алгоритмы обобщенного портрета (известные на Западе, как Support Vector Machines), синтез фишеровских классификаторов, модифицированные алгоритмы Козинца, решение обобщенных задач Андерсона, исходное решение Андерсона-Бахадур, ε -решение задачи Андерсона. Реализован синтез квадратичных дискриминантных функций с помощью отображения в спрямляющее пространство.

5.8 Библиографические заметки

Формулировка исходной задачи Андерсона-Бахадур и ее решение опубликовано в работе [Anderson and Bahadur, 1962]. Шлезингер [Schlesinger, 1972a; Schlesinger, 1972b] сформулировал и решил задачи более общего вида, когда каждое состояние характеризуется не одной гауссовой случайной величиной, а многими.

Одной из первых публикаций алгоритмов перцептрона в том виде, как они сформулированы в лекции, является основополагающая монография Розенблатта о перцептронах [Rosenblatt, 1962]. Теорема Новикова доказана в [Novikoff, 1962]. Доказательство этой теоремы приводится в лекции по публикации [Varnik and Chervonenkis, 1974]. К алгоритмам перцептрона близки по духу алгоритмы Якубовича [Jakubovich, 1966; Jakubovich, 1969]. Алгоритмы Козинца опубликованы в [Kozinec, 1973].

К сожалению, мы не включили в лекцию обширный круг вопросов, идейно близких к проблематике линейного разделения множеств, известный как метод потенциальных функций [Ajzerman et al., 1970]. Если бы мы это сделали, объем лекции, уже и так дольшой, еще более бы увеличился. А эти вопросы настолько интересны, что говорить о них скороговоркой просто невозможно.

Сведение задачи поиска оптимальной гиперплоскости к квадратичной минимизации описано в работах Козинца [Kozinec, 1973] и Валника и Червоненкиса [Varnik and Chervonenkis, 1974]. В русской литературе метод уже многие десятилетия известен как метод обобщенного портрета. В последнее время он стал широко известен на Западе как Support Vector Machine [Boser et al., 1992], [Varnik, 1995], [Varnik, 1998].

Класс фишеровских стратегий, которые сводятся к синтезу линейной дискриминантной функции, введен в работе [Fisher, 1936].

Модификация перцептронных алгоритмов и алгоритмов Козинца для ε -оптимального решения задач Андерсона опубликована в [Schlesinger et al., 1981].

В работе [Franc and Hlaváč, 2001] описано применение алгоритмов Козинца для отыскания гиперплоскости для случая, когда выпуклые оболочки

обучающих множеств пересекаются.

Основные понятия математической теории негладкой оптимизации приводятся по базовой монографии [Shor, 1979; Shor, 1998].

Глава 6

Самообучение

Если к правде святой
Мир дорогу найти не умеет,
Честь безумцу, который навеет
Человечеству сон золотой.

По безбрежным блуждая просторам,
Нам безумец открыл Новый Свет;
Нам безумец дал Новый Завет -
Ибо этот безумец был Богом.
Если б завтра Земли нашей путь
Осветить наше Солнце забыло -
Завтра целый бы мир осветила
Мысль безумца какого-нибудь.

Пьер-Жан Беранже, *"Безумцы"*, 1833, перевод В.Курочкина.

6.1 Вводные замечания об особом характере лекции

Эта лекция посвящена самообучению в распознавании образов, известному также как обучение без учителя, неконтролируемое обучение, а также под другими названиями. Содержание этой лекции мы изложим в несколько иной форме, чем в предыдущих лекциях. Наши предыдущие рассказы были основаны на понятиях, сформировавшихся вне распознавания и уже в своей завершённой форме перешедших в распознавание. Поэтому исходным моментом предыдущих лекций являлась однозначно понимаемая задача, за которой следовал ее анализ и алгоритмы решения. При таком построении лекции оставался за кадром длительный период жизни научной идеи, предшествующий ясно осознанной формулировке задачи. Этот этап научного исследования, как правило, имеет яркую эмоциональную окраску и наполнен драматизмом.

В отличие от фундаментальных понятий, перешедших в распознавание из других областей, идея самообучения родилась внутри самого распознавания. Она еще не достигла такого почтенного возраста, как байесовские или небайесовские подходы. Однако в сравнительно молодом возрасте самообучения есть что-то очень ценное. Мы все еще можем окинуть одним взглядом весь период его существования, увидеть все те рытвины и колдобины, по которым оно должно было пройти, прежде чем воплотиться в четкой постановке задачи и ее решении.

Драматические обстоятельства рождения новых знаний представляют столь привлекательную тему для обсуждения, что мы с трудом удерживаемся от соблазна показать на ряде примеров, как во все времена и на всех континентах научные открытия постигали очень сходные судьбы. Не совсем серьезно, но и не просто в шутку, можно сказать, что открытие Америки как бы вобрало в себя все то общее, что присутствует и при научных открытиях.

- Открытие Америки Колумбом стало возможным благодаря мощной финансовой поддержке проекта.
- Колумб получил заказ на работу только после того, как сформулировал ожидаемый результат, имеющий явную прикладную направленность и понятный заказчикам, - открытие нового пути из Европы в Индию.
- Уверенность Колумба, что он решит важную практическую задачу, основывалась на ложных предпосылках, и ожидаемого результата он не достиг. Наоборот, в процессе выполнения проекта обнаружилось существенное препятствие - целый континент, что убедительно доказывало невозможность достижения поставленной цели.
- Несмотря на явно отрицательные результаты, Колумбу удалось убедить заказчиков, что проект успешно выполнен в полном соответствии с заданием.
- После выполнения проекта никто, включая и исполнителя, не заметил того действительно важного положительного результата, который все же был достигнут. Более того, если бы кто-то и обратил на него внимание, едва ли было бы понято его значение: заказчики не интересовались новыми, до сих пор неизвестными территориями, их интересовали лишь новые пути к уже известным территориям.
- Когда возникла практическая потребность в знаниях, фактически полученных Колумбом, о его результатах уже никто не помнил и пришлось начинать новый проект с новыми заказчиками и новым финансированием.
- Континент, открытый Колумбом, назван вовсе не по его имени, а по имени совсем другого человека, который лишь повторил то, что сделал Колумб намного раньше.
- Сам Колумб до конца своих дней так и не понял, что же он на самом деле открыл, и был уверен (а может, только делал вид?), что открыл новый путь в Индию.
- Сейчас уже всем известно, что Америку открыл не Колумб. Задолго до Колумба о существовании Америки знали скандинавские народы и практически использовали это знание.

Форма этой лекции обусловлена не только ее содержанием, но и нашим желанием показать, насколько каменисты пути к новым знаниям, пусть даже почти незаметным.

6.2 Предварительное неформальное определение самообучения в распознавании

Распознающие алгоритмы можно разделить на три группы:

- алгоритмы, которые работают без обучения;
- обучаемые алгоритмы,
- самообучающиеся алгоритмы.

Алгоритм распознавания относится к той или иной группе в зависимости от того, насколько полны сведения о статистической модели распознаваемого объекта и каким образом восполняется недостаток этих сведений.

Исчерпывающим описанием статистической модели объекта является функция $p_{XK}: X \times K \rightarrow \mathbb{R}$, значение которой $p_{XK}(x, k)$, $x \in X$, $k \in K$, - это совместная вероятность, что объект находится в состоянии k и характеризуется признаком x . Распределение вероятностей p_{XK} служит основой для постановки ряда *байесовских задач* (в зависимости от конкретной функции потерь). Решением байесовской задачи является стратегия $q: X \rightarrow K$, реализованная в определенном, не обучаемом распознающем устройстве. Эта стратегия ориентирована на одну-единственную статистическую модель p_{XK} , но зато для этой модели она оптимальна. Формальные свойства байесовских стратегий рассмотрены нами в лекции 1.

Неполнота сведений о статистической модели выражается в том, что функция p_{XK} неизвестна, а известен лишь класс \mathcal{P} функций, которому p_{XK} принадлежит. Эта неполнота может иметь две принципиально различные причины.

В первом случае недостаток сведений восполнить невозможно, так как функция p_{XK} не фиксирована: она зависит от неслучайного, но меняющегося вмешательства. В этом случае стратегию распознавания невозможно ориентировать на какую-то одну модель p_{XK} . Стратегия должна принимать во внимание весь класс \mathcal{P} моделей. Эти задачи известны как *небайесовские задачи различения сложных гипотез* и являются содержанием лекции 2.

Во втором случае ситуация не столь жесткая. Функция p_{XK} может быть фиксированной, но неизвестной только потому, что объект распознавания еще не был достаточно тщательно изучен. В лекциях 4 и 5 показано, как можно формализовать необходимые в этом случае дополнительные исследования объекта распознавания. Результатом этой формализации являются процедуры обучения распознаванию, основанные на *обучающей выборке* вида $(x_1, k_1), (x_2, k_2), \dots, (x_n, k_n)$. Элементы этой выборки - это взаимно независимые случайные пары (x_i, k_i) , имеющие распределение вероятностей p_{XK} , которое неизвестно.

Напомним, что мы используем термин "обучающая выборка" вместо часто используемых терминов "обучающее множество" или "обучающая последовательность". Исходные для обучения данные не образуют множества, так как некоторые примеры в этих данных могут повторяться много раз. Эти данные не являются последовательностью, так как не имеет значения, в каком порядке расположены отдельные примеры.

Обучающую выборку получают из двух различных источников. Наблюдения x получают из того же источника, что и при собственно распознавании. Это наблюдаемые параметры объекта и их получить намного легче, чем скрытые состояния k . Вопрос, откуда берутся сведения о скрытых состояниях, обходят стороной при теоретических рассуждениях. От него отмахиваются скороговоркой, что эти сведения поступают от учителя. На практике же для их получения нередко требуются сложные и дорогостоящие мероприятия. Состояние k обычно скрыто и, чтобы сделать его наблюдаемым, нужно неординарно вмешаться в объект, нарушить его нормальный режим, а иногда даже разрушить его.

Эти трудности порождают ряд вопросов, выражающих одно и то же. Действительно ли при обучении нужна информация о скрытых параметрах? Нельзя ли за счет одних лишь наблюдений x восполнить недостающие сведения о статистической модели p_{XK} ? Нельзя ли при обучении обойтись без учителя? Эти вопросы породили гипотезу о некотором свойстве распознающих систем, названное самообучением еще до того, как было понято, что же это самообучение обозначает. Понималось лишь, что самообучающаяся распознающая система способна улучшать качество своей работы на основании одних лишь наблюдений, которые ей предъявляют для распознавания, без какого-либо участия учителя в этом процессе. Потребовалось некоторое время, чтобы размытые представления о таких распознающих системах выразились в конкретной формулировке задачи и алгоритмах ее решения.

Мы покажем, как формировалось сегодняшнее представление о возможности такого самосовершенствования распознающей системы. Оно сформировалось в результате слияния трех мощных потоков, развивавшихся независимо друг от друга в различных областях науки, пока они наконец не встретились в распознавании. Этими потоками являются перцептрон, эмпирический байесовский подход и кластерный анализ.

6.3 Самообучение в перцептроне

Несомненно прав был Халмош [Halmos, 1971], когда сказал, что лучшим стимулом для хорошей работы является выполненная кем-то другим плохая работа. Не приходится далеко ходить за примерами, подтверждающими это правило. В распознавании образов, например, значительно больше работ, стимулирующих дальнейшие исследования, чем работ, имеющих вид завершеного не улучшаемого результата. Среди работ, именно таким образом стимулирующих исследования в распознавании, несомненное пер-

венство принадлежит персептрону и всевозможным нейронным сетям. Эти направления в информатике ярко выделяются на общем фоне. В рамках этих направлений смело приступают к решению проблем, которые действительно головокружительно сложны, но результат этих попыток обычно вызывает большое желание сделать это иначе и лучше. Можно долго спорить, является ли это положительной или отрицательной характеристикой нейронных направлений, и так и не придти к общему мнению.

Разногласия и взаимное непонимание сторонников нейронных сетей и сторонников других направлений возникло сразу же после первых публикаций Ф.Розенблатта о персептроне в конце 50-ых [Rosenblatt, 1957]. Поскольку эти разногласия продолжаются уже более 40 лет, то повидимому здесь идет речь не о временных недоразумениях, а о более глубоких причинах. Их рассмотрел сам Ф.Розенблатт в своей монографии [Rosenblatt, 1962], в которой подводились итоги многолетних исследований персептрона.

Согласно Розенблатту, существуют две принципиально различные модели исследований в кибернетике, которые он назвал монотипической и генотипической. Целью исследования в обеих моделях является установление соответствия между определенным устройством (сегодня мы бы сказали, алгоритмом) и задачей, которую он решает. Эта цель достигается существенно различными путями в монотипической и генотипической модели. В монотипической модели исходной является задача, а результат - это алгоритм, который эту задачу решает. Это та модель, которой мы придерживались в наших предыдущих лекциях, и последующие лекции будут выдержаны в этом же стиле.

В генотипической модели это происходит в точности наоборот. Сначала создается некоторое устройство, по крайней мере мысленно, а затем оно исследуется, чтобы понять, какую задачу оно решает. С точки зрения обычной инженерной практики такой подход к делу представляется абсурдным. Инженер, в силу своего многолетнего воспитания, просто неспособен приступить к конструированию устройства раньше, чем он поймет, для решения какой задачи это устройство предназначено. Однако для физиологов и других исследователей живых организмов такой подход является обычным. Вначале, на основании исследования живых существ обнаруживается тот или иной механизм обработки информации, и только после этого осмысливается его назначение, достигается понимание, зачем такой механизм возник в данном исследуемом организме.

Розенблатт считает, что причиной непонимания персептрона является то, что приверженцы каждой из указанных двух моделей действуют так, будто именно их модель единственно возможная. Согласно Розенблатту, персептрон - это наиболее выразительный пример генотипического исследования, и основная причина его непонимания состоит в стремлении дать ему определение, как это принято в монотипической модели. Персептрон - это не устройство для распознавания изображений, не устройство для понимания речи, и вообще, персептрон не определяется как устройство с определенным назначением. Определение персептрона состоит не в фор-

мулироке задачи, которую он решает, а в описании того, как он решает некоторую свою, нам неизвестную задачу. Дадим это определение в том виде, как это сделано в публикациях Розенблатта, но с той степенью детальности, которая нужна в контексте данной лекции.

Персептрон - это устройство, которое состоит из множества элементов, называемых нейронами. Каждый нейрон может находиться в одном из двух состояний: возбужденном и подавленном. Состояние нейрона однозначно определяется изображением на его входе. Пусть множество входных для персептрона изображений состоит из l изображений. Это множество можно разделить на два класса 2^l способами. Один из классов называется положительным, а второй - отрицательным. Существует некто *учитель*, кто из 2^l классификаций выбирает одну и фиксирует ее для дальнейшего рассмотрения. Эта классификация называется *классификацией учителя* или *правильной классификацией*. Наряду с правильной классификацией существует классификация, выполняемая персептроном. Она, вообще говоря, не обязательно совпадает с правильной. Она выполняется следующим образом. Каждый нейрон характеризуется численным параметром, который называется его весом. Изображение на входе персептрона считается положительным или отрицательным в зависимости от знака суммы всех нейронов, возбужденных данным изображением.

Таким образом, классификация персептрона зависит от весов нейронов. Эти веса меняются с помощью специальных процедур *поощрения и наказания* нейронов. Поощрение и наказание - это, соответственно, увеличение или уменьшение веса нейрона на постоянную величину Δ . При наблюдении определенного изображения отдельный нейрон поощряется или наказывается в зависимости от того

- возбужден или подавлен этот нейрон данным изображением;
- к какому классу относит учитель текущее изображение;
- к какому классу относит текущее изображение сам персептрон.

Различные сочетания этих условий определяют следующие три процедуры изменения весов нейронов. Процесс изменения этих весов и есть обучение персептрона.

Algorithm 6.1 Обучение по классификации учителя

1. Если учитель относит текущее изображение к положительному классу, то все возбужденные нейроны поощряются.
 2. Если учитель относит текущее изображение к отрицательному классу, то все возбужденные нейроны наказываются.
-

Algorithm 6.2 Обучение по классификации учителя и по классификации персептрона

1. Если персептрон классифицирует текущее изображение так же, как и учитель, то веса нейронов не меняются.

2. Если персептрон относит текущее изображение к положительному классу, а учитель - к отрицательному, то все возбужденные нейроны наказываются.
3. Если персептрон относит текущее изображение к отрицательному классу, а учитель - к положительному, все возбужденные нейроны поощряются.

Algorithm 6.3 Самообучение - обучение по классификации персептрона

1. Если персептрон относит текущее изображение к положительному классу, все возбужденные нейроны поощряются.
2. Если персептрон относит текущее изображение к отрицательному классу, все возбужденные нейроны наказываются.

Как видно, алгоритм 6.3 отличается от алгоритма 6.1 только тем, что роль учителя в нем выполняет сам персептрон, и это различие существенно. Чтобы реализовать алгоритмы 6.1 или 6.2, персептрон должен иметь два входа: первый - для наблюдения изображения, а второй - для получения указаний учителя. Алгоритм 6.3 можно реализовать без каких-либо контактов с учителем. Поэтому, если первые два алгоритма 6.1 и 6.2 были названы алгоритмами обучения персептрона, то третьему алгоритму 6.3 естественно присвоилось название самообучение.

Гипотеза Розенблатта состояла в том, что персептрон, управляемый каждым из приведенных алгоритмов, достигает состояния, при котором он правильно классифицирует все входные изображения.

Сейчас уже трудно описать восторг, с которым были приняты первые публикации Розенблатта об этих гипотезах. В персептроне все представлялось чарующая простота алгоритмов, необычные для вычислительной техники того времени термины "нейрон", "обучение" и даже само слово "персептрон". Возникла пленительная атмосфера ожидания, что вот-вот откроется дверь, за которой находится нечто потрясающее, хотя и неизвестно что именно. Ведь если бы гипотеза Розенблатта оправдалась, это избавило бы от необходимости трудоемкого и тщательного изучения объектов, которые надо распознавать. Достаточно было бы построить персептрон, показать ему в нескольких примерах, как он должен работать, а дальше все пошло бы само собой. Более того, даже не надо было бы приводить примеры правильной классификации: учитывая способность персептрона к самообучению, достаточно было бы персептрону предъявить только некоторые изображения, чтобы он сам нашел, как их следует классифицировать.

И как почти всегда случается, только малая часть этой замечательной сказки осуществилась. Эта часть выражена в теореме Новикова, формулировка и доказательство которой приведены в предыдущей лекции. Теорема Новикова подтверждает предположение Розенблатта о работоспособности алгоритма 6.2. Отметим однако, что теорема Новикова существенно конкретизует, то есть сужает область компетентности алгоритма 6.2. Алгоритм приходит к правильной классификации, конечно же, только при условии,

что набор весов нейронов, обеспечивающий правильную классификацию, вообще существует.

Легко показать, что гипотеза Розенблатта о работоспособности алгоритма 6.1 ошибочна. Хотя это и не входит в основную цель данной лекции, покажем ошибочность этой гипотезы на конкретном примере.

Пример 6.1 Пример неработоспособности алгоритма 6.1. Пусть персептрон состоит из трех нейронов. Каждое положительное изображение возбуждает все три нейрона, а каждое отрицательное изображение возбуждает либо второй нейрон, либо третий, но не оба одновременно. Эта ситуация благоприятна для персептрона, потому что существует набор весов w_1, w_2, w_3 , при которых персептрон безошибочно классифицирует изображения. Например, это веса $w_1 = 3, w_2 = -1, w_3 = -1$. Сумма весов, возбужденных положительным изображением, равна $+1$, а возбужденных отрицательным изображением, равна -1 .

Предположим, что обучение персептрона началось именно с этих весов, то есть в состоянии, при котором входные изображения распознаются правильно. Посмотрим, что получится, если обучить персептрон на n положительных и n отрицательных изображениях. В процессе обучения все нейроны будут n раз поощрены, и кроме того, второй нейрон будет n_2 раз наказан, а третий нейрон n_3 раз наказан, $n = n_2 + n_3$. Веса нейронов станут равны

$$\begin{aligned}w_1 &= 3 + n \Delta, \\w_2 &= -1 + \Delta (n - n_2), \\w_3 &= -1 + \Delta (n - n_3).\end{aligned}$$

Сумма весов нейронов, возбужденных положительным изображением, будет равна $1 + 2n \Delta$, то есть останется положительной при любом n . Таким образом, коррекция весов нейронов не разрушила правильное распознавание положительных изображений. Однако сумма весов нейронов, возбужденных отрицательным изображением, будет либо $-1 + \Delta(n - n_2)$, либо $-1 + \Delta(n - n_3)$. Сумма этих двух чисел равна $-2 + n\Delta$, то есть положительна при достаточно большом n . Отсюда следует, что по крайней мере одно из чисел $-1 + \Delta(n - n_2)$ или $-1 + \Delta(n - n_3)$ положительно. Таким образом, обучение персептрона по алгоритму 6.1 привело к тому, что персептрон перешел из состояния, в котором он правильно распознавал все изображения, в состояние, в котором он распознает правильно не все изображения. ▲

Гипотеза Розенблатта оправдалась только относительно алгоритма 6.2 и оказалась ложной для алгоритма 6.1. Что касается гипотезы Розенблатта о самообучении в персептроне, то она высказана настолько нечетко, что о его истинности или ложности вообще невозможно судить. Покажем это на предельно упрощенном примере.

Утверждение $2 \cdot 2 = 4$ истинно, а утверждение $2 \cdot 2 = 5$ ложно. Но утверждение $2 \cdot 2 = \frac{\sqrt{c}}{2}$ не является ни ложным, ни истинным. Это вообще

не утверждение. Очевидно, что оно не может считаться истинным. В то же время нельзя его признать и ложным, так как в этом случае пришлось бы признать истинность утверждения $2 \cdot 2 \neq \frac{\sqrt{c}}{2}$. Утверждение $2 \cdot 2 = \frac{\sqrt{c}}{2}$ - это просто бессмыслица, потому что в нем что-то не дописано.

Гипотеза Розенблатта о самообучении в персептроне не полностью сформулирована, так как в ней ничего не сказано, какую из 2^l классификаций входных изображений следует считать правильной. О правильности этой гипотезы можно говорить только после ее надлежащей конкретизации.

Очевидно, что если правильной считать классификацию учителя, то гипотеза о самообучении ложна. Классификация, которую дает персептрон в результате самообучения, зависит только от множества входных изображений. Следовательно, фиксированное множество изображений будет классифицироваться персептроном всегда одним и тем же способом. Учитель же может классифицировать фиксированное множество по разному. Нечто постоянное не может быть тождественным тому, что меняется.

Если же правильной считается классификация, достигнутая персептроном, то гипотеза о самообучении верна. Однако после такой конкретизации эта гипотеза получает форму: персептрон достигает правильную классификацию, а правильной классификацией считается та, которую достигает персептрон. Это - безукоризненно правильное утверждение с пустым содержанием типа $2 \cdot 2 = 2 \cdot 2$.

Таким образом, надо было как-то вразумительно сформулировать свойства классификации, получаемой самообучающимся персептроном. Говоря другими словами, надо было в конечном итоге сформулировать задачу, которую решает предложенный Розенблаттом алгоритм самообучения. И хотя такой ретроспективный анализ алгоритмов вызывает чисто эстетическое неприятие людей с техническим образованием, такая практика обычна для генотипических исследований, ярким примером которых является персептрон.

Розенблатт выполнил ряд экспериментов, чтобы выяснить свойства классификации, получаемой в процессе самообучения в персептроне. Результат этих экспериментов был ошеломляюще плохим. Персептрон из любого состояния упорно переходил в состояние, при котором он все изображения относил к какому-то одному классу, положительному или отрицательному. Такая классификация не является правильной ни с какой точки зрения. И как это часто бывает, после того, как этот отрицательный результат был получен, сразу стало понятно, что результат и не мог быть иным. Если персептрон достиг состояния, при котором все изображения относятся к одному классу, пусть для определенности это - положительный класс, то никакое последующее самообучение не сможет изменить это положение. При обработке очередного изображения веса невозбужденных нейронов не меняются, а веса возбужденных увеличиваются. Таким образом, суммарный вес нейронов, возбужденных тем или иным изображением, не убывает, и следовательно, остается положительным.

Таким разочарованием закончился первый акт в истории становления самообучения, как направления в современном распознавании образов. Од-

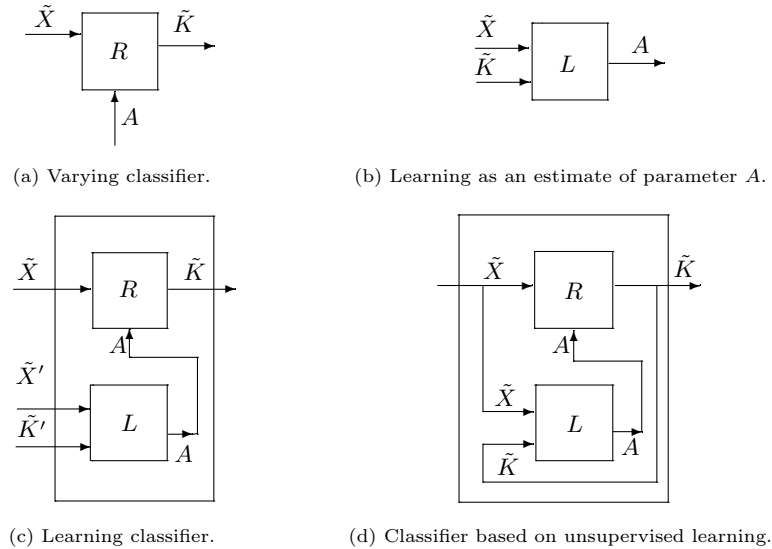


Figure 6.1 Different classifier configurations.

нако, оценивая содержание этого этапа с точки зрения сегодняшнего дня, мы видим здесь идею, которая стала одной из компонент современного самообучения. Это идея об определенном формате алгоритмов самообучения, который мы сейчас сформулируем. Однако эта идея была тщательно скрыта за нейронными, физиологическими, а вернее, псевдофизиологическими рассуждениями. Создается впечатление, будто это делалось специально, чтобы сделать непонятными те простые идеи, которые в конечном итоге оказались плодотворными. Мы очистим эти идеи от всего ненужного в контексте данной лекции и сформулируем определенный класс алгоритмов, которые в знак глубокого уважения к автору персептрона назовем розенблаттовскими алгоритмами.

Пусть X и K - два множества, элементами которых являются наблюдения x и результаты распознавания k . Функция $q: X \rightarrow K$ - это стратегия распознавания, а Q - определенное параметрически заданное множество стратегий. Пусть некое устройство (см. рис. 6.1a) предназначено для реализации любой стратегии из Q . Конкретная стратегия, которую устройство должно в данный момент реализовать, определяется значением параметра A , который поступает на вход, специально для этого предназначенный. Наблюдения же поступают на другой вход. Последовательность \tilde{X} наблюдений (x_1, x_2, \dots, x_n) , $x_i \in X$, преобразуется устройством в последовательность \tilde{K} результатов распознавания (k_1, k_2, \dots, k_n) , $k_i \in K$. Устройство выполняет то или другое преобразование в зависимости от значения параметра A . Это устройство мы назовем *переменным распознающим устройством* и обозначим R . Оно реализует функцию двух переменных \tilde{X} , A так, что $\tilde{K} = R(\tilde{X}, A)$.

Пусть некое другое устройство преобразует каждую пару \tilde{X} , \tilde{K} в значе-

ние параметра A (см. рис. 6.1b). Обозначим L функцию, которую реализует это устройство так, что $A = L(\tilde{X}, \tilde{K})$, и назовем его *устройством обучения*. Соединим эти два устройства, как показано на рис. 6.1c, и получим устройство, которое назовем *обучаемым устройством распознавания*. На это устройство поступают три последовательности \tilde{X} , \tilde{X}' и \tilde{K}' , которые преобразуются в выходную последовательность \tilde{K} . Это преобразование определяется двумя алгоритмами: *алгоритмом обучения* L и *алгоритмом переменного распознавания* R . Первый вычисляет значение параметра A на основании последовательностей \tilde{X}' и \tilde{K}' . Полученное значение A подается на вход распознающего устройства R и этим конкретизируется, как именно последовательность \tilde{X} должна преобразоваться в последовательность \tilde{K} , то есть конкретизируется алгоритм собственно распознавания.

Это чересчур общее описание ни в коем случае не претендует на определение обучаемого распознающего устройства. Здесь только устанавливается, из каких частей оно состоит, как они взаимодействуют и какими данными они оперируют. Понятно, что алгоритмы R и L должны быть в определенном смысле правильно устроены так, чтобы результат \tilde{K} распознавания последовательности \tilde{X} не сильно отличался (опять-таки, в определенном смысле) от результата \tilde{K}' распознавания последовательности \tilde{X}' . Точные формулировки этих требований и их анализ составляли содержание лекции 4.

Однако, и не конкретизуя, как должны выглядеть алгоритмы распознавания и обучения, можно выразить основную идею Розенблатта о самообучении, которая была безнадежно скрыта за нейрональными дробями. Предположим, что обучаемое распознающее устройство уже построено так, как представлено на рис. 6.1, и составлено оно из правильно сконструированных алгоритмов R и L . В таком случае *самообучающееся распознающее устройство* (алгоритм самообучения) можно определить, как устройство, которое состоит из тех же частей R и L , но соединенных по другому, см. рис. 6.1d. Эту новую конфигурацию надо понимать следующим образом.

При определенном начальном значении A_0 параметра A следует распознать последовательность \tilde{X} , то есть получить последовательность \tilde{K}_0 . Последовательность \tilde{K}_0 используется затем совместно с последовательностью \tilde{X} так, будто она вычислена не самим алгоритмом, а получена от учителя. Это значит, что вычисляется новое значение $A_1 = L(\tilde{X}, \tilde{K}_0)$. При этом новом значении параметра A опять распознается входная последовательность, то есть вычисляется $\tilde{K}_1 = R(\tilde{X}, A_1)$. Эта процедура повторяется и дальше так, что на основании результатов распознавания \tilde{K}_{t-1} получают новые результаты $\tilde{K}_t = R(\tilde{X}, L(\tilde{X}, \tilde{K}_{t-1}))$ или, что тоже самое, на основании значения A_{t-1} параметра A вычисляется новое значение $A_t = L(\tilde{X}, R(\tilde{X}, A_{t-1}))$.

Алгоритмы такого вида сформулированы с точностью до алгоритма распознавания R и алгоритма обучения L , то есть очень неточно. Обратной, положительной стороной этой неточности является то, что эти алгоритмы образуют очень обширный класс. Если оценивать вклад Розенблатта в теорию самообучения с точки зрения сегодняшнего дня, то он состоит

в гипотезе, что алгоритмы такого вида пригодны для решения некоторой интуитивно понимаемой, но еще не сформулированной задачи.

В то же время в непосредственном соседстве с распознаванием образов, в соседней научной дисциплине были сформулированы задачи, которые длительное время оставались без должного решения. Только после того, как эти задачи вошли в сферу распознавания, в этом новом для себя окружении они нашли приемлемые решения в виде алгоритмов, весьма близких по своему характеру к алгоритмам, предсказанным Розенблаттом. Отложим описание этой встречи до раздела 6.6.

6.4 Эмпирический байесовский подход Роббинса

Эмпирический байесовский подход Г.Роббинса является другим мощным потоком, который, влившись в распознавание, составил идейную основу современного самообучения. Эмпирический байесовский подход возник в результате стремления заполнить пробел между байесовскими и небайесовскими методами, которым мы посвятили лекции 1 и 2. Согласно Роббинсу, методы и задачи статистических решений состоят не из двух классов, байесовских и небайесовских, а из следующих трех классов.

Байесовский класс. Областью применения байесовских методов являются ситуации, при которых как состояние k , так и наблюдение x являются случайными, причем известны априорные вероятности $p_K(k)$ состояний и условные вероятности $p_{X|K}(x|k)$ наблюдений x при условии, что объект находится в состоянии k .

Небайесовский класс. Область применения небайесовских методов составляют ситуации, когда известны только условные вероятности $p_{X|K}(x|k)$. Состояние k является переменным, но оно не является случайным, и следовательно, априорные вероятности $p_K(k)$ не просто неизвестны, они не существуют. В лекции 2 мы рассмотрели ряд небайесовских задач. Далее в этом подразделе мы рассмотрим только частный случай небайесовских задач в их минимаксной постановке, потому что именно в сравнении с этими задачами определится область применения эмпирических байесовских методов.

Класс задач Роббинса. Наконец, областью применения методов Роббинса являются ситуации, когда случайны как состояния k , так и наблюдения x . Это их роднит с байесовскими методами. Однако, в отличие от байесовских ситуаций известны лишь условные вероятности $p_{X|K}(x|k)$, и это их роднит с небайесовскими методами. Априорные вероятности $p_K(k)$ состояний существуют, но они неизвестны. Для принятия решений в этих ситуациях Роббинс предлагает особые методы, которые должны заполнить пробел между байесовскими и небайесовскими подходами.

Основную идею подхода Роббинс объясняет на следующем простом примере.

Пусть X - множество вещественных чисел, множество K состоит из двух состояний, $K = \{1, 2\}$, а признак x - одномерная гауссова случайная величина с дисперсией $\sigma^2 = 1$ и математическим ожиданием, которое зависит

от состояния k . Если объект находится в первом состоянии, математическое ожидание равно 1, а если во втором, то -1 . Пусть $q: X \rightarrow \{1, 2\}$ - стратегия, а α - априорная вероятность первого состояния. Пусть $R(q, \alpha)$ - вероятность ошибочного решения о состоянии k . Эта вероятность зависит от вероятности α и от стратегии q и равна

$$R(q, \alpha) = \alpha \int_{\{x|q(x)=2\}} \frac{1}{\sqrt{2\pi}} e^{-\frac{1}{2}(x-1)^2} dx + (1 - \alpha) \int_{\{x|q(x)=1\}} \frac{1}{\sqrt{2\pi}} e^{-\frac{1}{2}(x+1)^2} dx .$$

Обозначим $q(\alpha)$ байесовскую стратегию, которая минимизировала бы вероятность ошибочного решения при известной вероятности α . Это значит, что

$$q(\alpha) = \operatorname{argmin}_{q' \in Q} R(q', \alpha) ,$$

где Q - множество всех возможных стратегий.

Пусть q^* - стратегия, которая принимает решение в пользу первого состояния при $x \geq 0$ и в пользу второго состояния в противном случае. Хотя это и достаточно очевидно, покажем, что стратегия q^* является решением минимаксной задачи. Вероятность $R(q^*, \alpha)$ ошибочного решения равна

$$\begin{aligned} R(q^*, \alpha) &= \alpha \int_{-\infty}^0 \frac{1}{\sqrt{2\pi}} e^{-\frac{1}{2}(x-1)^2} dx + (1 - \alpha) \int_0^{\infty} \frac{1}{\sqrt{2\pi}} e^{-\frac{1}{2}(x+1)^2} dx \\ &= \alpha \int_{-\infty}^{-1} \frac{1}{\sqrt{2\pi}} e^{-\frac{1}{2}x^2} dx + (1 - \alpha) \int_1^{\infty} \frac{1}{\sqrt{2\pi}} e^{-\frac{1}{2}x^2} dx . \end{aligned}$$

Так как интегралы

$$\int_{-\infty}^{-1} \frac{1}{\sqrt{2\pi}} e^{-\frac{1}{2}x^2} dx \quad \text{и} \quad \int_1^{\infty} \frac{1}{\sqrt{2\pi}} e^{-\frac{1}{2}x^2} dx$$

равны (они равны приблизительно 0.16), число $R(q^*, \alpha)$ не зависит от вероятности α , и таким образом,

$$R(q^*, \alpha) = \max_{\alpha} R(q^*, \alpha) \cong 0.16 . \quad (6.1)$$

С другой стороны, стратегия q^* совпадает с байесовской стратегией $q(0.5)$, построенной для случая априори равновероятных состояний. Это значит, что для любой стратегии q , не равной q^* , справедливо неравенство

$$R(q, 0.5) > R(q^*, 0.5), \quad (6.2)$$

которое в рассматриваемом примере выполняется строго. Объединяя равенство (6.1) и неравенство (6.2), получаем

$$\max_{\alpha} R(q, \alpha) \geq R(q, 0.5) > R(q^*, 0.5) = \max_{\alpha} R(q^*, \alpha),$$

то есть

$$\max_{\alpha} R(q, \alpha) > \max_{\alpha} R(q^*, \alpha) \approx 0.16.$$

Таким образом, стратегия q^* обладает следующими двумя свойствами.

1. Стратегия q^* принимает ошибочное решение о состоянии k объекта с вероятностью 0.16 независимо от априорных вероятностей состояний.
2. Любая другая стратегия не обладает таким свойством. Более того, для любой другой стратегии существуют такие априорные вероятности состояний, при которых вероятность ошибочного решения больше, чем 0.16.

В этом смысле стратегия q^* является наилучшей стратегией для случая, когда априорные вероятности состояний могут быть любыми. Однако, когда априорные вероятности фиксированы и известны, можно построить стратегию $q(\alpha)$ следующего вида. Стратегия принимает решение в пользу первого состояния, когда

$$\frac{\alpha}{\sqrt{2\pi}} e^{-\frac{1}{2}(x-1)^2} \geq \frac{1-\alpha}{\sqrt{2\pi}} e^{-\frac{1}{2}(x+1)^2},$$

или, что то же самое, когда

$$x \geq \frac{1}{2} \ln \frac{1-\alpha}{\alpha}. \quad (6.3)$$

В противном случае принимается решение в пользу второго состояния.

При использовании этой стратегии вероятность $R(q(\alpha), \alpha)$ ошибки равна 0.16 только при $\alpha = 0.5$. При любом другом значении α она меньше. На рис. 6.2 показано, как вероятности $R(q(\alpha), \alpha)$ и $R(q^*, \alpha)$ зависят от априорной вероятности α . Стратегия $q(\alpha)$ основана на знании вероятности α . Минимаксная стратегия q^* не использует эти знания. Рис. 6.2 еще раз иллюстрирует тот факт, что качество распознавания, которое обеспечивает стратегия q^* , не зависит от α и выдерживается на постоянном уровне 0.16. Однако это уровень, который достигается байесовской стратегией в наихудших условиях.

Указанное взаимоотношение минимаксной и байесовской стратегий оставляют место для глубокого неудовлетворения. Почему, собственно, если что-то неизвестно, то нужно настраиваться на наихудший вариант того, что неизвестно? Ведь действительность не всегда реализуется в своем наименее благоприятном виде. Стратегия же q^* ведет себя так, словно в действительности осуществляются только самые худшие варианты. Можно ли найти такой способ классификации наблюдений x , при котором не требовались бы знания об априорных вероятностях, но вероятность ошибочных

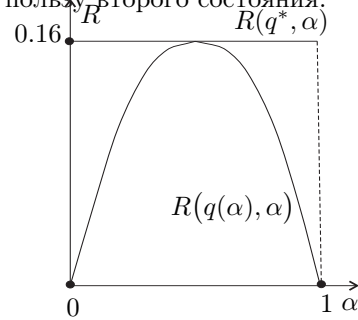


Figure 6.2 The dependence of a probability of a wrong decision R on the *a priori* probability α for the Bayesian and minimax strategy.

решений была бы равной 0.16 только в худшей ситуации? А в ситуации, которая не самая плохая, и результаты распознавания были бы лучше, чем в самой плохой ситуации? Все это лишь различные варианты одного и того же вопроса: можно ли без знания априорных вероятностей состояний построить стратегию, которая была бы не хуже, чем стратегия, построенная с учетом этих знаний?

В какой бы форме не ставился этот вопрос, на него возможен лишь отрицательный ответ. Байесовская стратегия $q(\alpha)$ существенно зависит от априорных вероятностей состояний. Эта зависимость явно выражается формулой (6.3). Следовательно, стратегия, которая не зависит от априорных вероятностей, не может совпадать с байесовской стратегией, которая от этих вероятностей зависит. Заметим, что мы пришли к этому отрицательному выводу на основании таких же рассуждений, которые привели нас к выводу о невозможности самообучения в перцептроне. В обоих случаях отрицательный ответ основывался на том очевидном факте, что нечто постоянное не может совпадать с чем-то переменным.

Значение прорыва, который осуществил Роббинс на байесовском фронте (мы цитируем восторженный отзыв Неймана [Neuman, 1962]), состоит в том, что Роббинс и не пытался ответить на вопрос, где отрицательный ответ очевиден. Вместо этого он разумно изменил сам вопрос так, что отрицательный ответ на него стал далеко не очевиден.

С точки зрения реального использования стратегии вполне естественно предположить, что эта стратегия будет использована не один раз, а много раз для распознавания n объектов в моменты времени $i = 1, 2, \dots, n$. Предположим также, что состояния объекта в эти моменты случайно и последовательность k_1, k_2, \dots, k_n состоит из взаимно независимых элементов k_i . При этом вероятность события $k_i = 1$ равна фиксированной величине α во все моменты времени, однако эта величина неизвестна. Предположим далее, что решение о состоянии k_1 не обязательно должно приниматься немедленно в момент наблюдения x_1 . Это решение можно задержать до того, как будут получены все остальные наблюдения x_2, x_3, \dots, x_n . Если допускается такой режим распознавания, то решение о состоянии k_1 можно принимать не на основании единственного наблюдения x_1 , а на основании всей последовательности наблюдений x_1, x_2, \dots, x_n . Таким же образом, на основании всей последовательности наблюдений, можно принимать решение о состояниях k_2, k_3, k_4 и т.д. Это значит, что в этом случае следует говорить не о стратегии вида $X \rightarrow K$, а о стратегии более общего вида $X^n \rightarrow K^n$.

Повторим теперь прежний вопрос, но в том разумно измененном виде, как это сделал Роббинс: можно ли, не зная априорных вероятностей состояний, построить стратегию вида $X^n \rightarrow \{1, 2\}^n$, которая была бы не хуже, чем байесовская стратегия вида $X \rightarrow \{1, 2\}$, построенная на основе знания этих априорных вероятностей? Этот вопрос уже далеко не тривиальный, и совсем не очевидно, что ответ на него отрицательный.

Роббинс указывает конкретную стратегию, которая по крайней мере в рассматриваемом примере и по крайней мере при больших n служит поло-

жительным ответом на поставленный вопрос. Стратегия принимает решение в пользу первого состояния, когда

$$x_i \geq \frac{1}{2} \ln \frac{n - \sum_{i=1}^n x_i}{n + \sum_{i=1}^n x_i}, \quad (6.4)$$

и в пользу второго состояния в противном случае. В рассматриваемом примере, при $n \rightarrow \infty$, стратегия (6.4) становится сколь угодно близкой к стратегии (6.3) в том смысле, что случайная величина в правой части (6.4) стремится к числу в правой части (6.3). Действительно,

$$\lim_{n \rightarrow \infty} \frac{n - \sum_{i=1}^n x_i}{n + \sum_{i=1}^n x_i} = \lim_{n \rightarrow \infty} \frac{1 - \frac{1}{n} \sum_{i=1}^n x_i}{1 + \frac{1}{n} \sum_{i=1}^n x_i} = \frac{1 - \lim_{n \rightarrow \infty} \frac{1}{n} \sum_{i=1}^n x_i}{1 + \lim_{n \rightarrow \infty} \frac{1}{n} \sum_{i=1}^n x_i}. \quad (6.5)$$

С учетом того, что $\lim_{n \rightarrow \infty} \frac{1}{n} \sum_{i=1}^n x_i$ есть математическое ожидание $E(x)$ случайной величины x , а математические ожидания величины x для объектов в первом и втором состоянии равны соответственно 1 или -1 , справедливо, что $E(x) = \alpha \cdot 1 + (1 - \alpha)(-1) = 2\alpha - 1$. Продолжая прерванную цепочку (6.5), получаем

$$\lim_{n \rightarrow \infty} \frac{n - \sum_{i=1}^n x_i}{n + \sum_{i=1}^n x_i} = \frac{1 - (2\alpha - 1)}{1 + (2\alpha - 1)} = \frac{2(1 - \alpha)}{2\alpha} = \frac{1 - \alpha}{\alpha}.$$

Роббинс трактует стратегию (6.4) как результат двухкратного использования последовательности наблюдений, которая содержит в себе информацию двух различных типов. С одной стороны, каждое наблюдение x_i есть случайная величина, зависящая от состояния k_i объекта в i -ый момент. Следовательно, в наблюдении x_i содержится информация о состоянии k_i . С другой стороны, последовательность x_1, x_2, \dots, x_n в целом есть случайная выборка из генеральной совокупности с распределением вероятностей p_X . Это распределение вероятностей зависит от априорных вероятностей $p_K(1) = \alpha$ и $p_K(2) = 1 - \alpha$, то есть от параметра α , так, что

$$p_X(x) = \frac{p_K(1)}{\sqrt{2\pi}} e^{-\frac{1}{2}(x-1)^2} + \frac{p_K(2)}{\sqrt{2\pi}} e^{-\frac{1}{2}(x+1)^2}.$$

В силу этого последовательность x_1, x_2, \dots, x_n содержит в себе и информацию о неизвестных априорных вероятностях. Последовательность наблюдений обрабатывается в два прохода:

1. На первом проходе оцениваются априорные вероятности, которые заранее неизвестны.
2. Эти более или менее точные оценки используются для решения о состояниях k_1, k_2, \dots, k_n с помощью байесовской стратегии так, будто эти оценки совпадают с действительными значениями.

Стратегия (6.4) получена на основании непрямого оценивания априорных вероятностей. Эта оценка основана на том, что математическое ожидание наблюдения x однозначно определяет неизвестные вероятности так, что

$$p_K(1) = \frac{1}{2} (1 + E(x)), \quad p_K(2) = \frac{1}{2} (1 - E(x)),$$

а число $\frac{1}{n} \sum_{i=1}^n x_i$ при достаточно больших n есть приемлемая оценка математического ожидания $E(x)$.

Такая непрямая оценка не является наилучшей. Роббинс формулирует задачу отыскания таких значений $p_K^*(k)$, $k \in K$, которые максимизируют вероятность выборки x_1, x_2, \dots, x_n , то есть, формулирует задачу, как наиболее правдоподобное оценивание

$$\begin{aligned} (p_K^*(k), k \in K) &= \operatorname{argmax}_{(p_K(k)|k \in K)} \prod_{i=1}^n \sum_{k \in K} p_K(k) p_{X|K}(x_i | k) \\ &= \operatorname{argmax}_{(p_K(k)|k \in K)} \sum_{i=1}^n \log \sum_{k \in K} p_K(k) p_{X|K}(x_i | k). \end{aligned} \quad (6.6)$$

Вычислительные алгоритмы решения задачи (6.6) длительное время были неизвестны и в этом смысле задача Роббинса оставалась нерешенной. Только в результате ее погружения в распознавание образов были найдены изящные алгоритмы для ее решения, очень сходные с розенблаттовскими. Мы приведем эти решения в конце лекции.

Однако до этого мы хотим обратить внимание, что задачи Роббинса совсем не следует понимать только в том узком смысле, как это показано здесь. Рассмотренный нами частный случай с одномерными гауссовыми случайными величинами есть только пример, иллюстрирующий основную идею эмпирического байесовского подхода. Уже сама формулировка задачи (6.6) есть нечто более общее, чем описанный нами (и Роббинсом) пример. Подход Роббинса, понимаемый в достаточно широком смысле, включает в себя ситуации, которые, формально говоря, выходят за рамки формулировки (6.6). Мы приведем пример такой ситуации, повторим для нее все те соображения, которые привели Роббинса к задаче (6.6), а затем сформулируем задачу, которая очерчивает рамки эмпирического байесовского подхода во всей его общности.

Вернемся к примеру с одномерными гауссовыми случайными величинами. На сей раз предположим, что априорные вероятности обоих состояний известны и равны 0.5. Однако неизвестны условные математические ожидания наблюдений при условии, что объект находится в том или ином состоянии. Известно только, что:

- Если объект находится в первом состоянии, то математическое ожидание величины x находится в интервале от 1 до 10;
- Если объект находится во втором состоянии, то математическое ожидание величины x находится в интервале от -1 до -10 .

В этом случае наилучшая в минимаксном смысле стратегия принимает решение $q^*(x) = 1$ при $x \geq 0$ и $q^*(x) = 2$ в противном случае. Эта стратегия ориентирована на худшую ситуацию, когда первое математическое ожидание равно 1, а второе равно -1 . Однако, действительная ситуация не обязательно должна оказаться наихудшей. Первое математическое ожидание может равняться, например, 1, а второе - быть равным -10 . Для этого случая можно было бы построить значительно лучшую стратегию

q^* . Стратегию, приспособливающуюся к различным ситуациям, можно реализовать тогда, когда нет необходимости распознавать наблюдение x немедленно после его поступления, а можно распознавание отложить до того, как накопится достаточно большое количество наблюдений. Эти наблюдения используются вначале для более точной оценки математических ожиданий, и только после этого эти наблюдения собственно распознаются.

Можно продолжить ряд все более и более сложных примеров задач, подход к решению которых основан на тех же идеях Роббинса, так как именно в этих идеях состоял "прорыв на байесовском фронте". Все эти задачи можно выразить в следующей *обобщенной форме*.

Пусть X и K - два множества. Их декартово произведение образует множество значений случайной пары (x, k) . Вероятность пары (x, k) определяется функцией $p_{XK}: X \times K \rightarrow \mathbb{R}$. Функция p_{XK} неизвестна, но известно множество \mathcal{P} , содержащее функцию p_{XK} . Кроме того, известна выборка x_1, x_2, \dots, x_n случайных и взаимно независимых наблюдений x . Вероятность наблюдения x есть сумма

$$\sum_{k \in K} p_{XK}(x, k).$$

Задача состоит в нахождении такого распределения вероятностей p_{XK}^* , которое принадлежит множеству \mathcal{P} и максимизирует вероятность выборки x_1, x_2, \dots, x_n . Это значит, что

$$p_{XK}^* = \operatorname{argmax}_{p_{XK} \in \mathcal{P}} \sum_{i=1}^n \log \sum_{k \in K} p_{XK}(x_i, k). \quad (6.7)$$

Решение этой задачи позволило бы конструировать распознающие системы, которые улучшают свою стратегию распознавания в процессе их нормальной эксплуатации, анализируя при этом только наблюдения, предъявленные для распознавания без каких-либо указаний учителя о том, как эти наблюдения следует распознавать. Таким образом, эта задача формализует размытые представления о самообучении, высказанные на интуитивном уровне автором перцептрона. Мы покажем в разделе 6.6, что и алгоритмы решения этой задачи весьма сходны с алгоритмами, предсказанными Розенблаттом. Однако, до долгожданной встречи алгоритмов Розенблатта с задачами Роббинса эти алгоритмы жили своей самостоятельной жизнью, в течение которой они соприкоснулись с давно известными задачами кластеризации. В следующем разделе мы покажем плодотворные результаты этой встречи.

6.5 Квадратичная кластеризация и формулировка общей задачи кластеризации

В конце раздела 6.3 мы описали класс алгоритмов, которые мы назвали розенблаттовскими. Мы не стремились к точному формальному описанию этих алгоритмов, а описали их наглядно с помощью схем на рис. 6.1. Алгоритмы самообучения, как и алгоритмы обучения с учителем, состоят из

двух частей: обучения и распознавания. Однако взаимодействие этих двух частей при обучении и самообучении существенно разное. При обучении на вход алгоритма поступают сведения от учителя о правильных результатах распознавания сигналов из обучающей выборки. При самообучении информация от учителя о правильной классификации заменяется результатами распознавания, полученными самой распознающей системой. С подачи Розенблатта сформировалась гипотеза, что если результаты собственного распознавания обрабатывать так, будто они получены от учителя, то это может привести к некоторым разумным результатам.

Эта гипотеза была настолько привлекательна, что ее не смогли поколебать даже явно отрицательные результаты экспериментов с самообучением в персептроне. Эти результаты могли объясняться еще и тем, что эксперименты выполнялись с теми алгоритмами персептрона, которые оказались неработоспособны даже в режиме обучения с учителем. В силу этого оставались надежды, что если в качестве составных частей самообучения использовать в определенном смысле "правильные" алгоритмы обучения и распознавания, то и результат самообучения окажется "правильным". Сейчас, по истечении достаточно длительного времени, нужно отдать должное этому оптимизму, так как он в конечном итоге привел к знаменитому алгоритму, известному как ИЗОДАТА. Опишем этот алгоритм.

Рассмотрим следующую простейшую статистическую модель распознаваемого объекта. Пусть распознаваемый объект может находиться в одном из двух состояний с равными вероятностями, то есть $K = \{1, 2\}$, $p_K(1) = p_K(2) = 0.5$. Пусть x - n -мерный вектор признаков. Если объект находится в первом состоянии, то x - это случайный гауссов вектор с математическим ожиданием μ_1 , причем компоненты этого вектора взаимно независимы и дисперсия каждой компоненты равна 1. Те же свойства имеет вектор x при условии, что объект находится во втором состоянии, с тем лишь различием, что в этом случае его математическое ожидание равно μ_2 .

Байесовская стратегия, минимизирующая вероятность ошибочного распознавания, должна при наблюдении x принимать решение в пользу первого состояния, если

$$(x - \mu_1)^2 \leq (x - \mu_2)^2, \quad (6.8)$$

и в пользу второго состояния в противном случае. Обозначим эту стратегию q . Эта стратегия зависит от математических ожиданий μ_1 и μ_2 . Обозначим эту зависимость как $q(\mu_1, \mu_2)$. Стратегия однозначно определена, если определены векторы μ_1 и μ_2 .

Алгоритм обучения должен вычислить наиболее правдоподобные оценки μ_1^* , μ_2^* математических ожиданий μ_1 , μ_2 на основе выборки (x_1, k_1) , (x_2, k_2) , \dots , (x_m, k_m) случайных пар (x, k) , имеющих распределение плотности вероятности

$$p_{XK}(x, k) = p_K(k) \frac{1}{(\sqrt{2\pi})^n} e^{-\frac{1}{2}(x - \mu_k)^2}.$$

Эти оценки наибольшего правдоподобия равны

$$\begin{aligned}
(\mu_1^*, \mu_2^*) &= \operatorname{argmax}_{(\mu_1, \mu_2)} \sum_{i=1}^m \log \left(p_K(k_i) \frac{1}{(\sqrt{2\pi})^n} e^{-\frac{1}{2}(x_i - \mu_{k_i})^2} \right) \\
&= \operatorname{argmax}_{(\mu_1, \mu_2)} \sum_{i=1}^m -(x_i - \mu_{k_i})^2 = \operatorname{argmin}_{(\mu_1, \mu_2)} \left(\sum_{i \in I_1} (x_i - \mu_1)^2 + \sum_{i \in I_2} (x_i - \mu_2)^2 \right) \\
&= \left(\operatorname{argmin}_{\mu_1} \sum_{i \in I_1} (x_i - \mu_1)^2, \operatorname{argmin}_{\mu_2} \sum_{i \in I_2} (x_i - \mu_2)^2 \right) \\
&= \left(\frac{1}{|I_1|} \sum_{i \in I_1} x_i, \frac{1}{|I_2|} \sum_{i \in I_2} x_i \right), \tag{6.9}
\end{aligned}$$

где I_k , $k = 1, 2$, обозначает множество индексов i , для которых $k_i = k$.

Авторы ИЗОДАТЫ Холл и Болл [Ball and Hall, 1967] соединили стратегию (6.8) и алгоритм обучения (6.9) так, что получился алгоритм, работающий следующим образом. Он многократно, итерация за итерацией, обрабатывает выборку x_1, x_2, \dots, x_m , получая каждый раз новые оценки математических ожиданий μ_1 и μ_2 , а заодно и новую классификацию предъявленной выборки.

Значения μ_1^0 и μ_2^0 (верхний индекс обозначает номер итерации) могут быть любыми, если только стратегия $q(\mu_1^0, \mu_2^0)$ не относит всю выборку к какому-то одному классу. Для этого достаточно, например, выбрать эти значения равными $\mu_1^0 = x_1$ и $\mu_2^0 = x_2$, если только x_1 не равно x_2 . Пусть после $(t-1)$ -ой итерации векторы μ_1 и μ_2 приняли значения $\mu_1^{(t-1)}$ и $\mu_2^{(t-1)}$. Тогда на очередной, t -ой итерации выполняются две процедуры.

1. *Классификация.* Наблюдение x_i , $i = 1, \dots, m$, относится к первому классу, если

$$\left(x_i - \mu_1^{(t-1)} \right)^2 \leq \left(x_i - \mu_2^{(t-1)} \right)^2,$$

и ко второму классу в противном случае. В результате получается разбиение множества $i = 1, 2, \dots, m$ на два подмножества $I_1^{(t)}$ и $I_2^{(t)}$.

2. *Обучение.* Вычисляются новые значения $\mu_1^{(t)}$ и $\mu_2^{(t)}$ как средние значения векторов, отнесенные предыдущей процедурой к одному классу,

$$\mu_k^{(t)} = \frac{1}{|I_k^{(t)}|} \sum_{i \in I_k^{(t)}} x_i, \quad k = 1, 2.$$

Конечно же, требовалась определенная смелость, чтобы предложить и исследовать такой алгоритм в условиях нокдауна, в котором находились идеи самообучения после неудачных экспериментов с самообучением в перцептроне. Эта смелость была вознаграждена следующими положительными экспериментальными результатами.

1. Алгоритм сходится за конечное количество итераций независимо от начальных значений μ_1^0 and μ_2^0 . Результирующие значения μ_1 и μ_2 и классификация I_1 и I_2 являются решением следующей системы уравнений,

$$\left. \begin{aligned} \mu_1 &= \frac{1}{|I_1|} \sum_{i \in I_1} x_i, \\ \mu_2 &= \frac{1}{|I_2|} \sum_{i \in I_2} x_i, \\ I_1 &= \{i \mid (x_i - \mu_1)^2 \leq (x_i - \mu_2)^2\}, \\ I_2 &= \{i \mid (x_i - \mu_1)^2 > (x_i - \mu_2)^2\}. \end{aligned} \right\} \quad (6.10)$$

2. Система уравнений (6.10) может иметь более чем одно решение. В одном из них классификация I_1, I_2 исходной выборки x_1, x_2, \dots, x_n оказывается близкой к классификации, которая была бы достигнута при обучении с учителем. К сожалению, ИЗОДАТА не обязательно сходится именно к этому хорошему решению. Но если длина m выборки достаточно велика, а отношение разности $|\mu_1 - \mu_2|$ действительных математических ожиданий велико по сравнению с дисперсией, ИЗОДАТА сходится с большой вероятностью именно к указанному хорошему решению.

Приведенный алгоритм находит (по крайней мере в некотором частном случае) правильную классификацию входных наблюдений и делает это без каких-либо указаний от учителя. К сожалению, алгоритм непосредственно не обобщается на более общие случаи, например, когда априорные вероятности состояний не одинаковы, или компоненты гауссового случайного вектора взаимно зависимы и т.п. Вместе с тем, система уравнений (6.10) тесно связана с решением некоторой вполне понятной задачи, которая, однако, не имеет ничего общего с задачей распознавания состояния k по наблюдению x . Эта задача принадлежит классу задач, имеющих следующую общую формулировку.

Пусть x - случайный объект из множества X , на котором задано распределение вероятностей $p_X: X \rightarrow \mathbb{R}$. Пусть D - множество, элементы которого называются *решениями*, так что каждому объекту $x \in X$ соответствует определенное решение $d \in D$. Пусть $W: X \times D \rightarrow \mathbb{R}$ - штрафная функция, значение которой $W(x, d)$ обозначает потери в случае, когда к объекту x применяется решение d . Обратим внимание на то, что введенная функция потерь имеет совсем другой формат, чем функция потерь в байесовских задачах принятия решений. Качество решения d зависит от известного наблюдения x , а не от скрытого состояния k . Понятие скрытого состояния в данной постановке задачи вообще отсутствует.

Предположим для начала, что задача состоит в построении стратегии $q: X \rightarrow D$, которая минимизирует математическое ожидание потерь

$$\sum_{x \in X} p(x) W(x, q(x)).$$

Легко увидеть, что решением этой задачи является стратегия

$$q(x) = \operatorname{argmin}_{d \in D} W(x, d),$$

которая не зависит от распределения вероятностей p_X вообще и которую нетрудно построить для некоторых штрафных функций W . Задача существенно усложняется, если потребовать, чтобы стратегия q принимала на множестве X не более, чем заданное количество значений, например, только два значения d_1 и d_2 . Искомая стратегия, следовательно, должна иметь вид $q: X \rightarrow \{d_1, d_2\}$, а не $q: X \rightarrow D$. При этом значения d_1, d_2 не заданы заранее и их следует оптимальным образом выбрать. Если бы они были заданы, то, очевидно, стратегия q^* должна была бы решать, что $q^*(x) = d_1$ при $W(x, d_1) \leq W(x, d_2)$ и $q^*(x) = d_2$ в противном случае. Величина штрафа, очевидно, была бы равна

$$W(x, q^*(x)) = \min(W(x, d_1), W(x, d_2)).$$

Если же значения d_1, d_2 заранее не заданы, требуется найти такие их значения d_1^* и d_2^* , которые минимизируют математическое ожидание

$$\sum_{x \in X} p(x) \min(W(x, d_1), W(x, d_2)). \quad (6.11)$$

В этой постановке величины d_1^* и d_2^* являются вспомогательными. Они служат лишь для получения окончательного решения, то есть разбиения множества X на подмножества X_1 и X_2 . Смысл этого разбиения состоит в том, что различие некоторых объектов игнорируется, а именно, различие объектов, отнесенных к одному и тому же подмножеству X_i . К таким объектам применяется одно и то же решение, хотя следовало бы применять разные. Вследствие этого потери возрастают, и задача состоит в том, чтобы эти потери возрасли как можно меньше. Рассмотрим объяснительный пример задачи такого вида.

Пример 6.2 Размещение колонок с водой. Пусть некоторый поселок надо снабжать водой. Он состоит из домов, которые достаточно малы по сравнению с размером поселка, так, что расположение того или иного дома достаточно точно определяется точкой x на двумерной плоскости. Пусть X - множество точек, в которых расположены дома, а $p_X(x)$ - количество жителей в доме, расположенном в точке x . Пусть $W(x, d)$ - потери жителя, который проживает в точке x и ходит за водой к колонке в точке d .

Стратегия $q: X \rightarrow D$ определяет для каждого жителя, к какой колонке он должен ходить за водой. Если на стратегию не накладывать никаких ограничений, то оптимальной окажется стратегия, при которой для каждого x величина $q(x)$ равна x . Естественно, суммарные потери жителей минимальны, если каждый житель может получить воду, не выходя из своего дома.

Если же во всем поселке должно быть не более двух колонок и можно только выбирать их расположение, задача существенно усложняется и принимает вид (6.11). Сначала следует определить места d_1 и d_2 расположения колонок. Затем следует решить значительно более простую задачу: множество домов в поселке следует разбить на два подмножества, которые определяют приписывание каждого дома к одной из двух колонок. ▲

В приведенном примере, равно как и в общей формулировке (6.11), задача сформулирована так, что ее решением является классификация множества X наблюдений. В этом отношении задача сходна с задачей распознавания, решением которой также является классификация наблюдений. Однако природа этих задач существенно различна.

В задачах статистического распознавания предполагается существование определенного фактора k , от которого зависят наблюдения x . Хотя этот фактор ненаблюдаем, но он реально существует и принимает определенное значение. При каком-то нестандартном вмешательстве в объект можно даже непосредственно измерить этот фактор. Этот фактор представляет собой некую реальность, и качество распознавания определяется именно тем, насколько результаты распознавания отличаются от реальности.

В задаче (6.11), проиллюстрированной приведенным примером, не предполагается существование какого-либо скрытого фактора, влияющего на наблюдения. Так, расположение домов в поселке вовсе не зависит от того, в каком месте уже сформировавшегося поселка будут построены колонки. Цель классификации наблюдений здесь совершенно иная. Наблюдения классифицируются единственно с целью упрощения последующей их обработки, которое достигается за счет того, что с различными объектами будут поступать одинаково. При этом, конечно же, такое закругление приведет к определенным потерям. Задача состоит в том, чтобы потери от такого закругления были как можно меньше.

На неформальном уровне задачи такого типа известны достаточно давно, как задачи *кластеризации*, задачи *таксономии*, *классификации* и т.п. Формулировку (6.11) можно рассматривать как формальное выражение задач такого типа. ИЗОДАТА есть частный случай этих задач, когда $W(x, d) = (x - d)^2$.

Экспериментально наблюдаемый факт, что результат ИЗОДАТЫ - частного случая кластеризации, совпадает с решением частной байесовской задачи, есть не более, чем случайное совпадение. Ситуация здесь подобна тому, как если бы кто-то хотел построить программу вычисления функции $\log_2 x$, но случайно получил программу вычисления \sqrt{x} . Первые эксперименты, когда программа тестировалась при значениях x , близких к $\sqrt{4} = 2 = \log_2 4$, вселяли надежду, что это именно та программа, что нужно. Однако при дальнейшем анализе выяснилось, что это программа вычисления совсем другой функции, тоже очень хорошей.

Мы увидели, что розенблаттовские алгоритмы оказались полезными для

решения некоторых задач кластеризации, формулировка которых существенно отличается от задач самообучения, как они были сформулированы в разделе 6.2. Для решения задач самообучения алгоритмы Розенблатта пришлось несколько модифицировать. Как мы увидим, эта модификация, с одной стороны, едва заметна, а с другой - очень существенна.

6.6 Алгоритмы самообучения и их анализ

Пусть X и K - множества наблюдений и состояний. На декартовом произведении X и K определено распределение вероятностей p_{XK} . Как и прежде, будем считать, что множества X и K конечные, и тем самым уклонимся от некоторых теоретических трудностей, которые несущественны в контексте данной лекции, но заслонили бы ее основную идею.

6.6.1 Задача распознавания

Пусть функция p_{XK} известна и для распознавания предъявлена выборка (x_1, x_2, \dots, x_n) наблюдений. Распознавание элементов этой выборки не может пониматься, как определение соответствующих состояний k_1, k_2, \dots, k_n , так как наблюдение x не определяет состояние k однозначно. Наблюдение x_i может иметь место при различных состояниях объекта, и можно вычислить лишь апостериорные вероятности $\alpha(i, k)$, $k \in K$, пребывания объекта в состоянии k в момент наблюдения x_i ,

$$\alpha(i, k) = \frac{p_{XK}(x_i, k)}{\sum_{k \in K} p_{XK}(x_i, k)} = \frac{p_K(k) p_{X|K}(x_i | k)}{\sum_{k \in K} p_K(k) p_{X|K}(x_i | k)}. \quad (6.12)$$

Совокупность величин $\alpha(i, k)$, $i = 1, 2, \dots, n$, $k \in K$, будем понимать как результат распознавания предъявленной выборки (x_1, x_2, \dots, x_n) наблюдений.

6.6.2 Задача обучения

Значение $p_{XK}(x, k)$ функции $p_{XK} : X \times K \rightarrow \mathbb{R}$ обозначает совместную вероятность наблюдения x и состояния k . Функция p_{XK} однозначно определяет пару функций $p_K : K \rightarrow \mathbb{R}$ и $p_{X|K} : X \times K \rightarrow \mathbb{R}$. Первая обозначает априорное распределение вероятностей состояний k , а вторая - условное распределение вероятностей наблюдений x при условии, что объект находится в состоянии k . Для дальнейшего изложения мы изменим обозначение этих условных вероятностей так, что они будут выражаться не одной функцией $p_{X|K} : X \times K \rightarrow \mathbb{R}$ двух переменных x и k , а с помощью совокупности $|K|$ функций вида $p_{X|k} : X \rightarrow \mathbb{R}$ одной переменной x . Функции из этой совокупности индексируются переменной k , которая принимает значения на множестве K . Условная вероятность наблюдения x при условии, что объект находится в состоянии k , будет обозначаться $p_{X|k}(x)$.

Предположим, что априорные вероятности $p_K(k)$ неизвестны, а условные вероятности $p_{X|k}(x)$ для каждого $k \in K$ известны частично, то есть с точностью до значения параметра $a \in A$, где A - множество его значений.

Мы выразим эту неполноту знаний в виде функции $f: X \times A \rightarrow \mathbb{R}$ двух переменных x и a . Эта функция выражает частичное знание статистической модели в том смысле, что для каждого $k \in K$ существует такое значение a_k (заранее неизвестное), что условная вероятность $p_{X|k}(x)$ равна $f(x, a_k)$. Обозначим символом m совокупность всех неизвестных статистических параметров объекта и будем называть ее *статистической моделью объекта* или коротко моделью, $m = ((p_K(k), a_k), k \in K)$. Модель однозначно определяет вероятности $p_{XK}(x, k)$, необходимые для решения задачи распознавания.

Предположим, что задана выборка $(x_1, k_1), (x_2, k_2), \dots, (x_n, k_n)$ случайных и взаимно независимых пар (x, k) . Каждая пара в этой выборке есть реализация случайной пары (x, k) , имеющей распределение вероятностей p_{XK} , которое неизвестно. Известно однако, как вероятность выборки $(x_1, k_1), (x_2, k_2), \dots, (x_n, k_n)$ в целом зависит от модели m , то есть, от априорных вероятностей $p_K(k)$ и значений $a_k, k \in K$. Эта вероятность равна

$$l(m) = \prod_{i=1}^n p_{XK}(x_i, k_i) = \prod_{i=1}^n p_K(k_i) \prod_{i=1}^n p_{X|k_i}(x_i) = \prod_{i=1}^n p_K(k_i) \prod_{i=1}^n f(x_i, a_{k_i}).$$

Обозначим $L(m)$ логарифм вероятности $l(m)$ и получим

$$L(m) = \sum_{i=1}^n \log p_K(k_i) + \sum_{i=1}^n \log f(x_i, a_{k_i}).$$

Информацию от учителя о состояниях k_1, k_2, \dots, k_n можно представить с помощью совокупности чисел $\alpha(i, k)$, $i=1, \dots, n, k \in K$, где $\alpha(i, k)$ равно 1, если $k = k_i$, и равно 0 в любом другом случае. Применяя такое обозначение, функцию $L(m)$ можно выразить в виде

$$L(m) = \sum_{i=1}^n \sum_{k \in K} \alpha(i, k) \log p_K(k) + \sum_{i=1}^n \sum_{k \in K} \alpha(i, k) \log f(x_i, a_k). \quad (6.13)$$

Под *задачей обучения* понимается поиск такой модели $m = ((p_K(k), a_k), k \in K)$, при которой достигается максимум логарифма правдоподобия $L(m)$. При этом числа $p_K(k), k \in K$, конечно же, должны удовлетворять условию $\sum_{k \in K} p_K(k) = 1$. Нетрудно убедиться (мы к этому еще вернемся), что наилучшие оценки вероятностей $p_K(k)$ определяются как

$$p_K(k) = \frac{\sum_{i=1}^n \alpha(i, k)}{n}, \quad k \in K, \quad (6.14)$$

а максимизация функции $L(m)$ по совокупности значений $(a_k, k \in K)$, сводится к независимым оптимизациям $|K|$ функций

$$\sum_{i=1}^n \alpha(i, k) \log f(x_i, a_k)$$

по переменной a_k . Таким образом, можно записать

$$a_k = \operatorname{argmax}_a \sum_i \alpha(i, k) \log f(x_i, a). \quad (6.15)$$

Отметим, что задача, сформулированная таким образом, имеет смысл не только для случая *полностью информированного учителя*, который для каждого наблюдения x_i в точности знает действительное состояние объекта. Величины $\alpha(i, k)$ не обязательно должны быть нулями или единицами. Они могут принимать вещественные значения в интервале $0 \leq \alpha(i, k) \leq 1$, сумма которых $\sum_{k \in K} \alpha(i, k)$ равна 1 для каждого i . В таком виде можно представить и сведения от *неполностью информированного учителя*, который не знает точно о состоянии объекта, но по каким-то дополнительным наблюдениям может судить о вероятности того или иного состояния. Сведения, получаемые от такого неполностью информированного учителя, представлены в том же формате, что и результаты распознавания, как это было определено в предыдущем подразделе.

6.6.3 Задача самообучения

Пусть $p_{XK}(x, k) = p_K(k) p_{X|k}(x)$, $x \in X$, $k \in K$, - совместные вероятности наблюдения x и состояния k . Априорные вероятности $p_K(k)$ состояний неизвестны, а условные вероятности $p_{X|k}(x)$ известны с точностью до значения параметра a_k . Это значит, что задана такая функция $f: X \times A \rightarrow \mathbb{R}$, что при определенном значении a_k параметра a условные вероятности $p_{X|k}(x)$ равны числам $f(x, a_k)$. Пусть x_1, x_2, \dots, x_n - выборка случайных наблюдений, имеющих распределение вероятностей

$$p_X(x) = \sum_{k \in K} p_{XK}(x, k) = \sum_{k \in K} p_K(k) f(x, a_k).$$

Вероятность этой выборки равна

$$l(m) = \prod_{i=1}^n \sum_{k \in K} p_K(k) f(x_i, a_k). \quad (6.16)$$

Вероятность $l(m)$ зависит от априорных вероятностей $p_K(k)$, $k \in K$, и совокупности значений a_k , $k \in K$. Логарифм этой вероятности равен

$$L(m) = \sum_{i=1}^n \log \sum_{k \in K} p_K(k) f(x_i, a_k). \quad (6.17)$$

Задача самообучения состоит в поиске таких значений априорных вероятностей и параметров

$$m^* = (p_K^*(k), a_k^* | k \in K),$$

которые максимизируют число (6.17), то есть максимизируют вероятность выборки x_1, x_2, \dots, x_n .

В частном случае, когда отыскиваются максимально правдоподобные оценки только для априорных вероятностей $p_K(k)$, задача совпадает с задачей Роббинса.

6.6.4 Алгоритм самообучения

Пусть $m^0 = (p_K^0(k), a_k^0 | k \in K)$ - совокупность начальных значений оцениваемых параметров. Алгоритм самообучения строит, итерация за итерацией, последовательность моделей $m^1, m^2, \dots, m^t, m^{t+1}, \dots$ по следующим правилам.

Пусть $m^t = (p_K^t(k), a_k^t | k \in K)$ - модель, полученная после t -ой итерации самообучения. Следующая модель m^{t+1} вычисляется в два этапа. На первом этапе вычисляются числа

$$\alpha^t(i, k) = \frac{p_K^t(k) f(x_i, a_k^t)}{\sum_{k' \in K} p_K^t(k') f(x_i, a_{k'}^t)} \quad (6.18)$$

для каждого $i = 1, 2, \dots, n$ и каждого $k \in K$. Числа $\alpha^t(i, k)$ подобны апостериорным вероятностям состояний k при условии наблюдения x_i . Их следовало бы вычислять при распознавании, если бы значения $p_K^t(k)$ и a_k^t совпадали с действительными. Однако эти числа не обязательно равны апостериорным вероятностям, так как текущие оценки $p_K^t(k)$ и a_k^t не обязательно равны действительным априорным вероятностям и значениям параметра a . Тем не менее, на втором этапе $t+1$ -ой итерации строится модель m^{t+1} с помощью алгоритма обучения (6.14) и (6.15) так, будто числа $\alpha^t(i, k)$ сообщил учитель, то есть

$$p_K^{t+1}(k) = \frac{\sum_{i=1}^n \alpha^t(i, k)}{\sum_{k' \in K} \sum_{i=1}^n \alpha^t(i, k')} = \frac{\sum_{i=1}^n \alpha^t(i, k)}{n}, \quad k \in K, \quad (6.19)$$

$$a_k^{t+1} = \operatorname{argmax}_a \sum_{i=1}^n \alpha^t(i, k) \log f(x_i, a), \quad k \in K. \quad (6.20)$$

Сходство приведенного алгоритма с розенблаттовскими алгоритмами, по-видимому, бросается в глаза. Приведенный алгоритм, как и розенблаттовские алгоритмы, устроен в виде повторяющихся процедур обучения и распознавания, причем обучение основано не на данных, полученных от учителя, а на результатах собственного, пусть неправильного распознавания.

В то же время приведенный алгоритм существенно отличается от алгоритмов, предсказанных Розенблаттом. Как распознавание, так и обучение в приведенном алгоритме происходят несколько иначе, чем в розенблаттовских алгоритмах. При распознавании не происходит отнесение какого-либо наблюдения к какому-то определенному классу. Алгоритм словно делит каждое наблюдение x_i на части, пропорциональные числам $\alpha(i, k)$, а затем относит это наблюдение частично в один класс, а частично - в другой. Подобные изменения произошли и в алгоритмах обучения, когда они становятся составной частью самообучения. В отличие от алгоритмов Розенблатта уточнение модели происходит не на основании однозначного указания, в каком состоянии находился объект в момент наблюдения x_i , а на основании лишь вероятностей того или иного состояния. Мы увидим, что

именно эти отличия окажутся существенными для работоспособности самообучения.

Приведенный алгоритм устанавливает взаимосвязь трех обширных классов задач статистического распознавания: задач собственно распознавания, обучения с учителем и самообучения. Эта взаимосвязь является универсальной, так как имеет место для статистических моделей самого общего вида. Как задача самообучения, так и алгоритм ее решения сформулированы без каких-либо упрощающих предположений о классе статистических моделей. Мы увидим далее, что и формальные свойства алгоритма самообучения имеют место либо в самом общем случае, либо при очень слабых дополнительных ограничениях. В силу этой общности, а также благодаря тому, что алгоритм самообучения допускает наглядную и легко запоминаемую формулировку, его открытие принадлежит к наиболее значительным результатам не только в распознавании образов, но и в информатике статистических данных вообще. Наука о распознавании демонстрирует таким образом свою зрелость, показывая, что она не только поглощает научные результаты из окрестных научных дисциплин, но и обогащает эту окрестность собственными результатами.

6.6.5 Анализ алгоритмов самообучения

Покажем наиболее важное свойство алгоритмов самообучения, которое имеет место в самом общем случае и доказывается при отсутствии каких-либо упрощающих предположений. Это свойство состоит в монотонном характере последовательности $m^0, m^1, \dots, m^t, m^{t+1}, \dots$, моделей, создаваемых алгоритмом. Каждая последующая модель m^t не хуже предыдущей, а если она не равна предыдущей, $m^t \neq m^{t-1}$, то она гарантированно лучше предыдущей. Таким образом, однажды достигнутое качество не может ухудшиться в процессе самообучения. Докажем сначала лемму, а затем теорему, в которой указанное свойство формулируется точно.

Лемма 6.1 Шеннон. Пусть $\alpha_i, i = 1, \dots, n$, - положительные константы, а $x_i, i = 1, \dots, n$, - положительные переменные, удовлетворяющие условию $\sum_{i=1}^n x_i = 1$. В этом случае справедливо неравенство

$$\sum_{i=1}^n \alpha_i \log x_i \leq \sum_{i=1}^n \alpha_i \log \frac{\alpha_i}{\sum_{j=1}^n \alpha_j}, \quad (6.21)$$

причем равенство выполняется только тогда, когда $x_i = \alpha_i / \sum_{j=1}^n \alpha_j$ для всех i . ▲

Доказательство. Обозначим $F(x_1, x_2, \dots, x_n) = \sum_{i=1}^n \alpha_i \log x_i$ и найдем значения x_i , максимизирующие F при условии $\sum_i x_i = 1$. Поскольку функция F вогнута, точка x_1, x_2, \dots, x_n есть точка максимума, если она удовле-

творяет равенства

$$\left. \begin{aligned} \frac{\partial \Phi(x_1, x_2, \dots, x_n)}{\partial x_i} &= 0, \quad i = 1, \dots, n, \\ \sum_{i=1}^n x_i &= 1, \end{aligned} \right\} \quad (6.22)$$

где Φ - функция Лагранжа

$$\Phi(x_1, x_2, \dots, x_n) = \sum_{i=1}^n \alpha_i \log x_i + \lambda \sum_{i=1}^n x_i.$$

В данном случае система уравнений (6.22) имеет вид

$$\left. \begin{aligned} \frac{\alpha_i}{x_i} + \lambda &= 0, \quad i = 1, \dots, n, \\ \sum_{i=1}^n x_i &= 1. \end{aligned} \right\}$$

Это значит, что

$$x_i = \frac{-\alpha_i}{\lambda}, \quad \sum_{j=1}^n x_j = \frac{-\sum_{j=1}^n \alpha_j}{\lambda}, \quad 1 = \frac{-\sum_{j=1}^n \alpha_j}{\lambda}, \quad \lambda = -\sum_{j=1}^n \alpha_j,$$

и в итоге

$$x_i = \frac{\alpha_i}{\sum_{j=1}^n \alpha_j}. \quad (6.23)$$

Легко видеть, что система уравнений (6.22) имеет единственное решение. Величины x_i , определяемые выражением (6.23), определяют единственную точку максимума на гиперплоскости $\sum_{i=1}^n x_i = 1$. ■

Теорема 6.1 **О монотонном характере самообучения.** Пусть $m^t = (p_K^t(k), a_k^t \mid k \in K)$ и $m^{t+1} = (p_K^{t+1}(k), a_k^{t+1} \mid k \in K)$ - модели, вычисленные на t -ой и $(t+1)$ -ой итерации самообучения. Пусть по крайней мере для одного i и одного k выполняется

$$\frac{p_K^t(k) f(x_i, a_k^t)}{\sum_{k' \in K} p_K^t(k') f(x_i, a_{k'}^t)} \neq \frac{p_K^{t+1}(k) f(x_i, a_k^{t+1})}{\sum_{k' \in K} p_K^{t+1}(k') f(x_i, a_{k'}^{t+1})}. \quad (6.24)$$

В таком случае выполняется неравенство

$$\sum_{i=1}^n \log \sum_{k' \in K} p_K^t(k') f(x_i, a_{k'}^t) < \sum_{i=1}^n \log \sum_{k' \in K} p_K^{t+1}(k') f(x_i, a_{k'}^{t+1}). \quad (6.25)$$

▲

Доказательство. Пусть $\alpha(i, k)$, $i = 1, \dots, n$, $k \in K$, - любые неотрицательные числа, такие, что для любого i выполняется равенство

$$\sum_{k \in K} \alpha(i, k) = 1. \quad (6.26)$$

В таком случае функцию $L(m)$ можно представить в виде алгебраической суммы трех слагаемых,

$$\begin{aligned} L(m) &= \sum_{i=1}^n \log \sum_{k' \in K} p_K(k') f(x_i, a_{k'}) = \sum_{i=1}^n \sum_{k \in K} \alpha(i, k) \log \sum_{k' \in K} p_K(k') f(x_i, a_{k'}) \\ &= \sum_{i=1}^n \sum_{k \in K} \alpha(i, k) \log p_K(k) + \sum_{i=1}^n \sum_{k \in K} \alpha(i, k) \log f(x_i, a_k) \\ &\quad - \sum_{i=1}^n \sum_{k \in K} \alpha(i, k) \log \frac{p_K(k) f(x_i, a_k)}{\sum_{k' \in K} p_K(k') f(x_i, a_{k'})}. \end{aligned} \quad (6.27)$$

Разложение (6.27) функции $L(m)$ на три слагаемые справедливо при любых числах $\alpha(i, k)$, удовлетворяющих ограничению (6.26), а следовательно, и при числах $\alpha^t(i, k)$. Представим в виде такого разложения числа $L(m^t)$ и $L(m^{t+1})$. В обоих разложениях мы используем одни и те же числа $\alpha^t(i, k)$. Изменив порядок суммирования $\sum_{i=1}^n$ и $\sum_{k \in K}$ в первом и втором слагаемом, получим

$$\begin{aligned} L(m^t) &= \sum_{k \in K} \sum_{i=1}^n \alpha^t(i, k) \log p_K^t(k) + \sum_{k \in K} \sum_{i=1}^n \alpha^t(i, k) \log f(x_i, a_k^t) \\ &\quad - \sum_{i=1}^n \sum_{k \in K} \alpha^t(i, k) \log \frac{p_K^t(k) f(x_i, a_k^t)}{\sum_{k' \in K} p_K^t(k') f(x_i, a_{k'}^t)}; \end{aligned} \quad (6.28)$$

$$\begin{aligned} L(m^{t+1}) &= \sum_{k \in K} \sum_{i=1}^n \alpha^t(i, k) \log p_K^{t+1}(k) + \sum_{k \in K} \sum_{i=1}^n \alpha^t(i, k) \log f(x_i, a_k^{t+1}) \\ &\quad - \sum_{i=1}^n \sum_{k \in K} \alpha^t(i, k) \log \frac{p_K^{t+1}(k) f(x_i, a_k^{t+1})}{\sum_{k' \in K} p_K^{t+1}(k') f(x_i, a_{k'}^{t+1})}. \end{aligned} \quad (6.29)$$

Значения $p_K^{t+1}(k)$, $k \in K$, априорных вероятностей по определению (6.19) равны

$$p_K^{t+1}(k) = \frac{\sum_{i=1}^n \alpha^t(i, k)}{\sum_{k' \in K} \sum_{i=1}^n \alpha^t(i, k')},$$

и в силу леммы 6.1 получаем

$$\sum_{k \in K} \sum_{i=1}^n \alpha^t(i, k) \log p_K^t(k) \leq \sum_{k \in K} \sum_{i=1}^n \alpha^t(i, k) \log p_K^{t+1}(k). \quad (6.30)$$

Это значит, что первое слагаемое в левой части (6.28) не превышает первое слагаемое в правой части (6.29).

В соответствии с определением (6.20) справедливы неравенства

$$\sum_{i=1}^n \alpha^t(i, k) \log f(x_i, a_k^t) \leq \sum_{i=1}^n \alpha^t(i, k) \log f(x_i, a_k^{t+1}), \quad k \in K.$$

Просуммировав эти неравенства по всем значениям $k \in K$, получим

$$\sum_{k \in K} \sum_{i=1}^n \alpha^t(i, k) \log f(x_i, a_k^t) \leq \sum_{k \in K} \sum_{i=1}^n \alpha^t(i, k) \log f(x_i, a_k^{t+1}), \quad (6.31)$$

и это обозначает, что второе слагаемое в левой части (6.28) не больше, чем второе слагаемое в правой части (6.29).

По определению (6.18),

$$\alpha^t(i, k) = \frac{p_K^t(k) f(x_i, a_k^t)}{\sum_{k' \in K} p_K^t(k') f(x_i, a_{k'}^t)}.$$

Поскольку $\sum_{k \in K} \alpha^t(i, k) = 1$, а также в силу леммы 6.1 неравенство

$$\sum_{k \in K} \alpha^t(i, k) \log \frac{p_K^t(k) f(x_i, a_k^t)}{\sum_{k' \in K} p_K^t(k') f(x_i, a_{k'}^t)} \geq \sum_{k \in K} \alpha^t(i, k) \log \frac{p_K^{t+1}(k) f(x_i, a_k^{t+1})}{\sum_{k' \in K} p_K^{t+1}(k') f(x_i, a_{k'}^{t+1})} \quad (6.32)$$

выполняется для всех $i = 1, 2, \dots, n$. В то же время по условию (6.24) по крайней мере для некоторых значений i и k выполняется строгое неравенство

$$\frac{p_K^t(k) f(x_i, a_k^t)}{\sum_{k' \in K} p_K^t(k') f(x_i, a_{k'}^t)} \neq \frac{p_K^{t+1}(k) f(x_i, a_k^{t+1})}{\sum_{k' \in K} p_K^{t+1}(k') f(x_i, a_{k'}^{t+1})}.$$

Таким образом, неравенство (6.32) можно записать строго,

$$\begin{aligned} & \sum_{i=1}^n \sum_{k \in K} \alpha^t(i, k) \log \frac{p_K^t(k) f(x_i, a_k^t)}{\sum_{k' \in K} p_K^t(k') f(x_i, a_{k'}^t)} \\ & > \sum_{i=1}^n \sum_{k \in K} \alpha^t(i, k) \log \frac{p_K^{t+1}(k) f(x_i, a_k^{t+1})}{\sum_{k' \in K} p_K^{t+1}(k') f(x_i, a_{k'}^{t+1})}. \end{aligned} \quad (6.33)$$

Неравенство $L(m^t) < L(m^{t+1})$ есть очевидное следствие неравенств (6.30), (6.31) (6.33). \blacksquare

Мы видим, что логарифм вероятности $L(m)$ монотонно возрастает в процессе самообучения, то есть, что $L(m^0) < L(m^1) < \dots < L(m^t) < L(m^{t+1})$,

если только $m^{t+1} \neq m^t$. Отсюда непосредственно следует, что последовательность $L(m^t)$ имеет предел при $t \rightarrow \infty$, так как $L(m)$ не может принимать положительные значения. Однако отсюда не следует, что последовательность моделей m^t стремится к пределу при $t \rightarrow \infty$. Более того, на том уровне общности, на котором мы сейчас анализируем алгоритм, не определено само понятие предела последовательности моделей. Ведь до сих пор не были определены никакие метрические свойства параметрического множества моделей, и следовательно, не определено понятие сходящейся последовательности. Параметром модели, например, может быть граф, и тогда непонятно, что обозначает утверждение, что последовательность графов стремится к некоторому фиксированному графу. В силу этого последующий анализ обязательно должен опираться на определенные дополнительные предположения, если этот анализ имеет целью показать, что последовательность моделей в процессе самообучения стремится к определенному пределу. Следовательно, результаты последующего анализа будут неизбежно менее общими, чем результат, сформулированный в предыдущей теореме. Одновременно с этим еще раз обратим внимание на исключительную общность предыдущей теоремы 6.1 о монотонности, справедливой в самом общем случае.

Мы покажем определенные асимптотические свойства самообучения, которые можно трактовать, как его сходимость. И тем не менее мы уклонимся от конкретизации вида параметров, определяющих модель распознаваемого объекта, то есть останемся на уровне анализа моделей достаточно общего вида. Мы достигнем этого, анализируя сходимость чисел $\alpha^t(i, k)$, а не сходимость моделей m^t . Для этого анализа нам придется ввести некоторые предположения о модели, которые, конечно же, несколько ограничат область действия полученных далее результатов, но не так сильно, как это бы имело место при анализе сходимости последовательности моделей m^t . С определенной точки зрения числа $\alpha(i, k)$ более важны, чем модель m , так как именно числа $\alpha(i, k)$ определяют, как в конечном итоге будут распознаны наблюдения x_1, x_2, \dots, x_n . Что касается модели m , то это промежуточный результат, вспомогательный при достижении этой главной цели.

Для дальнейшего анализа введем дополнительные обозначения. Совокупность чисел $\alpha(i, k)$, $i = 1, 2, \dots, n$, $k \in K$, обозначим символом α без последующих скобок. Символом α^t обозначим совокупность чисел $\alpha^t(i, k)$, полученную на t -ой итерации самообучения. Совокупность α подается на вход алгоритма обучения, определенного выражениями (6.19) и (6.20). На основе α алгоритм обучения вычисляет новую модель m . Обозначим эти вычисления символом U (обУчение), так, что $m^{t+1} = U(\alpha^t)$.

На основе модели m вычисляется новая совокупность α в соответствии с формулой (6.18). Обозначим эти вычисления R (Распознавание), так, что $\alpha^t = R(m^t)$. Обозначим S (Самообучение) применение операции R к результату операции U , так, что $\alpha^{t+1} = S(\alpha^t) = R(U(\alpha^t))$. Преобразование S реализуется одной итерацией самообучения, которая сейчас будет определена несколько иначе, чем в формулировке (6.18), (6.19) и (6.20). Каждая итерация начинается и заканчивается получением очередной совокупности

α , а не очередной моделью m , как это определялось раньше. Очевидно, что при этом в самом процессе самообучения ровным счетом ничего не меняется. Меняется лишь договоренность о том, на каком этапе вычислений заканчивается одна итерация и начинается следующая.

Совокупность α будет пониматься, как точка нормированного линейного пространства с естественно определенной нормой $|\alpha|$ вида

$$|\alpha| = \sqrt{\sum_i \sum_k (\alpha(i, k))^2}.$$

Основным в последующем анализе является условие *непрерывности функции* S . Именно, малым изменениям α , соответствуют малые изменения $S(\alpha)$. Выразим это условие более точно. Пусть α^i , $i = 1, 2, \dots, \infty$, - любая сходящаяся последовательность, не обязательно та, которая получается в процессе самообучения. Пусть α^* есть предел этой последовательности. Тогда последовательность $S(\alpha^i)$, $i = 1, 2, \dots, \infty$, также сходится и ее пределом является $S(\alpha^*)$.

Решение уравнения $\alpha = S(\alpha)$ будет называться *неподвижной точкой самообучения*. Последующий анализ основан еще на предположении о конечном количестве неподвижных точек алгоритма. Этих *двух предположений достаточно* для доказательства, что в процессе самообучения последовательность $\alpha^1, \alpha^2, \dots, \alpha^t, \dots$ *сходится к одной из неподвижных точек*. Для доказательства этого утверждения потребуется ряд вспомогательных утверждений.

Лемма 6.2 Кульбак. Пусть α_i и x_i , $i = 1, \dots, n$, положительные числа, такие, что $\sum_{i=1}^n \alpha_i = \sum_{i=1}^n x_i = 1$. В таком случае

$$\sum_{i=1}^n \alpha_i \ln \frac{\alpha_i}{x_i} \geq \frac{1}{2} \sum_{i=1}^n (\alpha_i - x_i)^2. \quad (6.34) \quad \blacktriangle$$

Доказательство. Пусть $\delta_i = x_i - \alpha_i$, $i = 1, \dots, n$. Определим две функции φ и ψ от переменной γ , которая принимает значения в интервале $0 \leq \gamma \leq 1$,

$$\varphi(\gamma) = \sum_{i=1}^n \alpha_i \ln \frac{\alpha_i}{\alpha_i + \gamma \delta_i},$$

$$\psi(\gamma) = \frac{1}{2} \sum_{i=1}^n (\gamma \delta_i)^2.$$

Очевидно, что $\varphi(0) = \psi(0) = 0$. Для всех i и γ выполняются неравенства $0 < \alpha_i + \gamma \delta_i \leq 1$. Производные функций φ и ψ по переменной γ равны

$$\frac{d\varphi(\gamma)}{d\gamma} = - \sum_{i=1}^n \frac{\alpha_i \delta_i}{\alpha_i + \gamma \delta_i},$$

$$\frac{d\psi(\gamma)}{d\gamma} = \sum_{i=1}^n \gamma \delta_i^2.$$

Поскольку $\sum_{i=1}^n \delta_i = 0$, справедлива следующая цепочка равенств и неравенств, устанавливающая отношение между этими производными,

$$\frac{d\varphi(\gamma)}{d\gamma} = -\sum_{i=1}^n \frac{\alpha_i \delta_i}{\alpha_i + \gamma \delta_i} + \sum_{i=1}^n \delta_i = \sum_{i=1}^n \frac{\gamma \delta_i^2}{\alpha_i + \gamma \delta_i} \geq \sum_{i=1}^n \gamma \delta_i^2 = \frac{d\psi(\gamma)}{d\gamma}.$$

Неравенство в этой цепочке следует из сказанного ранее утверждения, что число $\alpha_i + \gamma \delta_i$ не превосходит 1 для любых i и γ . Число $\varphi(1)$ равно $\sum_{i=1}^n \alpha_i \log(\alpha_i/x_i)$. Число $\psi(1)$ равно $\frac{1}{2} \sum_{i=1}^n (\alpha_i - x_i)^2$. Отношение этих двух чисел определяет следующая цепочка

$$\varphi(1) = \varphi(0) + \int_0^1 \frac{d\varphi}{dt} dt = \psi(0) + \int_0^1 \frac{d\varphi}{dt} dt \geq \psi(0) + \int_0^1 \frac{d\psi}{dt} dt = \psi(1),$$

из которой следует неравенство (6.34). ■

Лемма 6.3 Кульбака сформулирована и доказана для натуральных логарифмов, так как это делает формулировку более наглядной. Далее мы будем использовать логарифмы при произвольном основании и это не приведет к недоразумениям. Модификация леммы Кульбака для произвольных логарифмов примет вид

$$\sum_{i=1}^n \alpha_i \log \frac{\alpha_i}{x_i} \geq c \sum_{i=1}^n (\alpha_i - x_i)^2,$$

где основание логарифмов влияет лишь на величину постоянного множителя c . В последующем изложении при ссылке на кульбакову лемму имеется в виду только что приведенное неравенство.

Лемма 6.3 Пусть α^t и α^{t+1} - две совокупности, полученные после t -ой и $(t+1)$ -ой итераций самообучения. В этом случае число

$$\sum_i \sum_k (\alpha^t(i, k) - \alpha^{t+1}(i, k))^2 \tag{6.35}$$

стремится к нулю при $t \rightarrow \infty$. ▲

Доказательство. Последовательность $L(m^t)$ монотонно возрастает и не принимает положительные значения. Следовательно, она сходится, откуда следует, что $L(m^{t+1}) - L(m^t)$ стремится к нулю,

$$\lim_{t \rightarrow \infty} (L(m^{t+1}) - L(m^t)) = 0. \tag{6.36}$$

Запишем разность $L(m^{t+1}) - L(m^t)$ более детально, используя разложения, подобные (6.28) и (6.29):

$$\begin{aligned}
& L(m^{t+1}) - L(m^t) \\
&= \sum_k \sum_i \alpha^t(i, k) \log p_K^{t+1}(k) f(x_i, a_k^{t+1}) - \sum_i \sum_k \alpha^t(i, k) \log \alpha^{t+1}(i, k) \\
&\quad - \sum_k \sum_i \alpha^t(i, k) \log p_K^t(k) f(x_i, a_k^t) + \sum_i \sum_k \alpha^t(i, k) \log \alpha^t(i, k) \\
&= \left(\sum_k \sum_i \alpha^t(i, k) \log p_K^{t+1}(k) f(x_i, a_k^{t+1}) - \sum_k \sum_i \alpha^t(i, k) \log p_K^t(k) f(x_i, a_k^t) \right) \\
&\quad + \left(\sum_i \sum_k \alpha^t(i, k) \log \frac{\alpha^t(i, k)}{\alpha^{t+1}(i, k)} \right).
\end{aligned}$$

Разность $L(m^{t+1}) - L(m^t)$ состоит из двух слагаемых, заключенных в большие скобки в последней части приведенной цепочки. Оба слагаемые неотрицательны. Неотрицательность первого слагаемого следует из определений (6.19) и (6.20). Неотрицательность второго слагаемого следует из доказанной ранее леммы 6.1 Шеннона. Так как сумма двух неотрицательных слагаемых стремится к нулю, то каждое из этих слагаемых также стремится к нулю. Для нас важно, что именно второе слагаемое стремится к нулю, то есть

$$\lim_{t \rightarrow \infty} \sum_i \sum_k \alpha^t(i, k) \log \frac{\alpha^t(i, k)}{\alpha^{t+1}(i, k)} = 0.$$

Из этого равенства в силу леммы 6.2 Кульбака следует, что

$$\lim_{t \rightarrow \infty} \sum_i \sum_k (\alpha^t(i, k) - \alpha^{t+1}(i, k))^2 = 0. \quad \blacksquare$$

Лемма 6.4 Пусть S - непрерывная функция от совокупности

$$\alpha = (\alpha(i, k), i = 1, 2, \dots, n, k \in K);$$

$\alpha^1, \alpha^2, \dots, \alpha^t, \dots$ - бесконечная последовательность совокупностей α , получаемая в процессе самообучения, то есть $\alpha^t = S(\alpha^{t-1})$.

Тогда пределом любой сходящейся подпоследовательности

$$\alpha^{t(1)}, \alpha^{t(2)}, \dots, \alpha^{t(j)}, \dots, \quad t(j) > t(j-1),$$

является неподвижная точка самообучения. ▲

Доказательство. Последовательность $\alpha^1, \alpha^2, \dots, \alpha^t, \dots$ есть бесконечная последовательность точек в замкнутом и ограниченном подмножестве линейного пространства. В силу известной теоремы анализа такая последовательность обязательно содержит сходящуюся подпоследовательность $\alpha^{t(1)}, \alpha^{t(2)}, \dots, \alpha^{t(j)}, \dots$. Обозначим ее предел α^* ,

$$\lim_{j \rightarrow \infty} \alpha^{t(j)} = \alpha^*.$$

Построим последовательность $S(\alpha^{t(1)}), S(\alpha^{t(2)}), \dots, S(\alpha^{t(j)}), \dots$, составленную из тех элементов последовательности $\alpha^1, \alpha^2, \dots, \alpha^t, \dots$, которые следуют непосредственно после элементов $\alpha^{t(1)}, \alpha^{t(2)}, \dots, \alpha^{t(j)}, \dots$. Предел последовательности $S(\alpha^{t(j)})$, $j = 1, 2, \dots$, есть также точка α^* , то есть

$$\lim_{j \rightarrow \infty} S(\alpha^{t(j)}) = \alpha^*, \quad (6.37)$$

так как лемма 6.3 утверждает, что $\lim_{t \rightarrow \infty} |\alpha^t - S(\alpha^t)| = 0$. В силу непрерывности преобразования S из (6.37) следует, что $S(\alpha^*) = \alpha^*$. ■

Лемма 6.5 Пусть Ω - множество неподвижных точек самообучения; $\min_{\alpha^* \in \Omega} |\alpha - \alpha^*|^2$ - это расстояние от точки α до множества Ω . Если функция S непрерывна, то

$$\lim_{t \rightarrow \infty} \min_{\alpha^* \in \Omega} |\alpha^t - \alpha^*|^2 = 0. \quad (6.38)$$

▲

Доказательство. Предположим, что утверждение (6.38) ложно. Запишем формально утверждение (6.38) и его отрицание и рассмотрим следствия этого отрицания. Утверждение (6.38) есть лишь краткая запись более детального утверждения

$$\forall \varepsilon > 0 \exists T \forall t > T : \min_{\alpha^* \in \Omega} |\alpha^t - \alpha^*|^2 < \varepsilon,$$

а его отрицание есть утверждение

$$\exists \varepsilon > 0 \forall T \exists t > T : \min_{\alpha^* \in \Omega} |\alpha^t - \alpha^*|^2 \geq \varepsilon. \quad (6.39)$$

Утверждение (6.39) обозначает, что существует такое $\varepsilon > 0$ и такая бесконечная подпоследовательность

$$\alpha^{t(1)}, \alpha^{t(2)}, \dots, \alpha^{t(j)}, \dots, \quad \text{где } t(j) > t(j-1),$$

для которой неравенство

$$\min_{\alpha^* \in \Omega} |\alpha^{t(j)} - \alpha^*|^2 \geq \varepsilon \quad (6.40)$$

справедливо для всех элементов $t(j)$. Так как эта подпоследовательность есть бесконечная подпоследовательность в ограниченном и замкнутом подмножестве, она в свою очередь содержит сходящуюся подпоследовательность. В силу (6.40) предел этой новой подпоследовательности не принадлежит Ω и, таким образом, не является неподвижной точкой алгоритма. Этот вывод противоречит лемме 6.4 и поэтому утверждение (6.39) ложно. ■

Теорема 6.2 О сходимости самообучения. Если функция S , определяющая одну итерацию алгоритма самообучения, непрерывна, а множество неподвижных точек конечно, то последовательность

$$\alpha^1, \alpha^2, \dots, \alpha^t, \dots, \quad \alpha^t = S(\alpha^{t-1}),$$

сходится и ее пределом является неподвижная точка самообучения. \blacktriangle

Доказательство. Обозначим Δ расстояние между двумя ближайшими неподвижными точками. Поскольку количество неподвижных точек конечно, указанный минимум существует и число Δ положительно. Докажем, что событие

$$\operatorname{argmin}_{\alpha^* \in \Omega} |\alpha^t - \alpha^*| \neq \operatorname{argmin}_{\alpha^* \in \Omega} |\alpha^{t+1} - \alpha^*|$$

может произойти в последовательности $\alpha^1, \alpha^2, \dots, \alpha^t, \dots$ лишь конечное количество раз. Предположим, что это не так. Тогда сколь угодно раз будет иметь место ситуация, что расстояние от α^t до ближайшей неподвижной точки будет меньше, чем δ , а на следующем шаге ближайшей к α^{t+1} окажется другая неподвижная точка, тоже удаленная от α^{t+1} меньше, чем на δ . В силу леммы 6.5 эта ситуация будет иметь место при любом, сколь угодно малом δ . Это значит, что бесконечное количество раз произойдет, что расстояние между α^t и α^{t+1} окажется больше, чем $\Delta - 2\delta$. Это противоречит доказанной лемме 6.3, которая утверждает, что расстояние между α^t и α^{t+1} стремится к нулю.

Мы доказали, что начиная с некоторого шага t ближайшей к α^t становится одна и та же неподвижная точка

$$\operatorname{argmin}_{\alpha^* \in \Omega} |\alpha^t - \alpha^*|.$$

Обозначим ее α^{**} и запишем доказанное отношение (6.38) в виде $\lim_{t \rightarrow \infty} |\alpha^t - \alpha^{**}|^2 = 0$, или просто $\lim_{t \rightarrow \infty} \alpha^t = \alpha^{**}$. \blacksquare

При определенных дополнительных ограничениях на модель можно было бы показать, что неподвижные точки самообучения обладают экстремальными свойствами с точки зрения логарифма правдоподобия

$$\sum_i \log \sum_k p_K(k) p(x_i | a_k).$$

Например, можно было бы доказать, что значения $p_K(k)$, a_k , $k \in K$, характеризующие неподвижную точку, являются лучшими в своей окрестности, определенной тем или иным естественным образом. Анализ таких свойств для достаточно обширных классов моделей является не трудным, но трудоемким, и здесь мы его приводить не будем. Мы рекомендуем выполнять анализ этих свойств в каждом отдельном частном случае при рассмотрении конкретного приложения. В качестве примера такого анализа мы рассмотрим частный случай, когда условные вероятности $p_{X|K}(x|k)$ полностью известны, а неизвестны только априорные вероятности $p_K(k)$ скрытых состояний. Этот частный случай ценный сам по себе, так как это задача Роббинса во всей ее общности в исходной роббинсовой постановке. Мы покажем, что при некоторых легко проверяемых условиях самообучение сходится к глобально наиболее правдоподобным оценкам априорных вероятностей $p_K(k)$.

6.6.6 Алгоритм решения задачи Роббинса и ее анализ

Применительно к задачам Роббинса алгоритм самообучения имеет следующий вид. Пусть $p_K^0(k)$, $k \in K$, - начальные значения априорных вероятностей, а $p_K^t(k)$ - их значения после t -ой итерации самообучения. В соответствии с общим алгоритмом самообучения сначала вычисляются числа $\alpha^t(i, k)$, $i = 1, 2, \dots, n$, $k \in K$,

$$\alpha^t(i, k) = \frac{p_K^t(k) p_{X|K}(x_i | k)}{\sum_{k' \in K} p_K^t(k') p_{X|K}(x_i | k')}, \quad (6.41)$$

а затем новые значения $p_K^{t+1}(k)$, $k \in K$,

$$p_K^{t+1}(k) = \frac{\sum_i \alpha^t(i, k)}{n}. \quad (6.42)$$

Мы видим, что алгоритм решение задачи Роббинса сформулирован в явном виде, в отличие от общего алгоритма самообучения, в котором один из этапов сформулирован не явно, а через решение вспомогательной оптимизационной задачи (6.20). Мы видим также, что сам алгоритм, выраженный в виде формул (6.41), (6.42), невероятно прост. Наиболее трудной частью в нем является вычисление вероятностей $p_{X|K}(x_i | k)$. Все остальные вычисления настолько просты, что о них и говорить не стоит. Что касается вероятностей $p_{X|K}(x_i | k)$, то они так или иначе должны вычисляться при собственно распознавании, независимо от того, сопровождается ли распознавание обучением, самообучением или нет. Таким образом, к распознающей системе, решающей байесовскую задачу распознавания, следует добавить лишь самую малость, чтобы она стала способной решать роббинсову задачу.

Представляется правдоподобным, что алгоритм (6.41), (6.42) сходится к глобальному максимуму функции правдоподобия

$$\sum_i \log \sum_k p_K(k) p_{X|K}(x_i | k),$$

так как эта функция вогнута. Алгоритм, сходящийся к локальному максимуму вогнутой функции, обеспечивает ее глобальную максимизацию, так как эта функция имеет лишь один локальный максимум. Эти сугубо предварительные соображения не могут заменить следующую точную формулировку и ее доказательство.

Теорема 6.3 Решение задачи Роббинса в общем виде. Пусть числа $p_K^*(k)$, $k \in K$, являются неподвижной точкой алгоритма (6.41), (6.42), причем ни одно из них не равно 0. Тогда неравенство

$$\sum_{i=1}^n \log \sum_{k \in K} p_K^*(k) p_{X|K}(x_i | k) \geq \sum_{i=1}^n \log \sum_{k \in K} p_K(k) p_{X|K}(x_i | k)$$

справедливо для любых значений априорных вероятностей $p_K(k)$, $k \in K$. ▲

Доказательство. Исключим из формул (6.41), (6.42) вспомогательные переменные $\alpha^t(i, k)$ и выразим зависимость вероятностей $p_K^{t+1}(k)$, $k \in K$, непосредственно от вероятностей $p_K^t(k)$, $k \in K$,

$$p_K^{t+1}(k) = \frac{1}{n} \sum_i \frac{p_K^t(k) p_{X|K}(x_i | k)}{\sum_{k' \in K} p_K^t(k') p_{X|K}(x_i | k')}, \quad k \in K.$$

Подставим в полученное равенство неподвижную точку $p_K^*(k) = p_K^{t+1}(k) = p_K^t(k)$, $k \in K$, и, учитывая, что ни одна из вероятностей $p_K^*(k)$ не равна нулю, получим

$$n = \sum_i \frac{p_{X|K}(x_i | k)}{\sum_{k \in K} p_K^*(k) p_{X|K}(x_i | k)}, \quad k \in K. \quad (6.43)$$

Пусть $p_K(k)$, $k \in K$, - любые неотрицательные числа, сумма которых равна 1. Умножим каждое из равенств семейства (6.43) на число $p_K(k) - p_K^*(k)$, просуммируем их и получим

$$n \sum_{k \in K} (p_K(k) - p_K^*(k)) = \sum_{i=1}^n \frac{\sum_{k \in K} p_K(k) p_{X|K}(x_i | k) - \sum_{k \in K} p_K^*(k) p_{X|K}(x_i | k)}{\sum_{k \in K} p_K^*(k) p_{X|K}(x_i | k)},$$

что эквивалентно равенству

$$\sum_{i=1}^n \frac{\sum_{k \in K} p_K(k) p_{X|K}(x_i | k) - \sum_{k \in K} p_K^*(k) p_{X|K}(x_i | k)}{\sum_{k \in K} p_K^*(k) p_{X|K}(x_i | k)} = 0, \quad (6.44)$$

так как обе суммы $\sum_{k \in K} p_K(k)$ и $\sum_{k \in K} p_K^*(k)$ равны 1.

Совокупность чисел $(p_K(k), k \in K)$ обозначим для краткости p_K и введем обозначение $f_i(p_K)$,

$$f_i(p_K) = \sum_{k \in K} p_K(k) p_{X|K}(x_i | k). \quad (6.45)$$

Равенство (6.44) тогда приобретает более простой вид

$$\sum_{i=1}^n \frac{f_i(p_K) - f_i(p_K^*)}{f_i(p_K^*)} = 0. \quad (6.46)$$

Определим следующую функцию, зависящую от переменной γ ,

$$Q(\gamma) = \sum_{i=1}^n \frac{f_i(p_K) - f_i(p_K^*)}{f_i(p_K^*) + \gamma(f_i(p_K) - f_i(p_K^*))}. \quad (6.47)$$

Ясно, что $Q(0)$ - это левая часть равенства (6.46) и поэтому

$$Q(0) = 0. \quad (6.48)$$

При любом значении γ производная $dQ(\gamma)/d\gamma$ неположительна, так как

$$\frac{dQ(\gamma)}{d\gamma} = \sum_{i=1}^n - \left(\frac{(f_i(p_K) - f_i(p_K^*))^2}{(f_i(p_K^*) + \gamma(f_i(p_K) - f_i(p_K^*)))^2} \right) \leq 0.$$

Это с учетом (6.48) обозначает, что

$$Q(\gamma) \leq 0$$

при любом значении $\gamma \geq 0$. Далее отсюда следует, что $\int_0^1 Q(\gamma) d\gamma$ неположителен. В более подробной записи это обозначает, что (см.(6.47)),

$$\begin{aligned} \int_0^1 Q(\gamma) d\gamma &= \sum_{i=1}^n \log \left(f_i(p_K^*) + \gamma (f_i(p_K) - f_i(p_K^*)) \right) \Big|_{\gamma=0}^1 \\ &= \sum_{i=1}^n \log f_i(p_K) - \sum_{i=1}^n \log f_i(p_K^*) \leq 0. \end{aligned}$$

Еще более подробно, учитывая обозначение (6.45), получаем

$$\sum_{i=1}^n \log \sum_{k \in K} p_K(k) p_{X|K}(x_i | k) \leq \sum_{i=1}^n \log \sum_{k \in K} p_K^*(k) p_{X|K}(x_i | k).$$

Теорема 6.3 доказана. ■

6.7 Обсуждение

Мне кажется, что в лекции отсутствует завершающий аккорд, из которого было бы ясно, что представляет собой сухой выход лекции, а что служит лишь введением к основному результату и его обрамлению. Я принадлежу к той группе читателей, для которых конечный результат значительно важнее истории вопроса и тех путей, которые к нему привели. Мне представляется, что существенная часть лекции ограничивается формулировкой задачи самообучения в подразделе 6.6.3, алгоритмом самообучения в подразделе 6.6.4 и теоремой 6.1 в подразделе 6.6.5. Это сравнительно малая часть лекции. Но в таком случае позволите мне спросить без обиняков, что еще я могу почерпнуть из лекции, и не является ли все остальное ее содержание лишь излишне обширным введением к упомянутым мною основным результатам.

Ответим на Твой вопрос также без обиняков. Безусловно, Тебе следует знать алгоритм самообучения, описанный в подразделе 6.6.4. Мы используем его в последующих лекциях в качестве строительного блока при решении более сложных задач, которые Тебе сейчас еще неизвестны. Однако

вне зависимости от того, видишь ли Ты сейчас, где его применять или нет, этот алгоритм следует знать. В силу своей общности он входит в число фундаментальных результатов современной информатики случайных данных, и его должен знать каждый, кто претендует на профессиональное присутствие в этой области.

Что касается остальной части лекции, то мы согласны с Тобой, что это довольно обширное введение к указанным Тобой результатам. Но мы не так решительны, как Ты, и не считаем, что такое введение лишнее для любого читателя. Все зависит от того, с какой стороны смотреть на результаты, которые следует оценить. Ты ведь знаешь, что любой продукт можно оценивать с точки зрения того, кто этот продукт производит, и с точки зрения его потребителя. Даже на такой сравнительно простой продукт, как бифштекс, едок смотрит совсем иначе, чем повар. Ты должен сам определить свое место в кухне, называемой распознаванием, будет ли это место едока или место повара. В первом случае Тебе, конечно же, нет нужды знать, каким образом сформировались современные представления о самообучении. Во втором случае Ты рано или поздно поймешь, насколько малой может быть эффективность научной работы, насколько мала доля тех результатов, которым удается удержаться в науке, по отношению к массе неудачных попыток и незначительных результатов, которые должен перелопатить исследователь, чтобы получить что-то стоящее. От Тебя потребуются бесконечное терпение, чтобы вынырнуть свои догадки, беспомощные и легко уязвимые при своем зарождении, и вырастить их до научного результата, который может идти в люди и жить там своей собственной жизнью. Чем раньше Ты поймешь все разнообразие рытвин и колдобин в науке, тем лучше для Тебя, хотя это понимание и мало приятное. Мы написали эту лекцию еще и для того, чтобы Ты уже сейчас знал, что Тебя ожидает.

Мы уже Тебя немного знаем как творческую и ищущую личность. Скорее всего, Ты уже сейчас увлечен какими-то неясными предположениями, но не знаешь, с чего даже начать их обоснование. Более того, Ты не знаешь даже, как эти предположения однозначно сформулировать. В этих Твоих затруднениях Тебе никто не поможет. Одни не захотят вникать в Твои расплывчатые идеи, другие же тотчас их конкретизируют, но так, что получится нечто либо тривиальное, либо неосуществимое. В итоге Ты остаешься один на один со своими затруднениями. Зная Тебя, мы достаточно уверенно можем предсказать, что Твоя настойчивость будет вознаграждена и Ты в конце концов сформулируешь задачу, которая окажется и нетривиальной, и разрешимой. Ты ведь не принадлежишь к числу невезучих.

Но тогда наступит второй этап Твоего крестного пути. Пока Ты исследовал свою задачу, Ты прошел ее вдоль и поперек и увидел ее невероятную сложность. Поэтому разработанный Тобой алгоритм представляется Тебе исключительно простым: ведь Ты сравниваешь сложность своего алгоритма со сложностью задачи. Другие же не знают задачу так глубоко, как Ты. Они видят только алгоритм, и им он кажется неуклюжим, громозд-

ким, для его реализации нужна новая аппаратура, которой сейчас в лаборатории нет, и вообще, с точки зрения прикладных проблем, актуальных сейчас для лаборатории, все Твои предложения - это стрельба из пушки по воробьям. Таким будет второй этап жизни Твоей идеи.

Начало третьего этапа ознаменуется случайной встречей с умным собеседником, который проявит такую осведомленность в Твоей задаче и полученных Тобой результатах, будто он сам их получил. Да, он знаком с проблемой, понимает ее сложность и знает, что где-то на рубеже 19-го и 20-го веков этой проблемой уже занимался какой-то математик в Мораве, или Венгрии, или Парагвае. Его результаты сильно опережали свое время и не были восприняты современниками, а сейчас они почти забыты. Тебе стоит с ними познакомиться. Ты погружаешься в публикации многолетней давности и обнаруживаешь, что задолго до Твоего рождения Твои идеи увлекли кого-то так же как и Тебя и он получил почти те же результаты, что и Ты. Правда, есть некоторое отличие полученных Тобой результатов от полученных Твоим предшественником, но новизна касается очень незначительной части результата, а сам положительный результат составляет не более 1 процента от всей выполненной Тобой работы.

Пройдет еще несколько лет, появятся новые вычислительные средства, возрастет научный уровень распознавания, и мало помалу сформулированные Тобой и Твоим давним предшественником задачи приобретут все большую и большую актуальность. Вот тогда и окажется, что именно то новое, что Ты внес в решение задачи и что Тебе самому представлялось малозначительным, именно это определило реализуемость идеи и ее жизнеспособность. Кому-то может показаться, что это и есть Твоя победа. Но мы готовы поспорить, что в это время Ты будешь нуждаться скорее в дружеской поддержке, чем в поздравлениях. Возможно, к этому времени у Тебя еще сохранится наша с Тобой переписка. Тогда прочитай ее еще раз, начиная с того места, где мы не очень серьезно, но и не совсем шутя говорим об открытии Америки, и заканчивая этими строками.

Я не стал ждать так долго и прочитал все прямо сейчас. Сейчас я еще более укрепился во мнении, что все рассуждения исторического характера для меня бесполезны, потому что я уж как-нибудь смогу предотвратить прогнозируемое вами развитие. Более того, я уверен, что другим читателям затронутая вами тема может принести больше вреда, чем пользы. Ведь из каждой строчки нарисованной вами картина сквозит отчаяние. Когда я первый раз читал фрагмент об открытии Америки, я воспринимал ее не более, чем намеренно двусмысленное изложение фактов так, чтобы ради смеха они могли отнестись и к открытию Америки, и к научному исследованию. Однако стоит мне вообразить себя участником упомянутых событий, как меня охватывает ужас. Вы не бойтесь, что читателя, понявшего вашу лекцию именно так, как вы хотите, наука скорее отпугнет, а не привлечет?

Ни в малейшей степени. Во первых, мы будем рады, что такой читатель

своевременно понял, что существует много областей деятельности, которые привлекают его больше, чем наука. Во вторых, мало кто способен принять на веру, что научная карьера настолько неблагодарное занятие. Большинство верит, как и Ты, что как-то им удастся избежать рытвин и колдобин в своей научной карьере. И в третьих, самое грустное. Настоящий драматизм возникает только при рождении результатов действительно высокой научной значимости, а это происходит сравнительно редко. Таким образом, большинство из нас надежно защищено от действительно драматичного сценария научной карьеры.

Я бы хотел вернуться к обсуждению положительных результатов лекции с точки зрения едока. Алгоритм решения задачи Роббинса сформулирован в лекции в явном виде и вполне однозначно. В этом смысле это продукт, готовый для использования. Этого нельзя сказать об общем алгоритме самообучения, который является только полуфабрикатом, так как он не сформулирован явно. Он требует решения вспомогательной оптимизационной задачи, которое должно быть встроено в алгоритм самообучения. Я не вижу достаточно ясно, что мне делать с общей рекомендацией по конструированию алгоритмов самообучения, если равным счетом ничего не сказано, как нужно решать эти вспомогательные задачи. Ведь ниоткуда не следует, что они легче исходной задачи, которая также сформулирована, как оптимизационная.

Конечно, Ты прав. Рассматривая задачу в общем виде, можно лишь сказать, что решение задачи самообучения сводится к решению специальным образом построенных задач обучения с учителем, которые повидимому проще, чем задачи самообучения. Хотя и не удастся более точно сформулировать это утверждение, его справедливость очевидна при решении задачи самообучения для каждого частного класса статистических моделей объекта. Давай рассмотрим некоторые из них.

Пусть k принимает значения 1 или 2, $p_K(k)$ - априорные вероятности, x - одномерная гауссова случайная величина с условным распределением вероятностей $p_{X|k}(x)$, $k = 1, 2$,

$$p_{X|k}(x) = \frac{1}{\sqrt{2\pi}} e^{-\frac{1}{2}(x-\mu_k)^2}.$$

Предположим, что значения $p_K(k)$ и μ_k , $k = 1, 2$, неизвестны и их следует оценить на основе выборки x_1, \dots, x_n , где x_i - реализация случайной величины, имеющей распределение вероятностей $p_K(1)p_{X|1}(x) + p_K(2)p_{X|2}(x)$. Это значит, что следует найти числа $p_K(1), p_K(2), \mu_1$ и μ_2 , максимизирующие число

$$\sum_{i=1}^n \log \left(p_K(1) \frac{1}{\sqrt{2\pi}} e^{-\frac{1}{2}(x_i-\mu_1)^2} + p_K(2) \frac{1}{\sqrt{2\pi}} e^{-\frac{1}{2}(x_i-\mu_2)^2} \right). \quad (6.49)$$

Построй для этого случая алгоритм самообучения в соответствии с рекомендациями лекции.

Действительно, чтобы максимизировать функцию (6.49), я должен решить сравнительно простую задачу максимизации

$$\begin{aligned} \mu_k^* &= \operatorname{argmax}_{\mu} \sum_{i=1}^n \alpha(i, k) \log \frac{1}{\sqrt{2\pi}} e^{-\frac{1}{2}(x_i - \mu)^2} \\ &= \operatorname{argmax}_{\mu} \left(- \sum_{i=1}^n \alpha(i, k) (x_i - \mu)^2 \right) = \operatorname{argmin}_{\mu} \sum_{i=1}^n \alpha(i, k) (x_i - \mu)^2. \end{aligned} \quad (6.50)$$

Так как функция $\sum_{i=1}^n \alpha(i, k) (x_i - \mu)^2$ выпукла по переменной μ , точка минимума μ_k^* находится следующим образом:

$$\begin{aligned} \left. \frac{d \left(\sum_{i=1}^n \alpha(i, k) (x_i - \mu_k)^2 \right)}{d\mu_k} \right|_{\mu=\mu_k^*} &= 0, \\ -2 \sum_{i=1}^n \alpha(i, k) (x_i - \mu_k^*) &= 0, \\ \mu_k^* &= \frac{\sum_{i=1}^n \alpha(i, k) x_i}{\sum_{i=1}^n \alpha(i, k)}. \end{aligned}$$

Процедура максимизации числа (6.49) получает следующий вид. Начальные значения искомым параметров могут быть выбраны, например, $p_K^0(1) = p_K^0(2) = 0.5$, $\mu_1^0 = x_1$, $\mu_2^0 = x_2$. Алгоритм должен последовательно улучшать эти четыре числа. Пусть после t -ой итерации получены числа $p_K^t(1)$, $p_K^t(2)$, μ_1^t , μ_2^t . Новые числа $p_K^{t+1}(1)$, $p_K^{t+1}(2)$, μ_1^{t+1} , μ_2^{t+1} должны вычисляться по следующим явным формулам:

$$\left. \begin{aligned} \alpha(i, 1) &= \frac{p_K^t(1) e^{-\frac{1}{2}(x_i - \mu_1^t)^2}}{p_K^t(1) e^{-\frac{1}{2}(x_i - \mu_1^t)^2} + p_K^t(2) e^{-\frac{1}{2}(x_i - \mu_2^t)^2}}, & i = 1, 2, \dots, n; \\ \alpha(i, 2) &= \frac{p_K^t(2) e^{-\frac{1}{2}(x_i - \mu_2^t)^2}}{p_K^t(1) e^{-\frac{1}{2}(x_i - \mu_1^t)^2} + p_K^t(2) e^{-\frac{1}{2}(x_i - \mu_2^t)^2}}, & i = 1, 2, \dots, n; \\ p_K^{t+1}(1) &= \frac{\sum_{i=1}^n \alpha(i, 1)}{n}; & p_K^{t+1}(2) &= \frac{\sum_{i=1}^n \alpha(i, 2)}{n}; \\ \mu_1^{t+1} &= \frac{\sum_{i=1}^n \alpha(i, 1) x_i}{\sum_{i=1}^n \alpha(i, 1)}; & \mu_2^{t+1} &= \frac{\sum_{i=1}^n \alpha(i, 2) x_i}{\sum_{i=1}^n \alpha(i, 2)}. \end{aligned} \right\} \quad (6.51)$$

Я не указывал верхний индекс t при переменных $\alpha(i, k)$. Само собою разумеется, что они меняются от итерации к итерации.

Ты видишь, что для решения вспомогательной задачи (6.50) Тебе потребовались какие-то три строчки вывода, в то время как исходная задача (6.49) могла бы Тебя и отпугнуть.

Функция (6.49) действительно не очень приятная, но думаю, что при некоторых усилиях я бы смог найти приемлемый алгоритм ее оптимизации.

Нисколько в этом не сомневаемся. Но на основании общих рекомендаций Ты написал алгоритм (6.51) без каких-либо усилий, и это главный сухой выход лекции, если уж хочешь оценивать ее с точки зрения едока. Думаем, что подобным образом Ты сможешь сконструировать алгоритмы самообучения для более общих случаев, например, когда неизвестны еще и условные дисперсии наблюдения, или когда наблюдение является многомерной гауссовой величиной и т.п. Во всех этих случаях Ты увидишь, что вспомогательная задача существенно проще исходной.

Было бы хорошо, если бы Ты просмотрел еще один пример, который довольно прост, но дает повод для серьезных размышлений.

Пусть состояние k - случайная переменная, принимающая, как и прежде, два значения: $k = 1$ с вероятностью $p_K(1)$ и $k = 2$ с вероятностью $p_K(2)$. Пусть x - случайная переменная, принимающая значения из множества X с вероятностями $p_{X|1}(x)$, когда объект находится в первом состоянии, и с вероятностями $p_{X|2}(x)$, когда объект находится во втором состоянии. Пусть y - другая случайная величина, принимающая значения из множества Y с вероятностями $p_{Y|1}(y)$, когда объект находится в первом состоянии, и с вероятностями $p_{Y|2}(y)$, когда объект находится во втором состоянии. Вероятности $p_K(1)$, $p_K(2)$, $p_{X|1}(x)$, $p_{X|2}(x)$, $x \in X$, неизвестны, равно как и вероятности $p_{Y|1}(y)$, $p_{Y|2}(y)$, $y \in Y$. Это значит, что нет никаких априорных знаний о том, как зависят признаки x и y от состояния k . Известно однако, что при условии, что объект находится в первом состоянии, эти признаки не зависят друг от друга, равно как и при условии, что объект находится во втором состоянии. Это значит, что для любой тройки x, y, k , $x \in X$, $y \in Y$, $k = 1, 2$, имеет место равенство

$$p_{XY|k}(x, y) = p_{X|k}(x) p_{Y|k}(y),$$

где $p_{XY|k}(x, y)$ обозначает совместную вероятность значений x и y при условии, что объект находится в состоянии k .

Пусть дана выборка наблюдений $(x_1, y_1), (x_2, y_2), \dots, (x_n, y_n)$. На ее основе требуется найти наиболее правдоподобную модель, то есть числа $p_K(k)$, $p_{X|k}(x)$, $p_{Y|k}(y)$, которые максимизируют логарифм вероятности

$$\sum_{i=1}^n \log \left(\sum_{k=1}^2 p_K(k) p_{X|k}(x_i) p_{Y|k}(y_i) \right).$$

Вспомогательная задача состоит с отыскании чисел $p_{X|k}^*(x)$, $x \in X$, $p_{Y|k}^*(y)$, $y \in Y$, $k = 1, 2$, которые для данной выборки $(x_1, y_1), (x_2, y_2), \dots, (x_n, y_n)$ и

известных чисел $\alpha(i, k)$, $i = 1, \dots, n$, $k = 1, 2$, максимизируют число

$$\sum_{i=1}^n \alpha(i, k) \log \left(p_{X|k}(x_i) p_{Y|k}(y_i) \right). \quad (6.52)$$

Эта задача сравнительно легко решается:

$$\begin{aligned} (p_{X|k}^*, p_{Y|k}^*) &= \operatorname{argmax}_{(p_{X|k}, p_{Y|k})} \sum_{i=1}^n \alpha(i, k) \log p_{X|k}(x_i) p_{Y|k}(y_i) \\ &= \operatorname{argmax}_{(p_{X|k}, p_{Y|k})} \left(\sum_{i=1}^n \alpha(i, k) \log p_{X|k}(x_i) + \sum_{i=1}^n \alpha(i, k) \log p_{Y|k}(y_i) \right) \\ &= \left(\operatorname{argmax}_{p_{X|k}} \sum_{i=1}^n \alpha(i, k) \log p_{X|k}(x_i), \operatorname{argmax}_{p_{Y|k}} \sum_{i=1}^n \alpha(i, k) \log p_{Y|k}(y_i) \right) \\ &= \left(\operatorname{argmax}_{p_{X|k}} \sum_{x \in X} \sum_{i \in I_X(x)} \alpha(i, k) \log p_{X|k}(x), \operatorname{argmax}_{p_{Y|k}} \sum_{y \in Y} \sum_{i \in I_Y(y)} \alpha(i, k) \log p_{Y|k}(y) \right) \\ &= \left(\operatorname{argmax}_{p_{X|k}} \sum_{x \in X} \left(\sum_{i \in I_X(x)} \alpha(i, k) \right) \log p_{X|k}(x), \right. \\ &\quad \left. \operatorname{argmax}_{p_{Y|k}} \sum_{y \in Y} \left(\sum_{i \in I_Y(y)} \alpha(i, k) \right) \log p_{Y|k}(y) \right). \end{aligned}$$

Первое равенство в приведенной цепочке есть лишь повторение формулировки (6.52) вспомогательной задачи. Второе равенство есть эквивалентная запись логарифма произведения. Третье равенство имеет место в силу того, что требуется максимизировать сумму двух слагаемых, каждое из которых зависит от своих собственных переменных. Поэтому максимизация суммы достигается независимой максимизацией слагаемых. В четвертом равенстве использовано обозначение $I_X(x)$ для множества тех индексов i , при которых x_i равно x . Аналогичный смысл имеет обозначение $I_Y(y)$. Затем слагаемые $\alpha(i, k) \log p_{X|k}(x_i)$ сгруппированы так, что сначала производится суммирование по тем i , для которых признак x_i принял определенное значение x , а затем - суммирование по всем значениям x . Суммирование $\sum_{i=1}^n$ представлено, таким образом, в виде

$$\sum_{x \in X} \sum_{i \in I_X(x)} \quad \text{или} \quad \sum_{y \in Y} \sum_{i \in I_Y(y)} .$$

И наконец, последнее равенство основано на том, что значения $p_{X|k}(x)$ и $p_{Y|k}(y)$ в суммах

$$\sum_{i \in I_X(x)} \alpha(i, k) \log p_{X|k}(x) \quad \text{и} \quad \sum_{i \in I_Y(y)} \alpha(i, k) \log p_{Y|k}(y)$$

не зависят от индекса i , по которому производится суммирование, и их можно вынести за знак суммирования \sum_i .

В соответствии с леммой 6.1 числа $p_{X|k}^*(x)$ и $p_{Y|k}^*(y)$, которые максимизируют суммы

$$\sum_{x \in X} \left(\sum_{i \in I_X(x)} \alpha(i, k) \right) \log p_{X|k}(x) \quad \text{и} \quad \sum_{y \in Y} \left(\sum_{i \in I_Y(y)} \alpha(i, k) \right) \log p_{Y|k}(y),$$

а следовательно, и сумму (6.52), равны

$$p_{X|k}^*(x) = \frac{\sum_{i \in I_X(x)} \alpha(i, k)}{\sum_{x \in X} \sum_{i \in I_X(x)} \alpha(i, k)},$$

$$p_{Y|k}^*(y) = \frac{\sum_{i \in I_Y(y)} \alpha(i, k)}{\sum_{y \in Y} \sum_{i \in I_Y(y)} \alpha(i, k)}.$$

Алгоритм решения исходной оптимизационной задачи выражается в следующем явном виде. Пусть $p_K^t(k)$, $p_{X|k}^t(x)$, $p_{Y|k}^t(y)$, $k = 1, 2$, $x \in X$, $y \in Y$, - оценки неизвестных вероятностей, полученные после t -ой итерации самообучения. Новые оценки получают по формулам

$$\alpha(i, 1) = \frac{p_K^t(1) p_{X|1}^t(x_i) p_{Y|1}^t(y_i)}{p_K^t(1) p_{X|1}^t(x_i) p_{Y|1}^t(y_i) + p_K^t(2) p_{X|2}^t(x_i) p_{Y|2}^t(y_i)};$$

$$\alpha(i, 2) = \frac{p_K^t(2) p_{X|2}^t(x_i) p_{Y|2}^t(y_i)}{p_K^t(1) p_{X|1}^t(x_i) p_{Y|1}^t(y_i) + p_K^t(2) p_{X|2}^t(x_i) p_{Y|2}^t(y_i)};$$

$$p_K^{t+1}(1) = \frac{\sum_{i=1}^n \alpha(i, 1)}{n}; \quad p_K^{t+1}(2) = \frac{\sum_{i=1}^n \alpha(i, 2)}{n};$$

$$p_{X|1}^{t+1}(x) = \frac{\sum_{i \in I_X(x)} \alpha(i, 1)}{\sum_{i=1}^n \alpha(i, 1)}; \quad p_{X|2}^{t+1}(x) = \frac{\sum_{i \in I_X(x)} \alpha(i, 2)}{\sum_{i=1}^n \alpha(i, 2)};$$

$$p_{Y|1}^{t+1}(y) = \frac{\sum_{i \in I_Y(y)} \alpha(i, 1)}{\sum_{i=1}^n \alpha(i, 1)}; \quad p_{Y|2}^{t+1}(y) = \frac{\sum_{i \in I_Y(y)} \alpha(i, 2)}{\sum_{i=1}^n \alpha(i, 2)}.$$

Так же, как и в предыдущем упражнении (6.51), я опустил обозначения верхних индексов t в переменных $\alpha(i, k)$. Ясно, что они меняются от итерации к итерации.

Я не думаю, что где-то допустил ошибку, так как для вывода этого алгоритма я не использовал никаких сложных математических средств. Весь вывод основан на рекомендациях лекции и леммы 6.1, в которой, кстати, тоже нет ничего сложного. Но что-то в конечном полученном мною результате меня смущает. Меня удивляет, что самой трудоемкой частью алгоритма оказывается вычисление "апостериорных вероятностей" $\alpha(i, k)$, то есть та часть алгоритма, которой невозможно избежать независимо от того, идет ли речь о самообучении, или обучении с учителем, или распознавании без обучения. Та часть, которая превращает необучаемую систему в

самообучающуюся, настолько проста, что не стоит и речи. Вот эта простота и заставляет меня думать, что что-то очень важное ускользает от меня. Ведь при всей своей простоте алгоритм претендует на решение очень амбициозной задачи.

Речь идет об анализе поведения некоторой ненаблюдаемой случайной величины, то есть скрытого состояния k . Вообще говоря, в этом нет ничего необычного, если известно, как от этого скрытого параметра зависят другие, наблюдаемые параметры. При определенном упрощении можно считать, что это давно известная проблема косвенных измерений. Но здесь мы имеем принципиально иную ситуацию. Мы имеем два признака, x и y , о которых известно, что они как-то зависят от скрытого состояния, но ровным счетом ничего не известно о том, какова эта зависимость. И теперь вы утверждаете, что на основании наблюдения только этих двух признаков можно вскрыть, как они зависят от того, что никогда не наблюдалось. Более того, можно оценить статистические характеристики того, что никогда не наблюдалось. Как такие вещи вообще могут происходить? Действительно ли все это самообучение, а не самообман? Кстати, сам Роббинс в статье [Robbins, 1951] остроумно и самокритично (а он мог себе это позволить) сравнил свой подход с попыткой вытянуть себя самого за волосы. Мне трудно сформулировать свой вопрос конкретно, но обсуждение моего несформулированного вопроса было бы для меня очень полезно.

Ты прав, здесь все представляется бессмыслицей, но только на первый и второй взгляд. На третий и четвертый взгляд здесь обнаруживается определенный смысл, и он далеко не тривиален. Ты не совсем прав, когда говоришь, что нет никаких априорных знаний о зависимости признаков x и y от состояния k . Хотя и ничего неизвестно о том, как каждый в отдельности признак зависит от состояния, известно что-то очень существенное о их совместной зависимости от состояния. Именно, известно, что признаки x и y зависят от состояния k независимо друг от друга. Говоря иными словами, если бы состояние k не менялось, то есть было бы фиксированным, то признаки были бы независимы друг от друга. При меняющемся же состоянии они становятся зависимыми, но лишь постольку, поскольку оба они зависят от состояния.

Основной барьер в понимании самообучения состоит в том, что речь идет об эмпирическом исследовании взаимной зависимости двух величин в условиях, когда одна из них не наблюдаема. Мы с Тобой должны этот барьер преодолеть, признав, что несмотря на свою парадоксальность, такое исследование не является невозможным. Вся интеллектуальная деятельность как отдельных мыслящих существ, так и больших их сообществ состоит именно в исследовании таких характеристик, которые недоступны для непосредственного наблюдения. Можно было бы упомянуть такие грандиозные макропонятия, как добро и зло, но не будем входить глубоко в этом направлении. Выберем для наших рассуждений значительно более понятный параметр, каким является, например, температура объекта, понимаемая, как сумма квадратов скоростей молекул этого объекта. Хо-

тя эту температуру никто никогда не наблюдал, сейчас достаточно хорошо известно, как от этого параметра зависит объем тела, его яркость, как сама температура зависит от температуры окружения, кстати, тоже ненаблюдаемой. Обрати внимание еще и на то, что все эти свойства ненаблюдаемой (!!!) температуры были хорошо изучены задолго до понимания, что температура - это одна из интегральных характеристик скоростей большой совокупности молекул.

Введение в рассмотрение ненаблюдаемого параметра объекта есть не что иное, как анализ его наблюдаемых параметров и поиск механизма, объясняющего их взаимосвязь. Это значит, что взаимную зависимость наблюдаемых разрозненных признаков невозможно объяснить иначе (или проще), чем существованием невидимого фактора, влияющего на все видимые признаки и таким образом обуславливающего их взаимную зависимость. Все это - обычный метод исследования неизвестных явлений. Способность именно к такого рода исследованиям всегда считалась критерием интеллектуальности людей или существ, претендующих на интеллектуальность.

Можно это понимать как определенную декорреляцию признаков?

Словечко "декорреляция" здесь подошло бы идеально, если бы оно не было уже занято для обозначения обработок совсем другого типа, когда на основе анализа ковариационной матрицы признаков исходные признаки преобразуют в другие так, чтобы они оказались некоррелированными. Это преобразование называют еще разложением Карунена-Лоэва. Таким образом, нам с Тобой придется найти какое-то другое название для поиска скрытого параметра, который определяет зависимость между наблюдаемыми параметрами.

Я понял, что задачи, которые мы сейчас рассматриваем, не так уж бессмысленны, как это кажется на первый или второй взгляд. Но мы еще не дошли до действительной глубины, которая, как Вы сказали, открывается лишь на третий или четвертый взгляд.

Пусть $x, y, k = 1, 2$, три случайные величины, совместные вероятности $p_{XYK}(x, y, k)$ которых имеют вид произведения

$$p_K(k) p_{X|K}(x|k) p_{Y|K}(y|k).$$

Пусть дана выборка пар (x, y) , которая достаточно длинная, так что вероятность $p_{XY}(x, y)$ каждой пары (x, y) может быть оценена достаточно точно. Эти эмпирические данные находятся в отношении

$$p_{XY}(x, y) = \sum_{k=1}^2 p_K(k) p_{X|K}(x|k) p_{Y|K}(y|k) \quad (6.53)$$

с неизвестными вероятностями $p_K(k), p_{X|K}(x), p_{Y|K}(y)$. Именно эти вероятности нас интересуют, но их невозможно получить по наблюдениям, так

как параметр k непосредственно не наблюдается. Предположим, что мы нашли значения $p'_K(k)$, $p'_{X|K}(x|k)$, $p'_{Y|K}(y|k)$, приемлемые в том смысле, что они удовлетворяют условию

$$p_{XY}(x, y) = \sum_{k=1}^2 p'_K(k) p'_{X|K}(x|k) p'_{Y|K}(y|k) \quad (6.54)$$

и, таким образом, служат интерпретацией, объяснением эмпирических наблюдений, выраженных числами $p_{XY}(x, y)$. И сейчас возникает вопрос: в какой мере произвольные числа $p'_K, p'_{X|K}, p'_{Y|K}$, удовлетворяющие условию (6.54), соответствуют действительным числам $p_K, p_{X|K}, p_{Y|K}$?

Твой вопрос уместен и известен в литературе, как вопрос об идентифицируемости смеси. Условие (6.53) не всегда позволяет однозначно определить функции $p_{X|K}$ и $p_{Y|K}$ и числа $p_K(1)$ и $p_K(2)$ по эмпирически наблюдаемым вероятностям $p_{XY}(x, y)$. Все зависит от того, каковы в действительности функции $p_{X|K}$ и $p_{Y|K}$ и числа $p_K(1)$ и $p_K(2)$. В одних случаях вряд ли что-то можно сказать об искомым функциях. Однако, как мы увидим, это такие вырожденные случаи, что их можно не принимать во внимание. Это случаи, когда признаки не зависят от состояния. Даже если бы искомые функции были известны, на их основе невозможно было бы распознавать. В других, более интересных случаях можно вскрыть зависимость признаков от состояния с некоторой остаточной неопределенностью, которая, однако, не мешает решению практических задач. И наконец, при определенных ограничениях, не слишком обременительных с практической точки зрения, зависимость признаков от состояния можно определить однозначно.

Опишем один из возможных методов определения функций $p'_K, p'_{X|K}, p'_{Y|K}$ по вероятностям $p_{XY}(x, y)$. Ты ни в коем случае не должен понимать этот метод, как практическую рекомендацию. Он служит не для вычислений, а для объяснения, для более ясного понимания, каким образом отношение (6.53) определяет искомые функции. Наилучшим для практического использования является полученный Тобоем алгоритм. Однако, судя по Твоему недоумению, по этому алгоритму не становится достаточно ясным, каким образом информация об искомым статистических зависимостях содержится в эмпирических наблюдениях и извлекается из них.

На основании эмпирически полученных вероятностей $p_{XY}(x, y)$, $x \in X$, $y \in Y$, можно вычислить вероятности

$$\frac{p_{XY}(x, y)}{\sum_{x \in X} p_{XY}(x, y)}, \quad x \in X, \quad y \in Y,$$

то есть условные вероятности $p_{X|Y}(x|y)$ значения x первого признака при условии, что второй признак принял значение y . Как и прежде, выразим функцию $p_{X|Y}$ двух переменных x и y в виде семейства функций $p_{X|y}, y \in Y$, одной переменной x . Если отождествить каждую функцию

$p_{X|y}$ этого семейства с точкой $|X|$ -мерного линейного пространства, можно немедленно заметить, что все функции $p_{X|y}$, $y \in Y$, лежат на прямой линии, проходящей через точки $p_{X|1}$, $p_{X|2}$. Действительно, точка $p_{X|y}$ есть выпуклая линейная комбинация точек $p_{X|1}$ и $p_{X|2}$,

$$p_{X|y}(x) = p_{K|y}(1) p_{X|1}(x) + p_{K|y}(2) p_{X|2}(x), \quad y \in Y, \quad x \in X, \quad (6.55)$$

где $p_{K|y}(k)$ - апостериорная вероятность состояния k при наблюдении y . Обозначим эту прямую Γ . Прямая Γ непосредственно определяет зависимость наблюдаемых признаков x и y от скрытого состояния k . Подумай над этим хорошенько, прежде, чем мы продолжим.

В определенных случаях прямую Γ можно однозначно определить, не зная функций $p_{X|1}$ и $p_{X|2}$, а на основе одних лишь эмпирических наблюдений $p_{XY}(x, y)$. Если множество $\{p_{X|y} | y \in Y\}$ содержит более чем одну функцию, то любая пара не равных друг другу функций однозначно определяет прямую Γ . Если же множество $\{p_{X|y} | y \in Y\}$ содержит лишь одну функцию, то прямая Γ не определяется однозначно. Но это значит, что все функции $p_{X|y}$, $y \in Y$, одинаковы. Это и есть тот вырожденный случай, о котором мы говорили раньше. Рассмотрим его более детально.

Функция $p_{X|y}$ - одна и та же для всех значений y в следующих трех случаях (все время имей в виду (6.55)):

1. Функции $p_{X|1}$ и $p_{X|2}$ равны друг другу; в этом случае функция $p_{X|y}$ не зависит от вероятностей $p_{K|y}(2)$ и, следовательно, множество $\{p_{X|y} | y \in Y\}$ содержит лишь одну функцию.
2. Функции $p_{Y|1}$ и $p_{Y|2}$ равны друг другу; в этом случае апостериорные вероятности $p_{K|y}(k)$ не зависят от y и множество $\{p_{X|y} | y \in Y\}$ опять содержит лишь одну функцию.
3. Одна из априорных вероятностей $p_K(1)$ или $p_K(2)$ равна нулю; в этом случае вероятности $p_{K|y}(k)$ не зависят от y и, более того, одна из них все время равна нулю.

Все эти случаи вырождены в том смысле, что по крайней мере один из признаков не несет никакой информации о состоянии объекта. Рассмотрим далее нормальную ситуацию, что каждый из признаков x и y зависит от состояния, но априори неизвестно, как именно, и наша задача состоит в обнаружении этой зависимости по эмпирическим данным. В этой ситуации функция $p_{X|y}$ зависит от y , а это значит, что множество $\{p_{X|y} | y \in Y\}$ содержит в себе более чем одну функцию и прямая Γ определяется однозначно. Искомые функции $p_{X|1}$ и $p_{X|2}$ уже не могут быть любыми. Это могут быть лишь точки на прямой Γ .

Предположим на время, что положение точек $p_{X|1}$ и $p_{X|2}$ на прямой Γ известно. В этом случае можно было бы ввести координатную систему на этой прямой (состоящую всего лишь из одной координаты) так, что точка $p_{X|1}$ имеет координату 1, а точка $p_{X|2}$ - координату 0. В таком случае, в соответствии с выражением (6.55) можно утверждать, что координата точки $p_{X|y}$ - это апостериорная вероятность $p_{K|y}(1)$ первого состояния при условии наблюдения y . Если же точки $p_{X|1}$ и $p_{X|2}$ неизвестны, то, выбрав

любую систему координат, можно утверждать, что координата точки $p_{X|y}$ монотонно зависит от апостериорной вероятности $p_{K|y}(1)$ и отличается от нее лишь постоянным слагаемым и постоянным сомножителем, возможно, отрицательным.

Таким образом мы убедились, что множество Y наблюдений y можно естественно упорядочить в соответствии с положением точки $p_{X|y}$ на прямой Γ и это упорядочение совпадает с упорядочением в соответствии с апостериорной вероятностью $p_{K|y}(1)$ первого состояния. Отсюда следует, что байесова стратегия принятия решения о состоянии по признаку y имеет следующий вид. Следует выбрать определенное пороговое значение Θ в упорядоченном множестве Y . Для значений y , предшествующих этому порогу, принимается решение в пользу одного состояния объекта, а для всех других значений y - в пользу другого. Предшествование здесь понимается в соответствии с принятым упорядочением множества Y . Наиболее существенно здесь то, что для этого упорядочивания нет нужды знать функции $p_{X|1}$ и $p_{X|2}$, равно как и функции $p_{Y|1}$ и $p_{Y|2}$. Для указанного упорядочивания нужно знать только, как функции $p_{X|y}$ расположены на прямой Γ . А для этого нужно располагать только эмпирическими данными $p_{XY}(x, y)$.

Можно оценить, какое количество информации содержится в эмпирических наблюдениях об истинной стратегии распознавания состояния по признаку y . Пусть n - количество значений признака y . Множество Y можно разделить на два подмножества 2^n способами. Информацию об этом разделении можно выразить с помощью n бит, которые можно понимать как n бинарных ответов учителя о принадлежности каждого из n наблюдений к тому или иному подмножеству.

После надлежащего анализа эмпирических данных исключается подавляющее большинство этих 2^n классификаций и остаются только $2n$ классификаций, которые согласуются с эмпирическими данными. Чтобы выбрать одну из этих оставшихся, требуется только $1 + \log_2 n$ бит. Эту недостающую информацию следует извлечь из дополнительных источников, скажем, от учителя, который в этом случае должен указать требуемый ответ не для каждого из n возможных наблюдений, а лишь для специально выбранных $1 + \log_2 n$ наблюдений. Мы видим, что подавляющую часть информации о стратегии распознавания содержится уже в эмпирических данных, то есть в наблюдениях, предъявленных для распознавания. В этом и состоит самообучение.

Отметь, что даже при полностью известных функциях $p_{Y|1}$, $p_{Y|2}$ и числах $p_K(1)$, $p_K(2)$ стратегия распознавания состояния по признаку y не определяется однозначно. В этом случае можно говорить лишь о совокупности из $2n$ стратегий, каждая из которых может иметь место при определенной функции потерь. Таким образом, в эмпирических наблюдениях содержится не вся информация о требуемой стратегии распознавания, а только определенная очень значительная ее часть, но это именно та часть, которая бы имела в распоряжении, если бы статистическая модель распознаваемого объекта была полностью известна.

Рассмотрим теперь ситуацию, когда статистическая модель объекта

должна быть определена не обязательно ради последующего распознавания, а сама по себе. Когда такая модель определяется однозначно по эмпирическим наблюдениям? Мы укажем дополнительные ограничения, при которых система уравнений

$$p_{XY}(x, y) = \sum_{k=1}^2 p_K(k) p_{X|k}(x) p_{Y|k}(y), \quad x \in X, \quad y \in Y,$$

имеет единственное решение по функциям $p_K, p_{X|k}, p_{Y|k}$. Естественно, мы будем говорить о единственности модели с точностью до переименования состояний k . Говоря точнее, две модели $p_K, p_{X|k}, p_{Y|k}$ и $p'_K, p'_{X|k}, p'_{Y|k}$ будут считаться идентичными, если

$$\begin{aligned} p_K(1) &= p'_K(2), & p_K(2) &= p'_K(1), \\ p_{X|1} &= p'_{X|2}, & p_{X|2} &= p'_{X|1}, \\ p_{Y|1} &= p'_{Y|2}, & p_{Y|2} &= p'_{Y|1}. \end{aligned}$$

Дополнительно к условиям $p_K(1) \neq 0, p_K(2) \neq 0, p_{X|1} \neq p_{X|2}$, рассмотренным в предыдущем анализе, введем дополнительное условие, которое назовем существованием идеальных представителей. Предполагается, что существует такое значение y_1 признака y , который может иметь место только при условии, что объект находится в первом состоянии. Кроме того, предполагается существование такого значения y_2 , которое имеет ненулевую вероятность только при условии пребывания объекта во втором состоянии. Это значит, что

$$p_{Y|1}(y_1) \neq 0, \quad p_{Y|2}(y_1) = 0, \quad p_{Y|1}(y_2) = 0, \quad p_{Y|2}(y_2) \neq 0. \quad (6.56)$$

Из условия (6.56) следует, что

$$p_{K|y_1}(1) = 1, \quad p_{K|y_1}(2) = 0, \quad p_{K|y_2}(1) = 0, \quad p_{K|y_2}(2) = 1,$$

и таким образом, на основании (6.55) выполняется

$$p_{X|y_1} = p_{X|1}, \quad p_{X|y_2} = p_{X|2}. \quad (6.57)$$

Предположение об идеальных представителях обозначает лишь существование этих представителей, а не знание, каковы именно эти представители.

Если предположение об идеальных представителях выполняется, то функции $p_{X|1}$ и $p_{X|2}$, равно как и сами представители, могут быть восстановлены следующим простым способом. Функциями $p_{X|1}$ и $p_{X|2}$ могут быть лишь первая и последняя функция в множестве $\{p_{X|y} \mid y \in Y\}$, упорядоченного в соответствии с расположением его элементов вдоль прямой Γ . Функции $p_{X|1}$ и $p_{X|2}$, таким образом, определяются однозначно с точностью до переименования состояний k . Подобным образом можно восстановить функции $p_{Y|K}$ и p_K , если предположить существование в множестве X идеальных значений признака x .

Надеюсь, что я понял главное ядро ваших рассуждений. Оно состоит в том, что множество функций $\{p_{X|y} \mid y \in Y\}$ не может быть любым, а может быть только подмножеством точек на некоторой одномерной прямой. Я понимаю также, что это свойство легко может быть обобщено на случай, когда количество состояний объекта не обязательно равно двум. Если количество состояний равно n , то множество функций $\{p_{X|y} \mid y \in Y\}$ полностью содержится в некоторой $(n - 1)$ -мерной гиперплоскости, в которой содержатся и искомые, априори неизвестные функции $p_{X|k}$, $k \in K$.

Ты совершенно прав! Главная идея именно в этом!

Об этом следует еще достаточно долго размышлять, но я не буду сейчас затруднять вас этим. Сейчас я бы хотел использовать время, оставшееся для консультации с вами, чтобы выяснить еще один важный для меня вопрос.

Я понимаю, что рассуждения о том, как можно было бы восстановить функции p_K , $p_{X|K}$ и $p_{Y|K}$, ни в коей мере не претендуют на практическую применимость. Они имеют объяснительное, а не вычислительное назначение. Тем не менее я бы хотел перейти от этих идеальных представлений к реалистичной ситуации и вижу здесь следующую основную трудность. В идеальной ситуации можно найти прямую Γ , выбрав любые две различные точки $p_{X|y'}$, $p_{X|y''}$, потому что ВСЕ точки $p_{X|y}$ лежат на этой прямой. Но если эмпирические данные конечны, а они всегда конечны, то вероятности $p_{X|y}(x)$ не могут считаться известными. Известны лишь некоторые другие числа $p'_{X|y}(x)$, указывающие, как часто наблюдение x имело место одновременно с наблюдением y . Множество $\{p'_{X|y} \mid y \in Y\}$ функций, полученных таким путем, совсем необязательно лежит на одной прямой. Тогда и прямые линии, проведенные через две разные точки, не будут совпадать. И тогда задача отыскания прямой Γ становится недостаточно ясной, так как неясно, о какой собственно прямой идет речь. Здесь необходимо сначала дать численное определение, насколько та или иная прямая хорошо аппроксимирует заданное множество $\{p'_{X|y} \mid y \in Y\}$ функций. Затем следует формулировать задачу отыскания прямой Γ , наилучшим образом аппроксимирующей эмпирически наблюдаемые функции $p'_{X|y} \mid y \in Y$.

Только сейчас Ты пришел к пониманию задач, которые фактически были сформулированы и решены в лекции, когда самообучение было определено, как поиск модели, которая наилучшим образом аппроксимирует эмпирические данные, то есть конечную выборку наблюдений. Видя, что Ты пришел к этому пониманию, нам не совсем ясно, в чем же заключается Твой вопрос. То, о чем Ты спрашиваешь, уже было в лекции.

Постараюсь объяснить свои затруднения более аккуратно. С одной стороны, мы имеем дело с поиском чисел $p_K(k)$, $p_{X|K}(x|k)$, $p_{Y|K}(y|k)$, где $k = 1, 2$,

$x \in X, y \in Y$, которые максимизируют число

$$\sum_{i=1}^n \log \sum_{k=1}^2 p_K(k) p_{X|K}(x_i|k) p_{Y|K}(y_i|k)$$

при заданной последовательности (x_i, y_i) , $i = 1, 2, \dots, n$. Это значит, что следует найти

$$\operatorname{argmax}_{(p_K, p_{X|K}, p_{Y|K})} \sum_{i=1}^n \log \sum_{k=1}^2 p_K(k) p_{X|K}(x_i|k) p_{Y|K}(y_i|k). \quad (6.58)$$

Я чисто формально написал алгоритм решения этой задачи, как частного случая более общей задачи, которая, опять-таки формально, была сформулирована и решена в лекции. В лабиринте этих формальностей от меня скрылась главная идея того, как это может происходить, что можно изучать статистические свойства величин, которые никогда не наблюдались. После ваших объяснений ситуация стала чуть более ясной, но только для идеального случая, когда выборка наблюдений бесконечно большая. Эта ясность состоит в понимании, что при заданных числах $p_{XY}(x, y)$ решение системы уравнений

$$p_{XY}(x, y) = \sum_{k=1}^2 p_K(k) p_{X|K}(x|k) p_{Y|K}(y|k), \quad x \in X, \quad y \in Y, \quad (6.59)$$

по функциям $p_K, p_{X|K}, p_{Y|K}$ не может быть произвольным. Основным фактором, который внес здесь ясность, оказалась прямая Γ , построенная определенным образом. Но эту прямую никак не видно в формулировке (6.58). Поэтому мне трудно сейчас перенести свое понимание, достигнутое с вашей помощью для идеального случая (6.59), на реальную ситуацию, выраженную требованием (6.58). Сейчас мне следовало бы перебросить мостик от идеальных требований (6.59), которые я сейчас хорошо понимаю, к реальной задаче, которую я понимаю чисто формально. Но когда я буду пробиваться от задачи (6.59) к задаче (6.58), я бы не хотел потерять из виду прямую Γ , которая сейчас служит единственной опорной точкой для моего понимания.

Мне кажется, что я вижу первый шаг на этом пути. В реальной ситуации система уравнений (6.59) не имеет решения. Прямая Γ , которая в идеальном случае имеет вполне наглядный смысл, в этом случае просто не существует. Задачу (6.59) следует переформулировать так, чтобы прямая Γ определялась и в том случае, когда совокупность функций $p_{X|y}$, $y \in Y$, не лежит на одной прямой. Можете ли вы мне помочь сделать этот первый шаг, но так, чтобы прямая Γ не исчезла из рассмотрения?

Думаем, что можем. Предварительная формулировка могла бы иметь следующий вид. Пусть $(p'_{X|y} | y \in Y)$ - исходная совокупность точек, которые не лежат на одной прямой. Следует найти другую совокупность

точек $(p_{X|y} \mid y \in Y)$, лежащих на одной прямой, так, чтобы найденная совокупность была максимально "похожа" на исходную. Естественно было бы определить сходство двух совокупностей, как сумму сходств их элементов, то есть функцией вида

$$\sum_{y \in Y} p'_Y(y) L(p'_{X|y}, p_{X|y}), \quad (6.60)$$

где $L(p'_{X|y}, p_{X|y})$ - "сходство" функций $p'_{X|y}$ и $p_{X|y}$, а число $p'_Y(y)$ определяет частоту значения y в конечной выборке (x_i, y_i) , $i = 1, \dots, n$,

$$p'_Y(y) = \sum_{x \in X} p'_{XY}(x, y).$$

Число $p'_{XY}(x, y)$ определяет в свою очередь частоту пары (x, y) . Напомним, что

$$p'_{X|y}(x) = \frac{p'_{XY}(x, y)}{p'_Y(y)}.$$

Рассмотрим теперь, как можно было бы определить сходство L функций $p'_{X|y}$ и $p_{X|y}$. Функция $p'_{X|y}$ - это результат конечного наблюдения объекта, а $p_{X|y}$ - статистическая модель. Сходством результата $p'_{X|y}$ с моделью $p_{X|y}$ естественно считать логарифм вероятности этого результата,

$$L(p'_{X|y}, p_{X|y}) = \sum_{x \in X} p'_{X|y}(x) \log p_{X|y}(x). \quad (6.61)$$

Прямая Γ , которую Ты бы ни в коем случае не хотел упустить из виду, присутствует в следующей постановке задачи.

Следует найти функции $p_{X|y}$, $y \in Y$, лежащие на одной прямой (заранее не заданной) так, чтобы максимизировалось число (6.60), которое с учетом определения (6.61) равно

$$\sum_{y \in Y} p'_Y(y) \sum_{x \in X} p'_{X|y}(x) \log p_{X|y}(x), \quad (6.62)$$

где p'_Y и $p'_{X|y}$ - эмпирические данные, полученные по конечной выборке наблюдений.

Ты видишь, что число (6.62) напоминает число (6.58), к которому мы стараемся перебросить мостик. Прямую линию Γ , о которой идет речь в формулировке задачи (6.62), определяют точки $p_{X|1}$ и $p_{X|2}$, лежащие на этой прямой, а положение точек $p_{X|y}$, $y \in Y$, на этой прямой определяют числа $p_{K|y}(1)$ и $p_{K|y}(2)$. Используя функции $p_{X|1}$ и $p_{X|2}$ и числа $p_{K|y}(1)$ и $p_{K|y}(2)$, можно заменить функцию $p_{X|y}(x)$ в (6.62) эквивалентным выражением

$$p_{X|y} = p_{K|y}(1) p_{X|1} + p_{K|y}(2) p_{X|2} = \sum_{k=1}^2 p_{K|y}(k) p_{X|k}(x).$$

Таким образом, число (6.62), которое следует максимизировать, равно

$$\sum_{y \in Y} p'_Y(y) \sum_{x \in X} p'_{X|y} \log \sum_{k=1}^2 p_{K|y}(k) p_{X|k}(x). \quad (6.63)$$

Поиск наилучшей совокупности $(p_{X|y}, y \in Y)$ следует сейчас понимать так, что отыскивается положение прямой Γ (выраженной числами $p_{X|K}(x|k)$, $x \in X$, $k = 1, 2$) и положение точки $p_{X|y}$ для каждого $y \in Y$ на этой прямой (выраженное парой чисел $p_{K|y}(1)$ и $p_{K|y}(2)$). Ты видишь теперь, что максимизация (6.63) по числам $p_{K|y}(k), p_{X|K}(x|k)$, $x \in X$, $y \in Y$, $k = 1, 2$, есть не что иное, как поиск прямой Γ , которая в определенном смысле хорошо аппроксимирует эмпирические данные $(p'_{X|y}, y \in Y)$. Прямая Γ в формулировке (6.63) еще не потерялась. Она выражена функциями $p_{X|1}$ и $p_{X|2}$.

Покажем теперь, что исходная задача (6.58) включает в себя задачу (6.63). Максимизируемое в задаче (6.58) число можно представить в виде

$$\begin{aligned} & \sum_{i=1}^n \log \sum_{k=1}^2 p_K(k) p_{X|K}(x_i|k) p_{Y|K}(y_i|k) \\ &= n \sum_{x \in X} \sum_{y \in Y} p'_{XY}(x, y) \log \sum_{k=1}^2 p_K(k) p_{X|K}(x|k) p_{Y|K}(y|k) \\ &= n \sum_{x \in X} \sum_{y \in Y} p'_{XY}(x, y) \log \left(p_Y(y) \sum_{k=1}^2 \frac{p_K(k) p_{Y|K}(y|k)}{p_Y(y)} p_{X|K}(x|k) \right) \\ &= n \sum_{x \in X} \sum_{y \in Y} p'_Y(y) p'_{X|y}(x) \log \left(p_Y(y) \sum_{k=1}^2 p_{K|y}(k) p_{X|K}(x|k) \right) \\ &= n \sum_{y \in Y} p'_Y(y) \log p_Y(y) + n \sum_{y \in Y} p'_Y(y) \sum_{x \in X} p'_{X|y}(x) \log \sum_{k=1}^2 p_{K|y}(k) p_{X|K}(x|k). \end{aligned}$$

Мы видим, что максимизация (6.58) по функциям $p_K, p_{X|K}, p_{Y|K}$ эквивалентна максимизации числа

$$\sum_{y \in Y} p'_Y(y) \log p_Y(y) + \sum_{y \in Y} p'_Y(y) \sum_{x \in X} p'_{X|y}(x) \log \sum_{k=1}^2 p_{K|y}(k) p_{X|K}(x|k), \quad (6.64)$$

по функциям $p_Y, p_{K|y}, p_{X|K}$. Поскольку каждое из приведенных двух слагаемых в выражении (6.64) зависит от своих собственных переменных, максимизация их суммы сводится к максимизации каждого слагаемого. Первое слагаемое должно максимизироваться по функции p_Y , а второе - по функциям $p_{K|y}$ и $p_{X|K}$. Мы можем также видеть, что второе слагаемое совпадает с числом, которое должно максимизироваться при поиске наилучшей прямой Γ . Таким образом мы пробили путь от задачи (6.59) к задаче (6.58). Ты удовлетворен?

Почти, за исключением совершенной малости. Когда мы пробили путь к задаче (6.58), преобразованной к виду (6.64), мы увидели, что она, помимо поиска прямой Γ , содержит еще что-то, выраженное первым слагаемым в выражении (6.64). Что бы это могло быть?

Это совершенно очевидно. Помимо поиска прямой, которая хорошо аппроксимирует совокупность функций $\{p'_{X|y}, y \in Y\}$, следует находить еще и прямую, которая хорошо аппроксимирует совокупность функций $\{p'_{Y|x}, x \in X\}$. Без этой добавки процедура в целом была бы несимметричной относительно информации, содержащейся в признаках x и y .

У меня, наконец, появилось такое ощущение ясности, будто я сам все эти алгоритмы самообучения придумал. Думаю, что эти алгоритмы заслуживают того, чтобы их исследовать значительно тщательнее, чем это сделано сейчас. Например, я не уверен, что алгоритм самообучения для случая независимых признаков написан мною наилучшим образом. Я всего лишь применил для этого частного случая общий алгоритм самообучения, описанный в лекции. Но для общего алгоритма самообучения доказана лишь сходимость этого алгоритма к некоторой неподвижной точке и ничего не доказано по вопросу, является ли эта неподвижная точка глобальным максимумом функции, которая должна максимизироваться. Могу ли я считать, что в рассматриваемом частном случае достигается глобальный максимум числа (6.58), подобно тому, как это происходит в задачах Роббинса?

Ты можешь так считать, но это утверждение основано лишь на большом экспериментальном опыте. Теоретическая ясность здесь еще не достигнута. Возможно, это была бы прекрасная работа для Тебя.

Этого я бы решительно не хотел. Мне бы больше нравилось дать иную формулировку задачи самообучения, так чтобы задача глобальной максимизации оказалась разрешимой. Пусть даже критерий максимального правдоподобия пришлось бы заменить другим, менее респектабельным. Например, так.

Пусть числа $p'_{XY}(x, y)$, $x \in X$, $y \in Y$, следует аппроксимировать числами вида

$$\sum_{k=1}^2 p_K(k) p_{X|k}(x) p_{Y|k}(y),$$

когда система уравнений

$$p'_{XY}(x, y) = \sum_{k=1}^2 p_K(k) p_{X|k}(x) p_{Y|k}(y), \quad x \in X, \quad y \in Y,$$

не имеет решения. Что может быть в этом случае более естественным, чем поиск чисел $p_K(k)$, $p_{X|k}(x)$ и $p_{Y|k}(y)$, $x \in X$, $y \in Y$, $k = 1, 2$, которые

минимизируют сумму

$$\sum_{x \in X} \sum_{y \in Y} \left(p'_{XY}(x, y) - \sum_{k=1}^2 p_K(k) p_{X|k}(x) p_{Y|k}(y) \right)^2. \quad (6.65)$$

Задача, сформулированная подобным образом, не вытекает ни из каких статистических соображений, но имеет несомненное достоинство: она хорошо изучена. Ее решение основано на так называемом разложении Карунена-Лоэва. Правда, в контексте сегодняшней темы нашего обсуждения применение этого разложения несколько необычно.

Идея новая, и нам бы не хотелось подвергать ее суровым испытаниям сразу же в момент ее зарождения. Приведенная Тобой формулировка не кажется нам естественной. Трудно, например, дать статистическую интерпретацию квадрату разностей вероятностей. Далее, при формальном обращении с выражением (6.65) могут возникнуть такие понятия, как

‘длина’ $\sum_x (p_{X|k}(x))^2$,

‘скалярное произведение’ $\sum_x p_{X|1}(x) p_{X|2}(x)$,

‘матрица’ $|X| \times |Y|$, элементами которой являются вероятности $p'_{XY}(x, y)$,

‘ковариационная матрица’ размера $|X| \times |X|$ с элементами

$$\sum_y p'_{XY}(x', y) p'_{XY}(x'', y),$$

и другие формальные математические объекты, которые трудно интерпретируются в терминах исходной задачи. От Тебя потребуется масса терпения, чтобы вырастить эту идею из пеленок и дать ей жить самостоятельно.

Я извлек из этой лекции почти все, что смог, и теперь мне бы хотелось уложить свое понимание в контекст всего того, что я узнал из предыдущих лекций. Вы указали, что роббинсовы методы, обобщением которых является самообучение, возникли как попытка заполнить пробел между байесовскими и небайесовскими методами. Сейчас мне видится, что эта попытка лишь в очень малой степени оказалась успешной. Существенное отличие методов обучения и саобучения (включим сюда и методы Роббинса) от байесовских и небайесовских методов видно уже на стадии постановок задач. При всем разнообразии байесовских и небайесовских задач их объединяет то, что результатом их решения является стратегия распознавания. Результатом же решения задач обучения и самообучения является не стратегия распознавания, а наиболее правдоподобная оценка статистической модели объекта. При этом остается ясно видимый пробел между наиболее правдоподобным оцениванием модели и построением оптимальной стратегии при известной модели. Ведь ниоткуда не следует, что при неполноте известной модели стратегию распознавания следует строить, как байесову стратегию, в которой вместо действительной модели подставляется ее наиболее правдоподобная оценка. Однако именно это рекомендует современное распознавание, и эта рекомендация формулируется просто в

виде определенного постулата. Но постулат должен иметь более простую и более убедительную формулировку, такую, чтобы вопрос "А почему именно так" просто не возникал. В данном случае вопросы возникают, и ответы на эти вопросы слишком общи и расплывчаты. Нечто в том роде, что при большом объеме обучающей выборки наиболее правдоподобная модель мало отличается от действительной модели и поэтому построенная на ее основе стратегия также мало будет отличаться от наилучшей.

Меня же интересует вопрос, что следует делать, если обучающая выборка недостаточно велика. Ведь совершенно ясно, что если она очень мала, то небайесовская стратегия, решающая задачу распознавания сложных гипотез, может оказаться прагматически лучше, чем байесовская стратегия, построенная при предположении, что наиболее правдоподобная модель и есть действительная модель. Но тогда получается, что при малых объемах обучающей выборки лучше эту выборку игнорировать, чем использовать. Какое же это оптимальное использование информации, если в определенных ситуациях следует поступать так, будто ее нет вообще?

Ну, хорошо, допустим, что когда-нибудь этот парадокс будет объяснен, и станет ясно, что с выборками малого объема следует поступать так, будто их нет вообще. Но тогда возникает следующий, для меня очень трудный вопрос: при каком объеме выборку следует считать малой и при каком объеме можно считать, что она уже несущественно отличается от бесконечной. Я таким образом прихожу к выводу, что эмпирический подход Роббинса, равно как и методы обучения и самообучения совсем не заполняют пробел между байесовскими и небайесовскими задачами. Они заполняют лишь одну точку в этом пробеле, когда объем обучающей выборки бесконечный.

Я вижу, что статистическая теория распознавания в ее нынешнем виде состоит из трех почти не связанных друг с другом методов: (1) байесовских, (2) небайесовских и (3) методов обучения и самообучения. Между этими тремя группами методов зияют ясно видимые пробелы. Эти группы еще не стали составными частями хорошо структурированной иерархии методов. Это свидетельствует о незавершенности той конструкции, которую хотелось бы назвать теорией распознавания. Я бы хотел построить такую теорию статистического распознавания, которая бы не состояла из отдельных групп методов, а представляла бы один общий подход, который покрывал бы весь спектр задач от небайесовских до байесовских. При уменьшении объема обучающей выборки этот подход становился бы идентичным небайесовским методам и приближался бы к байесовым методам по мере увеличения объема обучающей выборки. Это должно было бы быть нечто подобное вашим предложениям по детерминированному обучению, о котором вы говорили в конце лекции 3, или статистическим обобщением этого детерминированного обучения.

Теперь нас уже трое, кто ставит перед собой такие амбициозные цели. Возможно, когда-нибудь такая статистическая теория распознавания будет построена. А пока будем следовать совету мудрых: смело идти с теми, кто ищет истину, но быстро убегать от тех, кто эту истину уже нашел.

Январь 1998.

6.8 Связь с TOOLBOX[®]ом

TOOLBOX "Статистические методы распознавания" написал В.Франц, как дипломную работу весной 2000 года. TOOLBOX доступен по адресу http://cmp.felk.cvut.cz/cmp/cmp_software.html. Построен на базе Matlab версии 5.3 и выше. Доступны исходные коды алгоритмов. Разработка TOOLBOX[®]а продолжается.

К содержанию данной лекции относятся алгоритмы самообучения для многомерной гауссовой модели распознаваемого объекта.

6.9 Библиографические заметки

Исследования по самообучению были инициированы Ф.Розенблаттом в публикации [Rosenblatt, 1957; Rosenblatt, 1959], породившей большой интерес к проблеме. Более сдержанное описание процедур самообучения в перцептроне представлено в монографии [Rosenblatt, 1962], подводящей итоги многолетних исследований перцептронов. Строгий анализ перцептронных алгоритмов обучения и самообучения содержится в работах В.М.Глушкова [Glushkov, 1962b; Glushkov, 1962a], показавших, что многие замечательные свойства перцептронов оказались такими только на словах. Несмотря на отрицательные результаты, полученные как Розенблаттом, так и Глушковым, исследования перцептронов продолжались [Minsky and Papert, 1969].

Важным моментом в формализации проблемы самообучения оказались работы [Schlesinger, 1965; Ball and Hall, 1967], в которых задачи самообучения были сформулированы как задачи кластеризации. В настоящее время проблеме кластеризации посвящено необозримое множество работ.

Следующий момент в становлении самообучения, как области научных исследований, состоял в формулировке задачи самообучения, как оценки параметров смешанной выборки наблюдений. Статистические основы этого подхода сформулированы задолго до применения их в распознавании. Наиболее ранней публикацией этого направления является, насколько известно, работа Пирсона 1894 года. Задача ставилась для смеси двух гауссовых случайных величин и решалась по методу моментов. В теорию и практику распознавания образов эти методы вошли благодаря работе [Cooper and Cooper, 1964].

Другой мощный поток, возникший вне распознавания и затем асимилированный в распознавании, состоит в фундаментальном результате Роббинса, известном как эмпирический байесовский подход [Robbins, 1951; Robbins, 1956; Neuman, 1962].

Задача максимально правдоподобного оценивания параметров смешанной выборки длительное время противостояла попыткам ее решения. Существенный прорыв в этом направлении определила работа [Schlesinger, 1968], в которой опубликован алгоритм самообучения в том виде, в котором он представлен в данной лекции. Несколько позже аналогичные результаты

были опубликованы другими авторами. Заметно позже после этого прорыва опубликованна широко цитируемая сейчас работа [Demster et al., 1977]. Из других публикаций следует указать [Wu, 1983; Grim, 1986].

Глава 7

Заключительные замечания по статистическому распознаванию и вводные замечания по структурному распознаванию

7.1 Общность статистического распознавания

В предыдущих рассказах мы неоднократно подчеркивали исключительную общность статистического распознавания. В силу этой общности выводы статистической теории распознавания имеют характер законов. Они должны учитываться при решении любой прикладной задачи, будь то диагноз человеческого сердца по электрокардиограмме, или оценка токарного инструмента по звуковому сигналу, который он издает во время работы, или обработка микроизображений кровяных телец, или исследование естественных ресурсов Земли по космическим снимкам. Сейчас, однако, мы хотим подчеркнуть не столько огромное разнообразие приложений статистической теории распознавания, сколько разнообразие ее формальных свойств. Множества X и K значений наблюдаемых и скрытых параметров могут быть весьма разнообразны в чисто математическом смысле этого слова. Мы скажем, пока еще очень неточно, что эти множества разнообразны по своей структуре.

Когда мы говорим, что такой параметр, как, например, вес объекта, принимает значения из хорошо структурированного множества, мы имеем в виду, что элементы этого множества можно складывать, умножать, сравнивать между собой, и каждая из этих операций имеет смысл в терминах исходной прикладной задачи. Оценки же, которые получает студент на экзамене, принадлежат множеству с более слабой структурой. Это - множество $\{2, 3, 4, 5\}$ (по крайней мере, в некоторых странах). Элементы этого множества можно сравнивать между собой, то есть говорить, какая оценка лучше и какая хуже. Но их нельзя складывать или умножать, так как два студента, получившие на экзамене двойку и тройку, ни в каком смысле не эквивалентны одному студенту, получившему пятерку. На множестве оценок определены отношения $=$, $<$, и $>$, а другие операции не определены, так как им ничего не соответствует в исходной прикладной задаче. Говоря иными словами, множество оценок является полностью упорядоченным

множеством и ничем иным. Еще более слабую структуру имеет множество номеров трамвайных маршрутов. Хотя эти номера и выражаются числами, структура множества маршрутов отличается от структуры множества чисел. В отличие от чисел, трамвай №12 никаким образом не заменяют два трамвая №6. В отличие от элементов упорядоченного множества, маршрут №2 ничуть не лучше маршрута №3 и не хуже маршрута №1.

Сейчас мы говорим об общности статистической теории распознавания, имея в виду именно разнообразие формальных свойств множеств X и K , а не разнообразие их прикладных интерпретаций.

Как правило, в распознавании образов наблюдение x состоит не в одном, а нескольких измерениях x_1, x_2, \dots, x_n . В этом случае можно говорить не только о структуре множества значений каждого отдельного признака, но и о структуре множества признаков. Множества признаков (именно признаков, а не их значений!) в различных задачах имеют различную структуру. Поясним это различие на следующих двух примерах.

В первом примере признаки x_1, x_2, \dots, x_n - это записи в истории болезни при первой регистрации пациента у врача. Они касаются температуры тела, давления крови, возраста, частоты пульса, пола, скажем, всего n признаков. Во втором примере речь идет о значениях какого-то одного признака, скажем, температуры, измеренных n раз через равные интервалы времени после определенного врачебного воздействия. Результатом такого наблюдения является опять-таки совокупность индексированных величин x_i , где индекс i принимает значения на множестве $\{1, 2, \dots, n\}$, как и в первом примере. Хотелось бы, однако, чтобы было очевидным, что структуры множеств индексов в приведенных двух примерах совершенно различны. Так, в первом примере несущественно, будет ли третьим признаком именно возраст пациента. В задаче ровным счетом ничего изменится при произвольной перенумерации признаков. Множество признаков в первой задаче лишено какой-либо структуры. Номера $1, 2, \dots, n$ признаков - это не числа, а просто символы в определенном абстрактном алфавите.

Ситуация во втором примере в корне иная. Здесь индекс признака понимается именно, как целое число. Совокупность измеренных значений температуры образует последовательность, и нумерация элементов в этой последовательности не может быть произвольной. Множество индексов здесь обладает ясно видимой структурой, которая отсутствовала в первом примере.

Мы еще не раз вернемся к этим вопросам. Сейчас же мы бросили только беглый взгляд на огромное разнообразие, которое кроется за словами "Пусть множество наблюдений X и множество состояний K - два конечных множества", которые рефреном повторялись в предыдущих лекциях, как исходный пункт для формальных рассуждений. Общность полученных в предыдущих рассказах результатов состоит в том, что они справедливы для множеств X и K произвольной структуры.

Отсюда, конечно же, не следует, что для решения каждой конкретной задачи достаточно знать лишь самые общие методы статистического распознавания. В каждом конкретном случае задачу следует формулировать

как можно более конкретно, то есть наделять множества X и K структурами, которые наиболее полно отражают ее прикладное содержание. Затем, уже для определенного частного случая, следует исследовать, во что выливаются общие рекомендации. Это нелегкий путь, и структурное распознавание, о котором пойдет речь в последующих рассказах, содержит рекомендации, как его проходить.

К счастью, некоторые приложения можно выразить в рамках формализма, который хорошо исследован в классической математической статистике в ее наиболее развитой части - статистике случайных чисел. Ее рекомендации основаны на таких понятиях, как математическое ожидание, дисперсия, корреляционная матрица, ковариационная матрица. Все эти понятия определены только для случайных объектов, представленных числами. Существует однако масса приложений, в которых результат наблюдения является не числом, а элементом какого-то другого множества, структура которого отличается от структуры множества чисел. К таким приложениям рекомендации численной статистики непосредственно не применимы. Стремление во что бы то ни стало решать эти задачи методами статистики случайных чисел приводит к недопустимой деформации исходной прикладной задачи. В конечном итоге разработанный алгоритм решает совсем не ту задачу, которую следовало решить.

В этом смысле самыми невезучими оказались задачи распознавания изображений. Мы имеем в виду не абстрактные изображения, а конкретно, результат наблюдения объекта с помощью, скажем, телевизионной камеры. Изображение - это исключительно своеобразный объект машинного анализа. Множества изображений, представляющие интерес в том или ином приложении, не принадлежат к классу множеств, тщательно изученных в процессе многовековых математических исследований. Это не выпуклые множества, не подпространства, не эллипсы, не шары, и вообще, это не то, что хорошо известно. Это совсем другие множества.

Существенные положительные результаты по распознаванию изображений не могут быть получены на основании одних лишь общих рекомендаций статистической теории распознавания. Необходимо изучать и учитывать исключительную специфичность изображения, как объекта формального анализа.

7.2 Почему для распознавания изображений нужно распознавания изображений?

7.2.1 Множество наблюдений

Общая теория распознавания не основана на каких-то определенных предположениях о математическом виде множества X наблюдений. Тем не менее, во многих исследованиях принимается, как нечто само собою разумеющееся, что множество X есть линейное пространство. Такая предпосылка мотивируется очень простыми рассуждениями. Распознавание ведь происходит на основе ряда измерений над объектом, результатом каждого из которых является число. Поэтому входной информацией при распознавании

является совокупность из n чисел. Эту совокупность можно отождествить с точкой n -мерного линейного пространства, i -ая координата которой есть результат i -ого измерения.

Если же распознавание состоит в оценке скрытого параметра, принимающего два значения, то стратегию распознавания можно понимать как разбиение пространства X на два подмножества X_1 и X_2 . Границу между этими двумя множествами можно понимать как поверхность в пространстве X , определенную уравнением $f(x) = 0$, где f - функция n переменных x_1, x_2, \dots, x_n , а x_i - результат i -ого измерения. При этом саму функцию следует выбрать так, чтобы она принимала положительные значения на множестве X_1 и отрицательные - на множестве X_2 . Таким наглядным образом можно представить базовые понятия распознавания образов и, исходя из такого представления, получать те или иные плодотворные результаты.

Приведенная формализация множества X , конечно же, имеет право на существование и вполне заслуживает того внимания, которое мы ей уделили в лекции 5. Однако было бы ошибкой считать ее единственно возможной и универсально применимой. В свое время такое ошибочное представление было очень распространенным. И хотя сейчас оно встречается все реже и реже, стоит его рассмотреть подробно, чтобы по отношению к нему создался стойкий иммунитет. Мы это сделаем на примере множества изображений и покажем, почему представление именно изображения в виде точки многомерного пространства особенно неуместно.

Изображение - это функция f , определенная на квадратном участке плоскости размером $D \times D$ и принимающая вещественные значения. Число $f(x, y)$ - это яркость точки с координатами x, y . Область определения изображения разбивается на N^2 меньших квадратов размера $\Delta \times \Delta$, $\Delta = D/N$, и результатом наблюдения считается совокупность средних яркостей в каждом из маленьких квадратов. Эта совокупность отождествляется с точкой в N^2 -мерном линейном пространстве. Класс изображений в таком случае - это определенное подмножество точек линейного пространства. Таким образом, основные понятия распознавания изображений формализуются, казалось бы, в достаточно наглядном виде, при котором для решения задач распознавания можно привлечь весь арсенал методов, тщательно исследованных в теории линейных пространств и многомерной статистики.

Потребовалось достаточно много времени и усилий, чтобы понять, что достигнутая ясность является в высшей степени обманчивой. Приведенная формализация служит не путеводной звездой, а прямо-таки дезориентирует. Отождествление изображения с точкой многомерного пространства поневоле провоцирует привлечение для распознавания именно тех множеств, функций, преобразований и случайных величин, которые хорошо изучены в теории линейных пространств и многомерной статистике, то есть, выпуклых множеств, подпространств, полупространств, линейных преобразований, гауссовых случайных величин и т.п. Однако именно такие множества, функции и случайные величины наименее подходят для описания свойств изображений. Поэтому многомерные пространственные аналогии не вызва-

ли такую лавину плодотворных результатов в распознавании изображений, как это произошло, например, в линейном программировании. Чисто геометрическое представление, например, что из любой вершины многогранника можно перейти в любую другую вершину, перемещаясь только по ребрам, плодотворно стимулируют интуицию исследователя в линейном программировании, позволяют убеждаться в правильности того или иного утверждения еще до того, как оно формально доказано. В распознавании изображений ситуация оказалась иной. Отождествление множества изображений с множеством точек многомерного пространства оказалось скорее разрушительным, чем созидательным. Известно достаточно резкое высказывание М.Минского и С.Пейперта по этому поводу [Minsky and Papert, 1969], что наибольший вред для машинного анализа изображений состоял именно в неразумной эксплуатации многомерных геометрических аналогий.

Покажем, в чем состоит фундаментальное отличие изображения и точки многомерного линейного пространства, то есть, многомерного вектора. Для этого посмотрим сначала, что их роднит. Как вектор, так и изображение - это функции вида $T \rightarrow V$ с областью определения T и областью значений V . Если эта функция представляет n -мерный вектор с координатами $x_i, i = 1, 2, \dots, n$, то T - это множество индексов, а V - это множество вещественных чисел. Если функция $T \rightarrow V$ представляет изображение, то множество T есть прямоугольный участок двумерной целочисленной решетки, то есть, $T = \{i, j \mid 1 \leq i \leq n; 1 \leq j \leq n\}$, а V - множество значений сигнала, наблюдаемого в точке с координатами i, j .

Если функция $T \rightarrow V$ представляет многомерный вектор, то на множестве ее значений определены операции сложения, умножения, существуют особые элементы 0 и 1 и имеют место другие свойства, в силу чего мы и говорим, что это функция с хорошо структурированной областью значений.

Если же функция $T \rightarrow V$ представляет изображение, то множество ее значений структурировано значительно слабее. К примеру, пусть $x: T \rightarrow V$ - изображение, а $f: V \rightarrow V$ некоторая монотонно возрастающая функция. Как правило, замена сигнала $x(t)$ в точке $t \in T$ на сигнал $f(x(t))$ не меняет равным счетом ничего с точки зрения информации, которую следует извлечь из изображения, если только ко всем точкам $t \in T$ применяется одно и то же преобразование f . Это значит, что структура области V значений значительно более слабая, чем структура действительных чисел. В рассматриваемом случае это структура упорядоченного множества. Передки приложения, в которых область значений V имеет еще более слабую структуру. Так, при распознавании цветных графических изображений множество V конечное и содержание изображения никак не меняется при произвольном переименовании цветов. Это значит, что множество V есть просто алфавит имен, лишенный какой-либо структуры.

Может показаться на первый взгляд, что нет особого вреда в том, что исходное прикладное понятие представляется в рамках более богатого формализма. Но это не так. Применение более богатого формализма произвольно вносит в исходную задачу свойства, которые в исходной задаче от-

существуют. Затем и алгоритм распознавания строится на основе этих внешних свойств, например, на основе минимизации среднеквадратичного отклонения яркости изображения, или линейной трансформации изображения, или чего-то другого, что имеет смысл в принятой формальной схеме, но не имеет разумной интерпретации в терминах исходной прикладной задачи.

Взглянем теперь, чем отличаются области определения функций, представляющих изображения и векторы. Переход от изображений к векторам приводит к потере именно тех важных свойств изображения, которые определяют его исключительность, как носителя информации. Когда функция $T \rightarrow V$ представляет вектор, то ее областью определения является абстрактное множество индексов, лишенное какой-либо структуры. В случае же, когда функция $T \rightarrow V$ представляет изображение, то область ее определения - это прямоугольный участок двумерной целочисленной решетки. Это множество с ясно видимой структурой, вне которой трудно сформулировать такие понятия, как связность, симметрия и другие специфически зрительные понятия. Сейчас можно только удивляться тому оптимизму, с которым распознавание образов в пору своего младенчества претендовало на решение задач и зрительного анализа, не учитывая при этом структуру области определения функций, представляющих изображения.

Для конструктивного применения статистической теории распознавания к задачам зрительного анализа следует прежде всего конкретизировать множество X наблюдений как множество функций, заданных на множествах с определенной, характерной именно для изображений структурой.

7.2.2 Множество скрытых параметров изображений

В предыдущих лекциях мы излагали статистическую теорию распознавания при постоянной предпосылке, что скрытый параметр k принимает значения из конечного множества K . Мы уже указывали, как много разнообразных ситуаций скрывается за этой краткой формулировкой. И тем не менее, в ряде исследований в явном или неявном виде сужают класс множеств K , и тем самым без необходимости сужают и область применимости рекомендаций статистической теории распознавания.

1. Предполагается, что K - это абстрактный алфавит символов, лишенный какой-либо структуры.
2. Предполагается, что количество элементов множества K не просто конечно, а малое настолько, что полный их перебор не приводит ни к каким вычислительным трудностям.
3. Предположение о малом количестве элементов множества $|K|$ немедленно приводит к отождествлению понятия стратегии с классификацией множества наблюдений, то есть, с разбиением множества X наблюдений на подмножества (классы) X_k , $k \in K$. Хотя любая стратегия $q: X \rightarrow K$ и определяет семейство подмножеств $X_k = \{x \in X | q(x) = k\}$, $k \in K$, для многих приложений (мы их укажем несколько ниже) поня-

тие классификации выглядит очень неестественно и такие приложения неявно исключаются из рассмотрения.

Введенные ограничения искусственно сужают область компетентности статистической теории распознавания. Выводы и рекомендации, которые мы изложили в предыдущих лекциях, вовсе не требуют привлечения таких ограничений. С другой стороны, в рамки указанных ограничений с трудом втискиваются те прикладные задачи, которые принято относить к распознаванию изображений. Все это вместе взятое приводит к ошибочному представлению о разрыве между статистической теорией распознавания и практикой машинной обработки изображений. Скажем прямо, это не разрыв между теорией и практикой распознавания, а разрыв между той малой частью теории и той малой частью практики, с которой знаком тот или иной автор.

В практике распознавания изображений сравнительно редки задачи, решение которых состоит в присвоении распознаваемому изображению того или иного имени из определенного конечного и малого алфавита. Даже в такой парадигматической задаче, как построение читающего автомата, результат распознавания имеет значительно более сложную структуру. При разработке действительно практического изделия, а не лабораторной диковинки сразу становится ясным, что речь идет не о распознавании изображений, на которых представлена одна-единственная буква, а о распознавании изображений, на которых представлен текст. И если результат распознавания одной-единственной буквы можно понимать, как классификацию входных изображений, то для результата распознавания страницы текста такое понимание совершенно неестественно.

В типичных прикладных задачах распознавания изображений результатом является, например, координата объекта (то есть, число или двумерный или трехмерный вектор), или его геометрические параметры (то есть, совокупность чисел), или последовательность имен букв при распознавании текстового документа (то есть, последовательность символов определенного алфавита), или список элементов электрической схемы с указанием связей между ними (то есть, граф), или карта местности, полученная по аэрофотоснимку (то есть, снова изображение). Получение такого рода результатов распознавания совершенно неестественно трактовать, как присвоение распознаваемому изображению имени из заранее заданного набора имен, или как отнесению изображения к одному из заранее заданных классов изображений.

Результат распознавания изображения совсем необязательно имеет такой простой вид, как символ из некоторого алфавита. Как правило, это сложный набор данных с той или иной структурой, зависящей от прикладного содержания задачи. Рекомендации статистической теории распознавания применимы и к таким задачам. Но для их конструктивного применения не следует игнорировать конкретный и зачастую довольно сложный вид множества K , то есть структуру данных, которые должны быть получены в результате распознавания.

7.2.3 Роль обучения в распознавании изображений

Мы уже говорили, что отождествление множества изображений с множеством точек многомерного линейного пространства может существенно исказить содержательный смысл задачи. Однако для многих приложений, как правило, не относящихся к зрительному анализу, такая формализация оказывается плодотворной. Если при этом есть основания предполагать, что стратегия реализуется с помощью линейной, или квадратичной функции, или полинома низкой степени от наблюдений, то построение алгоритма, реализующего эту стратегию, не наталкивается на какие-либо трудности вычислительного характера. В этом случае основную трудность составляет отыскание стратегии, которую следует реализовать. Эти трудности преодолевается методами обучения и самообучения, которым мы посвятили несколько предыдущих лекций. Однако, если опыт исследователя ограничивается только такими, вычислительно непроблематичными стратегиями, то легко скатиться к ошибочному представлению, что вся научная проблематика распознавания ограничивается задачами обучения и самообучения. В наиболее явной форме это представление выражается малопродуманной и высокомерной фразой типа "Если стратегия распознавания определена или определена статистическая модель объекта, то уже нет задачи". Это непростительная недооценка тех сложностей собственно распознавания, которые лежат вне проблематики обучения и самообучения. Такая недооценка следует из непонимания фундаментального различия между однозначным заданием функции, или множества, или отношения и построением алгоритма для вычисления значения функции, или принадлежности элемента заранее заданному множеству, или проверки однозначно сформулированного отношения. Функция, или множество, или отношение могут быть заданы одним способом, а для построения алгоритма их вычисления требуется их задание другим способом. Именно при распознавании изображений вскрываются трудности собственно распознавания, возникающие и тогда, когда стратегия распознавания вполне однозначно определена. Эти трудности невозможно преодолеть, если их не знать. Покажем на простом примере, насколько сильно отличается однозначное определение стратегии распознавания от алгоритма распознавания.

Пример 7.1 Вертикальные и горизонтальные линии.

Пусть множество $T = \{(i, j) \mid 1 \leq i \leq m; 1 \leq j \leq n\}$ - прямоугольный участок размером $m \times n$ двумерной целочисленной решетки, а $V = \{0, 1\}$ (то есть, белое и черное) - множество сигналов. Наблюдением является функция вида $T \rightarrow V$, которую назовем изображением. Обозначим h_i изображение, которое назовем i -ой горизонтальной линией, $i = 1, \dots, m$. В этом изображении сигнал $h_i(i', j)$ равен 1 тогда и только тогда, когда $i' = i$. Подобным образом обозначим v_j , $j = 1, \dots, n$, изображение, называемое j -ой вертикальной линией, в котором $v_j(i, j') = 1$ тогда и только тогда, когда $j' = j$. Обозначим h множество всех горизонтальных ли-

ний, а v - множество всех вертикальных линий, $h = \{h_i \mid i = 1, \dots, m\}$, $v = \{v_j \mid j = 1, \dots, n\}$.

Пусть k - подмножество горизонтальных и вертикальных линий, которое содержит в себе не все горизонтальные и не все вертикальные линии, то есть $k \subset v \cup h$, $v \not\subset k$, $h \not\subset k$. Множество всех подмножеств, полученных таким образом, обозначим K , $|K| = (2^m - 1)(2^n - 1)$. Для каждой совокупности $k \in K$ горизонтальных и вертикальных линий определим изображение x так, что $x(i, j) = 0$ тогда и только тогда, когда во всех изображениях из совокупности k сигнал в точке (i, j) равен нулю. Попросту говоря, изображение x получается в результате наложения друг на друга некоторой совокупности горизонтальных и вертикальных линий, но не всех горизонтальных и не всех вертикальных.

Пусть распознавание таких изображений состоит в том, чтобы по изображению x определить k , то есть, указать, из каких линий составлено предъявленное изображение. Эта задача тривиальна. Поскольку совокупность k содержит не все горизонтальные линии, существует строка i^* , на которой горизонтальная линия отсутствует. Поскольку совокупность k содержит не все вертикальные линии, существует столбец j^* , на котором отсутствует вертикальная линия. Следовательно, на изображении имеется по крайней мере одна белая точка. Обозначим ее координаты i^* , j^* и получим следующее правило распознавания: совокупность k содержит горизонтальную линию h_i тогда и только тогда, когда $x(i, j^*) = 1$, и содержит вертикальную линию v_j тогда и только тогда, когда $x(i^*, j) = 1$.

Введем теперь в рассмотрение случайные помехи. Мы увидим, что даже при простейшей модели помех задача не просто перестает быть тривиальной. Она становится неразрешимой за полиномиальное время.

Пусть изображение $x: T \rightarrow \{0, 1\}$ под воздействием помех превращается в изображение $x': T \rightarrow \{0, 1\}$ так, что для каждой точки $(i, j) \in T$ вероятность равенства $x(i, j) = x'(i, j)$ равна $1 - \varepsilon$, а вероятность неравенства $x(i, j) \neq x'(i, j)$ равна ε . Чтобы не отходить от простейшей модели помех, будем предполагать, что помехи действуют в каждой точке изображения независимо друг от друга. Если к тому же принять, что все совокупности k в множестве K имеют равные априорные вероятности, то совместные вероятности $p(x', k)$ однозначно определяются формулой

$$p(x', k) = \frac{(1 - \varepsilon)^{mn}}{(2^m - 1)(2^n - 1)} \prod_{i=1}^m \prod_{j=1}^n \left(\frac{\varepsilon}{1 - \varepsilon} \right)^{|x'(i,j) - \alpha(i)\beta(j)|},$$

где $\alpha(i) = 1$ при $h_i \in k$ и $\alpha(i) = 0$ при $h_i \notin k$, $\beta(j) = 1$ при $v_j \in k$ и $\beta(j) = 0$ при $v_j \notin k$. Вычисление вероятности пары x' и k по этой формуле настолько простое, что об этом не стоит даже говорить. Статистическая модель объекта, таким образом, однозначно задана и исключительно проста. Задача же распознавания исключительно сложна. Допустим, что распознавание состоит в указании совокупности k с наибольшей апо-

стериорной вероятностью при условии наблюдения x' . Оно сводится к минимизации

$$\left(\alpha^*(1), \dots, \alpha^*(m); \beta^*(1), \dots, \beta^*(n)\right) = \underset{\substack{(\alpha(1), \dots, \alpha(m), \\ \beta(1), \dots, \beta(n))}}{\operatorname{argmin}} \sum_{i=1}^m \sum_{j=1}^n |x'(i, j) - \alpha(i) \beta(j)|$$

при условии, что переменные $\alpha(i)$ и $\beta(j)$ принимают только значения 0 или 1. Это - задача фантастической сложности. В настоящее время неизвестны алгоритмы, которые решали бы эту задачу за время, существенно меньшее, чем $2^{\min(m, n)}$. ▲

Приведенный пример указывает, как сложна может быть задача собственно распознавания, когда статистическая модель распознаваемого объекта полностью определена и нет необходимости в обучении или самообучении. Естественно, мы здесь говорим о точном решении однозначно сформулированной задачи, а не о разного рода изобретениях, которые, как часто говорят, "не обеспечивают решения задачи, но в большинстве практических ситуаций себя оправдывают и т.п.". О практически приемлемых решениях можно говорить только в при рассмотрении задач действительно высокой практической значимости. Они неизмеримо сложнее рассмотренного здесь модельного примера. В них требуется обнаруживать значительно более замысловатые объекты, чем идеальные горизонтальные и вертикальные линии, на существенно более сложном фоне и при существенно более сложных и заранее неизвестных помехах. При рассмотрении конкретных прикладных задач такого уровня сложности можно и не настаивать на безукоризненном теоретическом обосновании принятого решения. Важнее в этом случае его чисто технологическая приемлемость. Но в приведенном, предельно упрощенном примере следовало бы достигнуть именно теоретическую ясность, будь то положительную или отрицательную.

Сложность приведенного примера состоит в том, что множество K значений скрытого состояния настолько велико, что невозможно осуществить полный перебор всех его элементов. Эта сложность определенным образом умалчивается в общей статистической теории распознавания, где исследуются другие, не менее важные проблемы. Но от этого умолчания эта сложность не исчезает. Для задач распознавания изображений эта сложность является правилом, а не исключением. В задачах распознавания изображений множество K , как правило, такое, что полный перебор его элементов невозможен.

Это, однако, не единственная особенность задач распознавания изображений. Они оказываются вычислительно трудными и в тех случаях, когда множество K малое, например, состоит всего из двух состояний. Для модели, рассмотренной в примере, можно сформулировать задачу так, что требуется указать не всю совокупность линий на изображении, а дать лишь бинарный ответ на вопрос, имеется ли горизонтальная линия, скажем, в пятой строке изображения. Распознавание в этом случае сводится к клас-

сификации входных изображений на два класса: на изображения, содержащие линию в пятой строке, и изображения, в которых такая линия отсутствует. К сожалению, такая модификация задачи ничуть ее не упрощает. Правда, поиск наиболее вероятного состояния в пятой строке перестает быть проблемой, потому что надо сделать выбор всего из двух состояний. Однако непреодолимые сложности возникают при вычислении вероятностей этих двух состояний.

Приведенный пример иллюстрирует типичные для распознавания изображений трудности. Они состоят не в недостатке знаний о тех или иных отношениях, множествах или вероятностях. Эти знания, как правило, имеются, но они не выражаются в терминах тех математических понятий, которые в результате многовекового исследования приобрели элегантную завершенность. Это просто что-то новое.

7.3 Основные понятия структурного распознавания

В последующих лекциях мы опишем специальные методы распознавания, учитывающие особенность изображения как носителя информации и объекта машинной обработки. Как и в предыдущих лекциях, материал будет излагаться в стиле, который принят в математических исследованиях, что предполагает определенный уровень абстракции. Известно, что абстрактные понятия имеют тенденцию жить своей собственной жизнью, в течение которой они приходят в соприкосновение не только с теми прикладными задачами, которые привели к их рождению. Поэтому идеи, изложенные в последующих лекциях, известны не как теория распознавания изображений, а как структурное распознавание образов, и это правильно. При всей общности вводимых далее формальных конструкций, они не могут покрыть все разнообразие проблем распознавания изображений. С другой стороны, в силу своей общности, результаты структурного распознавания применимы не только к распознаванию изображений. Введем основные понятия структурного распознавания.

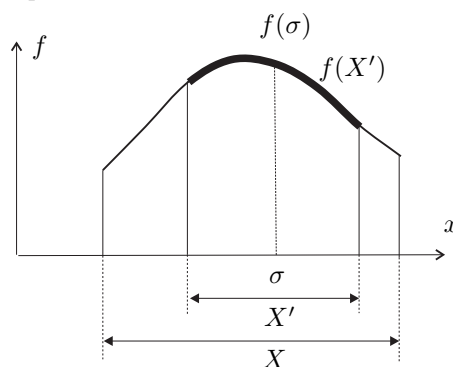
Пусть объект характеризуется определенным множеством параметров. Хотя это и не всегда естественно, будем считать, что значения параметров записаны в памяти, которая является составной частью самого объекта. Обозначим эту память T и будем называть ее полем объекта. Подчеркнем, что T обозначает только память, а не содержимое этой памяти. Память T состоит из ячеек t , соответствующих отдельным параметрам. Для простоты и без особой потери общности будем полагать, что все параметры принимают значения из одного и того же конечного множества S .

Объект считается полностью описанным (заданным), если для каждого параметра $t \in T$ указано его значение $s(t)$, $s(t) \in S$. Формально говоря, описание объекта - это функция $s: T \rightarrow S$, областью определения которой является множество T параметров (или, что то же самое, память T), со значениями из множества S .

Задача распознавания здесь и далее будет иметь следующий формат: на основании знания данных, записанных в определенной части памяти

$T' \subset T$, следует сказать нечто вразумительное о том, что записано в остальной ее части. Даже по этой весьма размытой формулировке можно усмотреть некоторое сходство будущих задач и задач статистического распознавания объекта, характеризуемого наблюдаемым признаком и скрытым состоянием. Однако сейчас с самого начала подчеркивается, что имеется не один наблюдаемый и не один скрытый параметр, а множество наблюдаемых и множество скрытых параметров.

Для дальнейшего изложения введем дополнительные обозначения. Пусть X и Y - два множества. Множество всех возможных функций вида $X \rightarrow Y$ обозначим Y^X . Ясно, что $|Y^X| = |Y|^{|X|}$. Пусть $f \in Y^X$ - некоторая функция вида $X \rightarrow Y$, а $X' \subset X$ - подмножество в X . Сужение функции $f: X \rightarrow Y$ на множество $X' \subset X$ определяется как такая функция $f': X' \rightarrow Y$, что $f'(x) = f(x)$ для всех $x \in X'$.



Сужение функции $f: X \rightarrow Y$ на подмножество $X' \subset X$ будем обозначать $f(X')$, а значение этой функции в точке $\sigma \in X$ обозначим, как обычно принято, $f(\sigma)$. Таким образом, $f(X')$ обозначает не одно значение, а функцию, определенную на подмноестве X' , см. рис. 7.1. Недоразумения здесь, как правило, не возникают, так как по аргументу в круглых скобках выражений $f(X')$ или $f(\sigma)$ ясно, идет ли речь о единственном значении (когда аргументом является точка) или о функции (когда аргументом является подмножество).

Пусть задано разбиение множества T (поля объекта) на два подмножества: подмножество Tx наблюдаемых параметров (наблюдаемое поле) и оставшееся подмножество Tk скрытых параметров (скрытое поле). Буквы x, k в обозначениях Tx и Tk следует понимать как части идентификаторов, а не как индексы, пробегающие некоторое множество значений.

Задача распознавания, все еще понимаемая неформально, приобретает следующий формат. Существует функция $s: T \rightarrow S$, определенная на заданном множестве T и принимающая значения из заданного множества S . Функция $s: T \rightarrow S$ неизвестна, но известно ее сужение $x: Tx \rightarrow S$ на заданное наблюдаемое поле $Tx \subset T$. Следует определить сужение $k: Tk \rightarrow S$ функции s на скрытое поле $Tk \subset T$. Функция $x: Tx \rightarrow S$ играет здесь ту же роль, что и наблюдение в предыдущих лекциях, а функция $k: Tk \rightarrow S$ соответствует скрытому параметру объекта. Функция $s: T \rightarrow S$, которая есть не что иное, как пара функций $x: Tx \rightarrow S$ и $k: Tk \rightarrow S$, будет называться *полным описанием объекта*.

На основании наблюдения $x: Tx \rightarrow S$ можно что-то определенное сказать о скрытой функции $k: Tk \rightarrow S$ только в том случае, когда известно, что они находятся в каком-то отношении друг с другом, или, что то же самое, когда известно, что полное описание $s: T \rightarrow S$ удовлетворяет неко-

торым априорным ограничениям. Мы будем определять эти ограничения двумя различными, но похожими способами. В первом случае мы будем определять подмножество $L \subset S^T$ функций $s: T \rightarrow S$, которые будем называть допустимыми. Во втором случае будем считать заданной функция $p_T: S^T \rightarrow \mathbb{R}$, которая для каждой функции $s: T \rightarrow S$ определяет вероятность $p_T(s)$, с которой функция s может оказаться полным описанием распознаваемого объекта.

Этот, пока что эскизный набросок *взаимосвязи наблюдаемых и скрытых параметров объекта* является тем мостиком, который соединяет статистическое и структурное распознавание. Структурное распознавание не является чем-то оторванным от статистического. И в том, и в другом случае распознавание основано на существовании взаимной зависимости наблюдаемой величины x и скрытой величины k . Специфика структурного распознавания состоит лишь в том, что множество допустимых пар (x, k) или распределение вероятностей на множестве этих пар задаются своими, специальными средствами. Опишем эти средства.

Некоторое множество $\mathcal{T} = \{T_1, T_2, \dots, T_m\}$ подмножеств поля T объекта будем называть *структурой поля*, а каждый элемент структуры - *фрагментом поля*. Число $\max_{T' \in \mathcal{T}} |T'|$ будем называть *порядком структуры*. В частности, когда структура \mathcal{T} состоит только из некоторых пар элементов из T , то речь идет о структуре второго порядка, то есть, попросту говоря, о (неориентированном) графе. Покажем сначала, какими средствами будет определяться подмножество $L \subset S^T$ допустимых описаний объекта, а затем, как будет определяться распределение вероятностей на множестве описаний объекта.

Пусть \mathcal{T} - структура и пусть для каждого фрагмента $T' \in \mathcal{T}$ определено подмножество $L_{T'} \subset S^{T'}$. Функция $s: T \rightarrow S$ будет определяться как допустимая, если ее сужение на каждый фрагмент T' из \mathcal{T} принадлежит $L_{T'}$. Таким образом подмножество $L \subset S^T$, то есть отношение наблюдаемых и скрытых параметров, выражено с помощью совокупности подмножеств $L_{T'} \subset S^{T'}$, $T' \in \mathcal{T}$. Очевидно, что не любое подмножество $L \subset S^T$ можно выразить таким способом. *Областью действия структурного распознавания являются только те случаи, когда сложное подмножество можно указанным способом представить с помощью совокупности более простых подмножеств*. Мы увидим, что эта область очень обширная.

Распределение вероятностей $p_T: S^T \rightarrow \mathbb{R}$ определяется иным, но подобным образом. Для каждого фрагмента T' из структуры \mathcal{T} определяется распределение вероятностей $p_{T'}: S^{T'} \rightarrow \mathbb{R}$. Взаимосвязь между глобальным распределением вероятностей p_T и локальными распределениями вероятностей на фрагментах T' определяется следующим образом. Пусть $s: T \rightarrow S$ - случайная функция с распределением вероятностей p_T . Сужение этой функции на фрагмент T' - это функция $s(T')$, которая также случайна, и ее распределение вероятностей есть $p_{T'}(s(T'))$.

Пример 7.2 Глобальные и локальные вероятности. Пусть поле объекта есть $T = \{1, 2, 3\}$, а структура \mathcal{T} состоит из пар $\{1, 2\}$, $\{1, 3\}$,

$\{2, 3\}$. В этом случае речь идет о трех случайных величинах s_1, s_2, s_3 . Их совместные вероятности $p_T(s_1, s_2, s_3)$ неизвестны. Известны однако совместные вероятности $p_{\{12\}}(s_1, s_2)$, $p_{\{13\}}(s_1, s_3)$ и $p_{\{23\}}(s_2, s_3)$. Эти вероятности ограничивают возможные значения вероятностей $p_T(s_1, s_2, s_3)$, так как они должны удовлетворять равенствам

$$p_{\{12\}}(s_1, s_2) = \sum_{s_3} p_T(s_1, s_2, s_3), \quad p_{\{13\}}(s_1, s_3) = \sum_{s_2} p_T(s_1, s_2, s_3),$$

$$p_{\{23\}}(s_2, s_3) = \sum_{s_1} p_T(s_1, s_2, s_3). \quad \blacktriangle$$

Локальные вероятности не определяют глобальные однозначно. Однозначность здесь достигается при дополнительных ограничениях о марковской природе функций, являющихся полным описанием объекта. Точная формулировка этих дополнительных ограничений будет дана в последующих лекциях. Сейчас же лишь еще раз подчеркнем характеристическую черту структурного распознавания: сложная функция на обширной области определения s^T определяется с помощью ряда более простых функций на значительно более узкой области определения. И как раньше, такое сведение сложного понятия к совокупности более простых не всегда возможно. Область действия структурного распознавания образуют именно те ситуации, когда такое сведение возможно. Мы увидим, что эта область очень обширна.

Проиллюстрируем введенные понятие на примере 7.1 о горизонтальных и вертикальных линиях.

Пример 7.3 Горизонтальные и вертикальные линии, иллюстрация основных понятий. Для хранения полного описания распознаваемого объекта требуется $(m * n + m + n)$ ячеек памяти, в которых содержится информация об изображении ($m * n$ ячеек), о совокупности горизонтальных линий (m ячеек) и совокупности вертикальных линий (n ячеек). Таким образом, поле T объекта - это множество $\{(i, j) \mid 0 \leq i \leq m, 0 \leq j \leq n, i + j \neq 0\}$. Только часть этого поля наблюдаема, а именно, та часть, в которой записана информация об изображении. Это - наблюдаемое поле $T_x = \{(i, j) \mid 1 \leq i \leq m, 1 \leq j \leq n\}$. Остальная часть поля, в которой записана информация о наличии или отсутствии линий на изображении, состоит из ячеек $(i, 0)$, $i = 1, \dots, m$, и $(0, j)$, $j = 1, \dots, n$. Таким образом, скрытое поле T_k - это множество $\{(i, 0) \mid 1 \leq i \leq m\} \cup \{(0, j) \mid 1 \leq j \leq n\}$. Наблюдаемые и скрытые параметры принимают значения из множества $\{0, 1\}$. Числа 0 или 1, записанные в ячейках наблюдаемого поля, обозначают, является ли ячейка черной или белой. Эти же числа, записанные в ячейках скрытого поля, обозначают, присутствует ли на изображении та или иная линия. Взаимосвязь всех параметров определяется структурой третьего порядка, состоящей из троек вида $((i, 0), (0, j), (i, j))$, $1 \leq i \leq m$, $1 \leq j \leq n$. Таким образом, структура - это множество $T = \{((i, 0), (0, j), (i, j)) \mid 1 \leq i \leq m, 1 \leq j \leq n\}$.

		$s(i, j) = 1$	$s(i, j) = 0$
$s(i, 0) = 0$	$s(0, j) = 0$	$\frac{1}{4}\varepsilon$	$\frac{1}{4}(1 - \varepsilon)$
$s(i, 0) = 0$	$s(0, j) = 1$	$\frac{1}{4}(1 - \varepsilon)$	$\frac{1}{4}\varepsilon$
$s(i, 0) = 1$	$s(0, j) = 0$	$\frac{1}{4}(1 - \varepsilon)$	$\frac{1}{4}\varepsilon$
$s(i, 0) = 1$	$s(0, j) = 1$	$\frac{1}{4}(1 - \varepsilon)$	$\frac{1}{4}\varepsilon$

Table 7.1 Eight probabilities describing noise in Example 7.2.

В случае, когда помехи не принимаются во внимание, ограничение на описание $s: T \rightarrow S$ задается списком допустимых троек $(s(i, 0), s(0, j), s(i, j))$ значений, которые допустимы на фрагменте $((i, 0), (0, j), (i, j))$. Этот список один и тот же для всех фрагментов и состоит из троек $\{(1, 1, 1), (1, 0, 1), (0, 1, 1), (0, 0, 0)\}$. Этот список формально выражает взаимную зависимость скрытых и наблюдаемых параметров, которые неформально были определены ранее. Он определяет, что ячейка (i, j) может быть белой в том и только в том случае, когда на рисунке отсутствует как i -ая горизонтальная линия, так и j -ая вертикальная.

В случае, когда должны учитываться помехи, следует указать вероятность каждой тройки значений $(s(i, 0), s(0, j), s(i, j))$. В нашем случае эти вероятности не зависят от координат (i, j) . Необходимо, таким образом, указать $|S|^3 = 8$ чисел. Они приведены в таблице 7.1. ▲

Введенные понятия, проиллюстрированные в предыдущем примере, еще не определяют задачу распознавания. Это только "действующие лица", которые будут участвовать в последующих постановках. Они должны быть определены до формулировки задачи подобно тому, как формулировке общих статистических задач предшествовали слова "Пусть X и K - два конечные множества, а $p_{XK}: X \times K \rightarrow \mathbb{R}$ - функция, определенная на декартовом их произведении". Теперь эти краткие вводные слова будут заменены следующим, более подробным введением. Пусть T и S - два конечные множества, где T - множество параметров, а S - множество значений, которые эти параметры принимают. Пусть $T_x \subset T$ - подмножество наблюдаемых, а $T_k = T \setminus T_x$ - подмножество скрытых параметров. Пусть \mathcal{T} - конечное множество подмножеств в T . Пусть для каждого подмножества $T' \in \mathcal{T}$ определено:

или подмножество $L_{T'} \subset S^{T'}$;

или распределение вероятностей $p_{T'}: S^{T'} \rightarrow \mathbb{R}$.

После этих вводных предложений будут формулироваться те или иные задачи распознавания. Задачи будут состоять в том, что по известной функции $x: T_x \rightarrow S$, определенной на наблюдаемом поле, надо будет принять разумное решение о функции $k: T_k \rightarrow S$, определенной на скрытом поле. Последующие лекции посвящены анализу задач этого класса и их конструктивному решению.

7.4 Обсуждение

Лекция 6 представляется мне водоразделом, которым завершается одна вашего курса и начинается другая. В связи с этим я прошу вас об одной любезности. По совету моего научного руководителя я веду факультативные занятия по распознаванию для небольшой группы студентов по специальности прикладная информатика. Ваши лекции я использовал, как основу этих занятий, и многие вопросы, которые я вам задавал, это лишь переадресация вопросов, которые задавали мне мои студенты.

У Тебя неплохие студенты.

Да, вполне может быть. Но чтобы убедиться в этом окончательно, я решил устроить зачет по пройденному материалу. Можете ли мне рекомендовать список тем, которые должен знать студент, прослушавший первые 6 лекций?

Предыдущие лекции должны были бы обеспечить хорошее понимание следующих понятий и задач.

1. Наблюдаемые и скрытые параметры объекта, стратегия, функция потерь, байесовский риск принятия решений.
2. Вероятность ошибочного распознавания как частный случай байесовского риска; стратегия, минимизирующая вероятность ошибочного распознавания.
3. Байесовская стратегия при возможном отказе от распознавания.
4. Детерминированный характер байесовских стратегий.
5. Задача Неймана-Пирсона и ее обобщения на случай, когда количество состояний объекта больше двух.
6. Минимаксная задача распознавания.
7. Вальдовская задача распознавания в виде, представленном в лекции 2.
8. Испытание сложных гипотез, статистические задачи распознавания при наличии мешающих параметров.
(Вид стратегий по вопросам 5-8 должен быть выведен на основании теорем двойственности).
9. Риск и эмпирический риск, теорема Вапника и Червоненкиса о необходимом и достаточном условии равномерной сходимости эмпирического риска к собственному риску.
10. Функция роста и емкость класса стратегий, достаточные условия сходимости эмпирического риска к собственному риску.
11. Обобщенные задачи Андерсона-Бахадура, необходимые и достаточные условия оптимальности стратегии в задачах Андерсона.
12. Линейные стратегии, алгоритм персептрона и теорема Новикова, алгоритмы Козинца и их обоснование.
13. Синтез фишеровских стратегий с помощью алгоритмов линейного разделения множеств.
14. ε -решение задач Андерсона-Бахадура с помощью алгоритмов линейного разделения множеств.

15. Формулировка задач кластеризации, алгоритм ISODATA, как пример алгоритма кластеризации.
16. Эмпирический байесовский подход Роббинса, задачи Роббинса и их обобщение.
17. Взаимосвязь обучения и самообучения, алгоритм самообучения, теорема о монотонном характере самообучения.

Это, пожалуй, все. Интересно, какие знания покажут Твои юные коллеги на зачете или экзамене.

Я ожидаю, что содержание последующих лекций будет существенно отличаться от предыдущих. Будут введены новые понятия, новые алгоритмы и определенный раздел начнет излагаться как бы опять от основ. Я бы хотел знать, можно ли изучать последующий материал, не полностью зная предшествующий. Некоторые студенты не записались раньше на мои занятия, но хотели бы присоединиться к моей группе. Не настал ли именно сейчас удобный момент, чтобы присоединиться к изучению ваших лекций?

Конечно, нет. Последующие лекции невозможно понять, не усвоив достаточно глубоко предшествующего материала. Твои новые студенты могут присоединиться к занятиям, но только, если самостоятельно изучат предыдущий материал. У Тебя же сохранились наши записки и даже материалы наших обсуждений.

Твой вопрос нам еще раз напомнил о распространенном заблуждении, что статистическое распознавание и структурное - это два независимых друг от друга раздела в распознавании. При таких представлениях можно овладеть лишь очень малой частью прикладных задач распознавания изображений. В распознавании изображений достаточно редки задачи, в которых не следовало бы учитывать сложные взаимосвязи составных частей изображения. Редки и задачи, в которых можно было бы игнорировать случайный характер изображений. Случайность обусловлена по крайней мере помехами, которым подвержено изображение. Таким образом, при решении прикладных задач распознавания изображений должны учитываться как сложная структура изображения, так и его вероятностная природа.

Боюсь, что вы меня неправильно поняли. Я не спрашиваю, необходимо ли для распознавания изображений знать и прошлый, и будущий материал. Хотя мне это сейчас еще тоже не вполне ясно, мой вопрос не об этом. Я спрашиваю, можно ли изучить будущий материал, не зная предыдущего.

Мы поняли Твой вопрос правильно и отвечаем еще раз: Последующий материал невозможно изучать без глубокого понимания предшествующего. Двойкая природа изображений должна учитываться при распознавании совсем не так банально, что на каком-то одном этапе обработки изображения используются чисто статистические методы, а на другом - чисто структурные. Эти две стороны распознавания совмещаются настолько огранично, что для какого-то фрагмента работающей программы просто невозможно

определить, минимизируется ли здесь вероятность ошибочного распознавания или учитываются ли сложные взаимоотношения отдельных частей изображения. И то, и другое делается одновременно.

Не представляю себе, как это может происходить.

Равно, как и многое другое. Поэтому мы и пишем эти лекции.

Хорошо. Согласимся, что весь предыдущий материал нужен для понимания следующих лекций. Можете ли вы указать минимальный набор ключевых понятий из предыдущих лекций, который послужил бы ориентацией для моих коллег, которые только сейчас присоединятся к занятиям, а первые шесть лекций изучат самостоятельно?

Ну, парень, Ты сейчас особенно настойчив. Тебе непременно нужно найти что-то лишнее в предыдущих лекциях. Ну что ж, мы будем менее настойчивы, чем Ты, и уступим Тебе.

Для понимания последующего материала непременно нужно знать байесовскую формулировку задач распознавания, формулировки задач обучения и самообучения как максимально правдоподобное оценивание модели. Конструирование алгоритмов самообучения для разного рода моделей должно быть доведено почти до автоматизма. Наконец, следует знать поиск линейных дискриминантных функций с помощью алгоритмов Козинца или пресептронных алгоритмов, то есть содержание почти всей лекции 5.

Лучше бы я не был так настойчив. Это довольно большой объем. Правда, это меня не очень пугает, так как этот материал, мне кажется, я хорошо изучил и понял. Меня удивляет, что нам все еще понадобятся методы разделения множеств точек линейного пространства с помощью гиперплоскости. Представление наблюдений в виде точек линейного пространства вы подвергли в лекции такой жесткой критике, что я думал, что их следует прямо-таки забыть, как нечто дезориентирующее. Моя настойчивость была вызвана именно желанием, чтобы вы явно подтвердили, что для понимания последующего материала необходимо несколько перестроиться и оторваться от привычного представления множества наблюдений в виде точек линейного пространства. Сейчас я вижу, что это не так, и это мне кажется странным.

Нам бы это тоже показалось странным, если бы мы не привыкли к стремлению научных идей жить самостоятельно и работать совсем не в той области, где они родились. Еще раз вспомним Минского и Пейперта [Minsky and Papert, 1969], которые сравнивали распознавание образов с математическим романом, в котором действующие лица появляются, надолго исчезают так, что о них почти забывают, и, наконец, появляются неожиданно, но в нужный момент и в нужном месте. Тогда-то и обнаруживается их сила, которая была едва заметна при их рождении. Критика

линейно-пространственных методов распознавания изображений, приведенная в предыдущей лекции, справедлива. Вместе с тем, при решении некоторых задач настройки алгоритмов структурного распознавания мы будем активно использовать методы разделения двух конечных подмножеств линейного пространства с помощью гиперплоскости. Не старайся разобраться в этом противоречии уже сейчас. В свое время Твои знания здесь уложатся нужным образом.

Мне уже интересно! Это неожиданный поворот событий. Ф.Розенблатт изобретает алгоритмы для поиска подходящей стратегии в определенном классе стратегий. Спустя некоторое время оказывается, что для задач зрительного распознавания этот класс стратегий недостаточно хорошо приспособлен, и рождается структурное распознавание, поначалу, как альтернативное направление в распознавании образов. Алгоритмы Розенблатта, а вместе с ними и проблематика обучения распознаванию уходят со сцены, и на ней начинают разыгрываться события структурного анализа. Но когда при развитии этих событий возникает необходимость настройки распознающих систем, опять начинают играть свою роль алгоритмы линейного разделения множеств. Понимаю ли я эту картину правильно?

В общем, да.

Но тогда об этом можно было бы писать книги!

А мы что делаем?

Математическое содержание предыдущих лекций мне было вполне понятно. Можете ли вы мне сказать об основных математических понятиях, которые будут использованы в последующих лекциях? Возможно, мне следовало бы подготовиться?

Как и раньше, нам нужно будет легко оперировать с вероятностями. Из новых понятий наиболее важными являются марковские цепи, графы, кратчайшие пути на графах, динамическое программирование, конечные автоматы, регулярные выражения, формализмы Хомского для формальных языков и грамматик, автоматные и контекстно свободные грамматики и языки. При более абстрактных рассмотрениях нам потребуются некоторые алгебраические конструкции, в частности, полукольца с идемпотентным сложением. Пожалуй, это все. Не беспокойся об этом заранее. Нам потребуются только самые основные, а следовательно, простейшие понятия. Мы их будем вводить по мере необходимости. Строго говоря, изложение последующего материала не предполагает наличие знаний, дополнительных по сравнению с теми, которые нужны были для понимания предыдущего материала. Требуется лишь желание эти знания приобрести.

Мне кажется, что я понял основные понятия, на которых будут строиться дальнейшие формальные конструкции. Объектом исследования будут

функции вида $T \rightarrow S$, где T - конечное множество, для которого определена структура $\mathcal{T} \subset 2^T$, где 2^T обозначает множество всех возможных подмножеств множества T . Почему ничего не было сказано о структуре множества S ?

В этих лекциях структура множества S не будет приниматься во внимание. Курс построен, как противопоставление теории, в которой наблюдение - это функция, принимающая значение на множестве чисел, но определенная на множестве, лишенной какой бы то ни было структуры. Противопоставление состоит в том, что мы увидим, какие результаты следуют из предположения, что область определения функции является хорошо структурированным множеством, но никакие предположения не делаются о множестве значений, которые принимают исследуемые функции. Конечно, такое исследование, диаметрально противоположное линейно-пространственному, имеет свои недостатки, как и любая другая крайность. Но полученные нами выводы будут обладать достаточно большой общностью именно в силу того, что они имеют место при любой структуре множества S . В то же время следует признать, что при ограничении класса множеств S можно было бы получить и дополнительные, более глубокие результаты, чем изложенные в данном курсе, но имеющие более узкую область действия.

Я обнаружил определенное несоответствие в описанных вами примерах с горизонтальными и вертикальными линиями. С одной стороны, было сказано, что рассматриваются только такие совокупности линий, в которых содержатся не все горизонтальные и не все вертикальные линии. Это ограничение было существенным, так как на нем базировалось распознавание изображений при отсутствии помех.

С другой стороны, когда множество допустимых изображений формулировалось средствами структурного распознавания, это ограничение не было принято во внимание. Структура \mathcal{T} и ограничения на фрагменты были определены так, что допустимыми оказывались любые совокупности линий, том числе и те, которые раньше считались недопустимыми. Я не хотел вас затруднять этой мелочью, полагая, что это просто ваш недосмотр. Но когда я постарался сам устранить этот недосмотр, я увидел, что это невозможно, и здесь что-то более серьезное, чем простой недосмотр. Дело в том, что совокупность значений $(s(i, 0), i = 1, 2, \dots, m)$ не может быть любой, а только такой, в которой по крайней мере одна переменная $s(i, 0)$ равна нулю. Это ограничение на m переменных невозможно выразить с помощью ряда ограничений на меньшее количество переменных. Я могу это доказать, но думаю, что вы это знаете и без меня. Но является ли тогда задачка с горизонтальными и вертикальными линиями именно тем примером, с которым невозможно справиться, даже располагая методами, о которых пойдет речь дальше?

И да, и нет, и не требуй от нас сейчас более точного ответа. Ты задал очень трудный вопрос. Сейчас мы только покажем, как с помощью средств, указанных в лекции, выразить ограничение, что совокупность чисел $(s(1, 0),$

$s(2, 0), \dots, s(m, 0)$) должна содержать по крайней мере один нуль. Введем вспомогательные переменные $z(0), \dots, z(m)$ и определим локальные ограничения для каждой тройки значений $(z(i-1), s(i, 0), z(i))$, $i = 1, 2, \dots, m$. Ограничения определим так, чтобы переменные $z(i)$ имели следующий содержательный смысл: число $z(i)$, $i = 1, 2, \dots, m$, равно нулю тогда и только тогда, когда по крайней мере одно из значений $(s(1, 0), s(2, 0), \dots, s(i, 0))$ равно нулю. Локальные ограничения должны состоять в том, что $z(0)$ должно равняться 1, $z(m)$ должно равняться 0, а тройка $(z(i-1), s(i, 0), z(i))$ должна совпадать с одной из следующих четырех троек: $(1, 1, 1)$, $(1, 0, 0)$, $(0, 1, 0)$ и $(0, 0, 0)$.

Я вижу, что я действительно затронул очень сложные вопросы. Я понимаю, что некоторые "многоместные" отношения можно заменить совокупностью "маломестных" и таким образом упростить их представление. Я понимаю, что такое упрощение не всегда возможно. Сейчас я вижу, что иногда такое упрощение становится возможным, если только к исходным переменным добавить некоторые вспомогательные переменные. Таким образом, чтобы упростить представление некоторого сложного "многоместного" отношения, нужно его для начала еще более усложнить, добавив к нему еще дополнительные фиктивные переменные. Так ли это?

Если закрыть глаза на некоторую расплывчатость Твоего утверждения, то Ты прав.

Мне кажется, что я готов приступить к изучению следующих лекций. Но до этого мне очень важно поднять еще один вопрос, и я прошу Вас понять меня правильно. Я преследую вполне приземленные цели, и сейчас самый удобный момент, чтобы вспомнить, зачем я, собственно говоря, принялся изучать ваши лекции. Если вы помните, мой интерес к распознаванию образов определился тем, что я написал программу для распознавания стандартных цифробуквенных символов. Я сделал это безо всякой теории, основываясь только на своих соображениях, казавшихся мне вполне разумными. Вопреки моим ожиданиям программа показывала неудовлетворительные результаты. Сейчас, благодаря вашим лекциям, я ознакомился с фундаментальными результатами распознавания образов, но я не имею малейшего представления, как вернуться назад, к моим вполне прозаичным целям. Рекомендации теории распознавания для меня слишком абстрактны, хотя и, не буду скрывать, некоторые из них очень интересны. Я надеялся, что во второй части курса мы опустимся, наконец, с абстрактного уровня на более конкретный, связанный именно с изображениями. Но сейчас я вижу, что, вместо того, чтобы спуститься с небес на грешную землю, мы уйдем в такие заоблачные дали, из которых мои скромные цели окажутся едва заметными. У меня такое чувство, будто я все более и более удаляюсь от своих первоначальных целей, ради которых я и пустился в путешествие с вами, которое, согласитесь, оказалось совсем не из легких. Могу ли я попросить вас указать конкретно, что из прошлого материала

ла, а возможно, и из будущего, способствует достижению моих не совсем возвышенных целей?

Попросить, конечно, можешь. Но Тебе ясно, что этот Твой вопрос сложнее всех предыдущих. Во первых, нам следовало бы ознакомиться с Твоей прикладной задачей, и во вторых, узнать, что Ты уже сделал и где Ты застрял. В любом случае, ответ на этот Твой вопрос не может быть кратким. Возможно, он не будет точным попаданием в яблочко, если можно так назвать Твои конкретные затруднения. Мы постараемся проанализовать пройденный Тобой путь и затруднения, в которые Ты попал, и предсказать, что Тебя еще ожидает. Если мы не в точности опишем именно Твою ситуацию, то давай считать, что мы говорим не о Тебе, а ком-то третьем.

Повидимому, сначала Ты постарался справиться с задачей при упрощающем предположении, что в поле зрения находится изображение только одной буквы, более того, являющейся либо буквой *A*, либо буквой *B*. Буква может находится в любом месте поля зрения, но имеет фиксированную ориентацию и размер. Ты догадываешься, что возникнут определенные затруднения, когда придется распознавать изображения, на которых представлены несколько букв, но их преодоление Ты оставляешь на будущее, когда Ты справишься с теми проблемами, которые лежат на поверхности.

Ты просканируешь несколько сотен изображений, скажем, размера $150 \cdot 100$ пикселей. Собственно буква занимает в этом поле зрения лишь $30 \cdot 20$ пикселей. Размер поля зрения у Тебя в пять больше размера букв, потому что Ты полагаешь, что очень скоро Тебе придется распознавать простейшие тексты, состоящие из пяти строк и пяти букв в каждой строке. Для начала Ты имеешь дело с изображением лишь одной буквы, произвольно расположенной в поле зрения. Буква может занимать любое из $(100 - 20) \cdot (150 - 30) = 9600$ положений. Просканировав изображение в бинарном режиме, Ты получаешь его в виде двумерного массива x , в ячейке $x(i, j)$ которого записаны 1 или 0 в зависимости от того, является ли пиксел с координатами (i, j) белым или черным. Отображая результаты сканирования на экране дисплея, Ты видишь вроде бы прекрасные изображения, которые, как Тебе кажется, не стоит особого труда автоматически распознать.

Исполненный оптимизма Ты конструируешь свою программу на основе следующих разумных соображений. Ты готовишь два изображения размером $20 \cdot 30$, которые соответствуют двум распознаваемым буквам *A* и *B*, и которые, как Тебе кажется, являются идеальными, неискаженными изображениями. Их Ты называешь эталонами. Затем Ты размножаешь каждый эталон 9600 раз, получая 9600 изображений, представляющих букву *A* или букву *B* во всех возможных положениях в поле зрения. Эти изображения можно пронумеровать так, что v_{tk} , $t = 1, 2, \dots, 9600$, $k = A, B$, обозначает букву k в положении t . Изображение x будет распознаваться так, что будет вычисляться его различие от каждого из $2 \cdot 9600$ эталонных изображений.

Твоя программа будет искать изображение $v_{t^*k^*}$, наименее отличающееся от изображения x , и считать, что изображение x представляет букву k^* в положении t^* . Ты недолго размышляешь над тем, что считать отличием эталона v_{tk} от изображения x . Тебе ясно, что это должно быть хеммингово расстояние

$$\sum_{i=1}^{150} \sum_{j=1}^{100} |x(i, j) - v_{tk}(i, j)|, \quad (7.1)$$

и таким образом, Твоя программа определяет результат k^* распознавания изображения x как

$$k^* = \operatorname{argmin}_k \left(\min_t \sum_{i=1}^{150} \sum_{j=1}^{100} |x(i, j) - v_{tk}(i, j)| \right). \quad (7.2)$$

Результаты распознавания, которые показывает эта программа, Тебя ошеломляют. Они не просто неправильные. Они необъяснимы. Программа допускает ошибки, но они происходят совсем не там, где Ты мог бы объяснить их появление. Тебе было бы понятно, если бы ошибки происходили на тех буквах, которые искажены помехами настолько, что их невозможно узнать. Но такие ошибки сравнительно редки. Очень часто алгоритм уверенно распознает изображение, как букву A , в то время, как Ты видишь на экране почти идеальную букву B . Поначалу Ты думаешь, что в программе содержится ошибка. После тщательной ее отладки Ты приходишь к выводу, что программа работает правильно, и чтобы понять причину ошибочного распознавания, Тебе следует выполнить более обширную экспериментальную проверку алгоритма распознавания. Для этого Тебе нужно на порядок ускорить быстроедействие программы, так как в ее теперешнем виде она очень медленная. При ее разработке Ты не обращал внимание на ее быстроедействие, намереваясь заняться этим после того, как убедишься в принципиальной работоспособности алгоритма. Теперь же Тебе придется заняться быстроедействием программы, реализующей алгоритм, в работоспособности которого Ты еще не уверен.

Стремясь ускорить работу программы, Ты прежде всего замечаешь, что, в соответствии с алгоритмом (7.2), различие (7.1) считается для всех возможных положений t буквы в поле зрения. Ты полагаешь, что если бы было известно действительное положение t_0 буквы, то ответ k^* можно было бы определить, вычисляя

$$k^* = \operatorname{argmin}_k \sum_{i=1}^{150} \sum_{j=1}^{100} |x(i, j) - v_{t_0k}(i, j)|, \quad (7.3)$$

а не (7.2). Вопрос, таким образом, состоит в выборе дешевого по объему вычислений алгоритма, который мог бы более или менее точно определять положение буквы в поле зрения. Не будем строить догадки, какому известному алгоритму такого рода Ты отдал предпочтение. Все они очень похожи и одинаково плохи. Ты выбираешь один из них.

Программа, реализующая (7.3), работает в тысячу раз быстрее, чем (7.2), и сейчас Ты можешь выполнить массовый эксперимент, а не только распознавание отдельных примеров. Естественно, программа сама по себе не стала работать лучше, но сейчас Ты начинаешь разбираться в причинах неправильного распознавания. Ты видишь, что хеммингово расстояние (7.1) очень грубо представляет Твое интуитивное понимание различия между изображениями, потому что в таком вычислении различия все пиксели принимают одинаковое участие. Но, наблюдая массу изображений, Ты видишь, что это не так. Некоторые пиксели в изображении A почти всегда белые, другие почти всегда черные, третьи - чаще белые, чем черные, и т.п. Тогда Ты пытаешься отсортировать пиксели по этому признаку, то есть выделить определенные зоны в поле зрения, причем отдельно для букв A и отдельно для букв B . Это каторжная работа, состоящая из проверки многих вариантов. Результаты каждой проверки неутешительны, но всякий раз Тебе кажется, что стоит еще немного подправить сортировку пикселей и уже все будет в порядке. Но постепенно Тобой овладевает чувство уходящей из-под ног почвы. Возможно, именно на этом этапе состоялось наше знакомство и началось наше сотрудничество, столь для нас приятное.

За исключением отдельных деталей вы почти точно реконструировали мои попытки. Именно на этом этапе я понял, что разнообразие пикселей невозможно учесть лишь тем, что множество пикселей делится на некоторое малое количество зон. Каждый пиксел должен характеризоваться своим собственным весом, с которым он должен присутствовать в формуле (7.1). Наиболее приятной в этом отношении оказалась третья лекция, в которой непосредственно указано, как следует выбирать эти весовые коэффициенты. Отличие изображения x от эталона v_A буквы A и от эталона v_B буквы B следует вычислять по формуле

$$\sum_i \sum_j \alpha_{ij}^k \left(x(i, j) - v_k(i, j) \right)^2, \quad k = A, B, \quad (7.4)$$

которая может пониматься двояко. Если и изображение x , и эталоны v_A и v_B бинарные, то коэффициенты α_{ij}^k должны быть равны

$$\log \left(\frac{p_{ij}^k}{1 - p_{ij}^k} \right),$$

где p_{ij}^k - вероятность того, что в изображении x , на котором представлена буква k , сигнал $x(i, j)$ отличается от значения $v_k(i, j)$.

Я применил также более точное сканирование, результатом которого является не бинарное, а многоуровневое изображение, при котором сигнал в пикселе принимает значения от 0 до 255. В этом случае я принял, что числа $v_k(i, j)$ в формуле (7.4) равны математическим ожиданиям сигналов $x(i, j)$ при условии, что на изображении представлена буква k . В соответствии

с рекомендациями третьей лекции, коэффициенты α_{ij}^k я выбрал равными $(\sigma_{ij}^k)^{-1}$, где σ_{ij}^k - условная дисперсия сигнала $x(i, j)$. Выполнение этих рекомендаций не привели ни к чему хорошему. В последующих лекциях я уже не нашел ничего, что смог бы использовать непосредственно для своих целей.

Разработанный к настоящему времени алгоритм ужасно примитивный, но он существенно лучше, чем алгоритм, разработанный Тобой до встречи с нами и основанный на вычислении расстояний Хемминга. Полезность третьей лекции совсем не в том, что она предлагает Тебе какие-то две стратегии распознавания, а в том, что она указывает, каким образом из статистической модели $p_{X|K}$ выводится стратегия того или иного вида. Если предположение о модели действительно имеют место, то выведенная стратегия обязана (!!!) работать не только правильно, но лучше любой другой стратегии.

Сейчас, когда Твой алгоритм продолжает работать неправильно, Тебе не следует исследовать сам алгоритм, не следует его ремонтировать или изобретать новый. Тебе следует исследовать сами изображения, до поры до времени не пытаясь их распознавать вообще, а с единственной целью определить, чем отличаются их статистическая модель $p_{X|K}$ от той модели, из которой следует стратегия (7.4).

Я вынужден вас прервать, даже рискуя показаться невежливым. До сих пор нам удавалось не упускать из виду мои заботы по построению алгоритма распознавания текстовых изображений. Сейчас вы предлагаете мне фактически прекратить разработку алгоритма распознавания и начать какие-то исследования. У меня возникает ощущение, что мы начинаем удаляться от моих целей. Вы втягиваете меня в обсуждение каких-то вопросов, которые, возможно, окажутся такими же интересными, как и ваши лекции. Но после обсуждения я опять буду в недоумении, как вернуться к своим изначальным проблемам и как применить все то интересное, что я от вас узнал, для своих конкретных целей.

Не беспокойся, Ты правильно сделал, что прервал нас. Мы не удаляемся от решения Твоих проблем, а вскрываем тот факт, что мы с Тобой еще далеки от этого решения. Согласись, что если два человека одинаково далеко находятся от цели, то тот, кто знает, насколько цель далека, имеет важное преимущество перед другим, который находится в приятном заблужденн, что цель находится уже на расстоянии вытянутой руки. Путь к алгоритму распознавания изображений обязательно проходит через исследование множества распознаваемых изображений. Этот путь иногда длинный, иногда очень длинный, но это путь, ведущий к алгоритму. Попытка же свернуть с этого пути и идти напрямиком к алгоритму ведет в такие дебри, из которых не выбраться иначе, кроме как вернуться в исходную точку.

Давай посмотрим на формулу (7.4) и постараемся понять, почему она не работает удовлетворительно, хотя и работает чуть лучше, чем формула

(7.3). Посмотрим, какие предпосылки, при которых процедура (7.4) может быть обоснована, в действительности не выполняются.

Если Ты просмотришь даже небольшое количество изображений, Ты немедленно обнаружишь по крайней мере одно предположение, которое необходимо для вывода формулы (7.4), а в действительности не выполняется. Это предположение об условной независимости сигналов $x(i, j)$ в различных пикселах изображения. Ты обнаружишь группы пикселей, черноты которых сильно коррелированы, причем положительно коррелированы. Этим мы имеем в виду, что сигналы в пикселах внутри этих групп, хотя и меняются от изображения к изображению, но меняются, оставаясь при этом приблизительно равными друг другу. Сигналы в некоторых других пикселах, наоборот, отрицательно коррелируют друг с другом. При большом значении сигнала в одном пикселе значение сигнала в другом пикселе, как правило, малое. Все эти наблюдения совершенно не согласуются с предположением о взаимной независимости сигналов в пикселах.

Зависимости такого рода можно было бы учесть в рамках многомерной гауссовой модели общего вида. Но для этого надо располагать ковариациями сигнала, вычисленными для каждой пары пикселей. Следовательно, речь идет о ковариационной матрице размером приблизительно 300 на 300. Надо иметь очень большую выборку изображений, чтобы матрицу таких размеров оценить достаточно точно. Кроме того, надо будет затем вычислить матрицу, обратную полученной ковариационной матрице. Эти перспективы нагоняют на меня страх.

Твои опасения вполне оправданы. Лучше избегать рассмотрения гауссовой модели общего вида в пространстве такой размерности. Мы с Тобой уже просмотрели много букв и достаточно хорошо знаем их свойства. Поэтому мы можем не только констатировать взаимную зависимость сигналов в пикселах, но и объяснить природу этой зависимости. Если бы изображения располагались в фиксированном положении в поле зрения, то зависимость между сигналами в пикселах была бы существенно меньше. Эта зависимость обусловлена в основном тем, что применяемый Тобой алгоритм определения положения буквы не точный. Действительное положение буквы может быть сдвинуто на один или два пиксела влево или вправо, вверх или вниз по сравнению с положением, который указал Твой алгоритм. Поэтому, при переходе от изображения к изображению, сигналы, наблюдаемые в отдельных группах пикселей, меняются не независимо, а одновременно во всех пикселах группы. Это происходит потому, что изображение в целом сместилось, скажем, на два пиксела вправо. Таким образом, зная изображения, Ты не просто констатируешь факт зависимости сигналов, но обнаруживаешь конкретный механизм этой зависимости. Поэтому Ты можешь выразить эту зависимость не с помощью общей многомерной гауссовой модели, а с помощью модели, которая непосредственно выражает механизм обнаруженной зависимости. Возможно, для этой цели хорошо подойдет представление распределения вероятностей $p_{X|K}$ в виде

смеси двадцати пяти частных распределений. Число 25 есть количество положений, в которых может находиться буква при условии, что Твой алгоритм указал для нее некоторое определенное положение. Таким образом,

$$p_{X|k} = \sum_{t=1}^{25} p^k(t) p_{X|k}^t, \quad (7.5)$$

откуда следует, что стратегия распознавания должна иметь вид: решение в пользу буквы A следует принимать, когда

$$\sum_{t=1}^{25} p^A(t) p_{X|A}^t(x) \geq \sum_{t=1}^{25} p^B(t) p_{X|B}^t(x), \quad (7.6)$$

и в пользу буквы B в противном случае. В формулах (7.5) и (7.6) число $p^A(t)$ обозначает, как часто действительное положение буквы A смещено на t от положения, определенного Твоим алгоритмом. Подобный смысл имеют числа $p^B(t)$. Они, конечно, отличаются от чисел $p^A(t)$, потому что применяемый Тобой алгоритм определяет положения букв A с одной точностью, а букв B - с другой. Функции $p_{X|k}^t$ определяют распределение вероятностей изображений буквы k в положении t . Об этих вероятностях можно уже с большим основанием утверждать, что они представляют совокупность независимых случайных величин. Ведь исключено влияние смещения изображения в поле зрения, которое по нашему мнению было основным фактором, обуславливающим зависимость сигналов в различных пикселах.

При применении формул (7.5) и (7.6) следует учитывать, что разные смещения t имеют разную вероятность. Ведь выбранный Тобой алгоритм, хотя и не совершенный, но и не совсем плохой. Достаточно часто он определяет положение буквы правильно, иногда дает ошибку в один пиксел, еще реже эта ошибка равна двум пикселям. Алгоритм распознавания буквы должен учитывать свойства алгоритма, определяющего положение буквы. Следовательно, должны быть известны 50 чисел $p^k(t)$, $t = 1, 2, \dots, 25$, $k = A, B$, чтобы применять их в формулах (7.6). Нахождение этих чисел - это значительно более конкретная задача, чем поиск алгоритма распознавания в необозримом множестве алгоритмов. Обращаем Твое внимание, что в данном случае речь идет лишь об оценке качества имеющегося алгоритма определения положения буквы, а не о том, чтобы его улучшить или изобрести новый. Для реализации алгоритма (7.5), (7.6) не обязательно нужен точный алгоритм определения положения буквы. Он может быть и неточным, но нужно точно знать, насколько он неточен.

Ваши объяснения показались мне настолько понятными, что я немедленно запрограммировал метод (7.6), чтобы его проверить. И вдруг, как гром среди ясного неба, обнаружилось, что я никаким образом не могу определить эти несчастные 50 вероятностей $p^k(t)$, которые следует включить в программу. Чтобы их определить, мне нужно выполнить серию измерений над выборкой случайных изображений буквы k и по каждому изображению получить два числа: положение t буквы, найденное моим алгоритмом,

и действительное положение t_0 . Из этих двух величин только величина t может считаться известной. Это то, что выдал мой алгоритм. Но я не знаю, откуда мне взять величину t_0 . Естественно, у меня нет программы для ее измерения. Если бы она у меня была, то именно ее я бы использовал в своем алгоритме распознавания. В качестве такого измерителя невозможно использовать даже человека. Когда я сам наблюдаю какое-то изображение, я с уверенностью могу определить, представляет ли оно букву A или B . Но я не в состоянии определить, совпадает ли положение буквы с положением, которое указала моя программа, или отличается ли оно на один пиксел или два пиксела и т.п.

На этом мои трудности не заканчиваются. Для применения алгоритма (7.6) мне нужно знать не только 50 чисел $p^k(t)$, но и совокупность функций $p_{X|k}^t$. Для того, чтобы их получить, мне нужно иметь выборку изображений одной и той же буквы (в этом нет проблем), но в одном и том же положении в поле зрения. А вот это уже проблема. У меня нет такой выборки, я не знаю, как сканировать изображения, чтобы быть уверенным, что они с точностью до одного пиксела попадают в одно и то же место в поле зрения, у меня нет программы, с помощью которой я бы мог безошибочно измерять такой скрытый параметр, как положение буквы в поле зрения. Таким образом, для реализации алгоритма мне необходимо многомерное распределение вероятностей $p_{X|k}^t$, которое невозможно получить экспериментально.

Так или иначе, я попал в дебри, если не на стадии придумывания алгоритма, то на стадии исследования модели наблюдаемых изображений. Разница лишь в том, что в первом случае я бы забрел в дебри самостоятельно (а может быть, и не забрел бы в них?), а во втором случае я туда попал, следуя вашим рекомендациям.

Не сердись на нас, но мы намеренно завели Тебя в ситуацию, которую Ты оцениваешь как дебри. Мы очень надеялись, что Ты сам увидишь, что здесь возникают довольно простые задачи самообучения, в которых Ты достаточно хорошо ориентируешься. Более того, здесь надо справиться именно с теми частными случаями самообучения, которые Ты лихо рассмотрел при обсуждении лекции 6. Мы не предупредили Тебя, что здесь Тебя ожидают именно эти задачи, чтобы Ты сам убедился, что самообучение - это не просто интеллектуальное упражнение и уж вовсе не витание в облаках. Самообучение предназначено для вполне прозаичного статистического анализа данных, с которым сталкивается статистик чуть ли не на первых шагах исследования сложного случайного объекта, примером которого является изображение. Конечно, если это исследование выполняется серьезно.

Пусть Тебя не пугают дебри, в которые Ты попал, потому что Ты знаешь, как из них выбраться. Для оценки параметров статистической модели (7.5) примени алгоритм, который Ты написал при обсуждении лекции 6. Ты еще удивлялся, как просты полученные Тобой алгоритмы. Нам уже любопытно, что из этого получится.

Вроде бы начинает получаться. Я оценил параметры модели с помощью алгоритмов самообучения и подставил эти параметры в алгоритм распознавания. Результаты распознавания позволяют мне надеяться, что я смогу добиться здесь практически приемлемых результатов. В течение достаточно длительной экспериментальной проверки я не получил ни одного (!!!) ошибочного результата распознавания.

Не хотим умалять Твой успех, но ... Не забывай, что емкость класса стратегий вида (7.6), очень велика и учти выводы статистической теории обучения, которые мы с Тобой обсуждали в лекции 4. Результаты распознавания, полученные в процессе обучения, могут существенно отличаться от результатов распознавания при эксплуатации алгоритма, полученного в результате обучения.

Как вы могли подумать, что я забыл о выводах лекции 4? Само собою разумеется, что параметры статистической модели (7.5) я настраивал по одной выборке данных, и совсем по другой выборке проверял качество распознавания, которое обеспечивает настроенный алгоритм.

Предварительно и с большими оговорками можешь себя поздравить. Сейчас Ты располагаешь не только алгоритмом распознавания, но и машинной моделью распознаваемых изображений. Ты можешь ее исследовать с различными целями, не только для конструирования алгоритмов распознавания.

Я уже это делаю. Прежде всего меня интересовало, как выглядят частные распределения вероятностей $p_{X|k}^t$. Я ожидал, что они будут построены так, будто алгоритм самообучения самостоятельно пришел к выводу, что изображения подвержены влиянию такого параметра, как их смещение в поле зрения. Это бы выразилось в том, что функции $p_{X|k}^t$ при различном значении t отличались бы друг от друга смещением. Но ничего подобного не произошло. Мне не удалось как-то разумно интерпретировать смысл параметра t в совокупности функций $p_{X|k}^t$, $k = A, B$, $t = 1, 2, \dots, 25$.

Видишь ли, мы с Тобой старались определить факторы, влияющие на все изображение в целом и таким образом обуславливающие взаимную зависимость его компонент. В первую очередь мы вспомнили о таком факторе, как смещение изображения в поле зрения. На самом деле существует еще много факторов, играющих ту же роль. Например, толщина линий знака, расположение знака относительно раstra. Если бы мы этому вопросу уделили больше времени, мы бы нашли еще что-то. Алгоритм самообучения посвятил этому вопросу столько времени, сколько счел нужным, и в рамках выделенных возможностей (ограниченных числом 25) учел все факторы, определяющие зависимость отдельных компонент изображения.

Я думаю, что полученная мною статистическая модель в нулевом приближении соответствует действительности, и выполнил следующее ее исследо-

вание. Я распознавал изображения, сгенерированные в соответствии с полученной моделью. Эксперименты с модельными изображениями зачастую более информативны, чем эксперименты с реальными изображениями. В таких экспериментах известна вся подноготная изображения, не только, какую букву оно представляет, но и значение параметра t , которое в реальном изображении было бы скрыто и не могло быть определено даже человеком.

Эксперименты показали, что вероятность ошибочного распознавания изображения существенно зависит от значения параметра t . Изображения, сгенерированное при более вероятном значении этого параметра, распознаются существенно лучше, чем при менее вероятном его значении. В этом нет ничего удивительного, так как стратегия построена так, чтобы минимизировалась средняя вероятность ошибки. Поэтому стратегия автоматически строится так, чтобы правильно распознавать изображения, соответствующие более вероятным значениям t .

Хоть я и понимаю все эти соображения, я бы не хотел им бездумно следовать. Мне претит трактовка параметра t , как случайного параметра по крайней мере потому, что я не знаю его содержательный смысл. Поэтому мне бы больше подошла не байесовская стратегия, минимизирующая вероятность ошибки в среднем по значениям параметра t , а небайесовская стратегия, обеспечивающая малую вероятность ошибки при любом значении этого параметра. Но это значит, что я должен всю задачу решить как бы заново. Следует ли мне это делать? Если я пойду по этому пути, то весь уже пройденный путь окажется напрасным. Можно будет выбросить все программное обеспечение, которым я сейчас располагаю. А это мне жаль делать. В него вложено уже много труда и, в конечном итоге, оно показывает вполне пристойные результаты.

Наконец мы узнаем Тебя прежнего. До сих пор нам казалось, что с нами беседует кто-то вместо Тебя. Ты приходишь к тому, на что мы сами хотели обратить Твое внимание. Но Ты неправ, что все, чем Ты сейчас располагаешь, можно выбросить в корзину. Если Ты еще раз внимательно просмотришь вторую лекцию, Ты увидишь, что решением сформулированной тобой небайесовской является стратегия того же вида (7.6), которую Ты уже запрограммировал. Отличие состоит лишь в том, что величины $p^k(t)$ должны вычисляться не алгоритмом самообучения, а как-то иначе. Не будем на этом долго останавливаться. Ты справишься с этим без нас.

Но это значит, что я напрасно прогаммировал алгоритм самообучения, а он мне тоже дорог.

Совсем не напрасно! Этот алгоритм вычисляет не только априорные вероятности значений параметра t , но и условные вероятности $p_{X|k}^t$, которые Ты по-прежнему активно используешь.

Ответим еще на один вопрос, который Ты и не задавал. Мы не исключаем, что Тебе удастся сконструировать алгоритм, который покажет вполне

приличные результаты при любых значениях t . В таком случае можно даже пойти на некоторое ухудшение результатов и добиться этим упрощения стратегии распознавания. Например, вместо стратегии вида (7.6), можно использовать линейные дискриминантные функции и искать наилучшее решение в этом суженном классе стратегий, как это описывалось в пятой лекции.

Таким образом, Ты видишь, что даже для Твоей сравнительно простой задачи и при довольно беглом просмотре нам пришлось использовать почти весь теоретический материал предыдущих лекций.

Да, это так, но установить соответствие между абстрактными понятиями и содержанием прикладной задачи оказалось довольно трудно. Это соответствие не лежит на поверхности.

Это не так. Если тщательно изучить прикладную задачу и хорошо ориентироваться в теории, то это соответствие достаточно часто просто бросается в глаза. Поэтому мы удивлены, что Тебе потребовались наши подробные объяснения по поводу того, как относится материал наших лекций к Твоей прикладной задаче. С содержанием лекций Ты знаком достаточно хорошо, значит, Ты недостаточно внимательно исследовал изображения, которые хотел распознавать. Возможно, Ты, как и многие другие, надеялся, что знание теории избавит Тебя от тщательного изучения объекта распознавания. Ты был неправ. Еще и еще раз напомним Тебе, что теория распознавания - это не чудесное средство типа скатерти-самобранки для лентяев.

Но если тщательно изучить свою прикладную задачу, то решить ее можно, не прибегая ни к какой теории. Разве я не прав?

Ты прав. Вопрос лишь в том, сколько времени на это потребуется. Ты мог бы спросить, например, можно ли вычислить площадь под графиком полиномиальной функции на интервале от 0 до 1, не зная интегрального исчисления. Конечно, можно, ведь в принципе ничто не мешает тому, чтобы самостоятельно сформулировать понятие производной, найти формулы дифференцирования полиномов, прийти к понятию интеграла и, наконец, сформулировать и доказать теорему Ньютона-Лейбница. Но если все это уже знать, вычислить площади можно скорее.

Я благодарен вам за подробное рассмотрение прикладной задачи, которой я занимаюсь. Как оказалось, для ее решения достаточно знать лишь материал предыдущих лекций. Мы нигде не натолкнулись на необходимость структурного анализа изображений. Обозначает ли это, что рассматриваемая задача входит в класс тех задач, которым повезло и решить которые можно без привлечения структурного распознавания?

То, что мы с Тобой сделали, лишь с большой натяжкой можно считать решением прикладной задачи. Мы ведь разобрали только ситуацию, когда

в поле зрения находится только одна буква. Программное обеспечение, которым Ты сейчас располагаешь, это всего лишь лабораторная диковинка, а не серьезное изделие.

Но я думаю, что сейчас уже выполнена наиболее проблематичная часть работы. Сейчас мне осталось только разбить изображение текста на маленькие прямоугольники, каждый из которых содержит одну и только одну букву. Могу ли я после этого утверждать, что решена практическая задача?

Ну, не полностью вся задача, но значительная ее часть. Не думай однако, что Тебе удастся разделить текст на буквы с помощью каких-то простых остроумных средств, основанных лишь на Твоей сообразительности, а не на знании структурных методов. Опять и опять призываем Тебя чаще и внимательнее смотреть на изображения, которые Ты собираешься распознавать. Это очень действенное отрезвляющее средство. На изображении Ты увидишь, что буквы сплошь и рядом прикасаются друг с другом так, что исчезает белый промежуток между ними; в результате непрочтено линии знака разрываются так, что множество черных пикселей, относящихся к одному знаку, перестает быть связным; прямоугольники, описанные вокруг одного знака, отличаются по высоте и ширине; на стыке двух знаков, составленном из двух половинок соседних знаков, образуется конфигурация, похожая на тот знак, который в этой строке отсутствует (таковы пары $oo \rightarrow x$, $xx \rightarrow o$, $cl \rightarrow d$, $ic \rightarrow k$, $lc \rightarrow k$); описанные вокруг знаков прямоугольники пересекаются друг с другом и т.д., и т.д., и т.д.

Когда Ты увидишь эти мелкие и не очень мелкие каверзы реальных изображений реальных текстов, Ты придешь к правильному выводу, что распознавание строки не сводится к распознаванию букв этой строки, потому что разделить строку на буквы невозможно, не распознав эти буквы. Этот замкнутый круг разрывается лишь тем, что Ты встанешь перед проблемой распознавания всей строки целиком. Эта перспектива приведет Тебя в ужас, так как количество всех возможных строк необозримо велико. Этот момент будет самым подходящим для того, чтобы изучить две следующие лекции и ознакомиться с методами распознавания последовательностей в тех ситуациях, когда это распознавание не сводится к независимому друг от друга распознаванию ее элементов. На основе этих методов Ты сможешь построить программу, которая могла бы годиться для решения некоторого ограниченного круга практических задач. Ограничение состоит в том, что программу распознавания строк можно было бы применять для распознавания только таких текстовых документов, в которых автоматическое разделение документа на строки не представляет собой проблему.

Распознавание документов, в которых строки тесно прилегают друг, не сводится к распознаванию строк этого документа, и Тебе придется решать задачу распознавания сразу всей страницы. Если Ты в продвижении своей прикладной задачи дойдешь до этого этапа, Ты будешь вполне готов к

тому, чтобы изучить последнюю, десятую лекцию нашего курса, где мы исследуем методы распознавания данных с такой двухмерной структурой.

Уже жду эти лекции с нетерпением. Но, возможно, еще одно небольшое замечание, которое не знаю, как выразить, чтобы не показаться невежливым. Было бы жаль, если бы все интегральное исчисление было пригодно лишь для того, чтобы вычислять площади под графиком полиномиальных функций. Пожалуй, это была бы стрельба из пушки по воробьям.

Эти слова мы уже где-то слышали и достаточно часто. Не помним случая, чтобы они принесли пользу.

Область действия структурного распознавания значительно шире, чем распознавание текстовых изображений. Но не требуй от нас, чтобы мы с Тобой рассмотрели еще одно приложение. Это уже должна быть Твоя, а не наша работа.

Февраль 1998.

7.5 Библиографические замечания

В лекции формулируются основные понятия структурного распознавания образов и представлена одна из возможных точек зрения на соотношение статистического и структурного распознавания. Судя по известным нам публикациям, изложенные в лекции взгляды разделяют многие исследователи структурного распознавания. В то же время нам известно, что многие придерживаются и других точек зрения, отличающихся от изложенных в лекции и тоже правильных. Нам неизвестны публикации, в которых в явном виде было бы сформулировано, что объединяет статистическое и структурное распознавание и что определяет специфику каждого из этих двух обширных разделов в современном распознавании. Исключением является публикация [Schlesinger and Gimmel'farb, 1987], в которой сформулирована часть положений, изложенных в лекции.

Представление множества изображений в виде множества точек линейного пространства резко критиковали Минский и Пейперт в известной монографии [Minsky and Papert, 1969].

Глава 8

Распознавание марковских последовательностей

8.1 Вводные замечания о последовательностях

Последовательность - это, пожалуй, простейшая структура, с которой уместно начать рассказ о структурном распознавании. Даже на таком простейшем примере, как распознавание последовательностей, можно достаточно выразительно показать, как следует распознавать сложный объект, состоящий из большого количества взаимозависимых частей, и как знание этой зависимости позволяет улучшить распознавание объекта в целом и каждой его части. Мы увидим, что хотя алгоритмы распознавания таких объектов не всегда тривиальны, их реализация не приводит к каким-либо непреодолимым вычислительным трудностям. Мы увидим также, что статистические задачи распознавания, обучения и самообучения, сформулированные ранее в общем виде, в исследуемом частном случае решаются точно и без каких-либо дополнительных упрощающих предположений.

Конечно же, последовательности интересны не только тем, что они прекрасный объект для формального исследования, но и тем, что они хорошо приспособлены для обширного слоя прикладных задач. Мы приведем два примера таких приложений. При этом мы ни в коем случае не утверждаем, что эти прикладные задачи полностью укладываются в рамки формальных методов, которые излагаются в данной лекции. Эти два примера приводятся лишь как иллюстрация абстрактных понятий, в терминах которых эти методы будут изложены.

Пример 8.1 Распознавание текстовой строки. Пусть x_1, x_2, \dots, x_n - строка текста, то есть последовательность изображений букв. Каждому изображению в этой последовательности соответствует наименование буквы из алфавита K . Цель распознавания состоит в том, чтобы на основании наблюдаемой последовательности x_1, x_2, \dots, x_n изображений сказать нечто убедительное о последовательности k_1, k_2, \dots, k_n букв, представленных на этих изображениях.

Если бы буквы k_i , $i = 1, \dots, n$, были взаимно независимы, решение о последовательности k_1, k_2, \dots, k_n свелось бы к n независимым решениям о каждой букве в отдельности. Решение о букве на i -ом месте принималось бы на основании одного лишь изображения x_i в последовательности x_1, x_2, \dots, x_n . Такое решение далеко не лучшее в реальной ситуации, когда буквы в строке зависят друг от друга. Эта зависимость (контекст) настолько сильная, что часто, на основании знания букв $k_1, k_2, \dots, k_{i-1}, k_{i+1}, \dots, k_n$ можно уверенно судить о букве k_i , даже когда изображение x_i очень сильно искажено или вообще неизвестно. Проблема заключается в том, что с самого начала неизвестна ни одна буква в последовательности k_1, k_2, \dots, k_n и решение о букве k_i надо принять, когда другие буквы тоже не известны. Следовательно, на основании всей последовательности x_1, x_2, \dots, x_n наблюдаемых изображений надо принять решение сразу о всей последовательности k_1, k_2, \dots, k_n букв, учитывая при этом априорные знания о их взаимной зависимости. ▲

Пример 8.2 Медицинская диагностика. Пусть x_1, x_2, \dots, x_n - результат измерения определенного параметра пациента в последовательные моменты времени, а k_1, k_2, \dots, k_n - последовательность его состояний в эти моменты. Эти состояния недоступны для непосредственного наблюдения. Цель распознавания состоит в принятии как можно более точного решения о состоянии пациента в тот или иной момент времени. Естественно, состояния в различные моменты времени сильно зависят друг от друга, и знание этой зависимости позволяет улучшить качество распознавания. ▲

К настоящему времени достаточно глубоко исследованы алгоритмы распознавания последовательностей при определенной формулировке задачи распознавания. Характеристической чертой этих алгоритмов является применение динамического программирования для нахождения оптимальной последовательности k_1, k_2, \dots, k_n скрытых состояний. Первоисточником этих методов является статья Ковалевского [Kovalevski, 1967], разработавшего их для распознавания текстовых строк. В работе Винцюка [Vincjuk,] этот подход впервые применен для распознавания речевых сигналов. С тех пор эти методы быстро распространились по всему миру, и сейчас эти методы пользуются таким уважением, которое становится для них уже вредным. Мало-помалу были забыты формулировки задач, которые уместно решать именно с помощью динамического программирования, и оно стало применяться без разбору и в тех приложениях, для которых уместны другие постановки, а следовательно, и другие решения.

В этой лекции мы определим класс статистических моделей распознаваемого объекта и для этого класса сформулируем различные байесовские задачи распознавания. Мы увидим, что одни задачи естественно решаются хорошо известным методом динамического программирования. Для других задач уместны менее известные, но тоже очень хорошие методы.

8.2 Марковская модель распознаваемого объекта

Пусть объект характеризуется $2n + 1$ параметрами, которые представлены двумя последовательностями $\bar{x} = (x_1, x_2, \dots, x_n)$ и $\bar{k} = (k_0, k_1, \dots, k_n)$. В данной лекции последовательности обозначаются символами с черточками сверху, в отличие от элементов последовательности, которые будут обозначены символами без черточек.

Параметры k_0, k_1, \dots, k_n являются *скрытыми*, а параметры x_1, x_2, \dots, x_n - *наблюдаемыми*. Последовательности \bar{x} и \bar{k} случайны и принимают значения из множеств X^n и K^{n+1} , где X - множество значений наблюдаемых параметров x_i , K - множество значений скрытых параметров k_i . Связная подпоследовательность $(x_{i_1}, x_{i_1+1}, \dots, x_{i_2})$ будет обозначаться $x_{i_1}^{i_2}$, а подпоследовательность $(k_{i_1}, k_{i_1+1}, \dots, k_{i_2})$ будет обозначаться $k_{i_1}^{i_2}$. Символ x_1^n , таким образом, обозначает то же, что и \bar{x} , символ x_i^i - то же, что и x_i . Аналогично, обозначения k_0^n и \bar{k} тождественны, равно как и обозначения k_i^i и k_i .

Мы будем говорить о совместных и условных вероятностях указанных параметров и отдельных их групп. Все вероятности будут обозначаться одной и той же буквой p . К примеру, $p(x_i, k_i, k_{i-1})$ обозначает совместную вероятность того, что i -ый наблюдаемый параметр примет значение x_i , $(i - 1)$ -ый скрытый параметр примет значение k_{i-1} , а i -ый скрытый параметр примет значение k_i . Тот же самый символ p будет использован для обозначения условной вероятности $p(x_i, k_i | k_{i-1})$ того, что i -ый наблюдаемый параметр примет значение x_i , а i -ый скрытый параметр примет значение k_i при условии, что $(i - 1)$ -ый скрытый параметр принял значение k_{i-1} . Таким образом, символ p используется для обозначения двух различных функций в выражениях $p(x_i, k_i, k_{i-1})$ и $p(x_i, k_i | k_{i-1})$ и это не совсем корректно. Тем не менее ради упрощения вида формул мы будем допускать эту некорректность. Эта неаккуратность не приведет к недоразумениям, так как идентификатор p будет записываться, как правило, с последующими скобками, внутри которых будут записаны идентификаторы переменных. По этим идентификаторам будет легко понять, о каких именно вероятностях говорится в той или иной формуле. В случае, если эта оговоренная неаккуратность все же может привести к недоразумениям, мы будем разнообразить обозначения всевозможными индексами, что приведет к некоторой громоздкости формул.

Статистическая модель объекта определяется функцией вида $X^n \times K^{n+1} \rightarrow \mathbb{R}$, которая для каждой последовательности \bar{x} и каждой последовательности \bar{k} указывает их совместную вероятность $p(\bar{x}, \bar{k})$. Будем считать, что эти вероятности удовлетворяют следующему условию: для каждого $i = 1, 2, \dots, n - 1$, для каждой последовательности $\bar{k} = (k_0^{i-1}, k_i, k_{i+1}^n)$ и для каждой последовательности $\bar{x} = (x_1^i, x_{i+1}^n)$ выполняется равенство

$$p(\bar{x}, \bar{k}) = p(k_i) p(x_1^i, k_0^{i-1} | k_i) p(x_{i+1}^n, k_{i+1}^n | k_i). \quad (8.1)$$

Из этого предположения следует, что равенство

$$p(\bar{k}) = p(k_i) p(k_0^{i-1} | k_i) p(k_{i+1}^n | k_i) \quad (8.2)$$

выполняется для каждого $i = 1, 2, \dots, n - 1$ и для каждой последовательности $\bar{k} \in K^{n+1}$, где $\bar{k} = (k_0^{i-1}, k_i, k_{i+1}^n)$. Равенство (8.2) получается в результате суммирования равенства (8.1) по всем последовательностям \bar{x} .

Случайная последовательность \bar{k} с вероятностями $p(\bar{k})$, которые удовлетворяют условию (8.2), называется марковской последовательностью или марковской цепью. В этой лекции мы будем иметь дело исключительно только с марковскими случайными последовательностями.

Если равенство (8.1) просуммировать по всем последовательностям k_{i+2}^n , а затем по всем последовательностям x_{i+2}^n , то получим

$$\sum_{x_{i+2}^n} \sum_{k_{i+2}^n} p(\bar{x}, \bar{k}) = p(x_1^i, k_0^i) \sum_{x_{i+2}^n} \sum_{k_{i+2}^n} p(x_{i+1}^n, k_{i+1}^n | k_i)$$

или, что то же самое,

$$p(x_1^{i+1}, k_0^{i+1}) = p(x_1^i, k_0^i) p(x_{i+1}, k_{i+1} | k_i).$$

Из этого рекурсивного правила следует рабочая формула

$$p(\bar{x}, \bar{k}) = p(x_1, x_2, \dots, x_n, k_0, k_1, \dots, k_n) = p(k_0) \prod_{i=1}^n p(x_i, k_i | k_{i-1}), \quad (8.3)$$

с помощью которой для каждой пары последовательностей \bar{k} и \bar{x} можно вычислить их совместную вероятность. Мы видим, что в силу допущения (8.1) о марковском характере случайной последовательности определение сложной функции от $2n + 1$ переменных k_i , $i = 0, \dots, n$, и x_i , $i = 1, \dots, n$, сводится к определению n функций $p(x_i, k_i | k_{i-1})$ трех переменных и одной функции $p(k_0)$ одной переменной. Вычисление же этой функции большого количества переменных выполняется с помощью явной формулы (8.3). Таким образом, условие (8.1) выделяет очень узкий, но важный класс статистических моделей, который мы собираемся исследовать. Свойства случайных последовательностей этого класса полезно понимать и неформально, и поэтому рассмотрим с иной точки зрения, пусть и не очень строго, что обозначает марковость случайной последовательности.

Пусть из генеральной совокупности пар последовательностей $(\bar{x}, \bar{k}) = (x_1, \dots, x_n, k_0, k_1, \dots, k_n)$, выбирается некоторая подсовокупность следующим образом. Выбирается произвольный номер i , $0 < i < n$, и произвольное значение σ скрытого параметра k_i , и для выбранных фиксированных значений i и σ из генеральной совокупности выбираются те и только те пары (\bar{x}, \bar{k}) , для которых $k_i = \sigma$. Марковость случайной последовательности обозначает, что в совокупности выбранных пар (\bar{x}, \bar{k}) группа параметров (x_1, x_2, \dots, x_i) , $(k_0, k_1, \dots, k_{i-1})$ окажется статистически независимой от группы параметров $(x_{i+1}, x_{i+2}, \dots, x_n)$, $(k_{i+1}, k_{i+2}, \dots, k_n)$.

Это правильное понимание марковости иногда выражается в вульгаризованной форме, что в марковских случайных последовательностях для любого i параметр k_i зависит только от k_{i-1} и не зависит от k_{i-2} , k_{i-3} и всех остальных k_{i-l} , $l > 2$. Это вульгаризованное определение марковости

неверно, и опасность его в том, что оно очень похоже на верное. Проиллюстрируем это на следующем неформальном примере.

Правильное неформальное понимание марковости дает следующая механическая модель. Пусть последовательность $(k_0, k_1, k_2, k_3, k_4)$ представляет положение некоторых точек на плоскости, равно как и последовательность (x_1, x_2, x_3, x_4) , рис. 8.1. Предположим, что некоторые точки соединены пружинами, которые на рисунке представлены прямолинейными отрезками. Предположим, что некоторая точка, скажем, точка x_3 , начинает вибрировать под воздействием определенных случайных воздействий. В этом случае, в результате механических связей между точками, начнут вибрировать все (!) остальные точки, а не только точки k_2 и k_3 , непосредственно связанные пружинами с точкой x_3 . В приведенной системе каждая точка зависит от каждой другой точки, и система в целом не распадается на какие-то взаимно независимые составные части. Далее, если зафиксировать положение точек x_1, x_2, x_3, x_4 , то точки k_0, k_1, k_2, k_3, k_4 установятся в некотором положении, и положение точки k_i при этом определится не только положением точек x_i и x_{i+1} , с которыми точка k_i непосредственно связана, а положением всех точек x_1, x_2, x_3, x_4 .

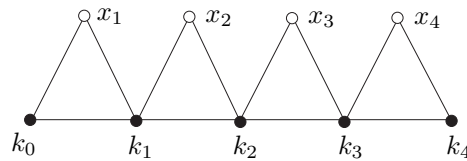


Figure 8.1 A mechanical model of a Markovian sequence.

Представим себе теперь, что положение какой-то точки, скажем, точки k_3 , зафиксировано. В этом случае вся механическая система распадается на две независимые части. Первая состоит из точек $k_0, k_1, k_2, x_1, x_2, x_3$, а вторая - из точек x_4 и k_4 . Теперь случайная

флуктуация, скажем, точки x_4 никак не влияет на положение точек k_1, x_2 и всех других точек слева от k_3 .

Дальнейшие рассуждения будут основаны на точном представлении марковских последовательностей в виде конечных автоматов. *Конечный автомат* определяется как шестерка $(K, V, X, \delta, k_0, F)$, где

K - конечное множество состояний автомата;

V - конечный алфавит входных символов;

X - конечный алфавит выходных символов;

k_0 - начальное состояние;

$F \subset K$ - подмножество состояний, называемых конечными состояниями;

$\delta: K \times V \rightarrow K \times X$ - функция, называемая функцией переходов.

Приведенная шестерка имеет следующий содержательный смысл. Если автомат находится в состоянии $k \in K$ и на его вход подается символ $v \in V$, то автомат переходит в состояние $k' \in K$ и генерирует выходной символ $x \in X$, где пара (k', x) определяется функцией переходов $\delta: K \times V \rightarrow K \times X$ так, что $(k', x) = \delta(k, v)$. В начальный момент автомат находится в состоянии k_0 .

Каждая последовательность \bar{v} входных символов v_1, v_2, \dots, v_n однозначно определяет последовательность $\bar{k} = (k_1, k_2, \dots, k_n)$ состояний и последовательность $\bar{x} = (x_1, x_2, \dots, x_n)$ символов. Это последовательность состояний, через которые проходит автомат под воздействием входной последовательности \bar{v} , и последовательность символов, которые автомат в это время подает на выход. При этом предполагается, что до подачи последовательности \bar{v} на вход автомата он находился в состоянии k_0 . Понятно, что состояние, в котором оказывается автомат после подачи на его вход последовательности \bar{v} , также однозначно определяется этой последовательностью. Таким образом с помощью автомата конструктивно представляются три отображения множества входных последовательностей: (1) в множество выходных последовательностей, (2) в множество последовательностей состояний и (3) в множество состояний. Эти три отображения реализуются, не прибегая к множеству F конечных состояний. Используя множество F , можно определить еще отображение множества входных последовательностей в множество $\{0, 1\}$, то есть выделить определенное подмножество входных последовательностей. Это множество тех последовательностей, которые переводят автомат из начального состояния k_0 в состояние из множества F .

Обобщением конечного автомата является *стохастический конечный автомат*. В этой более общей конструкции функция $\delta: K \times V \rightarrow K \times X$ заменяется более сложной функцией $\delta_s: X \times K \times K \times V \rightarrow \mathbb{R}$, а указание начального состояния k_0 заменяется функцией $p: K \rightarrow \mathbb{R}$. Эти функции имеют следующий смысл. Начальное состояние автомата случайное и любое состояние $k \in K$ может стать начальным с вероятностью $p(k)$. Дальнейшее поведение автомата также случайное. Если автомат находится в состоянии $k \in K$ и на его вход подается символ $v \in V$, автомат генерирует случайную пару $(x, k') \in X \times K$ с вероятностью $\delta_s(x, k' | k, v)$, а затем переходит в состояние k' и подает символ x на выход. Таким образом с помощью стохастического автомата для каждой последовательности входных символов конструктивно определяется распределение вероятностей на множестве выходных последовательностей, множестве состояний и множестве последовательностей состояний.

В случае, когда алфавит входных символов состоит из единственного символа, получается конструкция, называемая *автономным стохастическим автоматом*. Автономный стохастический автомат является точной моделью процессов, удовлетворяющих (8.1).

Пусть автомат характеризуется множеством K состояний и множеством X выходных символов. Поведение автомата определяется вероятностями $p(k_0)$, $p(x_i, k_i | k_{i-1})$, $k_0 \in K$, $k_i \in K$, $x_i \in X$, $i = 1, 2, \dots, n$. Автомат генерирует случайную последовательность символов x_1, x_2, \dots, x_n длины n следующим образом. Сначала автомат устанавливается в состояние k_0 , выбранное случайно в соответствии с распределением вероятностей $p(k_0)$. На i -ом шаге, $i = 1, 2, \dots, n - 1$, автомат генерирует случайную пару значений (x_i, k_i) в соответствии с распределением вероятностей $p(x_i, k_i | k_{i-1})$, пере-

ходит в состояние k_i и подает символ x_i на выход. При таком поведении автомата совместная вероятность последовательности k_0, k_1, \dots, k_n состояний и выходной последовательности x_1, x_2, \dots, x_n в точности такая, как это определяет формула (8.3).

Приведенная модель будет служить основой для изложения материала этой лекции. Это, конечно же, не значит, что распознаваемый объект должен быть непременно автоматом, а последовательности k_0, k_1, \dots, k_n и x_1, x_2, \dots, x_n должны быть непременно последовательностями во времени. Дальнейший анализ основан исключительно на предположении (8.1) о марковском характере параметров распознаваемого объекта, и результаты анализа пригодны для любых объектов, которые этому предположению удовлетворяют.

8.3 Распознавание стохастического автомата

8.3.1 Распознавание стохастического автомата; формулировка задачи

Пусть a и b - два автономных стохастических автомата. Они характеризуются одним и тем же множеством K состояний и одним и тем же алфавитом X выходных символов, но статистические характеристики этих автоматов различны. Понятно, что допущение об одном и том же множестве состояний и выходных символов не сузит общность наших рассуждений.

Первый автомат характеризуется вероятностями $p_a(k_0)$ и $p_a(x_i, k_i | k_{i-1})$, $k_0 \in K$, $k_i \in K$, $x_i \in X$, $i = 1, 2, \dots, n$, а второй - вероятностями $p_b(k_0)$ и $p_b(x_i, k_i | k_{i-1})$. Пусть x_1, x_2, \dots, x_n - последовательность, сгенерированная одним из этих двух автоматов, неизвестно, каким именно. Цель распознавания состоит в том, чтобы на основании знания статистических свойств автоматов принять разумное решение, какой из данных двух автоматов эту последовательность сгенерировал.

Эту цель можно конкретизировать, как отыскание байесовской стратегии, минимизирующей риск распознавания, или отыскание стратегии Неймана-Пирсона, или минимаксной стратегии, или как-то еще иначе. Однако, из первых двух лекций данного курса известно, что стратегии решения всех этих задач имеют следующий общий вид. Требуется вычислить вероятность $p_a(\bar{x})$ последовательности $\bar{x} = (x_1, x_2, \dots, x_n)$ при условии, что ее сгенерировал автомат a , и аналогичную вероятность $p_b(\bar{x})$ для автомата b . Решение в пользу первого или второго автомата принимается на основании отношения правдоподобия $p_a(\bar{x})/p_b(\bar{x})$. Вычисление вероятностей $p_a(\bar{x})$ и $p_b(\bar{x})$ в этой стратегии составляет ее наиболее трудоемкую часть. Посмотрим, как выглядит алгоритм этих вычислений.

8.3.2 Алгоритм распознавания стохастического автомата

Алгоритм для вычисления вероятности $p_a(\bar{x})$ такой же, как и для вычисления вероятности $p_b(\bar{x})$, и поэтому опишем его, опуская индекс a или b при символе p .

Вероятность $p(\bar{x})$ равна по определению сумме $\sum_{\bar{k}} p(\bar{k}, \bar{x})$ и, применяя (8.3), может быть выражена в виде многомерной суммы

$$p(\bar{x}) = \sum_{\bar{k}} p(\bar{k}, \bar{x}) = \sum_{k_0} \sum_{k_1} \cdots \sum_{k_{n-1}} \sum_{k_n} p(k_0) \prod_{i=1}^n p(x_i, k_i | k_{i-1}). \quad (8.4)$$

Выражение (8.4) непригодно для непосредственного вычисления вероятности $p(\bar{x})$, так как непосредственное вычисление потребовало бы суммирования $|K|^{n+1}$ слагаемых. Но его можно эквивалентно преобразовать к виду, при котором это вычисление станет реализуемым. Вынесем за знак суммирования по переменной k_i те множители, которые не зависят от k_i , и получим

$$\begin{aligned} p(\bar{x}) &= \sum_{k_0} p(k_0) \sum_{k_1} p(x_1, k_1 | k_0) \cdots \sum_{k_i} p(x_i, k_i | k_{i-1}) \\ &\cdots \sum_{k_{n-1}} p(x_{n-1}, k_{n-1} | k_{n-2}) \sum_{k_n} p(x_n, k_n | k_{n-1}). \end{aligned} \quad (8.5)$$

Для $i = 1, 2, \dots, n$ введем обозначение

$$\begin{aligned} f_i(k_{i-1}) &= \sum_{k_i} p(x_i, k_i | k_{i-1}) \sum_{k_{i+1}} p(x_{i+1}, k_{i+1} | k_i) \cdots \\ &\cdots \sum_{k_{n-1}} p(x_{n-1}, k_{n-1} | k_{n-2}) \sum_{k_n} p(x_n, k_n | k_{n-1}) \end{aligned}$$

и получим вычислительную процедуру

$$\left. \begin{aligned} f_n(k_{n-1}) &= \sum_{k_n} p(x_n, k_n | k_{n-1}); \\ f_i(k_{i-1}) &= \sum_{k_i} p(x_i, k_i | k_{i-1}) f_{i+1}(k_i), \quad i = n-1, \dots, 2, 1; \\ p(\bar{x}) &= \sum_{k_0} p(k_0) f_1(k_0). \end{aligned} \right\} \quad (8.6)$$

Можно увидеть, что количество операций для вычисления вероятности $p(\bar{x})$ имеет порядок $|K|^2 n$. С самого начала следует вычислить числа $f_n(k_{n-1})$ в соответствии с первой строкой в (8.6), затем последовательно числа $f_{n-1}(k_{n-2}), \dots, f_i(k_{i-1}), \dots, f_1(k_0)$ в соответствии со второй строкой в (8.6), и наконец, число $p(\bar{x})$ в соответствии с третьей строкой в (8.6). По этой процедуре вычисляются вероятности $p_a(\bar{x})$ и $p_b(\bar{x})$ для автоматов a и b , а затем на основании отношения правдоподобия $p_a(\bar{x})/p_b(\bar{x})$ принимается

решение в пользу одного из автоматов. Как правило, это решение принимается на основании сравнения отношения правдоподобия с определенным пороговым значением, хотя могут быть и другие, более замысловатые решения.

8.3.3 Матричное представление вычислительной процедуры

Хотя процедура (8.6) представляет необходимые вычисления вполне однозначно, мы выразим ее в более краткой форме, которая окажется полезной при дальнейшем анализе.

Процедура (8.6) представлена не в виде явной формулы для вычисления $p(\bar{x})$, так как она включает в себя вычисление вспомогательных величин $f_i(k_{i-1})$. Если же их исключить из системы (8.6), получится опять исходная формула (8.5). Мы все же выразим процедуру (8.6) в таком виде, в котором будут исключены промежуточные переменные и который будет иным, чем (8.5).

Совокупность вероятностей $p(x_i, k_i | k_{i-1})$, $k_i \in K$, $k_{i-1} \in K$, можно рассматривать, как функцию $K \times K \rightarrow \mathbb{R}$ двух переменных k_i и k_{i-1} . Величина x_i не является переменной, так как это результат измерения i -ого наблюдаемого параметра и в каждом акте распознавания он фиксирован. Эта функция двух переменных в свою очередь может пониматься как матрица размера $|K| \times |K|$, в которой в (k_i) -ом столбце и (k_{i-1}) -ой строке записано число $p(x_i, k_i | k_{i-1})$. Обозначим эту матрицу P_i . Она зависит от индекса i хотя бы потому, что значение x_i зависит от i . Представление совокупности вероятностей $(p(x_i, k_i | k_{i-1}), k_i \in K, k_{i-1} \in K)$ в виде матрицы оправдано потому, что далее эта совокупность чисел будет выступать в виде сомножителя определенных матричных произведений. Это приведет к большей наглядности вычислительной процедуры (8.6) и в конечном итоге к более эффективным вычислениям.

Числа $f_i(k_{i-1})$, $k_{i-1} \in K$, $i = 1, 2, \dots, n$, которые последовательно вычисляются в соответствии с процедурой (8.6), можно понимать, как последовательность $|K|$ -мерных векторов-столбцов f_i , $i = 1, \dots, n$, k -ой координатой которых является число $f_i(k)$. Пусть f - это $|K|$ -мерный вектор-столбец, все координаты которого равны 1, а φ - это $|K|$ -мерный вектор-строка, (k_0) -ая координата которого равна $p(k_0)$.

При таких введенных обозначениях процедуру (8.6) можно выразить в линейно алгебраической форме

$$\begin{aligned} f_n &= P_n f, \\ f_i &= P_i f_{i+1}, \quad i = n-1, n-2, \dots, 2, 1, \\ p(\bar{x}) &= \varphi f_1, \end{aligned}$$

или, после исключения вспомогательных векторов f_1, f_2, \dots, f_n , в виде

$$p(\bar{x}) = \varphi P_1 P_2 \cdots P_{n-1} P_n f. \quad (8.7)$$

Формулу (8.7) можно записать еще короче

$$p(\bar{x}) = \varphi \left(\prod_{i=1}^n P_i \right) f, \quad (8.8)$$

но это уже будет не совсем точно, так как матричное произведение не коммутативно. В выражении (8.7) порядок матриц-сомножителей записан в явном виде, а в кратком выражении (8.8) этот порядок уже не указан.

Хотя в матричном представлении (8.7) почти не просматривается статистический смысл выполняемых процедур, оно обладает большей наглядностью с вычислительной точки зрения. Например, ассоциативность матричного произведения в формуле (8.7) открывает разнообразие процедур вычисления вероятности $p(\bar{x})$, которое не очень очевидно в представлении (8.6) и еще более скрыто в формулах (8.5) и (8.4).

Пусть, к примеру, $n = 5$. Формула (8.7) эквивалентна двум следующим формулам, по которым очевидно, что вероятность $p(\bar{x})$ можно вычислять по крайней мере двумя различными способами,

$$p(\bar{x}) = \varphi \left(P_1 (P_2 (P_3 (P_4 (P_5 f)))) \right), \quad (8.9)$$

$$p(\bar{x}) = \left(((((\varphi P_1) P_2) P_3) P_4) P_5 \right) f. \quad (8.10)$$

Вычисления по формуле (8.9) в точности соответствует процедуре (8.6), а вычисления по формуле (8.10) - тоже правильные, но уже несколько отличные от процедуры (8.6).

Хотя формулы (8.9) и (8.10) достаточно наглядны в вычислительном отношении, их статистическое содержание оказалось тщательно скрытым. Раскроем его в следующем подразделе.

8.3.4 Статистическая интерпретация матричных произведений

Покажем, как следует вычислять условные вероятности $p(x_i^n | k_{i-1})$ для любого i , то есть вероятности, что автомат сгенерирует последовательность x_i, x_{i+1}, \dots, x_n при условии, что к началу этого генерирования автомат находился в состоянии k_{i-1} . Вычисление $p(x_n | k_{n-1})$ тривиально, так как

$$p(x_n | k_{n-1}) = \sum_{k_n \in K} p(x_n, k_n | k_{n-1}), \quad (8.11)$$

а числа $p(x_n, k_n | k_{n-1})$ - это известные вероятности, характеризующие автомат. Для вероятностей $p(x_{n-1}, x_n | k_{n-2})$ в самом общем случае справедливо равенство

$$\begin{aligned} p(x_{n-1}, x_n | k_{n-2}) &= \sum_{k_{n-1} \in K} p(x_{n-1}, x_n, k_{n-1} | k_{n-2}) \\ &= \sum_{k_{n-1} \in K} p(x_{n-1}, k_{n-1} | k_{n-2}) p(x_n | x_{n-1}, k_{n-1}, k_{n-2}). \end{aligned} \quad (8.12)$$

Оно справедливо для любых случайных величин, а не только для элементов марковской последовательности. Но если выполняется свойство (8.1) марковости (и если опираться на неформальное понимание марковости по механической модели на рис. 8.1), то

$$p(x_n | x_{n-1}, k_{n-1}, k_{n-2}) = p(x_n | k_{n-1}),$$

и выражение (8.12) приобретает вид

$$p(x_{n-1}, x_n | k_{n-2}) = \sum_{k_{n-1} \in K} p(x_{n-1}, k_{n-1} | k_{n-2}) p(x_n | k_{n-1}). \quad (8.13)$$

Таким образом, вероятности $p(x_i^n | k_{i-1})$, которые мы хотим вычислить для любого i , можно вычислить по крайней мере для $i = n$ и $i = n - 1$ по формулам (8.11) и (8.13). Сейчас мы покажем, как следует вычислять эти вероятности $p(x_{i-1}^n | k_{i-2})$, полагая, что вероятности $p(x_i^n | k_{i-1})$ уже вычислены.

Для вероятностей $p(x_{i-1}^n | k_{i-2})$ в самом общем случае справедливо

$$\begin{aligned} p(x_{i-1}^n | k_{i-2}) &= p(x_{i-1}, x_i^n | k_{i-2}) \\ &= \sum_{k_{i-1} \in K} p(x_{i-1}, x_i^n, k_{i-1} | k_{i-2}) \\ &= \sum_{k_{i-1} \in K} p(x_{i-1}, k_{i-1} | k_{i-2}) p(x_i^n | x_{i-1}, k_{i-1}, k_{i-2}). \end{aligned} \quad (8.14)$$

В силу свойства (8.1) (и неформального понимания марковости по механической модели), последовательность x_i^n при фиксированном состоянии k_{i-1} не зависит от предыдущего состояния k_{i-2} и от предыдущего наблюдения x_{i-1} . Следовательно, множитель $p(x_i^n | x_{i-1}, k_{i-1}, k_{i-2})$ в формуле (8.14) можно заменить на $p(x_i^n | k_{i-1})$, а выражение (8.14) на выражение

$$p(x_{i-1}^n | k_{i-2}) = \sum_{k_{i-1} \in K} p(x_{i-1}, k_{i-1} | k_{i-2}) p(x_i^n | k_{i-1}), \quad i = 2, 3, \dots, n. \quad (8.15)$$

Используя формулу (8.11) и многократно используя формулу (8.15), можно вычислить вероятности $p(x_1^n | k_0)$ для каждого состояния k_0 , а затем и вероятность $p(\bar{x})$,

$$p(\bar{x}) = \sum_{k_0 \in K} p(k_0) p(x_1^n | k_0). \quad (8.16)$$

Вычисления по формулам (8.11), (8.15) и (8.16) есть не что иное, как реализация процедуры (8.6). Таким образом, процедура (8.6), равно, как и ее матричное представление (8.9) не только формально обоснованы, но и интерпретированы со статистической точки зрения. Статистический смысл вектора

$$\left(\prod_{j=i}^n P_j \right) f$$

состоит в том, что его компонентами являются $|K|$ вероятностей $p(x_i, x_{i+1}, \dots, x_n | k_{i-1})$, $k_{i-1} \in K$, что автомат генерирует последовательность x_i, x_{i+1}, \dots, x_n при условии, что это генерирование началось в состоянии k_{i-1} .

Рассмотрим теперь статистические соображения, которые приводят к вычислению вероятности $p(\bar{x})$ по формуле (8.10). Эти соображения почти не отличаются от только что выполненных. Мы рассмотрим, как следует вычислить совместные вероятности $p(x_1^i, k_i)$, что автомат генерирует последовательность x_1, x_2, \dots, x_i и при этом окажется в состоянии k_i . Очевидно, для $i = 1$ эта вероятность равна

$$p(x_1, k_1) = \sum_{k_0 \in K} p(k_0) p(x_1, k_1 | k_0). \quad (8.17)$$

Предположим, что мы уже вычислили вероятности $p(x_1^{i-1}, k_{i-1})$ и хотели бы вычислить вероятности $p(x_1^i, k_i)$. В общем случае для вероятностей $p(x_1^i, k_i)$ справедливо, что

$$\begin{aligned} p(x_1^i, k_i) &= p(x_1^{i-1}, x_i, k_i) \\ &= \sum_{k_{i-1} \in K} p(x_1^{i-1}, x_i, k_{i-1}, k_i) \\ &= \sum_{k_{i-1} \in K} p(x_1^{i-1}, k_{i-1}) p(x_i, k_i | x_1^{i-1}, k_{i-1}). \end{aligned} \quad (8.18)$$

Но в данном частном случае, когда выполняется свойство (8.1) марковости (неформально выраженное в виде механической модели), мы утверждаем, что при фиксированном состоянии k_{i-1} пара (x_i, k_i) не зависит от наблюдения x_1^{i-1} и, следовательно,

$$p(x_i, k_i | x_1^{i-1}, k_{i-1}) = p(x_i, k_i | k_{i-1}).$$

Если полученное выражение включить в сумму (8.18), получается следующее рекурсивное выражение для вычисления $p(x_1^i, k_i)$,

$$p(x_1^i, k_i) = \sum_{k_{i-1} \in K} p(x_1^{i-1}, k_{i-1}) p(x_i, k_i | k_{i-1}). \quad (8.19)$$

После того, как по формулам (8.17) и (8.19) вычислены вероятности $p(x_1^n, k_n)$, вероятность $p(\bar{x})$ можно вычислить по формуле

$$p(\bar{x}) = \sum_{k_n \in K} p(x_1^n, k_n), \quad (8.20)$$

так как x_1^n есть просто $\bar{x} = (x_1, x_2, \dots, x_n)$.

Вычисление вероятности $p(\bar{x})$ по формулам (8.17), (8.19) и (8.20) есть не что иное, как вычисление матричного произведения (8.10). Формула (8.10)

ранее была обоснована лишь на основании формального свойства ассоциативности матричных произведений. Сейчас она интерпретирована и со статистической точки зрения. Статистический смысл матричного произведения

$$\varphi \left(\prod_{j=1}^i P_j \right)$$

в том, что это $|K|$ -мерный вектор, в котором k_i -ая компонента равна совместной вероятности $p(x_1, x_2, \dots, x_i, k_i)$, что автомат сгенерирует последовательность x_1, x_2, \dots, x_i и окажется в состоянии k_i .

8.3.5 Распознавание марковских объектов при неполных данных

В предыдущих рассмотрениях мы отождествляли распознаваемый объект с автономным стохастическим автоматом, и это могло создать впечатление, что последовательности x_1, x_2, \dots, x_n и $k_0, k_1, k_2, \dots, k_n$ есть непременно последовательности, развивающиеся во времени, а индекс i должен пониматься как время. Такое понимание сильно сужало бы область применимости методов, излагаемых в данной лекции. Для применимости рассматриваемых методов требуется, чтобы выполнялось условие (8.1) и вовсе не требуется, чтобы индекс i определял момент времени, в котором происходит измерение признака x_i . Признаки могут измеряться последовательно один за другим, но совсем в другой очередности, чем это определяется номером i . В этом случае возникают чисто технологические вопросы о том, в какой очередности обрабатывать элементы последовательности x_1, x_2, \dots, x_n и $k_0, k_1, k_2, \dots, k_n$. Ведь совсем не обязательно их обрабатывать справа налево по формуле (8.9) или слева направо по формуле (8.10). Поясним эту ситуацию на следующем примере.

Пусть объект описывается двадцатью признаками x_1, x_2, \dots, x_{20} и двадцать одним скрытым параметром k_0, k_1, \dots, k_{20} . Пусть в какой-то момент времени стали известны значения признаков $(x_5, x_6, \dots, x_{10})$ и $(x_{12}, x_{13}, \dots, x_{17})$, а остальные признаки будут измерены позже, причем через значительное время. Однако, когда значения остальных признаков станут известны, объект должен быть распознан как можно скорее. В этой ситуации возникает вопрос, как можно не тратить время попусту в ожидании остальных данных и что-то вычислить сейчас на основании уже имеющихся данных так, чтобы потом, после получения остальных данных, завершить распознавание как можно быстрее. На основании матричного представления (8.7) ответ на этот вопрос почти очевиден. Выражение (8.7) эквивалентно выражению

$$p(\bar{x}) = \varphi P_1 P_2 P_3 P_4 P^* P_{11} P^{**} P_{18} P_{19} P_{20} f, \quad (8.21)$$

где

$$P^* = P_5 P_6 P_7 P_8 P_9 P_{10}, \quad (8.22)$$

$$P^{**} = P_{12} P_{13} P_{14} P_{15} P_{16} P_{17}. \quad (8.23)$$

Из этого ясно, что при известных подпоследовательностях x_5^{10} и x_{12}^{17} можно вычислить матрицы P^* и P^{**} по формулам (8.22) и (8.23). После получения всей информации об объекте вероятность $p(\bar{x})$ вычисляется по формуле (8.21). Общее количество вычислений в этом случае увеличится по сравнению с вычислениями по формулам (8.9) или (8.10), но уменьшится количество вычислений, которые придется выполнить после получения исчерпывающих данных о распознаваемом объекте.

Представим себе теперь, что в описываемой ситуации ожидаемая дополнительная информация не поступила, а пришел запрос завершить распознавание на основании той информации, которая имеется. Укажем кратко, как в этом случае следует поступать.

Пусть I - это множество $\{1, 2, \dots, n\}$, на котором принимает значение индекс i в обозначении x_i . Пусть I' - это подмножество индексов и для каждого $i \in I'$ известно значение x_i . Совокупность известных значений обозначим как $(x_i, i \in I')$, а совокупность неизвестных значений - как $(x_i, i \notin I')$. Пусть \bar{k} - это последовательность k_0, k_1, \dots, k_n . Совместная вероятность совокупности $(x_i, i \in I)$ и последовательности \bar{k} равна

$$p((x_i, i \in I), \bar{k}) = p(k_0) \prod_{i \in I} p(x_i, k_i | k_{i-1}),$$

а искомая вероятность $p((x_i, i \in I'))$ равна

$$\begin{aligned} p((x_i, i \in I')) &= \sum_{\bar{k}} \sum_{(x_i, i \notin I')} p(k_0) \prod_{i \in I} p(x_i, k_i | k_{i-1}) \\ &= \sum_{\bar{k}} p(k_0) \prod_{i \in I'} p(x_i, k_i | k_{i-1}) \sum_{(x_i, i \notin I')} \prod_{i \notin I'} p(x_i, k_i | k_{i-1}) \\ &= \sum_{\bar{k}} p(k_0) \prod_{i \in I'} p(x_i, k_i | k_{i-1}) \prod_{i \notin I'} \sum_{x_i} p(x_i, k_i | k_{i-1}) \\ &= \sum_{\bar{k}} p(k_0) \prod_{i \in I'} p(x_i, k_i | k_{i-1}) \prod_{i \notin I'} p(k_i | k_{i-1}) \\ &= \sum_{\bar{k}} p(k_0) \prod_{i \in I} P_i(k_i | k_{i-1}), \end{aligned} \quad (8.24)$$

где

$$P_i(k_i | k_{i-1}) = \begin{cases} p(x_i, k_i | k_{i-1}), & \text{если значение } x_i \text{ известно,} \\ p(k_i | k_{i-1}) = \sum_{x_i} p(x_i, k_i | k_{i-1}), & \text{если значения } x_i \text{ неизвестно.} \end{cases}$$

Если отождествить функцию $P_i: K \times K \rightarrow \mathbb{R}$ с квадратной матрицей размера $|K| \times |K|$, то выражение (8.24) для вероятности $p((x_i, i \in I'))$ опять-таки можно представить в виде матричного произведения

$$p((x_i, i \in I)) = \varphi \left(\prod_{i=1}^n P_i \right) f.$$

Здесь вид матрицы P_i зависит от того, известно ли значение признака x_i или нет.

Рассмотренные задачи являются лишь немногочисленными примерами огромного разнообразия статистических задач распознавания в рамках марковских моделей объектов. И в свете рассмотренных задач удивляет широко распространенное в распознавании образов представление, что именно динамическое программирование является основным средством для распознавания марковских последовательностей. При тех вполне естественных формулировках задач, которые мы до сих пор рассмотрели, необходимость динамического программирования просто не возникает. Более того, вообще не возникают какие-либо оптимизационные проблемы, которые требовали бы привлечения современных методов оптимизации. На этом этапе данной лекции нам хотелось бы несколько распатать наивное и укоренившееся представление, что динамическое программирование является универсальным ключом в распознавании марковских последовательностей, которым открывается любая дверь.

Рассмотрим теперь именно ту дверь, наиболее подходящим ключом к которой является динамическое программирование.

8.4 Наиболее вероятная последовательность скрытых параметров

8.4.1 Различие между распознаванием объекта в целом и распознаванием его составных частей

Результаты анализа задачи распознавания марковских объектов в предыдущем подразделе 8.3.5 довольно поучительны. Задача распознавания была сформулирована как распознавание объекта в целом, а не как распознавание его составных частей. Однако на интуитивном уровне ожидалось, что формальное решение задачи приведет к алгоритму, который будет включать в себя нечто, что могло бы пониматься как распознавание составных частей объекта. Затем, на основании результатов распознавания частей принималось бы решение об объекте в целом. Однако в выведенных алгоритмах ничего подобного не обнаруживается. Мы старательно рассмотрели задачу с разных сторон, формально и менее формально, рассмотрели различные ее модификации и не обнаружили никакой иерархии в алгоритмах, при которой на нижнем уровне принимались бы решения о состояниях, через которые прошел автомат, а затем на верхнем уровне принималось бы решение о том, какой именно автомат породил распознаваемую последовательность. Формальное решение точно сформулированной задачи оказалось совсем иным, чем интуитивно ожидалось. Раз уж была сформулирована задача определить автомат, породивший заданную последовательность, то формально выведенный алгоритм занимается именно этим, а не вопросом, через какие состояния прошел автомат при порождении этой после-

довательности. Если же цель состоит в определении последовательности состояний автомата, сгенерировавшего определенную последовательность символов, то следует именно эту цель выразить в недвусмысленной формулировке задачи, которую затем следует решить. Решение одной задачи совсем не обозначает, что одновременно с ней будет решена другая.

8.4.2 Формулировка задачи поиска наиболее вероятной последовательности состояний

Пусть $\bar{x} = x_1^n$ - последовательность наблюдений x_1, x_2, \dots, x_n , а $\bar{k} = k_0^n$ - последовательность состояний автомата. Их совместная вероятность $p(\bar{x}, \bar{k})$ равна

$$p(\bar{x}, \bar{k}) = p(k_0) \prod_{i=1}^n p(x_i, k_i | k_{i-1}),$$

где $p(k_0), k_0 \in K$, и $p(x_i, k_i | k_{i-1})$, $x_i \in X$, $k_i \in K$, $i = 1, 2, \dots, n$, известные вероятности.

Задача формулируется как поиск последовательности \bar{k}^* с наибольшей апостериорной вероятностью при условии последовательности \bar{x} , то есть, поиск последовательности

$$\bar{k}^* = \operatorname{argmax}_{\bar{k} \in K^{n+1}} \frac{p(\bar{x}, \bar{k})}{\sum_{\bar{k} \in K^{n+1}} p(\bar{x}, \bar{k})} = \operatorname{argmax}_{\bar{k} \in K^{n+1}} p(\bar{x}, \bar{k}). \quad (8.25)$$

8.4.3 Сведение задачи к поиску кратчайшего пути на графе

Мы покажем, как можно свести оптимизационную задачу (8.25) к известной задаче поиска кратчайшего пути между двумя вершинами в графе специального вида. Задачи такого вида, как известно, решаются методами динамического программирования.

Обозначим q_i , $i = 1, 2, \dots, n$, функцию вида $K \times K \rightarrow \mathbb{R}$, значения $q_i(k_{i-1}, k_i)$, $k_{i-1} \in K$, $k_i \in K$, которой равны $-\log p(x_i, k_i | k_{i-1})$. Обозначим φ_0 функцию вида $K \rightarrow \mathbb{R}$, значения $\varphi_0(k_0)$, $k_0 \in K$, которой равны $-\log p(k_0)$. С использованием этих обозначений оптимизационную задачу (8.25) можно записать в виде

$$\bar{k}^* = \operatorname{argmin}_{k_0, k_1, \dots, k_n} \left(\varphi_0(k_0) + \sum_{i=1}^n q_i(k_{i-1}, k_i) \right) \quad (8.26)$$

и представить с помощью графа специального вида.

Граф состоит из множества V вершин и ориентированных стрелок, соединяющих некоторые вершины. Множество вершин состоит из начальной вершины α , конечной вершины β и еще $|K|(n+1)$ промежуточных вершин. Множество промежуточных вершин соответствует множеству пар вида (σ, i) , $\sigma \in K$, $i = 0, 1, \dots, n$.

Пример 8.3 Множество вершин графа. На рис. 8.2 показано множество V для случая, когда $n = 3$, а множество K состояний автомата

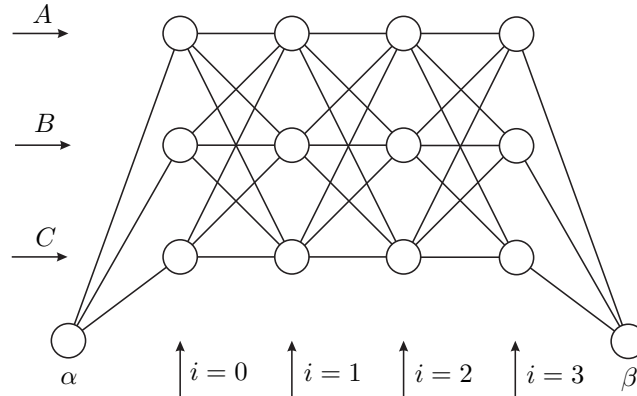


Figure 8.2 The optimisation task concerning the transition through the states is represented as seeking a path in an oriented graph. Edges of the graph are oriented from left to right.

состоит из трех состояний: A , B и C . Вершину (σ, i) можно отождествлять с точкой на плоскости с горизонтальной координатой i и вертикальной координатой, соответствующей значению σ . Это значение может быть A , B или C . Параметр σ вершины (σ, i) будет называться именем вершины. ▲

Укажем, как введенные вершины соединяются стрелками (направленными ребрами) и какие длины приписываются каждой стрелке.

- Из вершины α выходят $|K|$ стрелок к вершинам вида $(\sigma, 0)$, $\sigma \in K$.
- В вершину β ведут $|K|$ стрелок, которые выходят из вершин вида (σ, n) , $\sigma \in K$.
- Из каждой вершины вида (σ, i) , $\sigma \in K$, $i = 0, 1, \dots, n-1$ выходят $|K|$ стрелок, которые ведут в вершины вида $(\sigma', i+1)$, $\sigma' \in K$.
- Стрелкам вида $(\alpha, (\sigma, 0))$, $\sigma \in K$, приписывается длина $\varphi(\sigma)$.
- Стрелкам вида $((\sigma, n), \beta)$, $\sigma \in K$, приписывается длина 0.
- Стрелкам вида $((\sigma, i-1), (\sigma', i))$, $\sigma \in K$, $\sigma' \in K$, $i = 1, 2, \dots, n$, приписывается длина $q_i(\sigma, \sigma')$.

Построенный таким способом граф определяет множество путей из вершины α к вершине β , и для каждого такого пути определена его длина как сумма длин стрелок, из которых этот путь состоит. Каждому из этих путей соответствует последовательность состояний - имен вершин, через которые проходит путь. И наоборот, каждой последовательности состояний k_0, k_1, \dots, k_n соответствует путь на графе, проходящий через вершины

$$\alpha, (k_0, 0), (k_1, 1), (k_2, 2), \dots, (k_n, n), \beta.$$

При этом число $\varphi(k_0) + \sum_{i=1}^n q_i(k_{i-1}, k_i)$ оказывается как раз длиной пути, которому соответствует последовательность k_0, k_1, \dots, k_n . Таким образом оптимизационная задача (8.25) сведена к поиску кратчайшего пути, соединяющего две вершины на специально построенном графе. Алгоритм

решения таких задач хорошо известен и является одной из возможных реализаций динамического программирования Беллмана. В следующем подразделе ради полноты изложения приводится алгоритм нахождения кратчайшего пути в контексте наших задач.

8.4.4 Поиск кратчайшего пути на графе

Мы рассмотрим эту задачу на неформальном уровне, при котором, можно надеяться, наглядность достигается без потери убедительности. Представим себе, что некоторые гонцы должны доставить сообщение из вершины α в вершину β . В самом начале гонцы находятся в вершинах графа, причем в каждой вершине находится столько гонцов, сколько стрелок выходит из этой вершины. Таким образом, каждой стрелке графа соответствует свой гонец, и каждый гонец прикреплен к своей собственной стрелке. В начальный момент времени каждый гонец находится в вершине, из которой выходит его стрелка, и ожидает, когда ему станет известно сообщение. Гонцам, находящимся в вершине α , сообщение становится известным сразу же в начальный момент. Как только сообщение становится известным определенному гонцу, он начинает бег по своей стрелке, пока он не достигнет вершины в конце стрелки. Когда какой-то гонец достигает вершину в конце стрелки, сообщение становится известным всем гонцам в этой вершине. Все гонцы бегут с одной и той же скоростью, равной единице. Следовательно, время бега по стрелке равно длине этой стрелки. Время, когда первый гонец достигнет вершину β , очевидно, есть длина кратчайшего пути из α в β . Определим, чему равно это время.

Обозначим $f_i(\sigma)$ момент времени, когда сообщение становится известным для гонцов в вершине (σ, i) , или, что то же самое, длину кратчайшего пути из вершины α в вершину (σ, i) . Числа $f_0(\sigma)$, $\sigma \in K$, очевидно, равны $\varphi(\sigma)$, так как каждая вершина вида $(\sigma, 0)$ соединена с вершиной α единственной стрелкой,

$$f_0(\sigma) = \varphi(\sigma). \quad (8.27)$$

Значения $f_i(\sigma)$ для всех прочих вершин (σ, i) , $\sigma \in K$, $i > 0$, можно вычислить на основании следующих, хотя и неформальных, но достаточно точных рассуждений. Сообщение становится известным в вершине (σ, i) только тогда, когда оно доставлено одним из гонцов из вершин вида $(\sigma', i-1)$, $\sigma' \in K$. Информация из некоторой конкретной вершины $(\sigma', i-1)$ будет доставлена в вершину (σ, i) в момент времени $f_{i-1}(\sigma') + q_i(\sigma', \sigma)$, где $f_{i-1}(\sigma')$ - момент времени, когда сообщение стало известно в вершине $(\sigma', i-1)$, а $q_i(\sigma', \sigma)$ - время, необходимое для доставки сообщения из вершины $(\sigma', i-1)$ в вершину (σ, i) . Один из гонцов придет в вершину (σ, i) раньше всех, и это произойдет в момент времени $f_i(\sigma)$, который равен

$$f_i(\sigma) = \min_{\sigma' \in K} (f_{i-1}(\sigma') + q_i(\sigma', \sigma)). \quad (8.28)$$

В момент прихода этого гонца в вершину (σ, i) определяется, из какой именно вершины $(\sigma', i-1)$ поступило сообщение на (σ, i) раньше всех. Имя этой

вершины запоминается в виде указателя $\text{ind}_i(\sigma)$,

$$\text{ind}_i(\sigma) = \underset{\sigma' \in K}{\text{argmin}} (f_{i-1}(\sigma') + q_i(\sigma', \sigma)) . \quad (8.29)$$

Переменная $\text{ind}_i(\sigma)$ указывает, что кратчайший путь из вершины α до вершины (σ, i) проходит через вершину $(\text{ind}_i(\sigma), i-1)$. Отметим, что если таких путей несколько, то фиксируется один из них, безразлично какой.

Момент времени, когда сообщение становится известным в вершине β , то есть длина кратчайшего пути от α до β , определяется, очевидно, как

$$\min_{\sigma \in K} f_n(\sigma) , \quad (8.30)$$

потому что длины всех стрелок, ведущих в вершину β , равны нулю. Вершина из группы (σ, n) , $(\sigma \in K)$, через которую проходит этот путь, есть вершина

$$k_n = \underset{\sigma \in K}{\text{argmin}} f_n(\sigma) .$$

Формулы (8.27), (8.28), (8.29) и (8.30) составляют алгоритм поиска кратчайшего пути на графе. Соберем их всех вместе,

$$f_0(\sigma) = \varphi(\sigma) , \quad \sigma \in K ; \quad (8.31)$$

$$f_i(\sigma) = \min_{\sigma' \in K} (f_{i-1}(\sigma') + q_i(\sigma', \sigma)) , \quad i = 1, 2, \dots, n , \quad \sigma \in K ; \quad (8.32)$$

$$\text{ind}_i(\sigma) = \underset{\sigma' \in K}{\text{argmin}} (f_{i-1}(\sigma') + q_i(\sigma', \sigma)) , \quad i = 1, 2, \dots, n , \quad \sigma \in K ; \quad (8.33)$$

$$k_n = \underset{\sigma \in K}{\text{argmin}} f_n(\sigma) ; \quad (8.34)$$

и подведем итоги в виде конкретных указаний по вычислениям.

С самого начала определяются расстояния от вершины α до вершин вида $(\sigma, 0)$ в соответствии с (8.31). Затем последовательно по формуле (8.32) вычисляются кратчайшие расстояния от вершины α до вершин из совокупности $(\sigma, 1)$, $\sigma \in K$, до вершин из совокупности $(\sigma, 2)$, $\sigma \in K$, и так далее, и, в конечном итоге, до вершин из совокупности (σ, n) , $\sigma \in K$. Одновременно с вычислениями кратчайших расстояний $f_i(\sigma)$ для каждой вершины определяется указатель $\text{ind}_i(\sigma)$ по формуле (8.33). Он определяет имя σ' вершины $(\sigma', i-1)$, которая непосредственно предшествует вершине (σ, i) на кратчайшем пути из α в (σ, i) .

После вычислений по формулам (8.31), (8.32) и (8.33) длина кратчайшего пути от α до β определяется числом $\min_{\sigma} f_n(\sigma)$. Формула (8.34) указывает имя k_n последней вершины на этом пути, величина $\text{ind}_n(k_n)$ указывает имя k_{n-1} предпоследней вершины, величина $\text{ind}_{n-1}(k_{n-1})$ указывает имя k_{n-2} . В общем виде, величина $\text{ind}_i(k_i)$ указывает $(i-1)$ -ый элемент в искомой последовательности k_0, k_1, \dots, k_n , которая минимизирует (8.26) и максимизирует (8.25).

Мы советуем читателю проверить свое понимание алгоритма на примере, приведенном ниже, и убедиться в исключительной простоте алгоритма.

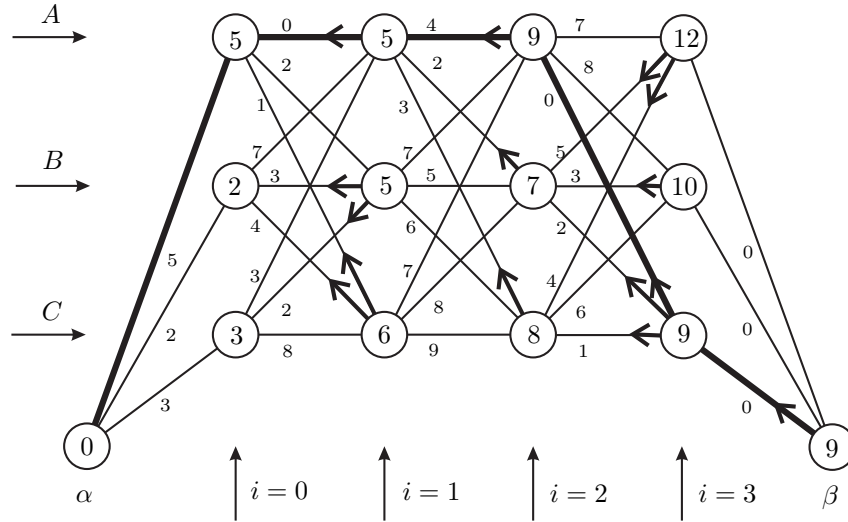


Figure 8.3 Seeking the shortest path in the Markovian sequence.

Пример 8.4 Поиск кратчайшего пути на графе. Рис. 8.3 отражает ту же самую ситуацию, которую мы рассматривали в примере 8.3. Количество состояний автомата равно трем, и это состояния A , B , C . Индекс i принимает значения $0, 1, 2, 3$. Числа на ребрах графа представляют их длины, то есть величины $q_i(\sigma', \sigma)$. Вершины графа обозначены кружочками, в которых проставлены числа $f_i(\sigma)$. Указатели $\text{ind}_i(\sigma)$ представлены в виде стрелок от вершин (σ, i) к вершинам $(\text{ind}_i(\sigma), i-1)$. Отметим, что в некоторых вершинах указатель не определяется однозначно, так как минимум в выражении (8.32) достигается одновременно при нескольких значениях имени σ' . В этом случае указателю $\text{ind}_i(\sigma)$ присваивается одно из этих значений. Но на рис. 8.3 стрелками указаны все эти значения. Движение по стрелкам от вершины β к вершине α указывает кратчайший путь на графе. Если из какой-то вершины выходят несколько стрелок, можно двигаться по любой из них. Например, переход по стрелкам от вершины β к вершине α возможен через последовательность вершин $(C, 3)$, $(A, 2)$, $(A, 1)$, $(A, 0)$. Эта траектория показана на рисунке утолщенной линией. Она обозначает, что последовательность $AAAC$ минимизирует (8.26) и максимизирует (8.25). Однако по конфигурации стрелок на рис. 8.3 видно, что наряду с последовательностью $AAAC$ оптимальными являются также последовательности $AABC$ и $AACC$. ▲

8.4.5 О формальном анализе задач

Алгоритм поиска кратчайшего пути на графе (и следовательно, поиска наиболее правдоподобной последовательности скрытых состояний) выводится обычно на основании достаточно ясных, но недостаточно формальных соображений, подобно тому, как это сделали мы в предыдущих под-

разделах 8.4.3 и 8.4.4. Именно благодаря своей содержательной понятности, согласующейся с интуитивным пониманием задачи, эти алгоритмы повсеместно известны и пользуются популярностью. Ни в коем случае не отрицая полезность иллюстративных, пусть и не очень строгих рассуждений при выводе алгоритмов, мы собираемся показать, что при анализе и решении многих задач нельзя ограничиваться только такого рода рассуждениями.

Тот путь, который мы сейчас хотим подвергнуть сомнению и которого мы придерживались в предыдущем подразделе 8.4.4, слишком часто апеллирует к очевидности, опирается на интуитивное понимание, а не на формальный вывод. Алгоритмы выводятся при помощи предложений естественного языка, а не в рамках какого-то формализма, при котором математические выражения преобразуются по тем или иным правилам, которые гарантируют эквивалентность. Такой путь вывода конечных результатов существенно отличается от того, что происходит, скажем, при решении алгебраических уравнений. Исходное уравнение задается в определенном, однозначно понимаемом виде, затем с ним выполняется ряд преобразований, гарантирующих эквивалентность и приводящих в конечном итоге к уравнению, решение которого очевидно, скажем, к уравнению $x = 2$.

Формулировка задач в рамках того или иного формализма, при котором гарантируется их однозначность, введение правил преобразования выражений, гарантирующих их эквивалентность, - все это само по себе не делает задачу тривиальной. Ведь известно, как трудно бывает найти ту конкретную последовательность преобразований уравнения, которые переводят его из исходного вида - постановки задачи - в конечный вид, при котором его решение становится очевидным. И все же формальное решение задачи путем применения конечного количества заранее оговоренных правил эквивалентного преобразования имеет свои преимущества. Такое решение исключает обман и самообман. Если уж задача решена в таком стиле, то его правильность может быть доказана кому-то другому, от кого уже не требуется ни интуиция, ни понимание содержательного смысла. Требуется только понимание, что выполненные преобразования задачи являются эквивалентными.

Задачи, рассмотренные нами до сих пор, были настолько прозрачны, что их решение оказалось возможным обосновать, пользуясь лишь предложениями естественного языка. Однако это стало возможным лишь потому, что мы имели дело пока что с простейшими задачами структурного анализа. Скоро мы перейдем к действительно сложным задачам, рассмотрение которых на уровне словесных рассуждений неизбежно станет слишком громоздким и неуклюжим, а словесные обоснования все менее и менее убедительными. Стремление не затуманивать сущность проблемы математическим формализмом приведет к тому, что она будет скрыта за многословием. Таким образом, при контакте с действительно сложными проблемами их формулировка, анализ и решение должны удовлетворять как можно более строгим формальным требованиям. Только тогда формулировка задачи будет однозначна, а ее решение - убедительным.

Конечно, с действительной сложностью задач можно обходиться и так, что она просто игнорируется. Тогда, имея дело с ранее неизвестной задачей, решение которой в силу ее новизны не совсем понятно, произвольно применяют к ней тот или иной известный алгоритм, нисколько не заботясь о том, решает ли он именно эту задачу или нет. Обоснование алгоритма в этом случае подменяют его известностью. Фактически же в лучшем случае вместо одной задачи решают совсем другую. В худшем случае известный алгоритм безжалостно деформируют, и в обиход запускается алгоритм, о котором невозможно сказать что-либо внятное, какую же задачу он решает. Покажем распространенный пример такого псевдорешения нечетко сформулированной задачи.

Пример 8.5 Неуместное отыскание кратчайшего пути на графе.

Пусть $\bar{x} = (x_1, x_2, \dots, x_n)$ - последовательность наблюдаемых символов, скажем, при $n = 100$. Эта последовательность зависит от последовательности $\bar{k} = (k_0, \dots, k_n)$ состояний, через которые прошел автомат при генерировании этой последовательности. В силу этой зависимости последовательность \bar{k} можно разумно оценить на основании наблюдения \bar{x} . Предположим теперь, что нас не интересует вся последовательность \bar{k} , а только ее последние десять элементов $k_{91}, k_{92}, \dots, k_{100}$. Эта задача не сводится к поиску кратчайшего пути на графе, как это имело бы место в случае, если бы требовалось указать наиболее вероятную последовательность $\bar{k} = (k_0, \dots, k_n)$. Задачу нужно формально ставить с самого начала, а затем формально ее решать.

Однако если знания конструктора ограничены лишь умением искать кратчайший путь на графе при непоколебимой вере, что это единственный ключ к решению любых задач, то рождается следующее псевдорешение. Ищется наиболее вероятная последовательность k_0, k_1, \dots, k_{100} с помощью динамического программирования, а затем в качестве ответа берутся десять последних элементов этой последовательности. Такая рекомендация необоснована, потому что не сопровождается утверждением о каких-либо хороших свойствах последних десяти элементов в последовательности, найденной таким способом. Если бы задача была сформулирована, как поиск наиболее вероятной последовательности $k_{91}, k_{92}, \dots, k_{100}$, рекомендуемый способ не являлся бы ее решением. Наиболее вероятная последовательность $k_{91}, k_{92}, \dots, k_{100}$ совсем необязательно равна последним десяти элементам в наиболее вероятной последовательности $k_0^*, k_1^*, \dots, k_{100}^*$. ▲

Мы определим основные математические понятия, уместные для формулировки задач структурного анализа. Для начала мы опять рассмотрим задачу поиска наиболее вероятной последовательности, которую можно, как мы видели, решать и без вводимых далее средств. Но на этом, уже знакомом нам примере легче понять смысл вводимых понятий, которые далее будут активно использоваться.

8.4.6 Обобщенные матрицы и их умножение

Мы займемся оптимизационной задачей

$$d = \min_{k_0} \min_{k_1} \min_{k_2} \cdots \min_{k_n} \left(\varphi(k_0) + \sum_{i=1}^n q_i(k_{i-1}, k_i) \right). \quad (8.35)$$

Чтобы избежать ненужных усложнений, которые затуманили бы основную идею, мы будем говорить только о нахождении значения минимума и не будем говорить о поиске последовательности $(k_0^n)^*$, которая обеспечивает этот минимум.

Форма (8.35), в которой выражена решаемая оптимизационная задача, подобна форме выражения

$$p(\bar{x}) = \sum_{k_0} \sum_{k_1} \sum_{k_2} \cdots \sum_{k_n} p(k_0) \prod_{i=1}^n p(x_i, k_i | k_{i-1}), \quad (8.36)$$

которое мы рассматривали при решении задачи распознавания автомата. Как в выражении (8.35), так и в выражении (8.36) речь идет о вычислении некоторого числа. Это число d в (8.35) и число $p(\bar{x})$ в (8.36). Это число вычисляется на основании функции вида $K \rightarrow \mathbb{R}$ (в (8.35) это $\varphi(k_0)$, а в (8.36) это $p(k_0)$, $k_0 \in K$) и n функций вида $K \times K \rightarrow \mathbb{R}$ (в (8.35) это функции q_i , $i = 1, 2, \dots, n$, а в (8.36) это $p(x_i, k_i | k_{i-1})$, $k_i \in K$, $k_{i-1} \in K$, $i = 1, \dots, n$). Значения выражений (8.35) и (8.36) вычисляются различными, но подобными программами. Различие состоит, во первых, в том, что везде, где в одной программе определяется сумма двух чисел, в другой программе вычисляется величина меньшего из двух чисел. Во вторых, везде, где в одной программе умножаются два числа, в другой программе они складываются.

При описании процедуры для вычисления числа $p(\bar{x})$, определяемого выражением (8.36), мы пришли к целесообразности интерпретации чисел $p(k_0)$ и $p(x_i, k_i | k_{i-1})$ как компонент вектора φ и матриц P_i , $i = 1, \dots, n$. При такой интерпретации вычисление по формуле (8.36) эквивалентно вычислению матричного произведения

$$p(\bar{x}) = \varphi P_1 P_2 \cdots P_n f. \quad (8.37)$$

Это матричное произведение определяет число, которое следует вычислить, и в этом смысле есть лишь эквивалентным представлением задачи (8.36) в несколько иной форме. Однако в этом виде число, которое следует вычислить, определено в явном виде формулой (8.37). В этом смысле матричное выражение (8.37) непосредственно представляет алгоритм необходимых вычислений. Таким образом, преобразование формулировки задачи из вида (8.36) к виду (8.37) есть фактически решение задачи, так как приведение задачи к виду (8.37) делает ее тривиальной.

Правомерность преобразования выражения (8.36) для исходной задачи в матричное произведение (8.37) вытекает из свойств сложения и умножения вещественных чисел. Эти свойства настолько само собою понимаются, что

о них, как правило, и не упоминают. Это ассоциативность обеих операций и дистрибутивность умножения относительно сложения. Для любых трех вещественных чисел x , y и z выполняется

$$\left. \begin{aligned} x + (y + z) &= (x + y) + z, \\ x (y z) &= (x y) z, \\ x (y + z) &= x y + x z. \end{aligned} \right\} \quad (8.38)$$

Иными словами говоря, сложение и умножение, понимаемые в обычном смысле этого слова, образуют на множестве вещественных чисел алгебраическую структуру, известную как *полукольцо*. Конечно, умножение и сложение вещественных чисел обладает необозримым множеством других свойств, но сейчас для нас важно именно то, что они образуют полукольцо. Кроме того, существенно, что сложение и умножение является далеко не единственной парой операций на множестве вещественных чисел, которая обладает свойствами (8.38). В частности, операции \min и $+$ также образуют полукольцо, и это ключевой момент наших рассуждений. Действительно,

$$\begin{aligned} \min(x, \min(y, z)) &= \min(\min(x, y), z), \\ x + (y + z) &= (x + y) + z, \\ x + \min(y, z) &= \min(x + y, x + z). \end{aligned}$$

На этом основании мы можем определить матрицы и операции над ними не в общепринятом смысле, а по иному, основываясь на полукольце из операций $(\min, +)$, а не на полукольце $(+, \times)$, как это принято в операциях над матрицами в обычном смысле этого слова. Введенные таким новым способом матрицы и операции над ними будут обладать всеми положительными свойствами обычных матриц, как то ассоциативность умножения и др.

Пусть $q_i: K \times K \rightarrow \mathbb{R}$, $i = 1, 2, \dots, n$, - это n функций двух переменных. Каждую функцию q_i можно понимать как матрицу размера $|K| \times |K|$, элемент которой в k -ой строке и k' -ом столбце равен $q_i(k, k')$. Пусть $\varphi: K \rightarrow \mathbb{R}$ и $f: K \rightarrow \mathbb{R}$ - две функции одной переменной. Функцию φ будем понимать как $|K|$ -мерный вектор-строку, а функцию f - как $|K|$ -мерный вектор-столбец. Пусть q' и q'' - две матрицы. Их произведением $q' \odot q''$ будет называться матрица q размерности $|K| \times |K|$, элемент $q(k, k')$ которой определяется как

$$q(k, k') = \min_{l \in K} (q'(k, l) + q''(l, k')). \quad (8.39)$$

Произведением $\varphi \odot q$ вектора-строки φ размерности $|K|$ на матрицу q размерности $|K| \times |K|$ будем считать вектор-строку φ' , k' -ая компонента которого определяется как

$$\varphi'(k') = \min_{k \in K} (\varphi(k) + q(k, k')). \quad (8.40)$$

И наконец, произведение $q \odot f$ матрицы q размера $|K| \times |K|$ на $|K|$ -мерный вектор-столбец f понимается как вектор-столбец f' , k -ая компонента которого равна

$$f'(k) = \min_{k' \in K} (q(k, k') + f(k')). \quad (8.41)$$

Если операцию $\min(x, y)$ обозначить $x \oplus y$, а операцию $x + y$ обозначить $x \odot y$, то определения (8.39), (8.40) и (8.41) примут вид

$$\left. \begin{aligned} q(k, k') &= \bigoplus_{l \in K} (q'(k, l) \odot q''(l, k')), \\ \varphi'(k') &= \bigoplus_{k \in K} (\varphi(k) \odot q(k, k')), \\ f'(k) &= \bigoplus_{k' \in K} (q(k, k') \odot f(k')) \end{aligned} \right\} \quad (8.42)$$

который представляет в общем виде и только что введенные операции над матрицами, и обычно применяемые. Если в качестве (\oplus, \odot) принимается пара $(+, \times)$, формулы (8.42) определяют умножение матриц в обычном смысле этого слова, то есть умножение в полукольце $(+, \times)$. Если же (\oplus, \odot) понимается как пара $(\min, +)$, эти же формулы определяют только что введенные операции, то есть матричное умножение в полукольце $(\min, +)$.

При использовании введенных обозначений \oplus и \odot выражение (8.35) для исходной оптимизационной задачи приобретает вид

$$d = \bigoplus_{k_0} \bigoplus_{k_1} \dots \bigoplus_{k_n} \left(\varphi(k_0) \odot \left(\bigodot_{i=1}^n q_i(k_{i-1}, k_i) \right) \right), \quad (8.43)$$

который является общим как для оптимизационной задачи (8.35), так и для вычисления (8.36), к которому сводится ранее рассмотренная задача распознавания автомата. Основываясь на тех же соображениях, что и при сведении многомерного суммирования (8.36) к матричному умножению (8.37), мы заключаем, что число (8.43), а следовательно, и число (8.35) есть матричное произведение

$$d = \varphi \odot q_1 \odot q_2 \odot \dots \odot q_n \odot f \quad (8.44)$$

в полукольце $(\min, +)$, где f есть вектор-столбец, все компоненты которого равны нулю.

Матричное произведение (8.44) - все та же исходная оптимизационная задача (8.35), но только лишь записанная в алгебраической форме. Однако ее преимущество в том, что при такой записи просто исчезает такой этап как конструирование алгоритма для решения задачи. Запись задачи в виде (8.44) является сразу же записью алгоритма для ее решения. В силу ассоциативности матричного умножения вычисление значения (8.44) можно выполнять по формуле

$$\left(\left(\left(\dots \left((\varphi \odot q_1) \odot q_2 \right) \odot \dots \odot q_{n-2} \right) \odot q_{n-1} \right) \odot q_n \right) \odot f,$$

которая предписывает буквально тот же порядок вычислений, что и алгоритм (8.31)–(8.34), выведенный ранее на основе неформальных, и как сейчас видно, более громоздких соображений. Выражение (8.44) открывает, однако, и другие способы вычислений, например, по формуле

$$d = \left(\varphi \odot \left(q_1 \odot \left(\dots \left(q_{n-1} \odot \left(q_n \odot f \right) \right) \dots \right) \right) \right)$$

или

$$\varphi \odot \left(\bigcirc_{i=1}^l q_i \right) \odot \left(\bigcirc_{i=l+1}^k q_i \right) \odot \left(\bigcirc_{i=k+1}^n q_i \right) \odot f.$$

Из них можно выбрать наиболее удобный, основываясь уже на тех или иных технологических соображениях, например, как это было рассмотрено в подразделе 8.3.5.

8.4.7 Распознавание подпоследовательности состояний

В примере 8.5 мы показали пример задачи, в которой неформальные рассуждения, кажущиеся вполне разумными, приводят к неправильному решению. Покажем теперь одну из возможных формализаций задачи и ее точное решение в принятой формулировке.

Пусть X и K - два конечных множества, $\bar{x} = (x_1, \dots, x_n) \in X^n$ и $\bar{k} = (k_0, \dots, k_n) \in K^{n+1}$ - две случайные последовательности, совместные вероятности которых определяются как

$$p(\bar{x}, \bar{k}) = p(k_0) \prod_{i=1}^n p(x_i, k_i | k_{i-1}), \quad (8.45)$$

где $p(k_0)$, $p(x_i, k_i | k_{i-1})$, $k_i \in K$, $x_i \in X$, $i = 1, 2, \dots, n$, - известные числа. Величины x_i , $i = 1, 2, \dots, n$, наблюдаемые, а k_i , $i = 0, 1, \dots, n$, скрытые параметры объекта.

Предположим, что хотя все признаки x_i наблюдаемы, в определенном эксперименте не все они были измерены. Обозначим Ix множество индексов тех признаков, значения которых были измерены. Таким образом, результатом наблюдения является совокупность значений $(x_i, i \in Ix)$. На основе экспериментальных данных можно оценить скрытые параметры k_0, k_1, \dots, k_n . Предположим однако, что экспериментатора не интересуют значения всех скрытых параметров, а только некоторые из них. Они определяются подмножеством индексов I_k . Задача состоит в том, чтобы на основании экспериментальных данных $(x_i, i \in Ix)$ определить совокупность $(k_i^*, i \in I_k)$ с наибольшей апостериорной вероятностью,

$$(k_i^*, i \in I_k) = \underset{(k_i, i \in I_k)}{\operatorname{argmax}} p((x_i, i \in Ix), (k_i, i \in I_k)).$$

Для вероятностей в общем случае справедливо равенство

$$p((x_i, i \in Ix), (k_i, i \in I_k)) = \sum_{(x_i, i \notin Ix)} \sum_{(k_i, i \notin I_k)} p(\bar{x}, \bar{k}).$$

С учетом марковских свойств (8.45) распознаваемого объекта имеем

$$p((x_i, i \in Ix), (k_i, i \in I_k)) = \sum_{(k_i, i \notin I_k)} \sum_{(x_i, i \notin Ix)} p(k_0) \prod_{i=1}^n p(x_i, k_i | k_{i-1}). \quad (8.46)$$

Указанная вероятность зависит только от значений $(k_i, i \in Ik)$, которые должны быть определены. Эта вероятность не зависит от значений $(x_i, i \notin Ix)$ и $(k_i, i \notin Ik)$, по которым производится суммирование. Она не зависит от значений $(x_i, i \in Ix)$, которые являются результатами измерения и, следовательно, фиксированы на данном акте распознавания. Число (8.46), зависящее от $(k_i, i \in Ik)$, обозначим $d((k_i, i \in Ik))$.

Выполним суммирование по переменным $(x_i, i \notin Ix)$ следующим образом,

$$\begin{aligned} d((k_i, i \in Ik)) &= \sum_{(k_i, i \notin Ik)} \sum_{(x_i, i \notin Ix)} p(k_0) \prod_{i=1}^n p(x_i, k_i | k_{i-1}) \\ &= \sum_{(k_i, i \notin Ik)} p(k_0) \prod_{i \in Ix} p(x_i, k_i | k_{i-1}) \prod_{i \notin Ix} \sum_{x_i \in X} p(x_i, k_i | k_{i-1}) \\ &= \sum_{k_i, i \notin Ik} \varphi(k_0) \prod_{i=1}^n q_i(k_{i-1}, k_i) \end{aligned}$$

где

$$q_i(k_{i-1}, k_i) = \begin{cases} p(x_i, k_i | k_{i-1}), & \text{если } i \in Ix, \\ \sum_{x_i \in X} p(x_i, k_i | k_{i-1}), & \text{если } i \notin Ix, \end{cases} \quad (8.47)$$

а число $\varphi(k_0)$ есть вероятность $p(k_0)$. Цель состоит в отыскании максимального значения $d((k_i, i \in Ik))$,

$$d = \max_{(k_i, i \in Ik)} \sum_{(k_i, i \notin Ik)} \varphi(k_0) \prod_{i=1}^n q_i(k_{i-1}, k_i), \quad (8.48)$$

и совокупности $(k_i^*, i^* \in Ik)$ значений, которые обеспечивают этот максимум,

$$(k_i^*, i \in Ik) = \operatorname{argmax}_{(k_i, i \in Ik)} \sum_{(k_i, i \notin Ik)} \varphi(k_0) \prod_{i=1}^n q_i(k_{i-1}, k_i). \quad (8.49)$$

Рассмотрим далее только задачу (8.48). При понимании ее решения становится ясным и решение задачи (8.49). Как и раньше, обозначим q_i матрицу размера $|K| \times |K|$, в которой в (k_{i-1}) -ой строке и (k_i) -ом столбце записано число $q_i(k_{i-1}, k_i)$, вычисляемое по формуле (8.47). Символом φ обозначим $|K|$ -мерный вектор-строку, k_0 -ая компонента которого равна $\varphi(k_0) = p(k_0)$, $k_0 \in K$. Обозначим \odot_i матричное произведение в полукольце $(+, \times)$ при $i \notin Ik$, и в полукольце (\max, \times) при $i \in Ik$. С учетом этих обозначений искомое число (8.48) есть матричное произведение

$$d = \varphi \odot_0 q_1 \odot_1 q_2 \odot_2 q_3 \odot_3 \cdots \odot_{n-1} q_n \odot_n f, \quad (8.50)$$

где f есть $|K|$ -мерный вектор-столбец, все компоненты которого равны 1.

Выражение (8.50) включает в себя и две задачи, рассмотренные до сих пор. Это распознавание марковского объекта в целом и распознавание значений скрытых параметров, одновременно с разнообразными модификациями этих двух задач - неполные данные и т.п.

При вычислении матричного произведения по формуле (8.50) необходимо учитывать, что в произведении (8.50) умножения матриц выполняются в разных полукольцах, в отличие от умножений в (8.7), где все они выполнялись в полукольце $(+, \times)$, и умножений в (8.44), где все они выполнялись в полукольце $(\min, +)$. В обоих этих случаях, в силу ассоциативности умножения, вычисления по формуле (8.7) или (8.44) могли выполняться в любом порядке, слева направо, справа налево, с середины к краям и т.п. Совсем иначе обстоит дело с формулой (8.50). Здесь разнообразие возможностей значительно меньше, так как умножение в полукольце $(+, \times)$ имеет больший приоритет по сравнению с умножением в полукольце (\max, \times) . Действительно, произведение $A \odot (B C)$ обозначает нечто отличающееся $(A \odot B) C$, хотя $A \odot (B \odot C) = (A \odot B) \odot C$ и $A (B C) = (A B) C$. Произведение $A \odot (B C)$ обозначает

$$\max_y \left(a(x, y) \left(\sum_z b(y, z) c(z, u) \right) \right),$$

в то время как $(A \odot B) C$ обозначает

$$\sum_z \left(\max_y (a(x, y) b(y, z)) \right) c(z, u),$$

а это разные функции.

В силу указанного ограничения на порядок умножений в произведении (8.50), сложность его вычисления может существенно отличаться от сложности вычислений по формулам (8.7) и (8.44), которая имеет порядок $\mathcal{O}(|K|^2 n)$. Если совокупность параметров, которые следует оценить, образует большое количество не связанных друг с другом подпоследовательностей, то есть, когда множество lk индексов сильно перемешано с множеством $\{0, 1, 2, \dots, n\} \setminus lk$, сложность вычислений произведения (8.50) может быть порядка $\mathcal{O}(|K|^3 n)$. Однако, если следует оценить связную подпоследовательность скрытых параметров, то есть когда следует оценить k_l^m , сложность вычисления по формуле (8.50) остается того порядка $\mathcal{O}(|K|^2 n)$, что и при вычислении по формуле (8.7) и (8.44). Произведение (8.50) можно в этом случае записать в виде

$$d = \varphi \left(\prod_{i=1}^{l-1} q_i \right) \odot \left(\bigodot_{i=l}^m q_i \right) \odot \left(\prod_{i=m+1}^n q_i \right) f,$$

который следует понимать как краткую запись следующих вычислений.

1. Построение вектора-строки φ' по формуле

$$\varphi' = \varphi q_1 q_2 \dots q_{l-1},$$

имеющее сложность порядка $\mathcal{O}(|K|^2 l)$.

2. Построение вектора-столбца f' по формуле

$$f' = q_{m+1} q_{m+2} \dots q_n f,$$

имеющее сложность порядка $\mathcal{O}(|K|^2 (n - m))$.

3. Вычисление искомого числа d по формуле

$$d = \varphi' \odot q_l \odot q_{l+1} \odot \dots \odot q_m \odot f'$$

имеющее сложность порядка $\mathcal{O}(|K|^2 (m - l))$.

Суммарная сложность вычислений имеет, следовательно, порядок $\mathcal{O}(|K|^2 n)$.

Мы видим, что если матрицы и операции над ними понимать в широком смысле, определенную группу задач, представляющихся существенно различными, можно единообразно и кратко записать в виде матричных произведений. Конечно, каждую из рассмотренных до сих пор задач можно было бы решать, не прибегая к предложенному формализму. Однако при таком решении задач не оптом, а в розницу они так бы и остались изолированными островками. Единообразное представление этих задач позволяет увидеть, что между ними нет никакой нейтральной полосы, а наоборот, есть богатое семейство задач, которые содержательно вполне осмысленны, и главное, конструктивно разрешимы.

8.5 Поиск последовательности наиболее вероятных состояний

Мы сформулировали задачу распознавания скрытой последовательности (или подпоследовательности) как поиск последовательности с наибольшей апостериорной вероятностью. Такая формулировка задачи вполне естественна и, конечно, заслуживает того внимания, которое мы ей уделили. Не меньшего внимания заслуживает и то, что задача в такой формулировке есть лишь частный случай более общей задачи минимизации байесовского риска при конкретной функции потерь. Рассмотрим теперь нашу задачу с этой более общей точки зрения и зададим себе вопрос, насколько естественной является именно та функция потерь, которая приводит к решению в пользу наиболее вероятной последовательности.

Решение в пользу наиболее вероятной последовательности следует принимать, когда функция потерь определена как $W(\bar{k}, \bar{k}') = 0$ при $\bar{k} = \bar{k}'$, и $W(\bar{k}, \bar{k}') = 1$ в противном случае. Это значит, что все ошибочные распознавания, то есть события $\bar{k}' \neq \bar{k}$, квалифицируются одинаково, независимо от того, отличаются ли последовательности \bar{k}' и \bar{k} только одним единственным элементом или всеми. Здесь игнорируется тот очевидный в ряде приложений факт, что ошибки могут быть меньшими или большими. Такое загроуненное требование к распознающей системе, при котором игнорируется различие между ошибками, не всегда естественно, а в некоторых приложениях просто неприемлемо. Таким образом, и решение в пользу

апостериори наиболее вероятного исхода совсем не является чем-то само собою разумеющимся. Все зависит от функции потерь, которая уместна в том или ином приложении. Например, для некоторых приложений естественно считать, что штраф зависит не просто от того, совпадают ли последовательности \bar{k} и \bar{k}' , а от количества несовпадающих элементов, то есть от количества индексов i , для которых $k_i \neq k'_i$. Это значит, что мы говорим о функции потерь

$$W(\bar{k}, \bar{k}') = \sum_{i=0}^n w(k_i, k'_i), \quad \text{где} \quad (8.51)$$

$$w(k_i, k'_i) = \begin{cases} 1, & \text{если } k_i \neq k'_i, \\ 0, & \text{если } k_i = k'_i, \end{cases}$$

противопоставляя ее функции

$$W(\bar{k}, \bar{k}') = \begin{cases} 1, & \text{если } \bar{k} \neq \bar{k}', \\ 0, & \text{если } \bar{k} = \bar{k}'. \end{cases} \quad (8.52)$$

Штрафные функции (8.51) и (8.52) различны. При функции потерь (8.52) минимизируется количество неправильно распознанных последовательностей (скажем, строк), а при функции потерь (8.51) минимизируется количество неправильно распознанных элементов последовательностей (скажем, букв). Только при очень поверхностном взгляде может показаться, что эти две оптимизации осуществляются одной и той же стратегией. Неправильность такого представления очевидна по следующему примеру.

Пример 8.6 Различие байесовских стратегий для двух функций потерь.

Пусть последовательности \bar{k} имеют длину $n = 2$, то есть состоят из двух букв (k_1, k_2) , которые следует оценить. Буква k_1 , равно как и буква k_2 может принимать значения из множества $\{A, B, C\}$. Существуют, следовательно, девять возможных последовательностей из двух букв. Предположим, что в результате тех или иных наблюдений было установлено, что апостериорные вероятности этих девяти пар такие, как указано в таблице.

$\bar{k}_2 \backslash \bar{k}_1$	A	B	C
A	0.30	0	0
B	0.20	0	0
C	0	0.25	0.25

Если требуется найти пару (k'_1, k'_2) , которая минимизирует вероятность события $(k_1, k_2) \neq (k'_1, k'_2)$, а это соответствует штрафной функции (8.52), то следует решить, что пара (k'_1, k'_2) есть (A, A) . При таком решении вероятность неравенства $(k_1, k_2) \neq (k'_1, k'_2)$ равна 0.7 и будет минимальной. Давайте посмотрим, каков будет риск этого решения при функции потерь (8.51), или, говоря иными словами, каково будет математическое ожидание количества неправильно распознанных букв. В

действительности может иметь место одна из четырех возможных пар: (A, A) , (A, B) , (B, C) и (C, C) , вероятности которых равны, соответственно, 0,3, 0,2, 0,25, 0,25. Количество неправильно распознанных букв будет равно, соответственно, 0, 1, 2, 2,. Следовательно, при решении $(k'_1, k'_2) = (A, A)$ в пользу наиболее вероятной пары математическое ожидание количества неправильно распознанных букв равно 1,2.

Посмотрим теперь, каким будет это математическое ожидание при решении $(k'_1, k'_2) = (A, C)$. Отметим, что это - решение в пользу пары с нулевой апостериорной вероятностью. Тем не менее при таком решении математическое ожидание количества неправильно распознанных букв будет равно 1. Как и прежде, в действительности может иметь место одна из четырех пар (A, A) , (A, B) , (B, C) , (C, C) , и при каждой из них количество неправильно распознанных букв будет равно 1.

Мы видим, что решение байесовской задачи при штрафной функции (8.51) даже приблизительно не совпадает с решением по штрафной функции (8.52). Следовательно, если в прикладной задаче требуется минимизировать количество неправильно распознанных букв в последовательности (в соответствии с функцией потерь (8.51)), то не следует находить наиболее вероятную последовательность. Она нужна при решении совсем другой задачи с функцией потерь (8.52). ▲

При решении задачи с функцией потерь (8.51) не следует легкомысленно натягивать на нее алгоритм, предназначенный для решения совсем другой задачи. Задачу следует решать с самого начала, например, так, как это показано далее.

Пусть X и K - два конечных множества, \bar{x} и \bar{k} две последовательности длиной n и $n + 1$ соответственно, составленные из элементов множеств X и K , $\bar{x} = (x_1, x_2, \dots, x_n)$, $\bar{k} = (k_0, k_1, \dots, k_n)$. Пара (\bar{x}, \bar{k}) случайна и принимает значения на множестве $X^n \times K^{n+1}$ так, что вероятность пары (\bar{x}, \bar{k}) равна

$$p(\bar{x}, \bar{k}) = p(k_0) \prod_{i=1}^n p(x_i, k_i | k_{i-1}), \quad (8.53)$$

где $p(k_0)$, $p(x_i, k_i | k_{i-1})$ - известные числа.

Пусть $W: K^{n+1} \times K^{n+1} \rightarrow \mathbb{R}$ - штрафная функция вида

$$W(\bar{k}, \bar{k}') = \sum_{i=0}^n w(k_i, k'_i), \quad \text{где} \quad (8.54)$$

$$w(k_i, k'_i) = 1, \quad \text{если } k_i \neq k'_i, \quad (8.55)$$

$$w(k_i, k'_i) = 0, \quad \text{если } k_i = k'_i. \quad (8.56)$$

При этих известных данных следует построить стратегию $q: X^n \rightarrow K^{n+1}$, которая для каждой последовательности x_1, x_2, \dots, x_n определяет последовательность $\bar{k}' = (k'_0, k'_1, \dots, k'_n)$ так, что минимизируется риск

$$\sum_{\bar{k} \in K^{n+1}} p(\bar{k} | \bar{x}) W(\bar{k}, \bar{k}'), \quad \text{то есть,}$$

$$\begin{aligned}
\bar{k}' &= (k'_0, k'_1, \dots, k'_n) = \operatorname{argmin}_{\bar{k}'} \sum_{\bar{k}} p(\bar{k} | \bar{x}) W(\bar{k}, \bar{k}') \\
&= \operatorname{argmin}_{\bar{k}'} \sum_{\bar{k}} p(\bar{x}, \bar{k}) W(\bar{k}, \bar{k}') \\
&= \operatorname{argmin}_{\bar{k}'} \sum_{\bar{k}} p(\bar{x}, \bar{k}) \sum_{i=0}^n w(k_i, k'_i). \quad (8.57)
\end{aligned}$$

Используя (8.53), получаем

$$\begin{aligned}
\bar{k}' &= (k'_0, k'_1, \dots, k'_n) \\
&= \operatorname{argmin}_{\bar{k}'} \sum_{k_0} \sum_{k_1} \dots \sum_{k_n} \left(p(k_0) \prod_{i=1}^n p(x_i, k_i | k_{i-1}) \right) \sum_{i=0}^n w(k_i, k'_i). \quad (8.58)
\end{aligned}$$

Эта задача обладает интереснейшим свойством, которое вытекает из адитивного вида (8.54) функции потерь, даже если не принимать во внимание ее конкретизацию (8.55) и (8.56) и не учитывать марковость (8.53) анализируемых последовательностей. Покажем это свойство.

Обозначим R риск $\sum_{\bar{k}} p(\bar{x}, \bar{k}) \sum_{i=0}^n w(k_i, k'_i)$, который следует минимизировать надлежащим выбором последовательности k'_0, k'_1, \dots, k'_n . Он равен

$$\begin{aligned}
R &= \sum_{\bar{k} \in K^{n+1}} p(\bar{x}, \bar{k}) \sum_{i=0}^n w(k_i, k'_i) = \sum_{i=0}^n \sum_{\bar{k} \in K^{n+1}} p(\bar{x}, \bar{k}) w(k_i, k'_i) \\
&= \sum_{i=0}^n \sum_{k_i \in K} w(k_i, k'_i) \sum_{(k_{i^*}, i^* \neq i)} p(\bar{x}, k_0, k_1, \dots, k_n) \\
&= \sum_{i=0}^n \sum_{k_i \in K} w(k_i, k'_i) p_i(\bar{x}, k_i),
\end{aligned}$$

где

$$p_i(\bar{x}, k_i) = \sum_{(k_{i^*}, i^* \neq i)} p(\bar{x}, k_0, k_1, \dots, k_n).$$

Мы видим, что функция от $n+1$ переменных $(k'_0, k'_1, \dots, k'_n)$, которую следует минимизировать, представляет собой сумму $n+1$ слагаемых, каждое из которых зависит только от одной переменной. При этом каждая переменная входит только в одно слагаемое. Вследствие этого оптимизационная задача (8.57) распадается на $n+1$ независимых оптимизаций по одной переменной,

$$k'_i = \operatorname{argmin}_{k'_i} \sum_{k_i} p_i(\bar{x}, k_i) w(k_i, k'_i). \quad (8.59)$$

Если теперь принять во внимание специфический вид (8.55), (8.56) частных функций w , то получается, что

$$k'_i = \operatorname{argmax}_{k_i \in K} p_i(\bar{x}, k_i). \quad (8.60)$$

В итоге мы имеем, что, хотя исходная задача формулировалась как оптимизационная (см. (8.57) и (8.58)), ее сложность сосредоточена совсем не в оптимизационной части, так как оптимизация сводится к ряду независимых друг от друга оптимизаций (8.59) или (8.60). Оптимизационные задачи, составляющие этот ряд, тривиальны, так как в каждой из них идет речь об оптимизации по значениям только одной переменной. Сложность сосредоточена в вычислении $(n+1)|K|$ чисел $p_i(\bar{x}, k_i)$, $i = 0, 1, \dots, n$, $k_i \in K$, по формуле

$$p_i(\bar{x}, k_i) = \sum_{(k_{i^*}, i^* \neq i)} p(\bar{x}, \bar{k}),$$

или, учитывая, наконец, марковость последовательностей,

$$p_i(\bar{x}, k_i) = \sum_{k_0} \sum_{k_1} \cdots \sum_{k_{i-1}} \sum_{k_{i+1}} \cdots \sum_{k_n} p(k_0) \prod_{i=1}^n p(x_i, k_i | k_{i-1}). \quad (8.61)$$

Процедуры для вычисления вероятностей (8.61) сходны с процедурами для вычисления вероятности

$$p(\bar{x}) = \sum_{k_0} \sum_{k_1} \cdots \sum_{k_{i-1}} \sum_{k_i} \sum_{k_{i+1}} \cdots \sum_{k_n} p(k_0) \prod_{i=1}^n p(x_i, k_i | k_{i-1}), \quad (8.62)$$

к которому сводится распознавание автомата, рассмотренное в разделе 8.3. Сложность вычисления вероятности $p(\bar{x})$, определяемой выражением (8.62), имеет порядок $\mathcal{O}(|K|^2 n)$, равно как и сложность вычисления вероятности $p_i(\bar{x}, k_i)$ при фиксированных значениях i и k_i . Поскольку следует вычислить не одну вероятность $p_i(\bar{x}, k_i)$, а $|K| \cdot (n+1)$ таких вероятностей для $i = 0, 1, \dots, n$ и $k_i \in K$, то (ВНИМАНИЕ! СЛЕДУЕТ ОШИБКА!) для вычисления всех этих вероятностей требуется время порядка $\mathcal{O}(|K|^3 n^2)$. Однако, такое время необходимо только при чересчур расточительных вычислениях. Несложные рассуждения позволяют снизить порядок вычислений до $\mathcal{O}(|K|^2 n)$. Это значит, что вычислить все $|K|(n+1)$ вероятностей $p_i(\bar{x}, k_i)$ можно за время того же порядка, которое нужно для вычисления какой-то одной вероятности из этой совокупности. Эта возможность особенно наглядна при представлении вероятностей (8.61) в матричном виде.

В соответствии с (8.61) для вероятности $p_i(\bar{x}, k_i)$ справедливо

$$\begin{aligned} p_i(\bar{x}, k_i) &= \sum_{k_0} \sum_{k_1} \cdots \sum_{k_{i-1}} \sum_{k_{i+1}} \cdots \sum_{k_n} p(k_0) \prod_{i=1}^n p(x_i, k_i | k_{i-1}) \\ &= \left(\sum_{k_0} \cdots \sum_{k_{i-1}} p(k_0) \prod_{j=1}^i p(x_j, k_j | k_{j-1}) \right) \times \left(\sum_{k_{i+1}} \cdots \sum_{k_n} \psi(k_n) \prod_{j=i+1}^n p(x_j, k_j | k_{j-1}) \right), \end{aligned} \quad (8.63)$$

где $\psi(k_n)$, $k_n \in K$, - числа, равные единице.

Введем обозначения

- φ для $|K|$ -мерного вектора-строки, k_0 -ая компонента которого равна $p(k_0)$;
- ψ для $|K|$ -мерного вектора-столбца, все компоненты которого равны 1;
- P_i для матрицы размера $|K| \times |K|$, в которой в k_i -ом столбце и k_{i-1} -ой строке записано число $p(x_i, k_i | k_{i-1})$.

Пусть функция $f_1: K \rightarrow \mathbb{R}$ представлена $|K|$ -мерным вектором-строкой f_1 , а функция $f_2: K \rightarrow \mathbb{R}$ - $|K|$ -мерным вектором-столбцом f_2 . Запись $f_1 \times f_2$ будем понимать как представление функции $f: K \rightarrow \mathbb{R}$, такой, что для каждого $k \in K$ выполняется $f(k) = f_1(k) \cdot f_2(k)$.

Функции p_i , определяемые выражением (8.63), можно следующим образом представить с использованием введенных обозначений:

$$\begin{aligned} p_0 &= P_1 \cdot P_2 \cdot \dots \cdot P_n \cdot \psi; \\ p_i &= (\varphi \cdot P_1 \cdot P_2 \cdot \dots \cdot P_i) \times (P_{i+1} \cdot P_{i+2} \cdot \dots \cdot P_n \cdot \psi), i = 1, \dots, n-1; \\ p_n &= \varphi \cdot P_1 \cdot P_2 \cdot \dots \cdot P_n. \end{aligned}$$

Функции p_i , $i = 0, 1, \dots, n$, то есть вероятности $p_i(\bar{x}, k_i)$, следует вычислять по такой процедуре:

1. Вычислить векторы-строки $\vec{f}_0, \vec{f}_1, \dots, \vec{f}_n$ по правилам

$$\vec{f}_0 = \varphi, \quad \vec{f}_i = \vec{f}_{i-1} \cdot P_i, \quad i = 1, 2, \dots, n.$$

2. Вычислить векторы-столбцы $\overleftarrow{f}_n, \overleftarrow{f}_{n-1}, \dots, \overleftarrow{f}_0$ по правилам

$$\overleftarrow{f}_n = \psi, \quad \overleftarrow{f}_i = P_{i+1} \cdot \overleftarrow{f}_{i+1}, \quad i = n-1, n-2, \dots, 0.$$

3. Вычислить функции p_0, p_1, \dots, p_n по правилам

$$p_i = \vec{f}_i \times \overleftarrow{f}_i, \quad i = 0, 1, 2, \dots, n.$$

Трудоёмкость вычислений по первым двум пунктам имеет порядок $\mathcal{O}(|K|^2 n)$, а по последнему - порядок $\mathcal{O}(|K| n)$. Общая трудоёмкость имеет, таким образом, порядок $\mathcal{O}(|K|^2 n)$.

Мы рассмотрели три задачи распознавания объектов, которые характеризуются марковскими последовательностями. Это - распознавание объекта в целом, отыскание наиболее вероятной последовательности скрытых параметров и отыскание последовательности наиболее вероятных значений скрытых параметров. Хотя мы и рассмотрели эти задачи в различных их модификациях, мы ни в коем случае не утверждаем, что рассмотренные задачи охватывают все многообразие приложений. Наоборот, мы хотим создать впечатление, что здесь есть еще очень и очень много разнообразных и еще не тронутых задач, и тем самым расшатать распространенный и приятный самообман, что для распознавания марковских последовательностей достаточно уметь отыскивать наиболее

вероятную последовательность путем нахождения кратчайшего пути на графе методами динамического программирования.

Обнаружение класса оптимизационных задач распознавания, которые конструктивно решаются методами динамического программирования, явилось в свое время прорывным достижением и фактически ознаменовало начало структурного распознавания. Это достижение заслуживает безусловного уважения и сейчас, спустя десятилетия. Вместе с тем мы не должны забывать, что даже очень значительные знания превращаются в свою противоположность, когда начинают восприниматься, как исчерпывающие и универсально применимые.

8.6 Марковские объекты с ациклической структурой

8.6.1 Статистическая модель объекта

Мы видели, что если распознаваемый объект адекватно представляется марковской последовательностью наблюдаемых и скрытых параметров, то классические байесовские задачи распознавания такого объекта оказываются конструктивно разрешимыми. Существенно при этом, что их решение не наталкивается на какие-либо непреодолимые вычислительные трудности. Например, сложность рассмотренных задач растет линейно, а не экспоненциально с ростом наблюдений.

Все эти хорошие свойства следуют из вида совместных вероятностей $p(\bar{x}, \bar{k})$, который выражается равенством (8.1) и эквивалентным ему равенством (8.3) и неформально иллюстрируется механической моделью (Рис. 8.1). Сейчас мы покажем, что конструктивное распознавание сложных объектов возможно и при более слабых предположениях, чем (8.1) и (8.3). Мы покажем, что сложный объект можно распознавать и тогда, когда составляющие его части образуют не одномерную последовательность, а более сложную структуру.

Последовательность $\bar{k} = (k_0, k_1, \dots, k_n)$ можно трактовать, как функцию вида $I \rightarrow K$ на множестве $\{0, 1, \dots, n\}$ индексов, которыми являются целые числа. Мы обобщим понятие последовательности так, что множество I будет не подмножеством целых чисел, а множеством вершин неориентированного ациклического графа, то есть дерева. Обозначим этот граф G . Скрытой частью \bar{k} , характеризующей объект, будет, как и раньше, функция $I \rightarrow K$, определенная на вершинах графа. Последовательность есть такой частный случай этой функции, когда граф G есть цепочка, то есть связный граф, в котором из каждой вершины выходит не менее одного ребра и не более двух ребер.

Понятие последовательности $\bar{x} = (x_1, x_2, \dots, x_n)$ наблюдений мы обобщим так, что \bar{x} будет рассматриваться как функция $H \rightarrow X$, определенная на множестве H ребер графа G . Таким образом, наблюдение \bar{x} - это совокупность $(x_h, h \in H)$, составленная из индексированных величин x_h , где h - это ребро в графе G , или, что тоже самое, определенная пара (i, i') вершин. Наблюдение \bar{x} есть последовательностью, когда граф G является цепочкой, и следовательно, и множество его ребер образует цепочку.

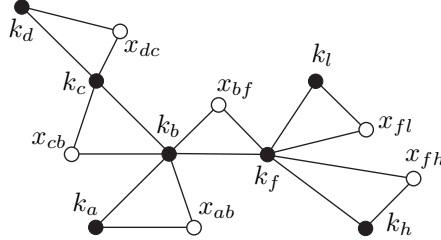


Figure 8.4 Mechanical model of a Markovian acyclic graph.

Основная предпосылка о совместной вероятности $p(\bar{x}, \bar{k})$ наблюдения $\bar{x}: H \rightarrow X$ и скрытых параметров $\bar{k}: I \rightarrow K$, которая обобщает марковскую модель (8.1), состоит в том, что для любой вершины $0 \in I$, назовем ее 0-ой, вероятность $p(\bar{x}, \bar{k})$ имеет вид

$$p(\bar{x}, \bar{k}) = p(k_0) \prod_{i \in I \setminus \{0\}} p(x_{\{i, g(i)\}}, k_i | k_{g(i)}), \quad (8.64)$$

где $g(i)$ - это вершина, соединенная с вершиной i ребром и лежащая на пути от 0-ой вершины к i -ой. Ранее сформулированное свойство (8.3) есть частный случай свойства (8.64) при $I = \{0, 1, 2, \dots, n\}$ и $g(i) = i - 1$.

Свойство (8.64) можно неформально представить в виде модели на рис. 8.4, которая обобщает механическую модель на рис. 8.1, иллюстрировавшую свойства марковских последовательностей.

На рис. 8.4 значения x_h , $h \in H$, и k_i , $i \in I$, представлены точками на плоскости и соединяющими их отрезками. Если представить себе каждый отрезок в виде упругой пружины, можно увидеть, что положение каждой точки влияет на положение всех остальных точек и отрезков. Однако, если зафиксировать некоторую точку, скажем, точку k_f , механическая модель распадается на три независимые части. Первая часть состоит из точек k_a, k_b, k_c, k_d и ребер x_{ab}, x_{cb}, x_{dc} и x_{bf} , вторая - из элементов k_l, x_{fl} , а третья - из k_h и x_{fh} . Это и есть условная независимость отдельных частей сложного объекта, которая формально выражена равенством (8.64). Эта условная независимость служит основой для формулировки и точного решения байесовских задач, подобных тем, которые мы рассматривали для последовательностей. Мы рассмотрим только две задачи, причем довольно кратко. Первой является задача вычисления вероятности $p(\bar{x})$ заданного наблюдения \bar{x} , а вторая - это вычисление числа $\max_{\bar{k} \in K^I} p(\bar{x}, \bar{k})$ при заданном наблюдении \bar{x} . При понимании этих двух задач становятся вполне ясными и другие задачи и их модификации, с которыми мы имели дело для случая последовательностей.

8.6.2 Вычисление вероятности наблюдения

Вычисление вероятности $p(\bar{x})$ состоит в вычислении многомерной суммы

$$p(\bar{x}) = \sum_{(k_i, i \in I)} p(k_0) \prod_{i \in I \setminus \{0\}} p(x_{\{i, g(i)\}}, k_i | k_{g(i)}) \quad (8.65)$$

для заданного наблюдения \bar{x} . Так как величины x_h , $h \in H$, фиксированы, выражение (8.65) можно переписать так, чтобы в них эти величины вообще не присутствовали. Обозначим $f_i(k_i, k_{g(i)})$ вероятность $p(x_{\{i, g(i)\}}, k_i | k_{g(i)})$. Чтобы добиться симметрии по вершинам i в выражении (8.65), а также из технологических соображений, которые станут ясны позже, введем переменные $\varphi_i(k_i)$, такие, что $\varphi_0(k_0) = p(k_0)$ при $i = 0$ и $\varphi_i(k_i) = 1$ для всех $i \neq 0$ и $k_i \in K$. Выражение (8.65) таким образом приобретает вид

$$d = \sum_{(k_i, i \in I)} \prod_{i \in I} \varphi_i(k_i) \prod_{i \in I \setminus \{0\}} f_i(k_i, k_{g(i)}). \quad (8.66)$$

В этом выражении каждая переменная k_i присутствует лишь в одном сомножителе вида $\varphi_i(k_i)$. Что касается сомножителей вида $f_i(k_i, k_{g(i)})$, то та или иная переменная k_{i^*} может присутствовать в одном, или двух, или более сомножителях в зависимости от индекса i^* . Переменная k_{i^*} , естественно, присутствует в сомножителе $f_{i^*}(k_{i^*}, k_{g(i^*)})$. Но она присутствует еще и в тех сомножителях $f_i(k_i, k_{g(i)})$, в которых $g(i) = i^*$. В то же время гарантированно существует такой индекс i^* , что переменная k_{i^*} присутствует только в одном сомножителе вида $f_i(k_i, k_{g(i)})$. Это тот индекс i^* , для которого при любом i выполняется неравенство $i^* \neq g(i)$. Существование такого индекса i^* следует из ацикличности графа G . В ациклическом графе всегда существует по крайней мере одна вершина, из которой выходит только одно ребро. Это и есть вершина i^* . Для индекса i^* формулу (8.66) можно переписать в виде

$$d = \sum_{(k_i | i \in I, i \neq i^*)} \prod_{i \in I \setminus \{i^*\}} \varphi_i(k_i) \prod_{i \in I \setminus \{0, i^*\}} f_i(k_i, k_{g(i)}) \sum_{k_{i^*}} \varphi_{i^*}(k_{i^*}) f_{i^*}(k_{i^*}, k_{g(i^*)}). \quad (8.67)$$

Обозначим $g(i^*)$ как i' и вычислим новые значения чисел $\varphi_{i'}(k_{i'})$ с помощью оператора присвоения

$$\varphi_{i'}(k_{i'}) := \varphi_{i'}(k_{i'}) \sum_{k_{i^*}} \varphi_{i^*}(k_{i^*}) f_{i^*}(k_{i^*}, k_{i'}). \quad (8.68)$$

Приведенное выражение следует понимать именно как оператор, а не как равенство. Обозначение $\varphi_{i'}(k_{i'})$ в правой части выражения (8.68) обозначает значение числа $\varphi_{i'}(k_{i'})$ до выполнения оператора; то же самое обозначение в левой части оператора обозначает новое значение этого числа, полученное после выполнения оператора. Оператор (8.68) имеет вычислительную сложность порядка $\mathcal{O}(|K|^2)$.

Полученное новое значение числа $\varphi_{i'}(k_{i'})$ можно подставить в выражение (8.67) и переписать это выражение в виде

$$\sum_{(k_i | i \in I_1)} \prod_{i \in I_1} \varphi_i(k_i) \prod_{i \in I_1 \setminus \{0\}} f_i(k_i, k_{g(i)}), \quad (8.69)$$

где $I_1 = I \setminus \{i^*\}$. Выражение (8.69) имеет тот же вид, что и исходное выражение (8.66), но количество переменных k_i в нем уже на единицу меньше.

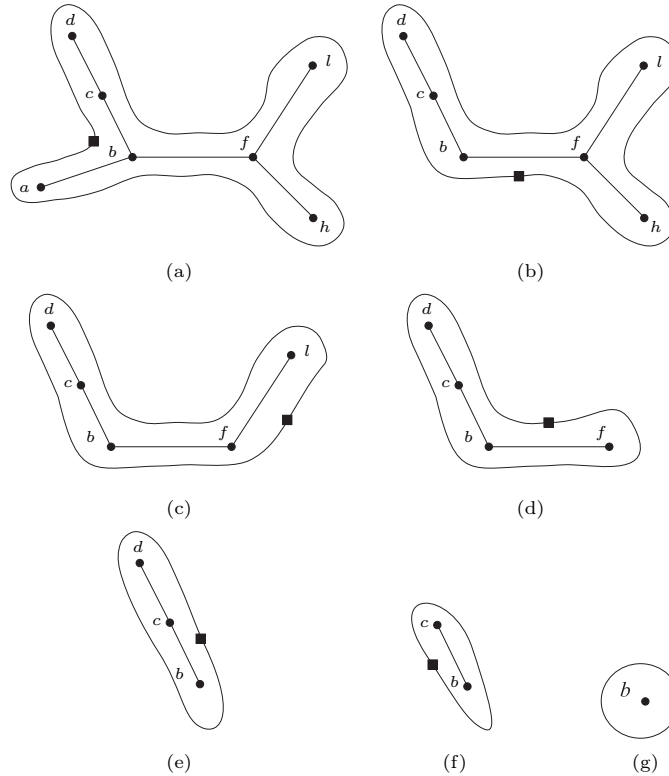


Figure 8.5 Reduction of the graph in calculating $p(x)$.

В этом сокращенном на единицу множестве переменных опять существует переменная, которая присутствует только в одном сомножителе вида $f_i(k_i, k_{g(i)})$ и которую можно исключить указанным ранее способом. После $(|I| - 1)$ -ого исключения выражение (8.67) преобразуется к виду

$$d = \sum_{k_0} \varphi(k_0),$$

при котором искомая величина d легко вычисляется, так как здесь речь идет о суммировании по значениям только одной переменной. Суммарная сложность вычисления вероятности $p(\bar{x})$ имеет порядок $\mathcal{O}(|K|^2 |I|)$, то есть тот же самый порядок, что и в аналогичной задаче для последовательностей.

Пример 8.7 Вычисление вероятности наблюдения в ациклическом графе. Покажем порядок вычислений по рекомендуемому алгоритму для графа на рис. 8.4. Структура исходных данных соответствует начальному графу, представленному на рис. 8.5(a). Каждой вершине i соответствует память для хранения $|K|$ чисел $\varphi_i(k_i)$, $k_i \in K$. Каждому ребру

(i, i') соответствует память для хранения $|K|^2$ чисел $f_i(k_i, k_{i'})$, $k_i \in K$, $k_{i'} \in K$.

Процесс вычислений можно объяснить следующим образом. Исходный граф G следует окаймить (обрисовать) замкнутой кривой, как это показано на рис. 8.5(a). На этой кривой следует выбрать начальную точку (представлена на рис. 8.5(a) черным квадратиком) и обойти контур против часовой стрелки. В процессе обхода контура зафиксировать последовательность вершин, рядом с которыми происходит обход. В данном примере это будет последовательность $(b, a, b, f, h, f, l, f, b, c, d, c, b)$. Вершины графа обходятся в указанной последовательности и на каждом шаге обхода либо происходят какие-то изменения данных, либо нет. Изменяются данные только в тех вершинах, из которых в текущем графе выходит только одно ребро. В данном примере будут изменяться данные последовательно для вершин (a, h, l, f, d, c) , причем в указанном порядке. Эти изменения состоят в том, что из графа исключается одна вершина и вычисляются новые значения чисел $\varphi_i(k_i)$ для некоторой вершины i , как это указано далее по шагам для каждой вершины из последовательности (a, h, l, f, d, c) .

Вершина a . Вычисляются новые значения чисел $\varphi_b(k_b)$ по формуле

$$\varphi_b(k_b) := \varphi_b(k_b) \sum_{k_a} \varphi_a(k_a) f_a(k_a, k_b).$$

Вершина a исключается из графа, и алгоритм продолжает работу на основе графе, представленном на рис. 8.5(b).

Вершина h . Вычисляются новые значения чисел $\varphi_f(k_f)$ по формуле

$$\varphi_f(k_f) := \varphi_f(k_f) \sum_{k_h} \varphi_h(k_h) f_h(k_h, k_f).$$

Вершина h исключается из графа, и алгоритм продолжает работу на основе графе, представленном на рис. 8.5(c).

Вершина l . Вычисляются новые значения чисел $\varphi_f(k_f)$ по формуле

$$\varphi_f(k_f) := \varphi_f(k_f) \sum_{k_l} \varphi_l(k_l) f_l(k_l, k_f).$$

Вершина l исключается из графа, и алгоритм продолжает работу на основе графе, представленном на рис. 8.5(d).

Вершина f . Вычисляются новые значения чисел $\varphi_b(k_b)$ по формуле

$$\varphi_b(k_b) := \varphi_b(k_b) \sum_{k_f} \varphi_f(k_f) f_f(k_f, k_b).$$

Вершина f исключается из графа, и алгоритм продолжает работу на основе графе, представленном на рис. 8.5(e).

Вершина d . Вычисляются новые значения чисел $\varphi_c(k_c)$ по формуле

$$\varphi_c(k_c) := \varphi_c(k_c) \sum_{k_d} \varphi_d(k_d) f_d(k_d, k_c).$$

Вершина d исключается из графа, и алгоритм продолжает работу на основе графе, представленном на рис. 8.5(f).

Вершина c . Вычисляются новые значения чисел $\varphi_b(k_b)$ по формуле

$$\varphi_b(k_b) := \varphi_b(k_b) \sum_{k_c} \varphi_c(k_c) f_c(k_c, k_b).$$

Вершина c исключается из графа, и алгоритм продолжает работу на основе графе, представленном на рис. 8.5(g), который состоит из единственной точки b .

Для полученного предельно упрощенного графа вероятность наблюдения x определяется как сумма $\sum_{k_b} \varphi_b(k_b)$. ▲

8.6.3 Наиболее вероятная совокупность скрытых параметров

Этот подраздел можно было бы и не писать, а просто заменить везде в предыдущем разделе слово "сложение" на слово "максимизация". Текст, полученный таким чисто формальным способом, был бы в той же степени правильным, как и текст предыдущего раздела. Правильность предыдущего раздела обусловлена дистрибутивностью умножения относительно сложения. Правильность данного раздела обусловлена дистрибутивностью умножения на неотрицательное число относительно максимизации. Действительно, $x \max(y, z) = \max(xy, xz)$ при любом неотрицательном x . Таким образом, алгоритм вычисления

$$d = \max_{(k_i, i \in I)} \prod_{i \in I} \varphi_i(k_i) \prod_{i \in I \setminus \{0\}} f_i(k_i, k_{g(i)}) \quad (8.70)$$

имеет ту же структуру и основан на тех же соображениях, как и алгоритм для вычисления

$$d = \sum_{(k_i, i \in I)} \prod_{i \in I} \varphi_i(k_i) \prod_{i \in I \setminus \{0\}} f_i(k_i, k_{g(i)}), \quad (8.71)$$

который мы только что рассмотрели. Мы очень кратко повторим эти соображения, но применительно к вычислению (8.70), а не (8.71).

Пусть i^* - такой индекс, что для любого $i \in I$ выполняется неравенство $g(i) \neq i^*$. Это значит, что переменная k_{i^*} присутствует только в одном сомножителе вида $f_i(k_i, k_{g(i)})$, а именно, в сомножителе $f_{i^*}(k_{i^*}, k_{g(i^*)})$. Следовательно, формулу (8.70) можно переписать в виде

$$d = \max_{(k_i, i \in I, i \neq i^*)} \prod_{i \in I \setminus \{i^*\}} \varphi_i(k_i) \prod_{i \in I \setminus \{0, i^*\}} f_i(k_i, k_{g(i)}) \max_{k_{i^*}} (\varphi_{i^*}(k_{i^*}) f_{i^*}(k_{i^*}, g(i^*))).$$

Обозначим $g(i^*)$ как i' и вычислим новые значения чисел $\varphi_{i'}(k_{i'})$ с помощью оператора

$$\varphi_{i'}(k_{i'}) := \varphi_{i'}(k_{i'}) \max_{k_{i^*}} (\varphi_{i^*}(k_{i^*}) f_{i^*}(k_{i^*}, i')). \quad (8.72)$$

Используя это вычисленное новое значение, число d можно записать как

$$d = \max_{(k_i | i \in I_1)} \prod_{i \in I_1} \varphi_i(k_i) \prod_{i \in I_1 \setminus \{0\}} f_i(k_i, k_{g(i)}), \quad (8.73)$$

где $I_1 = I \setminus \{i^*\}$. В выражении (8.73) максимизация производится по переменным, количество которых на единицу меньше, чем в исходном выражении (8.70). Оператор (8.72) используется $(|I| - 1)$ раз и при каждом его применении количество переменных уменьшается на единицу. Таким путем исходная задача сводится к вычислению

$$d = \max_{k_0} \varphi_0(k_0),$$

которое тривиально, потому что здесь речь идет о максимизации по всем значениям только одной переменной. Общее количество вычислений при решении задачи (8.70) имеет порядок $\mathcal{O}(|K|^2 n)$, то есть тот же, что и для случая последовательностей.

8.7 Формулировка задач обучения и самообучения

В лекции 4 были сформулированы в общем виде три задачи обучения распознаванию, как то или иное разумное оценивание статистической модели распознаваемого объекта. В лекции 6 мы сформулировали задачу самообучения и указали общий алгоритм ее решения. Сейчас мы конкретизируем эти задачи для случая марковских моделей, с которыми сейчас имеем дело. Мы проанализируем случай, когда совокупность параметров объекта имеет структуру последовательности, а не более общую структуру ациклического графа. Наиболее важные свойства решаемых задач проявляются уже на последовательностях, а рассмотрение их для более общего случая ациклических графов привело бы только к большей громоздкости формул. Обобщение результатов, которые мы покажем для случая последовательностей, на случай ациклических графов не выходит за рамки простого математического упражнения.

Как и раньше, мы предполагаем, что распознаваемый объект характеризуется последовательностью $\bar{x} = (x_1, x_2, \dots, x_n)$ наблюдаемых признаков длиной n и последовательностью $\bar{k} = (k_0, k_1, \dots, k_n)$ скрытых параметров длиной $n + 1$. Пара (\bar{x}, \bar{k}) случайна и принимает то или иное значение в соответствии с вероятностями $p(\bar{x}, \bar{k})$. Это распределение вероятностей не может быть произвольным, а должно иметь вид

$$p(\bar{x}, \bar{k}) = p(k_0) \prod_{i=1}^n p_i(k_i, x_i | k_{i-1}).$$

Это значит, что функция $p: X^n \times K^{n+1} \rightarrow \mathbb{R}$ однозначно определяется n функциями p_i , $i = 1, 2, \dots, n$, вида $K \times X \times K \rightarrow \mathbb{R}$. Значение $p_i(k', x, k)$ есть совместная вероятность того, что $(i - 1)$ -ый скрытый параметр примет значение k' , i -ый скрытый параметр - значение k , а i -ый наблюдаемый

параметр - значение x . Функция p_i однозначно определяет также вероятность, что i -ый скрытый параметр примет то или иное значение $k \in K$. Эта вероятность равна $\sum_{k' \in K} \sum_{x \in X} p_i(k', x, k)$. Эта же вероятность определяется и функцией p_{i+1} как $\sum_{x \in X} \sum_{k' \in K} p_{i+1}(k, x, k')$. Эта ситуация не должна приводить к противоречию, то есть функции $p_i, i = 1, \dots, n$, должны удовлетворять условию

$$\sum_{k' \in K} \sum_{x \in X} p_i(k', x, k) = \sum_{x \in X} \sum_{k' \in K} p_{i+1}(k, x, k'), \quad i = 1, 2, \dots, n-1, \quad k \in K.$$

Совокупность функций $(p_i, i = 1, \dots, n)$, удовлетворяющих этому условию, обозначим P , и будем называть ее статистической моделью объекта распознавания. Так как совокупность P однозначно определяет и функцию $p(\bar{x}, \bar{k})$, то есть распределение вероятностей пар $\bar{x} = (x_1, \dots, x_n)$ и $\bar{k} = (k_0, \dots, k_n)$, функцию p тоже будем называть *статистической моделью объекта*.

Если статистическая модель объекта известна, то могут формулироваться и решаться различные задачи распознавания, примеры которых мы рассмотрели в предыдущей части лекции. Когда модель неизвестна, ее следует найти либо на основании экспериментального исследования объекта, либо на основании информации от будущего пользователя о том, какие результаты распознавания он считает правильными. Такое формирование статистической модели объекта называется в распознавании обучением или самообучением, а данные, на основании которых строится модель, называются обучающей информацией. Та или иная точная формулировка этих задач зависит от характера обучающей информации. Далее приводятся эти формулировки.

8.7.1 Наиболее правдоподобное оценивание в режиме обучения

Предположим, что объект распознавания был переведен в специальный режим, при котором стало возможным измерять все его параметры, как наблюдаемые, так и те, которые обычно скрыты. Пусть выполнены l экспериментов с этим объектом, и результатом каждого, j -ого, $j = 1, 2, \dots, l$, эксперимента была пара последовательностей $\bar{x}^j = (x_1^j, x_2^j, \dots, x_n^j)$ и $\bar{k}^j = (k_0^j, k_1^j, \dots, k_n^j)$. Символом $x_i^j, i = 1, 2, \dots, n, j = 1, 2, \dots, l$, обозначен i -ый элемент j -ой последовательности, k_i^j имеет аналогичный смысл.

Если имеются веские причины считать, что результаты l экспериментов были взаимно независимыми реализациями случайной пары (\bar{x}, \bar{k}) с распределением вероятностей p , то модель $P = (p_1, p_2, \dots, p_n)$ можно оценить так, как это делается по методу наибольшего правдоподобия в классической статистике. Требуется найти такую функцию p , которая максимизирует вероятность экспериментально наблюдаемых результатов, то есть вероятность $\prod_{j=1}^l p(\bar{x}^j, \bar{k}^j)$. Более конкретно, требуется найти совокупность

функций

$$\begin{aligned} P^* &= (p_1^*, p_2^*, \dots, p_n^*) \\ &= \operatorname{argmax}_{p_1} \cdots \max_{p_n} \prod_{j=1}^l p_1(k_0^j, x_1^j, k_1^j) \prod_{i=2}^n \frac{p_i(k_{i-1}^j, x_i^j, k_i^j)}{\sum_{k \in K} \sum_{x \in X} p_i(k_{i-1}^j, x, k)}. \end{aligned} \quad (8.74)$$

Отметим, что в последнем выражении совместная вероятность $p(\bar{x}, \bar{k})$ выражена не через условные вероятности $p_i(k_i^j, x_i^j | k_{i-1}^j)$, как это делалось до сих пор, а эквивалентным образом через совместные вероятности $p_i(k_{i-1}^j, x_i^j, k_i^j)$.

8.7.2 Минимаксная оценка модели

Зачастую бывает очень трудно обеспечить или даже проверить, выполняются ли условия эксперимента, при которых оценка модели по методу максимального правдоподобия уместна. По этим условиям требуется, чтобы результатом эксперимента являлась случайная выборка взаимно независимых реализаций при том же распределении вероятностей, которое будет иметь место при последующей эксплуатации распознающей системы. Если эти условия не гарантируются, следует формулировать задачу обучения иначе, а именно, следующим образом.

Пусть имеется множество (а не случайная выборка) пар (\bar{x}^1, \bar{k}^1) , (\bar{x}^2, \bar{k}^2) , \dots , (\bar{x}^l, \bar{k}^l) , называемое обучающим множеством. Оно построено экспериментатором так, что он отобрал типичные по его мнению примеры, которые по его неформальным соображениям являются хорошими и весьма вероятными для распознаваемого объекта. Например, если речь идет о распознавании букв, то для каждой буквы экспериментатор отбирает в обучающее множество определенное количество разнообразных, но неискаженных изображений, которые по его мнению хорошо представляют все множество пар "изображение-буква", с которыми придется иметь дело. Целью обучения является поиск такой модели P^* , при которой ни одна из вероятностей $p^*(\bar{x}^j, \bar{k}^j)$ для отобранных примеров не окажется слишком малой. Более точно, следует найти максимальное значение ε , для которого еще существует такая модель P^* , что для любого примера (\bar{x}^j, \bar{k}^j) из обучающего множества выполнится неравенство $p^*(\bar{x}^j, \bar{k}^j) \geq \varepsilon$. Или, говоря по другому, требуется найти модель P^* ,

$$\begin{aligned} P^* &= (p_1^*, p_2^*, \dots, p_n^*) \\ &= \operatorname{argmax}_{p_1} \max_{p_2} \cdots \max_{p_n} \min_j \left(p_1(k_0^j, x_1^j, k_1^j) \prod_{i=2}^n \frac{p_i(k_{i-1}^j, x_i^j, k_i^j)}{\sum_{k \in K} \sum_{x \in X} p_i(k_{i-1}^j, x, k)} \right). \end{aligned}$$

8.7.3 Настройка алгоритма распознавания

Предположим, что уже построен алгоритм распознавания последовательности $\bar{k} = (k_0, k_1, \dots, k_n)$ по последовательности $\bar{x} = (x_1, x_2, \dots, x_n)$ в фор-

мулировке, приведенной в разделе 8.4.2. Это алгоритм вида

$$\bar{k} = \operatorname{argmax}_{k_0, k_1, \dots, k_n} \sum_{i=1}^n f_i(k_{i-1}, x_i, k_i), \quad (8.75)$$

где функции $f_i: K \times X \times K \rightarrow \mathbb{R}$ определенным образом зависят от статистической модели ($p_i: K \times X \times K \rightarrow \mathbb{R}$, $i = 1, \dots, n$). Вполне может случиться, что программа вычисления (8.75) уже написана, а статистическая модель $P = (p_i, i = 1, 2, \dots, n)$ неизвестна. В этом случае возникает вопрос, что следует записать в массив памяти, предназначенный для хранения чисел $f_i(k', x, k)$. Функции f_i можно построить различными путями в зависимости от характера обучающей информации (\bar{x}^j, \bar{k}^j) , $j = 1, \dots, l$. Если, например, можно рассматривать пары (\bar{x}^j, \bar{k}^j) как случайные и взаимно независимые реализации случайной величины, модель P можно построить в соответствии с требованиями (8.74). Полученные функции p_i затем пересчитываются в функции f_i по несложным формулам.

Однако природа примеров (\bar{x}^j, \bar{k}^j) , $j = 1, \dots, l$, может быть совершенно иной. Они могли быть получены не в результате наблюдения за реальным объектом, а как выражение требования пользователя к разрабатываемому распознающему алгоритму. В этом случае \bar{k}^j - это последовательность, которая должна быть выдана распознающим алгоритмом, когда на его вход подается последовательность \bar{x}^j . При таком характере обучающей информации функции f_i перестают быть статистическими характеристиками объекта распознавания, а становятся параметрами распознающего устройства. Эти параметры должны быть настроены так, чтобы на примерах, включенных в обучающее множество, алгоритм выдавал те результаты, которые указаны в обучающем множестве. Процессы, которые здесь происходят, уместно было бы называть прозаично "настройка распознающего алгоритма", а не поэтически "обучение распознаванию". Подобным образом, обучающее множество уместней было бы называть просто "тестовый материал". Настройка распознающего алгоритма (8.75) по заданному тестовому материалу (\bar{x}^j, \bar{k}^j) , $j = 1, \dots, l$, состоит в отыскании функций f_1, f_2, \dots, f_n , которые являются решением системы отношений

$$(k_0^j, k_1^j, \dots, k_n^j) = \operatorname{argmax}_{k_0} \max_{k_1} \dots \max_{k_n} \sum_{i=1}^n f_i(k_{i-1}, x_i^j, k_i), \quad j = 1, 2, \dots, l.$$

Обозначение "argmax" в этом требовании следует понимать как строгое. Это значит, что последовательность \bar{k}^j должна быть не одной из последовательностей, в которой достигается максимум, а единственной такой последовательностью. Без этой оговорки приведенная система имела бы тривиальное решение, когда все числа $f_i(k', x, k)$ равны нулю.

8.7.4 Задача самообучения

Совокупность P функций (p_1, p_2, \dots, p_n) однозначно определяет не только вероятность $p(\bar{x}, \bar{k})$ каждой пары (\bar{x}, \bar{k}) , но и вероятность каждой последовательности $\bar{x} = (x_1, x_2, \dots, x_n)$, так как она равна $\sum_{\bar{k} \in K^{n+1}} p(\bar{x}, \bar{k})$.

Пусть с объектом распознавания выполнены l экспериментов, в каждом из которых получены значения только наблюдаемых параметров. Задача самообучения состоит в том, чтобы на основании этих данных найти наиболее правдоподобную модель объекта

$$\begin{aligned}
 P^* &= (p_1^*, p_2^*, \dots, p_n^*) = \operatorname{argmax}_P \prod_{j=1}^l \sum_{\bar{k} \in K^{n+1}} p(\bar{x}^j, \bar{k}) \\
 &= \operatorname{argmax}_{p_1} \cdots \operatorname{max}_{p_n} \prod_{j=1}^l \sum_{k_0} \sum_{k_1} \cdots \sum_{k_n} p_1(k_0, x_1^j, k_1) \prod_{i=2}^n \frac{p_i(k_{i-1}, x_i^j, k_i)}{\sum_{x \in X} \sum_{k \in K} p_i(k_{i-1}, x, k)}.
 \end{aligned}$$

В этой задаче речь идет об обнаружении взаимной зависимости между скрытыми параметрами объекта и зависимости наблюдаемых признаков от скрытых параметров в ситуации, когда скрытые параметры не наблюдаются не только при распознавании, но даже при экспериментальном исследовании объекта.

8.8 Построение наиболее правдоподобной модели в режиме обучения

Наиболее правдоподобная модель определена требованием (8.74). Руководствуясь неформальными разумными соображениями, можно прийти к правильному выводу о том, как эта модель должна строиться по экспериментальным данным. Приведем эти разумные соображения, а затем покажем, что построенная на их основании модель удовлетворяет требованию (8.74). Предположим, что на основании экспериментально наблюдаемой выборки $((\bar{x}^j, \bar{k}^j), j = 1, 2, \dots, l)$ надо для каждой тройки значений $k' \in K, x \in X, k \in K$ и каждого i сделать разумный вывод о совместной вероятности того, что $(i-1)$ -ый скрытый параметр равен k' , i -ый наблюдаемый параметр равен x , а i -ый скрытый параметр равен k . Даже не уточняя, что понимается под разумностью в приведенной формулировке, вряд ли может быть что-то более естественное, чем простой подсчет, сколько раз произошло событие $k_{i-1} = k', x_i = x, x_i = k$ в процессе экспериментирования, и деление этого количества на общее количество l экспериментов.

Эту разумную рекомендацию можно обосновать, а именно, показать, что модель, полученная таким способом, максимизирует (8.74). Это доказательство нам необходимо, потому что далее мы используем максимально правдоподобное оценивание как составную часть более сложного минимаксного оценивания и самообучения. Анализ этих более сложных задач будет базироваться не на том, что приведенный способ оценки модели разумен, а на том, что он обеспечивает максимум (8.74).

С целью дальнейшего формального анализа задачи наиболее правдопо-

добного оценивания представим ее в следующем эквивалентном виде,

$$P^* = (p_1^*, p_2^*, \dots, p_n^*) \\ = \operatorname{argmax}_{p_1} \cdots \operatorname{max}_{p_n} \sum_{j=1}^l \left(\log p_1(k_0^j, x_1^j, k_1^j) + \sum_{i=2}^n \log \frac{p_i(k_{i-1}^j, x_i^j, k_i^j)}{\sum_{x \in X} \sum_{k \in K} p_i(k_{i-1}^j, x, k)} \right).$$

Выразим выборку $((\bar{x}^j, \bar{k}^j), j = 1, 2, \dots, l)$ в виде функции $g: X^n \times K^{n+1} \rightarrow \mathbb{Z}$, где \mathbb{Z} - множество целых чисел. Значение $g(\bar{x}, \bar{k})$ этой функции на паре последовательностей $\bar{x} \in X^n$ и $\bar{k} \in K^{n+1}$ обозначает, сколько раз пара (\bar{x}, \bar{k}) встретилась в выборке $((\bar{x}^j, \bar{k}^j), j = 1, 2, \dots, l)$. С использованием обозначения g предыдущее выражение для P^* преобразуется к виду

$$P^* = \operatorname{argmax}_P \left(\sum_{\bar{x} \in X^n} \sum_{\bar{k} \in K^{n+1}} g(\bar{x}, \bar{k}) \left(\log p_1(k_0, x_1, k_1) + \sum_{i=2}^n \log \frac{p_i(k_{i-1}, x_i, k_i)}{\sum_{x \in X} \sum_{k \in K} p_i(k_{i-1}, x, k)} \right) \right). \quad (8.76)$$

Содержание задачи не изменится, если целочисленную функцию g заменить функцией α , для которой $\alpha(\bar{x}, \bar{k}) = g(\bar{x}, \bar{k})/l$,

$$P^* = \operatorname{argmax}_P \left(\sum_{\bar{x} \in X^n} \sum_{\bar{k} \in K^{n+1}} \alpha(\bar{x}, \bar{k}) \left(\log p_1(k_0, x_1, k_1) + \sum_{i=2}^n \log \frac{p_i(k_{i-1}, x_i, k_i)}{\sum_{x \in X} \sum_{k \in K} p_i(k_{i-1}, x, k)} \right) \right). \quad (8.77)$$

Поскольку для функции α справедливо

$$\sum_{\bar{x} \in X^n} \sum_{\bar{k} \in K^{n+1}} \alpha(\bar{x}, \bar{k}) = 1,$$

она может пониматься как распределение вероятностей на множестве пар (\bar{x}, \bar{k}) , то есть на множестве всех возможных исходов эксперимента.

Введем обозначение $X_i(x')$, $i = 1, 2, \dots, n$, $x' \in X$, для множества всех тех последовательностей $\bar{x} = (x_1, x_2, \dots, x_n) \in X^n$, для которых $x_i = x'$, и обозначение $K_i(k', k'')$, $i = 1, 2, \dots, n$, $k' \in K$, $k'' \in K$, для всех тех последовательностей $\bar{k} = (k_0, k_1, \dots, k_n) \in K^{n+1}$, для которых $k_{i-1} = k'$, $k_i = k''$. Обозначим $\alpha_i(k', x, k)$ сумму

$$\alpha_i(k', x, k) = \sum_{\bar{x} \in X_i(x)} \sum_{\bar{k} \in K_i(k', k)} \alpha(\bar{x}, \bar{k}). \quad (8.78)$$

Сформулированная ранее рекомендация по оценке вероятностей $p_i(k', x, k)$ фактически утверждает, что функции p_i

должны равняться функциям α_i . Точная формулировка этого утверждения содержится в далее приведенной теореме, равно, как и доказательство, что при таком выборе функций p_i решается задача (8.77), а следовательно, и задача (8.74).

Теорема 8.1 Наиболее правдоподобное оценивание марковской модели. Пусть X и K - два конечных множества, а α - неотрицательно определенная функция вида $X^n \times K^{n+1} \rightarrow \mathbb{R}$, для которой выполняется

$$\sum_{\bar{x} \in X^n} \sum_{\bar{k} \in K^{n+1}} \alpha(\bar{x}, \bar{k}) = 1.$$

Пусть p_i^* , $i = 1, 2, \dots, n$, - функции вида $K \times X \times K \rightarrow \mathbb{R}$, для которых выполняется

$$p_i^*(k', x, k) = \sum_{\bar{x} \in X_i(x)} \sum_{\bar{k} \in K_i(k', k)} \alpha(\bar{x}, \bar{k}),$$

а p_i , $i = 1, 2, \dots, n$, - произвольные неотрицательно определенные функции $K \times X \times K \rightarrow \mathbb{R}$, для которых выполняется

$$\sum_{k' \in K} \sum_{x \in X} \sum_{k \in K} p_i(k', x, k) = 1.$$

В таком случае справедливо неравенство

$$\begin{aligned} & \sum_{\bar{x} \in X^n} \sum_{\bar{k} \in K^{n+1}} \alpha(\bar{x}, \bar{k}) \left(\log p_1^*(k_0, x_1, k_1) + \sum_{i=2}^n \log \frac{p_i^*(k_{i-1}, x_i, k_i)}{\sum_{x \in X} \sum_{k \in K} p_i^*(k_{i-1}, x, k)} \right) \\ & \geq \sum_{\bar{x} \in X^n} \sum_{\bar{k} \in K^{n+1}} \alpha(\bar{x}, \bar{k}) \left(\log p_1(k_0, x_1, k_1) + \sum_{i=2}^n \log \frac{p_i(k_{i-1}, x_i, k_i)}{\sum_{x \in X} \sum_{k \in K} p_i(k_{i-1}, x, k)} \right). \end{aligned}$$

▲

Доказательство. Рассмотрим соотношение между суммами

$$\sum_{\bar{x} \in X^n} \sum_{\bar{k} \in K^{n+1}} \alpha(\bar{x}, \bar{k}) \log p_1^*(k_0, x_1, k_1) \quad \text{и} \quad \sum_{\bar{x} \in X^n} \sum_{\bar{k} \in K^{n+1}} \alpha(\bar{x}, \bar{k}) \log p_1(k_0, x_1, k_1). \quad (8.79)$$

Для первой суммы справедливо

$$\begin{aligned} & \sum_{\bar{x} \in X^n} \sum_{\bar{k} \in K^{n+1}} \alpha(\bar{x}, \bar{k}) \log p_1^*(k_0, x_1, k_1) \\ & = \sum_{x_1 \in X} \sum_{\bar{x} \in X_1(x_1)} \sum_{k_0 \in K} \sum_{k_1 \in K} \sum_{\bar{k} \in K_1(k_0, k_1)} \alpha(\bar{x}, \bar{k}) \log p_1^*(k_0, x_1, k_1) \\ & = \sum_{k_0 \in K} \sum_{x_1 \in X} \sum_{k_1 \in K} \left(\sum_{\bar{x} \in X_1(x_1)} \sum_{\bar{k} \in K_1(k_0, k_1)} \alpha(\bar{x}, \bar{k}) \right) \log p_1^*(k_0, x_1, k_1) \\ & = \sum_{k_0 \in K} \sum_{x_1 \in X} \sum_{k_1 \in K} p_1^*(k_0, x_1, k_1) \log p_1^*(k_0, x_1, k_1). \end{aligned} \quad (8.80)$$

Подобным образом можно показать, что вторая сумма в (8.79) равна

$$\sum_{k_0 \in K} \sum_{x_1 \in X} \sum_{k_1 \in K} p_1^*(k_0, x_1, k_1) \log p_1(k_0, x_1, k_1). \quad (8.81)$$

Поскольку обе суммы

$$\sum_{k_0 \in K} \sum_{x_1 \in X} \sum_{k_1 \in K} p_1^*(k_0, x_1, k_1) \text{ и } \sum_{k_0 \in K} \sum_{x_1 \in X} \sum_{k_1 \in K} p_1(k_0, x_1, k_1)$$

равны 1, то на основании леммы 6.1 мы утверждаем, что (8.80) не меньше, чем (8.81), и таким образом,

$$\sum_{\bar{x} \in X^n} \sum_{\bar{k} \in K^{n+1}} \alpha(\bar{x}, \bar{k}) \log p_1^*(k_0, x_1, k_1) \geq \sum_{\bar{x} \in X^n} \sum_{\bar{k} \in K^{n+1}} \alpha(\bar{x}, \bar{k}) \log p_1(k_0, x_1, k_1). \quad (8.82)$$

Выясним теперь соотношение сумм

$$\sum_{\bar{x} \in X^n} \sum_{\bar{k} \in K^{n+1}} \alpha(\bar{x}, \bar{k}) \log \frac{p_i^*(k_{i-1}, x_i, k_i)}{\sum_{x \in X} \sum_{k \in K} p_i^*(k_{i-1}, x, k)}$$

и

$$\sum_{\bar{x} \in X^n} \sum_{\bar{k} \in K^{n+1}} \alpha(\bar{x}, \bar{k}) \log \frac{p_i(k_{i-1}, x_i, k_i)}{\sum_{x \in X} \sum_{k \in K} p_i(k_{i-1}, x, k)}. \quad (8.83)$$

Первая сумма равна

$$\begin{aligned} & \sum_{\bar{x} \in X^n} \sum_{\bar{k} \in K^{n+1}} \alpha(\bar{x}, \bar{k}) \log \frac{p_i^*(k_{i-1}, x_i, k_i)}{\sum_{x \in X} \sum_{k \in K} p_i^*(k_{i-1}, x, k)} \\ &= \sum_{k_{i-1} \in K} \sum_{x_i \in X} \sum_{k_i \in K} \sum_{\bar{x} \in X_i(x_i)} \sum_{\bar{k} \in K_i(k_{i-1}, k_i)} \alpha(\bar{x}, \bar{k}) \log \frac{p_i^*(k_{i-1}, x_i, k_i)}{\sum_{x \in X} \sum_{k \in K} p_i^*(k_{i-1}, x, k)} \\ &= \sum_{k_{i-1} \in K} \sum_{x_i \in X} \sum_{k_i \in K} \left(\sum_{\bar{x} \in X_i(x_i)} \sum_{\bar{k} \in K_i(k_{i-1}, k_i)} \alpha(\bar{x}, \bar{k}) \right) \log \frac{\alpha_i(k_{i-1}, x_i, k_i)}{\sum_{x \in X} \sum_{k \in K} \alpha_i(k_{i-1}, x, k)} \\ &= \sum_{k_{i-1} \in K} \sum_{x_i \in X} \sum_{k_i \in K} \alpha_i(k_{i-1}, x_i, k_i) \log \frac{\alpha_i(k_{i-1}, x_i, k_i)}{\sum_{x \in X} \sum_{k \in K} \alpha_i(k_{i-1}, x, k)}. \quad (8.84) \end{aligned}$$

На основании подобных рассуждений мы утверждаем, что вторая сумма в (8.83) равна

$$\sum_{k_{i-1} \in K} \sum_{x_i \in X} \sum_{k_i \in K} \alpha_i(k_{i-1}, x_i, k_i) \log \frac{p_i(k_{i-1}, x_i, k_i)}{\sum_{x \in X} \sum_{k \in K} p_i(k_{i-1}, x, k)}. \quad (8.85)$$

Поскольку сумма $\sum_{x_i \in X} \sum_{k_i \in K} \frac{p_i(k_{i-1}, x_i, k_i)}{\sum_{x \in X} \sum_{k \in K} p_i(k_{i-1}, x, k)}$ равна 1 при любом значении k_{i-1} , из леммы 6.1 следует, что неравенство

$$\begin{aligned} & \sum_{x_i \in X} \sum_{k_i \in K} \alpha_i(k_{i-1}, x_i, k_i) \log \frac{\alpha_i(k_{i-1}, x_i, k_i)}{\sum_{x \in X} \sum_{k \in K} \alpha_i(k_{i-1}, x, k)} \\ & \geq \sum_{x_i \in X} \sum_{k_i \in K} \alpha_i(k_{i-1}, x_i, k_i) \log \frac{p_i(k_{i-1}, x_i, k_i)}{\sum_{x \in X} \sum_{k \in K} p_i(k_{i-1}, x, k)} \end{aligned}$$

справедливо при любом значении k_{i-1} . Суммируя полученное неравенство по всем значениям k_{i-1} , получаем неравенство

$$\begin{aligned} & \sum_{k_{i-1} \in K} \sum_{x_i \in X} \sum_{k_i \in K} \alpha_i(k_{i-1}, x_i, k_i) \log \frac{\alpha_i(k_{i-1}, x_i, k_i)}{\sum_{x \in X} \sum_{k \in K} \alpha_i(k_{i-1}, x, k)} \\ & \geq \sum_{k_{i-1} \in K} \sum_{x_i \in X} \sum_{k_i \in K} \alpha_i(k_{i-1}, x_i, k_i) \log \frac{p_i(k_{i-1}, x_i, k_i)}{\sum_{x \in X} \sum_{k \in K} p_i(k_{i-1}, x, k)}, \end{aligned}$$

которое совместно с доказанными (8.84) и (8.85) дает справедливость неравенства

$$\begin{aligned} & \sum_{\bar{x} \in X^n} \sum_{\bar{k} \in K^{n+1}} \alpha(\bar{x}, \bar{k}) \log \frac{p_i^*(k_{i-1}, x_i, k_i)}{\sum_{x \in X} \sum_{k \in K} p_i^*(k_{i-1}, x, k)} \\ & \geq \sum_{\bar{x} \in X^n} \sum_{\bar{k} \in K^{n+1}} \alpha(\bar{x}, \bar{k}) \log \frac{p_i(k_{i-1}, x_i, k_i)}{\sum_{x \in X} \sum_{k \in K} p_i(k_{i-1}, x, k)} \end{aligned}$$

для любого $i = 2, 3, \dots, n$. Суммируя это неравенство по этим значениям i , получаем

$$\begin{aligned} & \sum_{\bar{x} \in X^n} \sum_{\bar{k} \in K^{n+1}} \alpha(\bar{x}, \bar{k}) \sum_{i=2}^n \log \frac{p_i^*(k_{i-1}, x_i, k_i)}{\sum_{x \in X} \sum_{k \in K} p_i^*(k_{i-1}, x, k)} \\ & \geq \sum_{\bar{x} \in X^n} \sum_{\bar{k} \in K^{n+1}} \alpha(\bar{x}, \bar{k}) \sum_{i=2}^n \log \frac{p_i(k_{i-1}, x_i, k_i)}{\sum_{x \in X} \sum_{k \in K} p_i(k_{i-1}, x, k)}. \end{aligned} \quad (8.86)$$

Из неравенства (8.86) и ранее доказанного неравенства (8.82) очевидным образом следует неравенство, которое следовало доказать. ■

Если известно, что статистическая модель объекта является марковской, теорема 8.1 доказывает правильность следующего алгоритма максимально правдоподобного оценивания модели.

Algorithm 8.1 Наиболее правдоподобное оценивание марковской модели

1. Результат экспериментального исследования модели, первоначально представленного в виде выборки $((\bar{x}^j, \bar{k}^j), j = 1, 2, \dots, n)$, $\bar{x}^j \in X^n$, $\bar{k}^j \in K^{n+1}$,

представить в виде функции $\alpha: X^n \times K^{n+1} \rightarrow \mathbb{R}$, значениями $\alpha(\bar{x}, \bar{k})$, $\bar{x} \in X^n$, $\bar{k} \in K^{n+1}$, которой являются относительные частоты этих пар в эксперименте.

2. Для каждого $i = 1, 2, \dots, n$ и каждой тройки $x' \in X$, $k \in K$ и $k' \in K$ вычислить вероятности $p_i(k', x, k)$ как суммы

$$p_i(k', x, k) = \sum_{\bar{x} \in X_i(x)} \sum_{\bar{k} \in K_i(k', k)} \alpha(\bar{x}, \bar{k}), \quad (8.87)$$

которые обозначают частоту, с которой в эксперименте происходило событие: i -ый элемент последовательности \bar{x} равен x и в то же время $(i-1)$ -ый и i -ый элементы последовательности \bar{k} равны k' и k соответственно.

3. Для любой предъявленной пары последовательностей $\bar{x} \in X^n$ и $\bar{k} \in K^{n+1}$ вероятность этой пары вычислять по формуле

$$p(\bar{x}, \bar{k}) = p_1(k_0, x_1, k_1) \prod_{i=2}^n \frac{p_i(k_{i-1}, x_i, k_i)}{\sum_{x \in X} \sum_{k \in K} p_i(k_{i-1}, x, k)} \quad (8.88)$$

с использованием полученных в п.2 чисел $p_i(k', x, k)$

Доказанная теорема 8.1 утверждает, что вероятность экспериментально наблюдаемой выборки $((\bar{x}^j, \bar{k}^j), j = 1, 2, \dots, l)$ в полученной таким образом модели будет не меньше, чем в любой другой марковской модели.

При дальнейшем изложении окажется удобным трактовать алгоритм 8.1, то есть формулы (8.87) и (8.88), как преобразование функции $\alpha: X^n \times K^{n+1} \rightarrow \mathbb{R}$ в функцию $p: X^n \times K^{n+1} \rightarrow \mathbb{R}$. Это - преобразование распределения вероятностей α , которое может быть любым, не обязательно марковским, в распределение вероятностей p , которое обязательно марковское. Распределение вероятностей p , полученное на основе распределения вероятностей α , будет называться марковской аппроксимацией α и обозначаться α^M . Индекс M в обозначении α^M следует понимать как оператор, преобразующий функцию α в функцию $p = \alpha^M$. Поэтому запись, например, $(\alpha + \beta)^M$ будет обозначать марковскую аппроксимацию суммы функций α и β , или запись $\alpha^M(x)$ будет обозначать значение в точке x функции, являющейся марковской аппроксимацией функции α , и т.п.

Из определения марковской аппроксимации немедленно следует

$$\sum_{\bar{x} \in X^n} \sum_{\bar{k} \in K^{n+1}} \alpha(\bar{x}, \bar{k}) \log \alpha^M(\bar{x}, \bar{k}) \geq \sum_{\bar{x} \in X^n} \sum_{\bar{k} \in K^{n+1}} \alpha(\bar{x}, \bar{k}) \log p(\bar{x}, \bar{k})$$

где $p: X^n \times K^{n+1} \rightarrow \mathbb{R}$ - любое распределение вероятностей марковского вида.

Мы еще раз напомним, что хотя понятие марковской аппроксимации, определенное с помощью (8.87) и (8.88), представляется неконструктивным из-за многомерной суммы в правой части (8.87), на самом деле вычисление (8.87) - это совершенно естественная и легко реализуемая обработка экспериментально наблюдаемой выборки. Сложность этой обработки пропорциональна объему l выборки, так как именно столько ненулевых слагаемых

присутствует в правой части (8.87). В алгоритмах для решения последующих задач марковская аппроксимация будет использоваться в качестве элементарной повторяющейся операции и поэтому важно сейчас осознать ее естественность и вычислительную реализуемость.

8.9 Минимаксное оценивание марковской модели

8.9.1 Формулировка алгоритма и его свойств

В этом подразделе мы приводим без доказательства обзор основных результатов анализа задачи минимаксного оценивания. Обоснование этих результатов приводится в следующих подразделах. Основной результат состоит в том, что *минимаксное оценивание модели сводится к последовательности наиболее правдоподобных оцениваний, то есть к построению шаг за шагом определенных марковских аппроксимаций*.

Исходным при минимаксном оценивании является конечное обучающее множество L , состоящее из пар (\bar{x}, \bar{k}) , которые представляют типичное (достаточно вероятное) поведение распознаваемого объекта. Задача состоит в поиске совокупности $P^* =$ функций $(p_1^*, p_2^*, \dots, p_n^*)$ вида $p_i^*: K \times X \times K \rightarrow \mathbb{R}$,

$$P^* = (p_1^*, p_2^*, \dots, p_n^*) \\ = \operatorname{argmax}_{p_1, \dots, p_n} \min_{(\bar{x}, \bar{k}) \in L} \left(\log p_1(k_0, x_1, k_1) + \sum_{i=2}^n \log \frac{p_i(k_{i-1}, x_i, k_i)}{\sum_{x' \in X} \sum_{k' \in K} p_i(k_{i-1}, x', k')} \right).$$

Мы определим алгоритм, который решает задачу для любой заранее заданной точности $\varepsilon > 0$.

Алгоритм меняет шаг за шагом целые числа $n(\bar{x}, \bar{k})$ для каждой пары последовательностей $\bar{x} = (x_1, \dots, x_n)$ и $\bar{k} = (k_0, k_1, \dots, k_n)$ из обучающего множества L , а также числа $n_i(k', x', k'')$ для каждого $i = 1, 2, \dots, n$ и каждой тройки $k' \in K, x' \in X, k'' \in K$. На основании чисел $n(\bar{x}, \bar{k})$ и $n_i(k', x', k'')$ вычисляются текущие значения вероятностей $\alpha(\bar{x}, \bar{k})$ и $p_i(k', x', k'')$ следующим образом,

$$\alpha(\bar{x}, \bar{k}) = \frac{n(\bar{x}, \bar{k})}{\sum_{(\bar{x}, \bar{k}) \in L} n(\bar{x}, \bar{k})}, \quad p_i(k', x', k'') = \frac{n_i(k', x', k'')}{\sum_{k' \in K} \sum_{x' \in X} \sum_{k'' \in K} n_i(k', x', k'')}.$$

На каждом шаге алгоритм вычисляет марковскую аппроксимацию текущего распределения α . Начальные значения чисел $n^1(\bar{x}, \bar{k})$ и $n_i^1(x', k', x'')$ могут быть любыми(!!!). Для определенности примем, что $n^1(\bar{x}, \bar{k}) = 1$ для любой пары (\bar{x}, \bar{k}) из обучающего множества L . Для тройки k', x, k начальные значения $n_i^1(k', x, k)$ выбираются равными количеству таких пар (\bar{x}, \bar{k}) в множестве L , что $x_i = x, k_{i-1} = k', k_i = k$.

Пусть к началу t -ого шага получены числа $n^t(\bar{x}, \bar{k})$, $(\bar{x}, \bar{k}) \in L$, и числа $n_i^t(k', x', k'')$, $i = 1, 2, \dots, n$. Новые значения этих чисел вычисляются по следующим правилам:

Algorithm 8.2 Минимаксное оценивание марковской модели

1. Для $i = 1, 2, \dots, n$, $k' \in K$, $x' \in X$, $k'' \in K$, вычисляются вероятности

$$\alpha^t(\bar{x}, \bar{k}) = \frac{n^t(\bar{x}, \bar{k})}{\sum_{(\bar{x}, \bar{k}) \in L} n^t(\bar{x}, \bar{k})}, \quad (\bar{x}, \bar{k}) \in L,$$

$$p_i^t(k', x', k'') = \frac{n_i^t(k', x', k'')}{\sum_{k' \in K} \sum_{x' \in X} \sum_{k'' \in K} n_i^t(k', x', k'')}.$$

2. Вычисляются вероятности $p^t(\bar{x}, \bar{k})$ по формуле (8.88), то есть,

$$p^t(\bar{x}, \bar{k}) = p_1^t(k_0, x_1, k_1) \prod_{i=2}^n \frac{p_i^t(k_{i-1}, x_i, k_i)}{\sum_{x' \in X} \sum_{k' \in K} p_i^t(k_{i-1}, x', k')}, \quad (\bar{x}, \bar{k}) \in L. \quad (8.89)$$

3. Проверяется, выполняется ли неравенство

$$\sum_{(\bar{x}, \bar{k}) \in L} \alpha^t(\bar{x}, \bar{k}) \log p^t(\bar{x}, \bar{k}) - \min_{(\bar{x}, \bar{k}) \in L} \log p^t(\bar{x}, \bar{k}) \leq \varepsilon. \quad (8.90)$$

4. Если неравенство (8.90) выполняется, алгоритм заканчивает свою работу, а текущие значения $p_i^t(k', x', k'')$ объявляются ε -решением задачи.

5. Если неравенство (8.90) не выполняется, то продолжают следующие вычисления.

(а) Выбирается любая пара $(\bar{x}^*, \bar{k}^*) \in L$, для которой

$$p^t(\bar{x}^*, \bar{k}^*) = \min_{(\bar{x}, \bar{k}) \in L} p^t(\bar{x}, \bar{k}).$$

(б) Вычисляются новые значения чисел $n(\bar{x}, \bar{k})$ и $n_i(k_{i-1}, x_i, k_i)$,

$$\begin{aligned} n^{t+1}(\bar{x}, \bar{k}) &= n^t(\bar{x}, \bar{k}) + 1, & \text{если } \bar{x} = \bar{x}^*, \bar{k} = \bar{k}^*, \\ n^{t+1}(\bar{x}, \bar{k}) &= n^t(\bar{x}, \bar{k}), & \text{если } (\bar{x}, \bar{k}) \neq (\bar{x}^*, \bar{k}^*), \\ n_i^{t+1}(k', x', k'') &= n_i^t(k' x', k'') + 1, & \text{если } k_{i-1}^* = k', x_i^* = x', k_i^* = k'', \\ n_i^{t+1}(k', x', k'') &= n_i^t(k' x', k''), & \text{если } (k_{i-1}^*, x_i^*, k_i^*) \neq (k', x', k''). \end{aligned}$$

6. Переходят к $(t+1)$ -ому шагу алгоритма, начиная с п. 1.

Для этого алгоритма справедливы следующие две теоремы.

Теорема 8.2 О сходимости алгоритма за конечное количество шагов.

Для любой заранее заданной величины ε алгоритм 8.2 приходит в состояние, при котором выполняется неравенство (8.90), и выходит на остатов.

Пусть $F(P)$ обозначает число

$$F(P) = \min_{(x, k) \in L} \left(\log p_1(k_0, x_1, k_1) + \sum_{i=2}^n \log \frac{p_i(k_{i-1}, x_i, k_i)}{\sum_{x' \in X} \sum_{k' \in K} p_i(k_{i-1}, x', k')} \right),$$

которое должно максимизироваться в соответствии с постановкой задачи.

Теорема 8.3 Об ε -решении задачи. Пусть P' - модель, построенная алгоритмом 8.2 в момент его останова, а P^* - решение минимаксной задачи оценивания. Тогда

$$F(P^*) - F(P') \leq \varepsilon. \quad \blacktriangle$$

Оставшаяся часть данного раздела 8.9 посвящена доказательству свойств алгоритма 8.2, равно как и важных свойств задачи в целом. Уже по приведенному алгоритму видно, что минимаксное оценивание статистической модели сводится к последовательности наиболее правдоподобных оценок. Такое сведение рассматриваемой задачи к уже рассмотренной возможно не только в рамках марковской модели, но и для значительно более широкого класса моделей. В силу этой общности последующие результаты выходят далеко за пределы проблематики распознавания марковских процессов. Далее мы покажем, что, вне зависимости от класса моделей, задача минимаксного оценивания сводится к специальным задачам выпуклой оптимизации. Следовательно, для минимаксного оценивания может применяться не только приведенный выше алгоритм, но и весь арсенал хорошо исследованных методов выпуклой оптимизации.

Последующий анализ опирается на некоторые известные математические результаты, которые мы приведем без доказательства.

Пусть X - выпуклое подмножество линейного пространства, а $f: X \rightarrow \mathbb{R}$ - вещественнозначная функция, определенная на этом подмножестве. Выпуклость функции определяется следующими тремя эквивалентными способами.

1. Функция f выпукла, если множество

$$\{(x, y) \mid y \geq f(x)\}$$

выпукло.

2. Функция f выпукла, если для любых $x_1 \in X$, $x_2 \in X$ и α , $0 \leq \alpha \leq 1$, справедливо неравенство,

$$f(\alpha x_1 + (1 - \alpha) x_2) \leq \alpha f(x_1) + (1 - \alpha) f(x_2).$$

3. Функция f выпукла, если для любой точки $x_0 \in X$ существует такая линейная функция $L_{x_0}: X \rightarrow \mathbb{R}$, что неравенство

$$L_{x_0}(x - x_0) \leq f(x) - f(x_0) \quad (8.91)$$

выполняется при любом $x \in X$.

Пусть вектор $g(x_0) \in X$ соответствует линейной функции L_{x_0} в том смысле, что $L_{x_0}(x)$ есть скалярное произведение $\langle g(x_0), x \rangle$. Такой вектор $g(x_0)$ называется *обобщенным градиентом* f в точке x_0 . Известно, что обобщенный градиент существует в любой точке выпуклой функции. Если функция f

такова, что в точке x_0 в соответствии с определением (8.91) линейная функция L_{x_0} , а следовательно, и обобщенный градиент $g(x_0)$, определяются однозначно, то функция f называется дифференцируемой (или гладкой) в точке x_0 , а обобщенный градиент $g(x_0)$ называется просто градиентом.

Любая выпуклая функция на конечномерном множестве *непрерывна*. Это значит, что разность $f(x) - f(x_0)$ стремится к нулю, когда x стремится к x_0 . Линейная функция есть частный случай выпуклой функции и поэтому $\lim_{x \rightarrow x_0} g(x_0)(x - x_0) = 0$.

Если же функция f к тому же гладка и $g(x_0) \neq 0$, то

$$\lim_{x \rightarrow x_0} \frac{f(x) - f(x_0)}{g(x_0)(x - x_0)} = 1.$$

Эта ситуация есть частный случай ситуации, в которой говорят о бесконечно малых одинакового порядка. Напомним точную формулировку этого понятия. Пусть $u_1, u_2, \dots, u_i, \dots$ и $v_1, v_2, \dots, v_i, \dots$ - две переменные величины, бесконечно малые в том смысле, что $\lim_{i \rightarrow \infty} u_i = \lim_{i \rightarrow \infty} v_i = 0$. Эти две бесконечно малые величины называют *малыми одного порядка*, если существует предел

$$\lim_{i \rightarrow \infty} \frac{u_i}{v_i},$$

который не равен нулю и не равен бесконечности. Бесконечно малые одного порядка обладают следующим важным свойством.

Пусть u_i и v_i , $i = 1, 2, \dots, \infty$, - две бесконечно малые одного порядка. В таком случае, если последовательность чисел $\sum_{i=1}^n u_i$, $n = 1, 2, \dots, \infty$, сходится к конечному пределу, то последовательность чисел $\sum_{i=1}^n v_i$, $n = 1, 2, \dots, \infty$, также сходится к конечному пределу (возможно, другому). И конечно, если $\lim_{n \rightarrow \infty} \sum_{i=1}^n u_i = \infty$, то справедливо также, что $\lim_{n \rightarrow \infty} \sum_{i=1}^n v_i = \infty$.

Свойства минимаксного оценивания, которые делают возможной его конструктивную реализацию, имеют место не только для марковских моделей. Более того, их анализ и обоснование легче выполнить в общем виде, так как в этом случае они не заслонены дополнительными подробностями. Поэтому в следующем подразделе мы исследуем задачу минимаксного оценивания в общем виде, а полученные результаты затем почти без труда применим для минимаксного оценивания марковских моделей.

8.9.2 Анализ задачи минимаксного оценивания

Задача минимаксного оценивания в общем виде уже была нами сформулирована в лекции 3. Для ее анализа переформулируем ее в несколько ином виде.

Пусть X - множество, для которого определен класс \mathcal{P} функций вида $p: X \rightarrow \mathbb{R}$. Пусть $(x^j, j = 1, \dots, n)$ - мультимножество элементов из X . Задача состоит в отыскании функции p^* в классе \mathcal{P} , которая максимизирует

число $\min_j p(x^j)$, то есть

$$p^* = \operatorname{argmax}_{p \in \mathcal{P}} \min_j p(x^j). \quad (8.92)$$

Так как решение задачи, то есть функция p^* , не зависит от того, сколько раз тот или иной элемент входит в мультимножество $(x^j, j = 1, \dots, n)$, а зависит лишь от того, входит ли он по крайней мере один раз, входные данные можно рассматривать как конечное множество $L \subset X$, а не как мультимножество $(x^j, j = 1, \dots, n)$. Задача (8.92) принимает вид

$$p^* = \operatorname{argmax}_{p \in \mathcal{P}} \min_{x \in L} p(x)$$

или, что то же самое,

$$p^* = \operatorname{argmax}_{p \in \mathcal{P}} \min_{x \in L} \log p(x). \quad (8.93)$$

Напомним задачу наиболее правдоподобного оценивания модели p , то есть поиска

$$p^* = \operatorname{argmax}_{p \in \mathcal{P}} \sum_{x \in L} \alpha(x) \log p(x). \quad (8.94)$$

Это значит, что следует найти функцию p^* , при которой максимизируется вероятность заданной выборки. В выражении (8.94) эта выборка представлена множеством L значений, попавших в выборку, и совокупностью чисел $\alpha(x)$, указывающих, сколько раз тот или иной элемент присутствует в выборке. Искомая функция p^* в соответствии с (8.94) зависит от коэффициентов $\alpha(x)$, и поэтому алгоритм вычисления (8.94) можно трактовать, как оператор, преобразующий функцию $\alpha: L \rightarrow \mathbb{R}$ в функцию $\alpha^M: X \rightarrow \mathbb{R}$ так, что

$$\alpha^M = \operatorname{argmax}_{p \in \mathcal{P}} \sum_{x \in L} \alpha(x) \log p(x).$$

В таком случае $\max_{p \in \mathcal{P}} \sum_{x \in L} \alpha(x) \log p(x)$ - это число

$$\sum_{x \in L} \alpha(x) \log \alpha^M(x),$$

которое зависит только от $\alpha: L \rightarrow \mathbb{R}$ и будет обозначаться $Q(\alpha)$. Для числа $Q(\alpha)$ по его определению справедливо неравенство

$$Q(\alpha) = \sum_{x \in X} \alpha(x) \log \alpha^M(x) \geq \sum_{x \in X} \alpha(x) \log p(x),$$

которое выполняется для любой функции $p \in \mathcal{P}$. Как и раньше, в разделе 8.8, функция $\alpha^M: X \rightarrow \mathbb{R}$ будет называться аппроксимацией функции $\alpha: L \rightarrow \mathbb{R}$ на классе \mathcal{P} или просто аппроксимацией. Когда \mathcal{P} есть класс марковских моделей, α^M есть марковская аппроксимация функции α .

Принципы наиболее правдоподобного оценивания (8.94) сейчас хорошо исследованы вплоть до разработанных программ для многих наиболее излюбленных классов \mathcal{P} . В частности, для класса марковских моделей наиболее правдоподобное оценивание исследовано в предыдущем разделе 8.8. Хотя минимаксное оценивание имеет определенные преимущества перед наиболее правдоподобным оцениванием, этот тип оценивания изучен значительно меньше и поэтому эти задачи представляются и более трудными. В этом отношении важным является вывод, что минимаксное оценивание можно свести к наиболее правдоподобному оцениванию в следующем смысле.

Допустим, что в нашем распоряжении имеется программа наиболее правдоподобного оценивания (8.94). Тогда можно формально, определенным стандартным способом, построить и программу минимаксного оценивания (8.93). Она будет включать программу решения (8.94) в качестве подпрограммы. Этот трюк возможен в силу того, что мы докажем, что вне зависимости от множеств X и \mathcal{P} решение задачи (8.93) обязательно имеет вид α^M для некоторой функции α . Или, говоря иными словами, минимаксная модель обязательно совпадает с наиболее правдоподобной моделью при некоторых коэффициентах $\alpha(x)$, $x \in L$. Коэффициенты $\alpha(x)$, при которых эти две задачи становятся идентичными, являются экстремальными в том смысле, что они минимизируют число $Q(\alpha) = \sum_{x \in L} \alpha(x) \log \alpha^M(x)$. Другой важный результат состоит в том, что, вне зависимости от множеств X , L и \mathcal{P} , функция $Q(\alpha)$ всегда выпукла. Следовательно, ее минимум можно находить различными известными сейчас методами выпуклой оптимизации. Сформулируем эти утверждения точно и докажем их.

Лемма 8.1 *Оценка сверху для $\min_{x \in L} \log p(x)$. Пусть p - любая функция $X \rightarrow \mathbb{R}$ из класса \mathcal{P} , а $\alpha: L \rightarrow \mathbb{R}$ - любая функция, для которой выполняется*

$$\left. \begin{aligned} \sum_{x \in L} \alpha(x) &= 1, \\ \alpha(x) &\geq 0, \quad x \in L. \end{aligned} \right\} \quad (8.95)$$

В таком случае

$$\min_{x \in L} \log p(x) \leq \sum_{x \in L} \alpha(x) \log \alpha^M(x). \quad (8.96) \blacktriangle$$

Доказательство. Неравенство (8.96) следует из очевидных неравенств

$$\min_{x \in L} \log p(x) \leq \sum_{x \in L} \alpha(x) \log p(x), \quad (8.97)$$

$$\sum_{x \in L} \alpha(x) \log p(x) \leq \sum_{x \in L} \alpha(x) \log \alpha^M(x). \quad (8.98)$$

Неравенство (8.97) следует из условия (8.95). Число в правой части (8.97) есть выпуклая комбинация чисел $\log p(x)$, а число в левой части (8.97) - наименьшее из этих чисел. Конечно, наименьшее число в некоторой совокупности чисел не превосходит выпуклую комбинацию чисел из этой совокупности.

Неравенство (8.98) следует из определения функции α^M . Из неравенств (8.97) и (8.98) мы получаем (8.96). ■

Обозначим A множество функций $\alpha: L \rightarrow \mathbb{R}$, удовлетворяющих (8.95).

Лемма 8.2 **О выпуклости функции Q .** *Функция*

$Q(\alpha) = \sum_{x \in L} \alpha(x) \log \alpha^M(x)$ *выпукла на множестве A .* ▲

Доказательство. Обозначим $\text{lin } A$ линейное замыкание множества A . Для любой точки $\alpha_0 \in A$, то есть для любой функции $\alpha_0: L \rightarrow \mathbb{R}$, определим линейную функцию $G_{\alpha_0}: \text{lin } A \rightarrow \mathbb{R}$

$$G_{\alpha_0}(\alpha) = \sum_{x \in L} \alpha(x) \log \alpha_0^M(x).$$

Для функции G_{α_0} выполняется

$$G_{\alpha_0}(\alpha_0) = \sum_{x \in L} \alpha_0(x) \log \alpha_0^M(x),$$

а для любого $\alpha \in A$ выполняется

$$G_{\alpha_0}(\alpha) \leq \sum_{x \in L} \alpha(x) \log \alpha^M(x).$$

Функция G_{α_0} является именно той функцией, существование которой в соответствии с (8.91) определяет выпуклость функции $Q(\alpha)$. Действительно, на множестве A выполняется следующее неравенство

$$G_{\alpha_0}(\alpha - \alpha_0) \leq \sum_{x \in L} \alpha(x) \log \alpha^M(x) - \sum_{x \in L} \alpha_0(x) \log \alpha_0^M(x). \quad \blacksquare$$

Теорема 8.4 **Необходимые и достаточные условия минимаксной модели.**

1. Если

$$\min_{x \in L} \log \alpha^{*M}(x) = \sum_{x \in L} \alpha^*(x) \log \alpha^{*M}(x), \quad (8.99)$$

то

$$\alpha^{*M} = \operatorname{argmax}_{p \in \mathcal{P}} \min_{x \in L} \log p(x) \quad (8.100)$$

и

$$\alpha^* = \operatorname{argmin}_{\alpha \in A} \sum_{x \in L} \alpha(x) \log \alpha^M(x). \quad (8.101)$$

2. Если функция $Q(\alpha) = \sum_{x \in L} \alpha(x) \log \alpha^M(x)$ гладкая и выполняется

$$\alpha^* = \operatorname{argmin}_{\alpha \in A} \sum_{x \in L} \alpha(x) \log \alpha^M(x), \quad (8.102)$$

то

$$\min_{x \in L} \log \alpha^{*M}(x) = \sum_{x \in L} \alpha^*(x) \log \alpha^{*M}(x). \quad (8.103) \quad \blacktriangle$$

Доказательство. Первая часть теоремы 8.4 доказывается достаточно легко. В силу леммы 8.1 неравенство (8.96) справедливо для любых функций $p \in \mathcal{P}$ и $\alpha \in A$, следовательно, для любой функции $p \in \mathcal{P}$ и для той функции α^* , которая удовлетворяет (8.99). Поэтому можно написать неравенство

$$\min_{x \in L} \log p(x) \leq \sum_{x \in L} \alpha^*(x) \log \alpha^{*M}(x), \quad p \in \mathcal{P},$$

которое вместе с условием (8.99) приводит к неравенству

$$\min_{x \in L} \log p(x) \leq \min_{x \in L} \log \alpha^{*M}(x), \quad p \in \mathcal{P}.$$

Последнее неравенство совпадает с утверждением (8.100), только записанным в другой форме.

Запишем неравенство (8.96) для $p = \alpha^{*M}$ и любого $\alpha \in A$,

$$\min_{x \in L} \log \alpha^{*M}(x) \leq \sum_{x \in L} \alpha(x) \log \alpha^M(x), \quad \alpha \in A.$$

Это неравенство совместно с условием (8.99) приводит к неравенству

$$\sum_{x \in L} \alpha^*(x) \log \alpha^{*M}(x) \leq \sum_{x \in L} \alpha(x) \log \alpha^M(x), \quad \alpha \in A,$$

которое эквивалентно утверждению (8.101). Таким образом, первая часть теоремы 8.4 доказана.

Докажем вторую часть теоремы 8.4 от противного. Очевидным является неравенство

$$\min_{x \in L} \log \alpha^{*M}(x) \leq \sum_{x \in L} \alpha^*(x) \log \alpha^{*M}(x).$$

Предположим, что (8.103) не выполняется, то есть выполняется строгое неравенство

$$\min_{x \in L} \log \alpha^{*M}(x) < \sum_{x \in L} \alpha^*(x) \log \alpha^{*M}(x). \quad (8.104)$$

Докажем, что в этом случае существует такая функция $\alpha \in A$, что

$$\sum_{x \in \alpha} \alpha(x) \log \alpha^M(x) < \sum_{x \in \alpha} \alpha^*(x) \log \alpha^{*M}(x),$$

то есть, что (8.102) не выполняется.

Обозначим $x' = \operatorname{argmin}_{x \in L} \alpha^{*M}(x)$ и выберем функцию $\alpha' : L \rightarrow \mathbb{R}$ так, что $\alpha'(x') = 1$ и $\alpha'(x) = 0$ для всех $x \in L$, не равных x' . Мы можем записать

$$\min_{x \in L} \log \alpha^{*M}(x) = \sum_{x \in L} \alpha'(x) \log \alpha^{*M}(x)$$

и в силу (8.104)

$$\sum_{x \in L} (\alpha'(x) - \alpha^*(x)) \log \alpha^{*M}(x) < 0. \quad (8.105)$$

Исследуем поведение функции $Q(\alpha) = \sum_{x \in L} \alpha(x) \log \alpha^M(x)$ на отрезке, который соединяет точки α^* и α' , то есть зависимость числа $Q(\alpha^*(1 - \gamma) + \alpha' \gamma)$ от коэффициента γ . Зафиксировав точки α^* и α' , функцию $Q(\alpha^*(1 - \gamma) + \alpha' \gamma)$ можно рассматривать как функцию одной переменной γ . Производная этой функции по переменной γ в точке $\gamma = 0$ равна

$$\frac{dQ}{d\gamma} = \sum_{x \in L} (\alpha'(x) - \alpha^*(x)) \log \alpha^{*M}(x).$$

В силу (8.105) эта производная отрицательна. Это значит, что по крайней мере при малых значениях γ число $Q(\alpha^*(1 - \gamma) + \alpha' \gamma)$ строго меньше, чем $Q(\alpha^*)$, и таким образом,

$$\alpha^* \neq \underset{\alpha \in A}{\operatorname{argmin}} Q(\alpha).$$

Вторая часть теоремы 8.4 также доказана. \blacksquare

Доказанная теорема 8.4 указывает путь отыскания минимаксной модели $p \in \mathcal{P}$. Необходимо найти веса $\alpha^*(x)$, $x \in L$, минимизирующие число $Q(\alpha) = \sum_{x \in L} \alpha(x) \log \alpha^M(x)$. Из второй части теоремы 8.4 следует, что эти веса $\alpha^*(x)$ удовлетворяют (8.103). Из первой же части теоремы 8.4 следует, что решением минимаксной задачи (8.93) является аппроксимация $p^* = \alpha^{*M}$, то есть, наиболее правдоподобная оценка

$$p^* = \operatorname{argmax}_{p \in \mathcal{P}} \sum_{x \in L} \alpha^*(x) \log p(x).$$

Поскольку функция $Q(\alpha)$ выпукла, ее минимум можно находить различными путями. Однако здесь нет необходимости применять непременно стандартные методы, так как в доказательстве теоремы 8.4 уже содержатся определенные рекомендации по минимизации.

Если найдена пара $\alpha \in A$ и $p \in \mathcal{P}$, для которой выполняется

$$\left. \begin{aligned} \min_{x \in L} \log p(x) &= \sum_{x \in L} \alpha(x) \log p(x), \\ p &= \operatorname{argmax}_{p \in \mathcal{P}} \sum_{x \in L} \alpha(x) \log p(x), \end{aligned} \right\} \quad (8.106)$$

задача уже решена. Если же равенство (8.106) не выполняется, то можно видеть, как следует менять числа $\alpha(x)$. Следует увеличить вес $\alpha(x)$ элемента $x \in L$ из обучающего множества с наименьшей текущей вероятностью $p(x)$.

Выразим эти идеи точно в виде однозначно понимаемого алгоритма и затем приведем его обоснование.

Algorithm 8.3 Минимаксное оценивание модели в классе $p \in \mathcal{P}$

1. Пользователь определяет требуемую точность $\varepsilon > 0$ решения задачи. Устанавливаются числа $n(x) = 1$ для всех примеров $x \in L$ из обучающего множества L и устанавливается номер $t = 1$ итерации.

2. Вычисляются величины

$$\alpha^t(x) = \frac{n^t(x)}{\sum_x n^t(x)}.$$

3. Вычисляется наиболее правдоподобная модель

$$p^t = \operatorname{argmax}_{p \in \mathcal{P}} \sum_{x \in L} \alpha^t(x) \log p(x). \quad (8.107)$$

4. Если выполняется неравенство

$$\sum_{x \in L} \alpha^t(x) \log p^t(x) - \min_{x \in L} \log p^t(x) < \varepsilon, \quad (8.108)$$

алгоритм заканчивает свою работу и p^t объявляется решением задачи.

5. Если неравенство (8.108) не выполняется, то

(a) Находится

$$x' = \operatorname{argmin}_{x \in L} p^t(x).$$

(b) Определяются новые значения $n(x)$, $x \in L$,

$$\begin{aligned} n^{t+1}(x') &= n^t(x') + 1, \\ n^{t+1}(x) &= n^t(x), \quad x \in L, \quad x \neq x'. \end{aligned}$$

6. Выполняется очередная, $(t+1)$ -ая итерация алгоритма, начиная с пункта 2.

Условие останова алгоритма 8.3 несколько отличается от условия (8.106), которое в силу теоремы 8.4 гарантирует точный оптимум максимизируемой функции. Здесь применяется более слабое условие (8.108). Можно предполагать, что коль скоро неравенство (8.108) приблизительно совпадает с условием (8.106) точного решения задачи, то оно может служить условием приближенного решения задачи, точность которого увеличивается с уменьшением величины ε . Это предположение верно в силу следующей теоремы.

Теорема 8.5 О точности минимаксного оценивания. Если $\alpha \in A$ и $p \in \mathcal{P}$ удовлетворяют неравенство

$$\sum_{x \in L} \alpha(x) \log p(x) - \min_{x \in L} \log p(x) < \varepsilon, \quad (8.109)$$

причем

$$p = \operatorname{argmax}_{p' \in \mathcal{P}} \sum_{x \in L} \alpha(x) \log p'(x),$$

то

$$\min_{x \in L} \log p^*(x) - \min_{x \in L} \log p(x) < \varepsilon$$

где

$$p^* = \operatorname{argmax}_{p \in \mathcal{P}} \min_{x \in L} p(x). \quad (8.110)$$

▲

Доказательство. В силу леммы 8.1 неравенство (8.96) справедливо для любых $\alpha \in A$ и $p \in \mathcal{P}$. Следовательно, оно выполняется и для α , которое

удовлетворяет условию (8.109), и для p^* , которое удовлетворяет (8.110). Поэтому можно записать

$$\min_{x \in L} \log p^*(x) - \sum_{x \in L} \alpha(x) \log p(x) \leq 0. \quad (8.111)$$

Складывая (8.111) и (8.109), получаем

$$\min_{x \in L} \log p^*(x) - \min_{x \in L} \log p(x) < \varepsilon. \quad \blacksquare$$

Покажем теперь, что при любом положительном ε алгоритм 8.3 выходит на условие (8.108) останова и, таким образом, гарантирует решение задачи со сколь угодно малой (но ненулевой) погрешностью в том смысле, как это утверждает теорема 8.5. Это утверждение доказывается при дополнительных, хотя и не очень обременительных условиях.

Будем говорить, что обучающее множество L не противоречит классу \mathcal{P} , если существует такая модель $p \in \mathcal{P}$, что $p(x) \neq 0$ для любого $x \in L$. Понятно, что если L противоречит \mathcal{P} , то любая модель плоха, так как для любой модели $\min_{x \in L} p(x) = 0$. Конечно же, в таком классе моделей не стоит искать наилучшую. Из того факта, что L не противоречит \mathcal{P} , непосредственно следует, что функция $Q(\alpha) = \sum_{x \in L} \alpha(x) \log \alpha^M(x)$ на множестве A ограничена снизу.

Теорема 8.6 **О сходимости алгоритма минимаксного оценивания.** *Если множество $L \subset X$ не противоречит классу \mathcal{P} , а функция $Q(\alpha) = \sum_{x \in L} \alpha(x) \cdot \log \alpha^M(x)$ гладкая, то алгоритм 8.3 за конечное количество итераций достигает состояния, при котором выполняется условие (8.108) останова.* \blacktriangle

Доказательство.

1. Предположим, что теорема 8.6 не верна, и алгоритм 8.3 выполняет неограниченное количество итераций, причем на каждой из них выполняется условие

$$\sum_{x \in L} \alpha^t(x) \log p^t(x) - \min_{x \in L} \log p^t(x) \geq \varepsilon. \quad (8.112)$$

Докажем, что в этом случае при достаточно большом t величина $Q(\alpha) = \sum_{x \in L} \alpha^t(x) \log p^t(x)$ может стать меньше любого заранее заданного числа. Это доказательство выполним по следующей схеме

- (a) Будет доказано, что

$$\lim_{t \rightarrow \infty} |\alpha_t - \alpha_{t-1}| = 0. \quad (8.113)$$

- (b) Будет определена линейная функция $G_t: A \rightarrow \mathbb{R}$

$$G_t(\alpha) = \sum_{x \in L} \alpha(x) \log p^t(x)$$

которая находится с функцией $Q(\alpha) = \sum_{x \in L} \alpha(x) \log \alpha^M(x)$ в отношении

$$G_t(\alpha - \alpha^t) \leq Q(\alpha) - Q(\alpha^t), \quad \alpha \in A,$$

и соответствует, таким образом, градиенту функции $Q(\alpha)$ в точке α^t .

(с) Коль скоро будет доказано (8.113), будет доказано также, что

$$\lim_{t \rightarrow \infty} G_t(\alpha^{t+1} - \alpha^t) = 0,$$

$$\lim_{t \rightarrow \infty} (Q(\alpha^{t+1}) - Q(\alpha^t)) = 0,$$

так как функция G_t линейна на конечномерном пространстве и, следовательно, непрерывна, а функция Q выпукла на конечномерном пространстве, следовательно, тоже непрерывна.

(d) Поскольку функция Q по условию теоремы гладкая, то величины $G_t(\alpha^{t+1} - \alpha^t)$ и $Q(\alpha^{t+1}) - Q(\alpha^t)$ стремятся к нулю, как бесконечно малые одного порядка. Поэтому, если будет доказано, что

$$\sum_{t=1}^{\infty} G_t(\alpha^{t+1} - \alpha^t) = -\infty, \quad (8.114)$$

тем самым будет также доказано, что

$$\sum_{t=1}^{\infty} (Q(\alpha^{t+1}) - Q(\alpha^t)) = -\infty. \quad (8.115)$$

Число в левой части последнего равенства есть не что иное, как $\lim_{t \rightarrow \infty} (Q(\alpha^t) - Q(\alpha^1))$. Выражение (8.115) таким образом утверждает, что с ростом t величина $Q(\alpha^t)$ может стать меньше любой отрицательной величины. Это, однако, противоречило бы условию, что функция $Q(\alpha)$ ограничена снизу. Таким образом, теорема была бы доказана от противного.

В этой схеме доказательства необходимо доказать, что из условия (8.112) следует (8.113) и (8.114).

2. Обозначим $n^t = \sum_{x \in L} n^t(x)$, $x^t = \operatorname{argmin}_{x \in L} p^t(x)$, и α^{t+1} функцию $L \rightarrow \mathbb{R}$, такую, что $\alpha^{t+1}(x) = 1$ при $x = x^t$ и $\alpha^{t+1}(x) = 0$ при $x \neq x^t$. С использованием этих обозначений неравенство (8.112) принимает вид

$$\sum_{x \in L} (\alpha^t(x) - \alpha^{t+1}(x)) \log p^t(x) \geq \varepsilon, \quad (8.116)$$

и веса $\alpha^{t+1}(x)$ можно выразить как

$$\alpha^{t+1}(x) = \alpha^t(x) \frac{n^t}{n^t + 1} + \alpha^{t+1}(x^t) \frac{1}{n^t + 1}.$$

Разность весов α на двух последовательных итерациях равна

$$\alpha^{t+1}(x) - \alpha^t(x) = \frac{\alpha^{t+1}(x^t) - \alpha^t(x)}{n^t + 1}.$$

3. Число n^t на первой итерации равно $|L|$ и затем увеличивается на единицу на каждой итерации. Поэтому $n^t = |L| + t - 1$, и следовательно,

$$\alpha^{t+1}(x) - \alpha^t(x) = \frac{\alpha^{t+1}(x) - \alpha^t(x)}{|L| + t}. \quad (8.117)$$

Числитель в этом выражении ограничен, и поэтому

$$\lim_{t \rightarrow \infty} |\alpha^{t+1}(x) - \alpha^t(x)| = 0$$

для любого $x \in L$, и утверждение (8.113) доказано.

4. Разность $G_t(\alpha^{t+1} - \alpha^t)$ равна

$$G_t(\alpha^{t+1}) - G_t(\alpha^t) = \sum_{x \in L} (\alpha^{t+1}(x) - \alpha^t(x)) \log p^t(x).$$

В силу (8.117) выполняется

$$G_t(\alpha^{t+1}) - G_t(\alpha^t) = \sum_{x \in L} \frac{\alpha^{t+1}(x) - \alpha^t(x)}{|L| + t} \log p^t(x).$$

И наконец, в силу (8.116), эта разность отрицательна и, более того,

$$G_t(\alpha^{t+1}) - G_t(\alpha^t) \leq -\frac{\varepsilon}{|L| + t}.$$

5. Сумма $\sum_{t=1}^T G_t(\alpha^{t+1} - \alpha^t)$ не превосходит величину $-\varepsilon \sum_{t=1}^T \frac{1}{|L| + t}$, и следовательно, с ростом T может стать меньше любой отрицательной величины. Утверждение (8.114) доказано. ■

Таким образом, минимаксное оценивание может выполняться достаточно простым приведенным алгоритмом. Конечно, этот алгоритм может считаться простым только при условии, что имеется в распоряжении простой алгоритм наиболее правдоподобного оценивания. Если такой алгоритм программно реализован, то программа минимаксного оценивания строится чисто механически. Имеющуюся программу наиболее правдоподобного оценивания дополняют некоторыми операциями типа добавление единицы, организация цикла и условия выхода из него, которые настолько просты, что о них не стоит и говорить. Существенным является то, что эта добавляемая надстройка не зависит от характера множества X наблюдений и от класса моделей \mathcal{P} . Таким образом, через этот алгоритм обнаруживается тесная взаимосвязь двух обширных задач оценивания модели, которые до сих пор представлялись никак не связанными друг с другом. Описанный алгоритм представляет интерес именно в таком познавательном аспекте. Что касается чисто прагматических свойств алгоритма, то он, повидимому, один из многих возможных, пусть сейчас и неизвестных. В силу универсальности алгоритма ему можно отдавать предпочтение на первых этапах работы с моделью, которая еще мало исследована. По мере увеличения

знаний о конкретной модели приведенный алгоритм следует разумно модифицировать тем или иным способом или применять другие алгоритмы. Эти возможности открываются в силу следующих двух основных результатов, показанных в лекции.

- Минимаксное оценивание статистической модели объекта по обучающему множеству L совпадает с наиболее правдоподобным оцениванием по обучающему мультимножеству L^* , в котором каждый элемент $x \in L$ присутствует с определенным весом $\alpha(x)$.
- Веса $\alpha(x)$, $x \in L$, которые обеспечивают идентичность этих двух оцениваний, обеспечивают минимум вполне определенной выпуклой функции, и поэтому для их поиска могут применяться многие хорошо развитые методы выпуклой оптимизации.

8.9.3 Минимаксное оценивание марковской модели

Общий алгоритм минимаксного оценивания описан в подразделе 8.9.2 с точностью до операции

$$p^t(x) = \operatorname{argmax}_{p \in \mathcal{P}} \sum_{x \in L} \alpha^t(x) \log p(x),$$

которая должна быть той или иной в зависимости от класса \mathcal{P} моделей. Для случая, когда \mathcal{P} - это множество марковских моделей, эта операция определена формулами (8.87) и (8.88) и названа марковской аппроксимацией. Включение вычислений (8.87), (8.88) из раздела 8.8 в общий алгоритм 8.3 приводит как раз к частному алгоритму 8.2.

Марковская модель \mathcal{P} удовлетворяет условиям теорем 8.4, 8.5 и 8.6. Это два условия: обучающее множество L не должно быть противоречивым в классе \mathcal{P} моделей, а функция $\sum \alpha(\bar{x}, \bar{k}) \log \alpha^M(\bar{x}, \bar{k})$ должна быть гладкой. Первое свойство выполняется, так как для любого обучающего множества $((\bar{x}^j, \bar{k}^j), j = 1, \dots, l)$ существует марковская модель, в которой каждая пара (\bar{x}^j, \bar{k}^j) имеет ненулевую вероятность. В частности, это модель, в которой все элементы в паре (\bar{x}, \bar{k}) последовательностей являются взаимно независимыми случайными величинами. Можно убедиться и в том, что для марковской модели зависимость числа $\max_{p \in \mathcal{P}} \sum_{(\bar{x}, \bar{k}) \in L} \alpha(\bar{x}, \bar{k}) \log p(\bar{x}, \bar{k})$ от коэффициентов $\alpha(\bar{x}, \bar{k})$ не только выпуклая, но и дифференцируемая. Таким образом, алгоритм 8.2 есть частный случай алгоритма 8.3, а теоремы 8.2 и 8.3 - частные случаи теорем 8.5 и 8.6, соответственно.

8.10 Настройка алгоритма распознавания последовательностей

Задача настройки алгоритмов распознавания была сформулирована в общем виде в лекции 4, а для конкретного случая марковских моделей - в подразделе 8.7.3. Задача состоит в том, что для данного тестового множества $L = \{(\bar{x}^j, \bar{k}^j) | j = 1, \dots, l\}$ следует найти совокупность $f_i: K \times X \times K \rightarrow \mathbb{R}$,

$i = 1, 2, \dots, n$, функций, для которых выполняется

$$\bar{k}^j = \operatorname{argmax}_{\bar{k} \in K^{n+1}} \sum_{i=1}^n f_i(k_{i-1}, x_i^j, k_i), \quad j = 1, \dots, l. \quad (8.118)$$

В несколько в ином виде можно написать, что значения функций $f_i(k_{i-1}, x_i, k_i)$ должны удовлетворять системе неравенств

$$\sum_{i=1}^n f_i(k_{i-1}^j, x_i^j, k_i^j) > \sum_{i=1}^n f_i(k_{i-1}, x_i^j, k_i), \quad \bar{k} \neq \bar{k}^j, \quad j = 1, 2, \dots, l. \quad (8.119)$$

Система неравенств (8.119) состоит из необозримо большого количества $(|K|^{n+1} - 1)l$ линейных неравенств, которые ограничивают значения $n \times |K|^2 \times |X|$ переменных $f_i(k', x', k'')$. Несмотря на огромное количество неравенств, систему (8.119) можно решить, пользуясь методами, изложенными в лекции 4. Естественно, эту систему можно решить только при условии, что ее решение вообще существует. Основное преимущество изложенных ранее методов, будь то перцептронные алгоритмы, или алгоритмы Козинца, состоит в том, что в этих алгоритмах совсем не требуется проверять все неравенства, входящие в систему. Достаточно иметь в распоряжении метод, позволяющий конструктивно находить какое-нибудь одно неравенство в системе (8.119), которое при текущих значениях чисел $f_i(k', x', k'')$ не выполняется. Коррекция чисел $f_i(k', x', k'')$ на том или ином шаге алгоритма происходит на основе единственного, найденного на этом шаге неравенства, даже если на этом шаге было много неравенств, которые не выполнялись. Для системы неравенств (8.119), которая на самом деле получена на основе (8.118), такой конструктивный способ существует. Для каждого $j = 1, \dots, l$ достаточно найти последовательность

$$\bar{k}^{*j} = \operatorname{argmax}_{\bar{k}} \sum_{i=1}^n f_i(k_{i-1}, x_i^j, k_i), \quad (8.120)$$

то есть распознать все примеры из обучающего множества с помощью распознающего алгоритма с текущими значениями параметров, как это было описано в подразделе 8.4.4. Затем следует проверить, выполняются ли равенства

$$\bar{k}^{*j} = \bar{k}^j, \quad j = 1, 2, \dots, n. \quad (8.121)$$

Если равенство (8.121) выполняется для всех j , то задача настройки уже решена. Если для какого-то j равенство (8.121) не выполняется, это немедленно указывает на то неравенство в системе (8.119), которое не выполняется. Это - неравенство, правая часть которого соответствует неправильному ответу $k_0^{*j}, k_1^{*j}, \dots, k_n^{*j}$, выданному алгоритмом в его текущем состоянии, а левая часть - требуемому ответу $k_0^j, k_1^j, \dots, k_n^j$, указанному в тестовом множестве. Этих двух последовательностей достаточно для коррекции чисел $f_i(k', x', k'')$ с помощью алгоритма перцептрона или Козинца в следующей их адаптации.

Адаптация алгоритма персептрона к данной решаемой задаче имеет вид: числа $f_i(k', x', k'')$ следует увеличить на единицу, если $k_{i-1}^j = k', x_i^j = x', k_i^j = k''$, и уменьшены на единицу, если $k_{i-1}^{*j} = k', x_i^j = x', k_i^{*j} = k''$. Это простое правило включает в себя и ситуацию, когда $k_{i-1}^{*j} = k_{i-1}^j = k'$ и $k_i^{*j} = k_i^j = k''$. В этом случае к соответствующему числу $f_i(k', x', k'')$ следует добавить единицу и тут же ее вычесть, то есть оставить число $f_i(k', x', k'')$ без изменения. Трудно себе представить более простой алгоритм решения задачи 8.118. Еще труднее было ожидать, что алгоритм окажется настолько простым, до того, как он был найден. Уж слишком замысловатой представляется формулировка 8.118 задачи. Модификация алгоритма Козинца применительно к нашей задаче оказывается тоже достаточно простой.

Теорема Новикова 5.5 является в точности применимой и к рассматриваемой ситуации: если требования 8.118 непротиворечивы, то через конечное количество коррекций будут найдены значения параметров алгоритма, при которых требования 8.118 будут выполнены.

8.11 Наиболее правдоподобное оценивание модели в режиме самообучения

Пусть X^n - множество всех возможных последовательностей вида $\bar{x} = (x_1, x_2, \dots, x_n)$, $x_i \in X$, длины n . Пусть K^{n+1} - множество всех возможных последовательностей вида $\bar{k} = (k_0, k_1, \dots, k_n)$, $k_i \in K$, длины $n+1$. Пусть функция $p: X^n \times K^{n+1} \rightarrow \mathbb{R}$ для любых двух последовательностей $\bar{x} \in X^n$ и $\bar{k} \in K^{n+1}$ определяет их совместную вероятность $p(\bar{x}, \bar{k})$. Пусть функция p имеет вид

$$p(\bar{x}, \bar{k}) = p_1(k_0, x_1, k_1) \prod_{i=2}^n \frac{p_i(k_{i-1}, x_i, k_i)}{\sum_{k' \in K} \sum_{x' \in X} p_i(k_{i-1}, x', k')} . \quad (8.122)$$

Это значит, что функция $p: X^n \times K^{n+1} \rightarrow \mathbb{R}$ однозначно определяется n функциями p_i , $i = 1, \dots, n$, вида $K \times X \times K \rightarrow \mathbb{R}$. Обозначим \mathcal{P} множество всех возможных функций $p: X^n \times K^{n+1} \rightarrow \mathbb{R}$ вида (8.122), а совокупность $(p_i, i = 1, \dots, n)$ функций обозначим P .

Пусть $(\bar{x}^1, \bar{x}^2, \dots, \bar{x}^l)$ - выборка взаимно независимых последовательностей, причем длина каждой последовательности имеет длину n . Это значит, что $\bar{x}^j \in X^n$, $j = 1, \dots, l$. Будем считать, что каждая последовательность \bar{x}^j является реализацией случайной последовательности с распределением $\sum_{\bar{k} \in K^{n+1}} p(\bar{x}, \bar{k})$. В этом случае вероятность выборки $(\bar{x}^1, \bar{x}^2, \dots, \bar{x}^l)$ есть произведение $\prod_{j=1}^l \sum_{\bar{k} \in K^{n+1}} p(\bar{x}^j, \bar{k})$, которое зависит от функции $p \in \mathcal{P}$. Максимально правдоподобное оценивание модели в режиме самообучения состоит в отыскании модели $p^* \in \mathcal{P}$, которая максимизирует эту вероятность,

$$\begin{aligned} p^* &= \operatorname{argmax}_{p \in \mathcal{P}} \prod_{j=1}^l \sum_{\bar{k} \in K^{n+1}} p(\bar{x}^j, \bar{k}) \\ &= \operatorname{argmax}_{p \in \mathcal{P}} \sum_{j=1}^l \log \sum_{\bar{k} \in K^{n+1}} p(\bar{x}^j, \bar{k}) . \end{aligned} \quad (8.123)$$

Сходство этой задачи с задачей самообучения, описанной в лекции 6, достаточно очевидно, хотя, строго говоря, это различные задачи. Раньше мы рассматривали ситуацию, при которой поиск модели $p(x, k)$ в режиме обучения распадался на $|K|+1$ независимых задач. Первой из них была оценка априорных вероятностей $p_K(k)$ для каждого значения скрытого параметра k . Остальные $|K|$ задач состояли в оценке условных распределений $p_{X|k}(x)$ для каждого значения $k \in K$. При этом предполагалось, что выбор функции $p_{X|k'}$ из данного класса \mathcal{P} для одного значения k' никаким образом не ограничивает выбор функции $p_{X|k''}$ для какого-то другого значения k'' . Иными словами говоря, каждая функция $p_{X|k}$, $k \in K$, полностью определялась своим собственным значением a_k параметра a , и на выбор значений a_k , соответствующих различным k , априори не накладывались никакие ограничения. В этом отношении рассматриваемая здесь ситуация совершенно иная. Если \bar{k}' и \bar{k}'' две разные последовательности, то условные вероятности наблюдения \bar{x} при условии \bar{k}' и \bar{k}'' - это числа

$$p_{X^n|\bar{k}'}(\bar{x}) = \prod_{i=1}^n \frac{p_i(k'_{i-1}, x_i, k'_i)}{\sum_{x' \in X} p_i(k'_{i-1}, x', k'_i)}$$

и

$$p_{X^n|\bar{k}''}(\bar{x}) = \prod_{i=1}^n \frac{p_i(k''_{i-1}, x_i, k''_i)}{\sum_{x' \in X} p_i(k''_{i-1}, x', k''_i)}.$$

В правых частях этих двух выражений должны присутствовать одни и те же функции $p_i: K \times X \times K \rightarrow \mathbb{R}$, $i = 1, \dots, n$. Особенность рассматриваемой сейчас задачи, следовательно, в том, что параметры (p_1, p_2, \dots, p_n) статистической модели распознаваемого объекта определяют одновременно условные распределения $p_{X^n|\bar{k}}$ для всех $\bar{k} \in K^{n+1}$ и они же определяют априорные вероятности $p_{K^{n+1}}(\bar{k})$ каждой последовательности $\bar{k} \in K^{n+1}$.

Вопреки указанному формальному различию между задачей (8.123) и задачей самообучения, рассмотренной в лекции 6, эти две задачи близки в идейном плане, если позволительно так сказать. Задачу (8.123) можно решить, несколько модифицировав рассуждения, выполненные при анализе самообучения. Повторим кратко этот анализ, но уже в модифицированном виде в рамках марковской модели объекта распознавания.

Построим последовательность моделей $p^t \in \mathcal{P}$ для $t = 1, 2, \dots, \infty$, то есть последовательность совокупностей чисел $p_i^t(k', x', k'')$, $i = 1, \dots, n$, $k' \in K$, $x' \in X$, и $k'' \in K$. Каждая совокупность в этой последовательности определяет распределение вероятностей на множестве пар $(\bar{x}, \bar{k}) \in X^n \times K^{n+1}$,

$$p^t(\bar{x}, \bar{k}) = p_1^t(x_0, x_1, k_1) \prod_{i=2}^n \frac{p_i^t(k_{i-1}, x_i, k_i)}{\sum_{x' \in X} \sum_{k' \in K} p_i^t(k_{i-1}, x', k')} . \quad (8.124)$$

Обозначим $J_i(x')$ подмножество индексов j , для которых $x_i^j = x'$, а подмножество последовательностей \bar{k} , для которых $k_{i-1} = k'$, $k_i = k''$, обозначим $K_i(k', k'')$.

Укажем алгоритм изменения параметров $p_i^t(k', x', k'')$ статистической модели объекта. Этот алгоритм работает по шагам. Пусть после выполнения t -ого шага получены числа $p_i^t(k', x', k'')$. Новые значения $p_i^{t+1}(k', x', k'')$ этих чисел строятся в два этапа.

Распознавание. Для каждой последовательности \bar{x}^j и каждой последовательности $\bar{k} \in K^{n+1}$ вычисляются числа

$$\alpha^t(\bar{x}^j, \bar{k}) = \frac{p^t(\bar{x}^j, \bar{k})}{\sum_{\bar{k}' \in K^{n+1}} p^t(\bar{x}^j, \bar{k}')}, \quad (8.125)$$

где числа $p^t(\bar{x}^j, \bar{k})$ вычисляются по формуле (8.124).

Обучение. Для каждого $i = 1, 2, \dots, n$ и каждой тройки $k' \in K$, $x' \in X$, $k'' \in K$ вычисляются новые значения

$$p_i^{t+1}(k', x', k'') = \frac{1}{l} \sum_{j \in J_i(x')} \sum_{\bar{k} \in K_i(k', k'')} \alpha^t(\bar{x}^j, \bar{k}). \quad (8.126)$$

Приведенный алгоритм, конечно же, непригоден для практического использования, так как множество K^{n+1} необозримо большое. Поэтому невозможно посчитать числа $\alpha(\bar{x}^j, \bar{k})$ для каждой последовательности $\bar{k} \in K^{n+1}$. По этой же причине невозможно вычислить сумму по всем возможным последовательностям $\bar{k} \in K_i(k', k'')$ в формуле (8.126). Позже мы сформулируем реализуемый алгоритм, эквивалентный только что приведенному. Алгоритм же в только что приведенной неконструктивной форме полезен тем, что для него сравнительно легко доказывается следующая фундаментальная теорема о монотонном характере самообучения.

Теорема 8.7 О самообучении марковских моделей. Пусть $p_i^t(k', x', k'')$ и $p_i^{t+1}(k', x', k'')$, $i = 1, 2, \dots, n$, $k' \in K$, $x' \in X$, $k'' \in K$, - две совокупности, вычисленные по формулам (8.125) и (8.126). Пусть p^t , p^{t+1} - две модели, то есть функции вида $X^n \times K^{n+1} \rightarrow \mathbb{R}$, определенные формулой (8.124). В этом случае справедливо неравенство

$$\prod_{j=1}^l \sum_{\bar{k} \in K^{n+1}} p^{t+1}(\bar{x}^j, \bar{k}) \geq \prod_{j=1}^l \sum_{\bar{k} \in K^{n+1}} p^t(\bar{x}^j, \bar{k}). \quad \blacktriangle$$

Доказательство. В силу того, что

$$\sum_{\bar{k} \in K^{n+1}} \alpha^t(\bar{x}^j, \bar{k}) = 1, \quad j = 1, \dots, l,$$

справедливы следующие два равенства,

$$\begin{aligned} & \sum_{j=1}^l \log \sum_{\bar{k} \in K^{n+1}} p^t(\bar{x}^j, \bar{k}) \\ &= \sum_{j=1}^l \sum_{\bar{k} \in K^{n+1}} \alpha^t(\bar{x}^j, \bar{k}) \log p^t(\bar{x}^j, \bar{k}) - \sum_{j=1}^l \left(\sum_{\bar{k} \in K^{n+1}} \alpha^t(\bar{x}^j, \bar{k}) \log \frac{p^t(\bar{x}^j, \bar{k})}{\sum_{\bar{k} \in K^{n+1}} p^t(\bar{x}^j, \bar{k})} \right), \end{aligned}$$

$$\begin{aligned} & \sum_{j=1}^l \log \sum_{\bar{k} \in K^{n+1}} p^{t+1}(\bar{x}^j, \bar{k}) \\ &= \sum_{j=1}^l \sum_{\bar{k} \in K^{n+1}} \alpha^t(\bar{x}^j, \bar{k}) \log p^{t+1}(\bar{x}^j, \bar{k}) - \sum_{j=1}^l \left(\sum_{\bar{k} \in K^{n+1}} \alpha^t(\bar{x}^j, \bar{k}) \log \frac{p^{t+1}(\bar{x}^j, \bar{k})}{\sum_{\bar{k} \in K^{n+1}} p^{t+1}(\bar{x}^j, \bar{k})} \right). \end{aligned}$$

В силу (8.125) для $j = 1, 2, \dots, l$ выполняется

$$\alpha^t(\bar{x}^j, \bar{k}) = \frac{p^t(\bar{x}^j, \bar{k})}{\sum_{\bar{k} \in K^{n+1}} p^t(\bar{x}^j, \bar{k})}$$

Поэтому из леммы 6.1 следует неравенство

$$\sum_{\bar{k} \in K^{n+1}} \alpha^t(\bar{x}^j, \bar{k}) \log \frac{p^t(\bar{x}^j, \bar{k})}{\sum_{\bar{k} \in K^{n+1}} p^t(\bar{x}^j, \bar{k})} \geq \sum_{\bar{k} \in K^{n+1}} \alpha^t(\bar{x}^j, \bar{k}) \log \frac{p^{t+1}(\bar{x}^j, \bar{k})}{\sum_{\bar{k} \in K^{n+1}} p^{t+1}(\bar{x}^j, \bar{k})},$$

справедливое для каждого $j = 1, 2, \dots, l$. Поэтому запишем

$$\sum_{j=1}^l \sum_{\bar{k} \in K^{n+1}} \alpha^t(\bar{x}^j, \bar{k}) \log \frac{p^t(\bar{x}^j, \bar{k})}{\sum_{\bar{k} \in K^{n+1}} p^t(\bar{x}^j, \bar{k})} \geq \sum_{j=1}^l \sum_{\bar{k} \in K^{n+1}} \alpha^t(\bar{x}^j, \bar{k}) \log \frac{p^{t+1}(\bar{x}^j, \bar{k})}{\sum_{\bar{k} \in K^{n+1}} p^{t+1}(\bar{x}^j, \bar{k})}. \quad (8.127)$$

Числа $p_i^{t+1}(k', x', k'')$, вычисленные по формуле (8.126), удовлетворяют теореме 8.1, откуда следует, что

$$\sum_{j=1}^l \sum_{\bar{k} \in K^{n+1}} \alpha^t(\bar{x}^j, \bar{k}) \log p^{t+1}(\bar{x}^j, \bar{k}) \geq \sum_{j=1}^l \sum_{\bar{k} \in K^{n+1}} \alpha^t(\bar{x}^j, \bar{k}) \log p^t(\bar{x}^j, \bar{k}). \quad (8.128)$$

Из неравенств (8.127) и (8.128) немедленно следует неравенство

$$\sum_{j=1}^l \log \sum_{\bar{k} \in K^{n+1}} p^{t+1}(\bar{x}^j, \bar{k}) \geq \sum_{j=1}^l \log \sum_{\bar{k} \in K^{n+1}} p^t(\bar{x}^j, \bar{k}),$$

что и доказывает теорему. ■

Покажем теперь, что приведенный алгоритм можно эквивалентно преобразовать так, что он становится конструктивно реализуемым. Числа $\alpha^t(\bar{x}^j, \bar{k})$ и вероятности $p^t(\bar{x}^j, \bar{k})$ являются в алгоритме лишь вспомогательными, промежуточными данными, которые облегчают доказательство теоремы 8.7. В конструктивном исполнении алгоритма эти числа можно исключить и представить алгоритм таким образом, что он будет оперировать

только с параметрами $p_i^t(k', x', k'')$ и числами $\alpha_i^t(k', x^j, k'')$, определенными как

$$\alpha_i^t(k', x_i^j, k'') = \sum_{\bar{k} \in K_i(k', k'')} \alpha^t(\bar{x}^j, \bar{k}).$$

Формула (8.126) приобретает в этом случае совершенно простой вид

$$p_i^{t+1}(k', x', k'') = \frac{1}{l} \sum_{j \in J_i(x')} \alpha_i^t(k', \bar{x}^j, k''),$$

и вычисление по ней не приводит ни к каким непреодолимым трудностям, так как суммирование в ней выполняется только по элементам, которые присутствуют в обучающей выборке.

Вычисления же по формуле (8.125) заменяются вычислением сумм $\alpha_i^t(k', \bar{x}^j, k'')$ на основе чисел $p_i^{t-1}(k', x', k'')$. Механизм вычисления этих чисел детально рассмотрен в разделе 8.5. Ведь число $\alpha_i^t(k', \bar{x}^j, k'')$ - это не что иное, как совместная апостериорная вероятность события $k_{i-1} = k'$, $k_i = k''$ при условии наблюдения последовательности \bar{x}^j .

8.12 Обсуждение

У меня не возникли вопросы по данной лекции, и это меня тревожит. Я тоже веду педагогическую работу и знаю, что если в конце занятия у слушателей не возникли вопросы, то это совсем не значит, что учебный материал усвоен. И тем не менее, задачи, рассмотренные в лекции, представляются мне достаточно понятными и даже простыми. Более того, отталкиваясь от материала лекции, мне не приходят на мысль задачи головоломной сложности, которые заставляли бы глубоко и надолго задуматься. Отсюда впечатление, возможно, кажущееся, что с распознаванием марковских последовательностей я так или иначе справлюсь, если в этом возникнет практическая необходимость.

Повидимому, Твое впечатление не обманчивое, и материал лекции Ты усвоил. А не вышел Ты на трудные задачи только потому, что мало об этом думал. Если бы Ты рассмотрел ряд постепенно усложняющихся задач, начиная с задач, рассмотренных в лекции, то на пятом-шестом шаге Ты бы обязательно вышел на задачу, которая потребовала бы от Тебя значительных усилий. Раз Ты этого не сделал, давай сделаем это вместе, начиная с довольно простой задачи.

Пусть $\bar{x} \in X^n$ и $\bar{k} \in K^{n+1}$ - это две случайные последовательности с распределением совместных вероятностей $p(\bar{x}, \bar{k})$ марковского типа. Мы уже знаем, что для любой последовательности \bar{x} можно конструктивно найти последовательность $\bar{k}^* = \operatorname{argmax}_{\bar{k} \in K^{n+1}} p(\bar{x}, \bar{k})$. Как бы Ты построил алгоритм, если бы Тебя не интересовала вся последовательность \bar{k}^* целиком, а нужно было бы только определить, сколько раз в этой последовательности встречается определенный символ $\sigma \in K$?

В вашем вопросе кроется какая-то ловушка, которую я не обнаружил. Коль скоро известно, как находить последовательность \bar{k}^* , то нужно просто подсчитать, сколько раз в этой последовательности встречается заданный символ σ . Я не вижу здесь никаких трудностей.

Нельзя сказать, что Ты неправ, но сформулированная задача все же дает определенный повод для размышлений. Решение, о котором Ты говоришь и которое первое приходит в голову, не всегда реализуемо. Ты принимаешь, как само собою разумеющееся, что для решения задачи непременно нужно построить последовательность \bar{k}^* . Но при формулировке задачи мы подчеркнули, что вся последовательность в целом Тебя не интересует. В рассматриваемой задаче эта последовательность - это только вспомогательная переменная, в терминах которой формулируется задача. Но это совсем не значит, что ее непременно нужно вычислять.

Да, было сказано, что последовательность \bar{k}^* можно не вычислять, но не было сказано, что ее нельзя вычислять. Я не вижу, почему следует избегать ее вычисления.

Потому что это требование может возникнуть само собой в результате технологических условий применения Твоего решения. Пусть, например, анализируется последовательность очень большой длины n , так, что в тот момент, когда она закончилась, ее начальная часть уже потеряна. В этом случае Тебе просто негде хранить указатели $\text{ind}_i(k)$ для каждого $i = 1, 2, \dots, n$ и каждого $k \in K$, так как для этого требуется память в $n |K| \log |K|$ бит, которая пропорциональна n . А без этих указателей Ты не построишь последовательность \bar{k}^* . Вспомни процедуру, описанную в подразделе 8.4.4. Следовательно, в этой ситуации Тебе невозможно построить последовательность \bar{k}^* . Тем не менее можно вычислить, сколько раз в этой последовательности встречается заданный символ σ .

Я понял. Я введу в рассмотрение числа $h_i(k)$, $i = 0, 1, \dots, n$, $k \in K$, которые обозначают, сколько раз символ σ содержится в последовательности k'_0, k'_1, \dots, k'_i , которая максимизирует вероятность $p(x_1^i, k_0^i)$ и для которой $k'_i = k$. Числа $h_i(k)$ вычисляются по следующей процедуре:

$$\begin{aligned} h_0(k) &= \begin{cases} 1, & \text{если } k = \sigma, \\ 0, & \text{если } k \neq \sigma. \end{cases} \\ h_i(k) &= \begin{cases} h_{i-1}(\text{ind}_i(k)) + 1, & \text{если } k = \sigma, \\ h_{i-1}(\text{ind}_i(k)), & \text{если } k \neq \sigma. \end{cases} \end{aligned} \quad (8.129)$$

Если $k^* = (k_0^*, k_1^*, \dots, k_n^*)$ - наиболее вероятная последовательность, то число $h_n(k_n^*)$ есть решение задачи, и для его вычисления нет нужды знать всю последовательность k^* , а достаточно знать лишь последний ее элемент.

Числа $\text{ind}_i(k)$ в выражении (8.129) используются только один раз, причем непосредственно после их вычисления, следовательно, для их запоминания достаточно всего $\log |K|$ бит. Числа $h_i(k)$, $k \in K$, вычисляемые алгоритмом, также используются только один раз при вычислении числа $h_{i+1}(k)$. Это значит, что числа $h_i(k)$, вычисленные на i -ом шаге, потребуются только на $i + 1$ -ом шаге и больше никогда. Следовательно, для них потребуется память в $2|K| \log n$ бит, то есть значительно меньше, чем $n|K| \log |K|$ бит, которые были бы необходимы при восстановлении всей последовательности k^* . Этот более продуманный алгоритм можно применять для анализа практически сколь угодно длинной последовательности.

Построй теперь алгоритм, который для заданного стохастического автомата определяет, сколько раз этот автомат прошел через заданное состояние σ , когда он генерировал известную последовательность (x_1, x_2, \dots, x_n) .

Разве я не сделал это только что?

Конечно, нет. Наиболее вероятная последовательность совсем необязательно совпадает с действительной. Более того, при больших n вероятность их совпадения ничтожно мала.

Но ведь действительная последовательность мне неизвестна. Самое большее, что я могу сделать, - это вычислить вероятность, что некоторая заданная последовательность совпадает с действительной. Таким образом, и число, указывающее, сколько раз автомат прошел через заданное состояние, тоже случайное. Поставленный вами вопрос сродни следующему: известно распределение вероятностей случайной величины требуется сказать, чему равна случайная величина. Но ведь это бессмыслица. Случайная величина не равна никакому фиксированному своему значению.

Мы очень рады столь жесткой Твоей критике. Это действительно бессмыслица. Сформулируем вопрос правильно. Для заданного стохастического автомата, заданного состояния σ , заданного числа l и заданной последовательности (x_1, x_2, \dots, x_n) указать вероятность того, что автомат l раз побывал в состоянии σ при генерировании указанной последовательности.

Я обозначу x_1^i последовательность (x_1, x_2, \dots, x_i) и буду считать, что x_1^0 - это пустая последовательность. Я обозначу $g_i(x_1^i, l, k)$, $i = 0, 1, \dots, n$, $l = 0, 1, 2, \dots, i + 1$, $k \in K$, совместную вероятность следующих трех событий:

1. Первые i элементов, сгенерированных автоматом, равны x_1^i .
2. В последовательности k_0, k_1, \dots, k_i состояний, через которые прошел автомат, состояние σ содержится l раз.
3. Состояние k_i есть k .

В соответствии с этим определением выполняется

$$g_0(x_1^0, l, k_0) = \begin{cases} p_0(k_0), & \text{если } l = 1 \text{ и } k_0 = \sigma, \\ p_0(k_0), & \text{если } l = 0 \text{ и } k_0 \neq \sigma, \\ 0, & \text{в прочих случаях.} \end{cases} \quad (8.130)$$

Обозначу символом $g'_i(x_1^i, l, k)$, $i = 1, 2, \dots, n$, $l = 0, 1, \dots, i$, $k \in K$, совместную вероятность несколько иных событий, чем тех, вероятность которых была обозначена g .

1. Первые i элементов в сгенерированной автоматом последовательности символов равны x_1^i .
2. В последовательности $(k_0, k_1, \dots, k_{i-1})$ состояний, через которые прошел автомат, состояние σ содержится l раз.
3. Состояние k_i равно k .

Вероятности g_i и g'_i связаны соотношением

$$g_i(x_1^i, l, k) = \begin{cases} g'_i(x_1^i, l, k), & \text{если } k \neq \sigma, \\ g'_i(x_1^i, l-1, k), & \text{если } k = \sigma. \end{cases} \quad (8.131)$$

а частное

$$\frac{\sum_{k \in K} g_n(x_1^n, l, k)}{\sum_{l=0}^n \sum_{k \in K} g_n(x_1^n, l, k)}$$

есть искомая апостериорная вероятность, что автомат l раз побывал в состоянии σ при генерировании известной последовательности x_1^n . Конструируемый алгоритм должен последовательно вычислять числа $g_0(), g_1(), \dots, g_n()$, начиная с чисел g_0 , которые определяются согласно (8.130). Для формулировки алгоритма этих вычислений я ввожу в рассмотрение вспомогательные числа $g''_i(x_1^i, l, k', k'')$, которые равны совместной вероятности следующих четырех событий.

1. Первые i элементов сгенерированной автоматом выходной последовательности равны x_1^i .
2. В последовательности k_0, k_1, \dots, k_{i-1} состояний, через которые прошел автомат, состояние σ встречается l раз.
3. Состояние k_{i-1} равно k' .
4. Состояние k_i равно k'' .

Вероятности g' и g'' удовлетворяют следующему равенству, которому удовлетворяют любые вероятности:

$$g'_i(x_1^i, l, k) = \sum_{k' \in K} g''_i(x_1^i, l, k', k). \quad (8.132)$$

Кроме того, для вероятностей g''_i выполняется

$$g''_i(x_1^i, l, k', k) = g''_i((x_1^{i-1}, x_i), l, k', k) = g_{i-1}(x_1^{i-1}, l, k') p_i(x_i, k | k'), \quad (8.133)$$

где $p_i(x_i, k | k')$ - известные вероятности, которые характеризуют автомат. Выражение (8.133) справедливо, так как при фиксированном состоянии k' в $(i - 1)$ -ый момент случайные величины x_i и k , реализованные после этого момента, не зависят от случайных величин, реализованных до этого момента. После включения (8.133) в (8.132) я получаю

$$g'_i(x_0^i, l, k) = \sum_{k' \in K} g_{i-1}(x_1^{i-1}, l, k') p_i(x_i, k | k').$$

Используя (8.131), я могу то же самое записать в виде

$$g_i(x_1^i, l, k) = \begin{cases} \sum_{k' \in K} g_{i-1}(x_1^{i-1}, l, k') p_i(x_i, k | k'), & \text{если } k \neq \sigma, \\ \sum_{k' \in K} g_{i-1}(x_1^{i-1}, l - 1, k') p_i(x_i, k | k'), & \text{если } k = \sigma. \end{cases} \quad (8.134)$$

Однократное вычисление по формуле (8.134) имеет сложность порядка $\mathcal{O}(l |K|^2)$, а весь алгоритм в целом - сложность порядка $\mathcal{O}(l^2 |K|^2)$. Эта сложность представляется мне довольно большой. Нельзя ли ее как-то снизить?

Вряд ли. Ты справился с задачей вполне прилично. Хотим добавить только, что если бы требовалось вычислить не распределение вероятностей случайной величины l , а только некоторые характеристики этой случайной величины, такие, как ее математическое ожидание или дисперсию, потребовались бы существенно меньше вычислений. Было бы слишком расточительным вычислять вероятности всех значений случайной величины лишь для того, чтобы затем вычислить только ее математическое ожидание.

Рассмотрим теперь следующую по сложности задачу, так как скорее всего мы еще не вышли на предел Твоих ресурсов.

Предположим, что Ты интересуешься не только тем, сколько раз автомат побывал в состоянии σ , но и в какие моменты времени это произошло. Это значит, что результатом распознавания является множество I' моментов времени, при которых $k_i = \sigma$. Эта задача интересна тем, что она может считаться предельно упрощенной задачей *сегментации*. В данном случае это *сегментация интервала времени* $(0, 1, 2, \dots, n)$ на подинтервалы, отделенные друг от друга состоянием σ . Эта задача по своему коварству может сравниться разве что с задачей о вертикальных и горизонтальных линиях, о которой Ты, наверное, помнишь. На первый взгляд она кажется настолько простой, что не заслуживает и внимания. Например, если речь идет об изображении текстовой строки, а под состоянием σ понимается символ пробела, то в сформулированной задаче требуется разбить текстовое изображение на сегменты, соответствующие отдельным словам. Мы сформулируем эту задачу точно, и Ты увидишь, что она может быть решена в общем виде, хотя построение алгоритма ее решения потребует от Тебя определенные усилия.

Назовем сегментацией подмножество $I' \subset \{0, 1, \dots, n\}$. Для каждой сегментации I' определим множество $K(I')$ таких последовательностей, что $k_i = \sigma$ для $i \in I'$ и $k_i \neq \sigma$ для $i \notin I'$. Вероятность сегментации I' есть вероятность того, что последовательность \bar{k} принадлежит $K(I')$. При заданной последовательности \bar{x} вероятность сегментации I' есть сумма $\sum_{\bar{k} \in K(I')} p(\bar{k} | \bar{x})$, а наиболее вероятная сегментация I^* - это

$$I^* = \operatorname{argmax}_{I'} \sum_{\bar{k} \in K(I')} p(\bar{k} | \bar{x}) = \operatorname{argmax}_{I'} \sum_{\bar{k} \in K(I')} p(\bar{x}, \bar{k}). \quad (8.135)$$

А теперь построй алгоритм, который для каждой последовательности \bar{x} указывает наиболее вероятную сегментацию (8.135).

Я справился с задачей, но, возможно, это самая трудная задача, с которой я еще смог справиться. Я нашел, что задачу (8.135) можно свести к задаче динамического программирования. Но для подготовки исходных данных для этой задачи требуются довольно нетривиальные вычисления.

Прежде всего я выяснил, как выглядит функция

$$F(I') = \sum_{\bar{k} \in K(I')} p(\bar{x}, \bar{k}),$$

которую следует максимизировать согласно требованию (8.135).

Так как I' - это подпоследовательность в упорядоченной последовательности $I = (0, 1, \dots, n)$, то I' также упорядочена и может быть представлена в виде последовательностей индексов $(i_0, i_1, \dots, i_q, \dots, i_Q)$, $i_q > i_{q-1}$, $q = 1, 2, \dots, Q$, $Q = |I'| - 1$. Далее я предположил, что индексы i_0 и i_Q известны, а именно, $i_0 = 0$ и $i_Q = n$. Это значит, что автомат начинает и заканчивает генерирование наблюдаемой последовательности в состоянии σ . Не буду отвлекать вас доказательством правомерности такого допущения. Я просто утверждаю, что этим допущением я не сужаю общность дальнейших рассуждений.

Для некоторой заданной сегментации $I' = (i_0, i_1, \dots, i_Q)$ пара (\bar{x}, \bar{k}) может быть представлена как конкатенация определенного количества подпоследовательностей в виде

$$(\bar{x}, \bar{k}) = \begin{array}{l} k_0, \\ x_1^{i_1}, \quad k_1^{i_1-1}, \quad k_{i_1}, \\ x_{i_1+1}^{i_2}, \quad k_{i_1+1}^{i_2-1}, \quad k_{i_2}, \\ \vdots \\ x_{i_{q-1}+1}^{i_q}, \quad k_{i_{q-1}+1}^{i_q-1}, \quad k_{i_q}, \\ \vdots \\ x_{i_{Q-1}+1}^n, \quad k_{i_{Q-1}+1}^{n-1}, \quad k_n. \end{array}$$

В силу марковости исследуемого объекта, при фиксированном состоянии k_{i_q} переменные $x_i, k_i, i < i_q$, не зависят от переменных $x_i, k_i, i > i_q$. Следовательно, совместная вероятность пары (\bar{x}, \bar{k}) имеет вид произведения

$$p(\bar{x}, \bar{k}) = p_0(k_0) \prod_{q=1}^Q p'_q \left(x_{i_{q-1}+1}^{i_q}, k_{i_{q-1}+1}^{i_q-1}, k_{i_q} \mid k_{i_{q-1}} \right). \quad (8.136)$$

В этом произведении число $p_0(k), k \in K$, есть вероятность того, что состояние k окажется начальным состоянием автомата. Число $p'_q(x_{i_{q-1}+1}^{i_q}, k_{i_{q-1}+1}^{i_q-1}, k_{i_q} \mid k_{i_{q-1}})$ есть условная вероятность того, что при условии состояния $k_{i_{q-1}}$ в (i_{q-1}) -ый момент последующие $i_q - i_{q-1}$ выходные символы будут равны $x_{i_{q-1}+1}^{i_q}$, а автомат пройдет через $i_q - i_{q-1}$ состояний $k_{i_{q-1}+1}^{i_q}$.

В сумме $\sum_{\bar{k} \in K(I')} p(\bar{x}, \bar{k})$, зависящей от сегментации $I' = (i_0, i_1, \dots, i_Q)$, учитываются только те последовательности \bar{k} , в которых $k_{i_q} = \sigma, q = 0, 1, \dots, Q$, а $k_i \neq \sigma$ для всех прочих индексов i . Поэтому произведение (8.136) принимает вид

$$p(\bar{x}, \bar{k}) = p_0(\sigma) \prod_{q=1}^Q p'_q \left(x_{i_{q-1}+1}^{i_q}, k_{i_{q-1}+1}^{i_q-1}, \sigma \mid \sigma \right). \quad (8.137)$$

Это произведение следует просуммировать по всем последовательностям из множества $K(I')$. Это значит, что должно быть произведено суммирование $\sum_{k \in K(I')}$, или, в более подробном виде, суммирование

$$\sum_{k_1^{i_1-1}} \sum_{k_{i_1+1}^{i_2-1}} \cdots \sum_{k_{i_{q-1}+1}^{i_q-1}} \cdots \sum_{k_{i_{Q-1}+1}^{i_Q-1}}. \quad (8.138)$$

Это суммирование произведений (8.137) имеет тот же самый вид, что и суммирование

$$\sum_{z_1} \sum_{z_2} \cdots \sum_{z_m} \prod_{i=1}^m \varphi_i(z_i)$$

произведений $\prod_{i=1}^m \varphi_i(z_i)$, которое эквивалентно произведению $\prod_{i=1}^m \sum_{z_i} \varphi_i(z_i)$. Равенство

$$\sum_{z_1} \sum_{z_2} \cdots \sum_{z_m} \prod_{i=1}^m \varphi_i(z_i) = \prod_{i=1}^m \sum_{z_i} \varphi_i(z_i)$$

справедливо для любых функций $\varphi_i, i = 1, 2, \dots, m$. Следовательно, функцию $F(I')$, которая есть сумма (8.138) произведений (8.137), можно представить в виде

$$\begin{aligned} F(I') &= \sum_{\bar{k} \in K(I')} p(\bar{x}, \bar{k}) \\ &= p_0(\sigma) \prod_{q=1}^Q \sum_{k_{i_{q-1}+1}^{i_q-1}} p'_q \left(x_{i_{q-1}+1}^{i_q}, k_{i_{q-1}+1}^{i_q-1}, \sigma \mid \sigma \right). \end{aligned} \quad (8.139)$$

Введем обозначение $p'_{ij}(x_{i+1}^j, k_{i+1}^{j-1}, \sigma | \sigma)$ для вероятности того, что при условии $k_i = \sigma$ автомат сгенерирует последовательность x_{i+1}^j и пройдет через состояния (k_{i+1}^{j-1}, σ) . Таким образом, ранее введенное обозначение p'_q эквивалентно обозначению p'_{i_{q-1}, i_q} . Каждый сомножитель в произведении (8.139) приобретает теперь вид

$$\sum_{k_{i+1}^{j-1}} p'_{ij}(x_{i+1}^j, k_{i+1}^{j-1}, \sigma | \sigma) \quad (8.140)$$

и зависит только от i и j . Сумма (8.140) очевидно не зависит от подпоследовательности k_{i+1}^{j-1} , потому что суммирование выполняется по множеству этих подпоследовательностей. Сумма (8.140) не зависит от подпоследовательности x_{i+1}^j , потому что она фиксирована. Введем символ $\Phi(i, j)$ для обозначения суммы (8.140), то есть,

$$\Phi(i, j) = \sum_{k_{i+1}^{j-1}} p'_{ij}(x_{i+1}^j, k_{i+1}^{j-1}, \sigma | \sigma). \quad (8.141)$$

В приведенном выражении суммирование выполняется по всем тем последовательностям k_{i+1}^{j-1} , в которых не присутствует символ σ . Выражение (8.139) для $F(I')$ можно кратко записать, используя введенное обозначение $\Phi(i, j)$,

$$F(I') = p_0(\sigma) \prod_{q=1}^Q \Phi(i_{q-1}, i_q). \quad (8.142)$$

Таким образом мне ужалось разложить решаемую задачу на две отдельные части. В первой части должны вычисляться числа $\Phi(i, j)$ для каждой пары индексов (i, j) , $i = 0, 1, \dots, n-1$, $j = 1, 2, \dots, n$, $i < j$, в соответствии с определением (8.141). Во второй части следует найти сегментацию I^* , то есть последовательность $i_0^*, i_1^*, \dots, i_Q^*$ (при заранее не заданном Q), которая максимизирует (8.142),

$$\begin{aligned} I^* &= (i_0^*, i_1^*, i_2^*, \dots, i_{Q-1}^*, i_Q^*) \\ &= \operatorname{argmax}_{i_1, i_2, \dots, i_{Q-1}} \prod_{q=1}^Q \Phi(i_{q-1}, i_q), \end{aligned} \quad (8.143)$$

при условии $0 = i_0 < i_1 < i_2 < \dots < i_{Q-1} < i_Q = n$.

Вычисления в первой части подобны вычислениям, которые упоминались в лекции. Пусть $\Phi'(i, j, k)$ вспомогательные переменные, определенные как

$$\Phi'(i, j, k) = \sum_{k_{i+1}^{j-1}} p'_{ij}(x_{i+1}^j, k_{i+1}^{j-1}, k | \sigma).$$

Искомые числа $\Phi(i, j)$ равны $\Phi'(i, j, \sigma)$. Числа $\Phi'(i-1, i, k)$ вычисляются по следующим рекуррентным формулам

$$\Phi'(i, j, k) = \sum_{k' \neq \sigma} \Phi'(i, j-1, k') p_j(x_j, k | k'). \quad (8.144)$$

Вычисления начинаются с определения $\Phi'(i-1, i, k)$, которые для каждого i равны известным вероятностям $p_i(x_i, k | \sigma)$, характеризующим автомат. Вычисление совокупности чисел $\Phi'(i, j, k)$ для всех троек (i, j, k) по формуле (8.144) имеет сложность порядка $\mathcal{O}(n^2 |K|^2)$. Когда скоро будут вычислены числа $\Phi'(i, j, k)$, станут известны и числа $\Phi(i, j) = \Phi'(i, j, \sigma)$.

Покажу теперь, как следует находить сегментацию I^* , удовлетворяющую требованию (8.143). Пусть $i^* > 0$ - некоторый индекс, а $J(i^*)$ - множество всех последовательностей вида

$$i_0, i_1, i_2, \dots, i_Q, \quad 0 = i_0 < i_1 < i_2 < \dots < i_{Q-1} < i_Q = i^*.$$

Каждая последовательность такого вида характеризуется числом $\prod_{q=1}^Q \Phi(i_{q-1}, i_q)$. Обозначу $F^*(i^*)$ наибольшее из этих чисел,

$$F^*(i^*) = \max_{(i_0, i_1, \dots, i_Q) \in J(i^*)} \prod_{q=1}^Q \Phi(i_{q-1}, i_q). \quad (8.145)$$

Пусть $i_0^*, i_1^*, \dots, i_{Q-1}^*, i_Q^*$ - последовательность, которая максимизирует (8.145). Последний индекс в этой последовательности есть i^* . Для $i^* > 0$ в общем случае верно, что

$$\begin{aligned} F^*(i^*) &= \max_{\substack{i_1 \\ 0 < i_1 < \dots < i_{Q-1} < i^*}} \dots \max_{i_{Q-1}} \prod_{q=1}^Q \Phi(i_{q-1}, i_q) \\ &= \max_{i_{Q-1} < i^*} \underbrace{\max_{i_1} \max_{i_2} \dots \max_{i_{Q-2}}}_{0 < i_1 < i_2 < \dots < i_{Q-2} < i_{Q-1}} \Phi(i_{Q-1}, i^*) \prod_{q=1}^{Q-1} \Phi(i_{q-1}, i_q) \\ &= \max_{i_{Q-1} < i^*} \Phi(i_{Q-1}, i^*) F^*(i_{Q-1}). \end{aligned}$$

Исключая промежуточные выкладки в этом выражении, получаю

$$F^*(i^*) = \max_{i < i^*} \Phi(i, i^*) F^*(i) \quad (8.146)$$

и

$$\text{ind}(i^*) = \operatorname{argmax}_{i < i^*} \Phi(i, i^*) F^*(i). \quad (8.147)$$

По формулам (8.146) и (8.147) я последовательно вычисляю $F^*(1)$, $F^*(2)$, \dots , $F^*(n)$ и $\text{ind}(1)$, $\text{ind}(2)$, \dots , $\text{ind}(n)$. Сложность этих вычислений имеет порядок $\mathcal{O}(n^2)$. Индекс $\text{ind}(n)$ есть предпоследний индекс i_{Q-1} в искомой сегментации, индекс $\text{ind}(i_{Q-1})$ определяет i_{Q-2} , и так с помощью указателей ind я нахожу все меньшие и меньшие индексы, пока не дойду до индекса 0.

Я уверен, что разобрался с этой задачей. Уверен также, что эта задача находится уже близко к границе моих возможностей.

Как нам кажется, именно к этому Ты стремился. Осталось теперь сделать последний шаг. Можешь ли Ты сформулировать задачу, которая находится за пределами Твоих возможностей?

Пока еще нет, но мне уже ясны идеи этой будущей формулировки, которые возникают из понимания того, что только что решенная задача является самым примитивным вариантом задачи сегментации. Прежде всего, о задачах сегментации говорят обычно применительно к изображениям, а не к последовательностям. Пока что я еще не представляю себе, каким образом следует обобщить рассматриваемые в лекции марковские модели, чтобы они стали пригодны для распознавания не только одномерных последовательностей, но и двумерных массивов. Правда, я знаю, что в этом направлении сделано сейчас довольно много. Однако, даже если рассматривать сегментацию одномерных последовательностей, то только что исследованная мною задача слишком примитивна. Эта формализация беднее приложений, потому что в ней игнорируются те особенности задачи, которые в ее прикладном, неформальном звучании прямо-таки бросаются в глаза. Формализация этих прикладных задач требует значительно большей тщательности.

Из ваших лекций мне стало достаточно ясно, что только на первый взгляд представляется естественным принимать решения в пользу апостериори наиболее вероятного значения. При первом же серьезном погружении в прикладную задачу становится очевидным, что такое решение очень отдаленно соответствует требованиям, естественным в том или ином приложении. Такое решение равносильно молчаливому допущению, что любое отличие двух сегментаций друг от друга имеет одинаковую значимость. Это недопустимое загробление задачи. Следовало бы учитывать по крайней мере, сколько раз принято ошибочное решение, что в какой-то i -ый момент автомат находился в состоянии σ , а в действительности он находился в каком-то другом состоянии, или наоборот. Однако и эти ошибочные решения следовало бы квалифицировать дифференцировано. К примеру, пусть принято ошибочное решение, что в i -ый момент автомат находился в состоянии σ , а в действительности он там не находился. Ошибка такого рода имеет одну значимость в случае, если автомат находился в состоянии σ в $i-1$ -ый или в $i+1$ -ый моменты времени, и совсем другую значимость, если автомат не находился в состоянии σ на протяжении достаточно большого интервала времени - окрестности i -ого момента.

Я вижу, таким образом, что формулировка задачи сегментации должна начинаться с тщательного определения штрафов $W(I', I'')$, которые адекватно выражают, насколько опасно в данном приложении, что решение I'' о сегментации принимается вместо сегментации I' , которая имеет место в действительности. Этим, собственно говоря, формулировка задачи и закончится, так как после задания функции потерь задача будет состоять в

минимизации математического ожидания потерь. Алгоритм сегментации, полученный в результате решения именно таким образом поставленной задачи, представлял бы несомненный интерес. Но это и есть то, что сейчас выходит за рамки моих возможностей. Я догадываюсь, что при тщательной продуманной функции потерь W даже простое вычисление ее значения $W(I', I'')$ для заданных двух сегментаций окажется не совсем тривиальным. Однако для решения задачи распознавания надо понимать не только функцию штрафов W , но и математическое ожидание штрафа $W(I', I'')$ по сегментации I' . Это математическое ожидание зависит от сегментации I'' , и эту функцию от I'' надо изловчиться выразить в таком виде, чтобы затем удалось находить ее минимальное значение. Решенная именно в таком духе задача сегментации представляла бы, по моему мнению, значительный интерес.

Не исключено, что Ты сможешь пройти тот путь, который Ты наметил. Как было бы замечательно, если бы в конце пути Ты был вознагражден и, вопреки Твоим опасениям, решение правильно сформулированной задачи оказалось бы проще, чем полученное Тобой решение задачи, которая является чересчур примитивной формализацией прикладной задачи сегментации.

Вряд ли я пойду по тому пути, который я указал, и совсем не потому, что себя недооцениваю. Более серьезная причина состоит в уважении к тому, что сейчас делается в подавляющем большинстве исследований по распознаванию, и в частности, в сегментации, и к тому, что никто не идет по тому пути, который сейчас представляется мне естественным. Это не может быть простой случайностью.

Первоисточником всех новых формулировок задач, которые мы разобрали, и их решений явилось ясное понимание того бесспорного факта, что наиболее вероятное значение случайной величины - это нечто совсем иное, чем ее действительное значение. Отсюда следует, что если надо оценить значение какой-то характеристики случайного объекта, то следует принимать во внимание не только наиболее вероятный объект, но и все прочие объекты с меньшей вероятностью. Хотя эта идея и проходила рефреном через многие лекции, по настоящему я понял ее только сейчас.

Но тогда почему в подавляющем большинстве прикладных задач не делается ничего другого, чем поиск наиболее вероятной последовательности скрытых параметров? После того, как она найдена, с ней поступают так, будто она совпадает с действительной. Если бы этот последний ход был правомерен, то все рассмотренные нами задачи, включая и последнюю задачу о сегментации, сводились бы к отысканию наиболее вероятной последовательности. Однако этот ход неправомерен. Но если почти все так делают, то этому должно быть более серьезное объяснение, чем простое утверждение, что так делать нельзя. При отсутствии такого объяснения я не осмеливаюсь покинуть протоптанную тропку, по которой ходят все. Я могу показаться вам излишне консервативным, но в конце концов, кон-

сервативность - не самое плохое качество. В данном случае это уважение к признанным достижениям в распознавании и боязнь разрушить то, что уже достигнуто. Я поневоле затронул более широкие вопросы, но если вернуться в русло нашей беседы, то зачем мне нужно придумывать новые пути в сегментации, если в этом направлении уже и без меня так много сделано?

Очень приятно видеть Твое уважение к работе Твоих предшественников в распознавании и поэтому на Твой прямой вопрос не можем серьезно ответить. Но не очень серьезно - можем, сославшись на одесский фольклор.

Некто заказал у портного брюки, а портной выполнил этот заказ только через неделю. Заказчику показался этот срок слишком длинным, и он упрекнул портного, напомнив ему, что одной недели хватало всевышнему для сотворения всего мира. На это портной лишь ответил: "Да, но Вы только посмотрите на этот мир и посмотрите на эти брюки". Так вот, посмотри на это необозримое множество алгоритмов сегментации и. . .

Спасибо.

Я бы хотел еще с вами обсудить вторую часть лекции, посвященную обучению и самообучению. Меня захватил уровень общности и абстрактности, на котором выполнен анализ проблем и вскрыты отношения между разными задачами. Я имею в виду, например, соотношение между минимаксным и правдоподобным оцениванием, которое было обнаружено, не прибегая к алгоритмам решения этих задач. Совершенно неожиданной для меня оказалась возможность настройки алгоритмов распознавания марковских последовательностей с помощью алгоритмов персептрона или Козинца. Я уже не говорю о взаимосвязи обучения и самообучения, с которой я уже познакомился по лекции 6. Сейчас я еще раз убедился в ее плодотворности, когда увидел, как задача оценки параметров скрытой марковской последовательности, первоначально представляющаяся головокружительно сложной, преобразуется в настолько простую задачу, что об этом не стоит и говорить. Когда я вижу манипуляции с такими обширными пластами задач (а я надеюсь, что я эти манипуляции понял), я как бы начинаю чувствовать в своих руках отбойный молоток, которым я смогу долбить скалу, называемую распознаванием. Конечно, это еще совсем не динамит, которым можно эту скалу взорвать, но это уже не перочинный ножик, которым я ковырял эту скалу до сих пор.

С определенным опасением ожидаем, что скажешь дальше. После таких восторгов обязательно следует нечто вроде "тем не менее", или "в то же время", или нечто подобное.

Это действительно так. В лекции тщательно исследована только одна сторона оценки статистической модели объекта, с моей точки зрения, не самая главная. Очень много внимания было уделено оценке численных параметров статистической модели, но полностью оставлен в стороне вопрос

об отыскании структуры сложного объекта, который мне кажется самым важным. Поясню, что я имею в виду. Когда мы знаем, что придется распознавать объект, состоящий из частей, то прежде всего нужно упорядочить эти составные части так, чтобы они образовали последовательность и чтобы оказалось возможным применять методы, описанные в лекции. Такое упорядочение иногда не вытекает непосредственно из прикладного содержания задачи и поэтому не задано заранее. Даже когда последовательность $k_0, k_1, k_2, \dots, k_i, \dots$ есть процесс во времени, эта упорядоченность может быть той или иной.

Пусть, например, индекс i обозначает время, а k_i обозначает поведение определенного человека в i -ый день. Только при беглом взгляде может показаться, что поведение человека сегодня полностью зависит от его поведения вчера. Представим себе однако, что человек имеет две различные сферы интересов: все будние дни он проводит на работе, а выходные - на даче. В этом случае его поведение в субботу в большей степени определяется тем, чем он занимался на даче в предыдущее воскресенье, а не тем, чем он занимался вчера на работе. Наряду с естественным упорядочением во времени здесь имеет место и другая упорядоченность, которая априори не задана.

В теории обучения, как она изложена в лекции, слишком сильное ударение делается на оценку численных параметров марковской последовательности и совсем не затрагивается вопрос, как эту последовательность следует сформировать. Я считаю, что именно структура взаимной зависимости составных частей сложного объекта является самой главной его характеристикой и обнаружение этой структуры по экспериментальным данным является неотъемлемой частью структурного распознавания.

Я попытался самостоятельно восполнить этот пробел в вашей лекции, но пришел к пессимистическому выводу, что разумно сформулированная задача о поиске наилучшей структуры эквивалентна известной задаче коммивояжера. Я знаю об очень плохой репутации этой задачи, знаю, что задачи этого типа не поддаются решению за приемлемое время. Таким образом, налицо явный отрицательный результат моих попыток. Я бы хотел у вас спросить, действительно ли дела здесь обстоят настолько плохо. Возможно, задача о нахождении структуры обладает какой-то спецификой, в силу которой ей соответствует не задача коммивояжера во всей ее общности, а некоторый частный ее подкласс, для которого возможен общий алгоритм решения?

Пожалуй, ситуация здесь абсолютно безнадежна. Мы даже не будем спрашивать Тебя, каким образом Ты сводил задачу к задаче коммивояжера. Мы неоднократно пробовали разобраться в вопросах, которые Ты сейчас поднял, и всякий раз наталкивались на необходимость решения либо задачи коммивояжера, либо родственной с ней задачи о гамильтоновом цикле.

Прошу прощения, я не знаю, что такое гамильтонов цикл. Вас не затруднит объяснить мне это?

Эту задачу обычно формулируют следующим наглядным образом. Предположим, что Ты пригласил множество I людей на банкет. Некоторые из этих людей знакомы друг с другом, а некоторые - нет. Это можно выразить с помощью неориентированного графа с множеством I вершин. Две вершины в этом графе соединены ребром тогда и только тогда, когда эти две вершины соответствуют двум Твоим гостям, которые знакомы друг с другом. Тебе нужно рассадить гостей за круглым столом так, чтобы любые два человека, сидящие рядом за столом, оказались знакомы друг с другом. В графовой терминологии это обозначает отыскание цикла в графе, в который каждая вершина входит один и только один раз. В некоторых графах такой цикл содержится, и такие графы называются гамильтоновыми. Задача состоит в построении алгоритма, который для каждого поданного на его вход графа определяет, является ли он гамильтоновым или нет. При положительном ответе на этот вопрос требуется также указать цикл, проходящий через каждую вершину один и только один раз.

И эту ерунду невозможно решить?

Еще хуже. До сих пор никто не предложил алгоритм, решающий эту задачу за полиномиальное время, но никто и не доказал, что такого алгоритма нет. Искренне советуем Тебе не пытаться разобраться с этими проблемами. Это пропасть, подобная теореме Ферма.

С той существенной разницей, что эта теорема уже доказана.

Этого мы даже не знали. Ну что ж, подождем пару сотен лет, пока будет внесена ясность в отыскании гамильтоновых циклов.

А что связывает задачу коммивояжера с задачами о гамильтоновых циклах?

Пусть на множестве ребер графа определена неотрицательно определенная вещественнозначная функция. Значение этой функции на определенном ребре назовем весом этого ребра. Задача коммивояжера состоит в отыскании гамильтоновского цикла с минимальной суммой весов входящих в него ребер.

Но ведь задача о поиске наилучшей структуры свелась у меня не к поиску наилучшего цикла, проходящего через все вершины графа, а к поиску наилучшей цепочки, проходящей через все вершины. Возможно, такая задача не является вычислительно безнадежной?

Тебе достаточно подумать всего лишь несколько минут, чтобы увидеть, что это задачи одинаковой сложности.

Очень жаль. Сведение поиска наилучшей структуры к задаче коммивояжера не совсем тривиально и потребовало определенных усилий, которые оказались напрасными.

Еще больше Ты бы сожалел, если бы узнал, что Ты находился на расстоянии вытянутой руки от прекрасной задачи, которую тоже можно понимать как поиск наилучшей структуры, но которая вполне разрешима. Скажи, пожалуйста, зачем Тебе нужно было формулировать задачу как упорядочение составных частей непременно в виде последовательности? Ведь в лекции мы специально обратили внимание на то, что большая часть результатов, полученных для последовательностей, оказывается справедливой и для более общего вида структур. . .

Я понял!!! Рассмотренные в лекции задачи распознавания оказываются конструктивно разрешимыми и в случае, когда структура объекта образует ациклический граф. Поэтому я могу сейчас попробовать еще раз рассмотреть задачу построения наилучшей структуры, отказавшись от требования, что это должна быть последовательность, а заменив его более слабым требованием, что она должна быть деревом. Возможно, в этом случае я приду к задаче, которая окажется конструктивно разрешимой.

Да, Ты прав, но теперь уже никуда не торопись, потому что Ты вышел на задачу, для решения которой потребовалось собрать результаты, получаемые в течение почти столетнего периода. В 1968 американец Чау (Chow) [Chow, 1965; Chow and Liu, 1968] сформулировал задачу, которую мы теперь понимаем как построение наилучшей структуры. Он показал, что сформулированная им задача сводится к определенной задаче из теории графов, сформулированной 1889 математиком Кейли (Cauley) [Cauley, 1889]. Не путай эту фамилию с Келли, который также является выдающимся исследователем теории графов. Задача Кейли, известная сейчас как задача о минимальной связной сети, длительное время считалась трудно разрешимой. Прошло без малого 40 лет от опубликования задачи до ее блестящего решения моравским математиком Борувкой [Boruvka, 1926]. Решение Борувки до сих пор поражает своей безукоризненной элегантностью. К сожалению, оно не стало общеизвестным, и еще много лет спустя и даже совсем недавно настойчиво появлялись новые и новые статьи, посвященные решению задачи Кейли. Сейчас эти три потока, сформированные работами Кейли, Борувки и Чау, успешно встретились друг с другом.

А сейчас старательно сформулируй интересующую Тебя задачу и укажи путь к ее решению, то есть повтори то, что сделал Чау.

Пусть I - конечное множество составных частей сложного объекта. Каждая часть $i \in I$ характеризуется скрытым параметром k_i и наблюдаемым параметром x_i . Совокупность $\bar{k} = (k_i, i \in I)$ образует ненаблюдаемое состояние объекта в целом, а $\bar{x} = (x_i, i \in I)$ - результат наблюдения объекта.

Как и прежде, будем считать, что параметры k_i , $i \in I$, принимают значения из конечного множества K , а параметры x_i , $i \in I$, - из конечного множества X .

Множество I будем понимать как множество вершин определенного неориентированного дерева. Множество ребер этого дерева обозначим G . Запись $(i, j) \in G$ будем понимать как обозначение того, что вершины i и j , $i \in I$, $j \in I$, соединены ребром из G . Предположим, что при фиксированном параметре k_i , характеризующем i -ую часть, наблюдаемый параметр x_i не зависит ни от каких других параметров объекта. В силу этого $p(\bar{x} | \bar{k}) = \prod_{i \in I} p_i(x_i | k_i)$. Будем считать также, что распределение вероятностей $p(\bar{k})$ является марковским на дереве, сформированном множеством G ребер,

$$p(\bar{k}) = \frac{\prod_{(i,j) \in G} g_{ij}(k_i, k_j)}{\prod_{i \in I} (g_i(k_i))^{h_i-1}}, \quad (8.148)$$

где h_i обозначает множество ребер, выходящих из вершины i . Совместная вероятность $p(\bar{x}, \bar{k})$ равна

$$\begin{aligned} p(\bar{x}, \bar{k}) &= \frac{\prod_{(i,j) \in G} g_{ij}(k_i, k_j) \prod_{i \in I} p_i(x_i | k_i)}{\prod_{i \in I} (g_i(k_i))^{h_i-1}} \\ &= \frac{\prod_{(i,j) \in G} g_{ij}(k_i, k_j) \prod_{i \in I} p_i(x_i | k_i) g_i(k_i)}{\prod_{i \in I} (g_i(k_i))^{h_i}} \\ &= \frac{\prod_{(i,j) \in G} g_{ij}(k_i, k_j) \prod_{i \in I} p_i(x_i, k_i)}{\prod_{i \in I} (g_i(k_i))^{h_i}}. \end{aligned} \quad (8.149)$$

В формулах (8.148) и (8.149) число $g_{ij}(k, k')$, $k \in K$, $k' \in K$, обозначает совместную вероятность того, что скрытый параметр i -ой части примет значение k и скрытый параметр j -ой части примет значение k' . Число $p_i(x, k)$, $x \in X$, $k \in K$, представляет совместную вероятность того, что наблюдаемый и скрытый параметры i -ой части примут значения x и k соответственно. Это значит, что

$$g_i(k) = \sum_{k' \in K} g_{ij}(k, k')$$

для всех j , таких, что $(i, j) \in G$, и в то же время

$$g_i(k) = \sum_{x \in X} p_i(x, k).$$

Статистическая модель распознаваемого объекта определяется совокупностью функций $(p_i, i \in I)$, множеством G ребер дерева и совокупностью функций $(g_{ij}, (i, j) \in G)$. Если известна статистическая модель, то можно формулировать различные статистические задачи распознавания скрытых параметров объекта на основе знания наблюдаемых параметров. Если же эта модель не известна, то ее следует оценить на основе экспериментального исследования объекта. Я рассмотрю задачу наиболее правдоподобного оценивания модели на основе случайной обучающей выборки (\bar{x}^j, \bar{k}^j) ,

$j = 1, \dots, l$, пар, имеющих распределение вероятностей $p(\bar{x}, \bar{k})$. Это значит, что я займусь сейчас задачей обучения в ее простейшей постановке. Как было показано в лекции, эта задача является в определенном смысле базовой. Как только я построю алгоритм решения этой задачи, я смогу чисто формально построить алгоритмы минимаксного оценивания или наиболее правдоподобного оценивания в режиме самообучения.

Пусть числа $n(\bar{x}, \bar{y})$ обозначают, сколько раз пара (\bar{x}, \bar{k}) имела место в эксперименте, то есть, в обучающей выборке. Пусть

$$\alpha(\bar{x}, \bar{y}) = \frac{n(\bar{x}, \bar{y})}{l}.$$

В решаемой задаче требуется найти множество G ребер дерева с вершинами в I , и совокупность $(g_{ij}, (i, j) \in G)$, $(p_i, i \in I)$ функций, которые максимизируют вероятность обучающей выборки $((\bar{x}^j, \bar{k}^j), j = 1, 2, \dots, l)$,

$$(G^*, (p_i^*, i \in I), (g_{ij}^*, (i, j) \in G^*)) \quad (8.150)$$

$$\begin{aligned} &= \operatorname{argmax}_G \max_{(p_i, i \in I)} \max_{(g_{ij}, (i, j) \in G)} \sum_{\bar{x}, \bar{k}} \alpha(\bar{x}, \bar{k}) \log p(\bar{x}, \bar{k}) \\ &= \operatorname{argmax}_G \max_{(p_i, i \in I)} \max_{(g_{ij}, (i, j) \in G)} \sum_{\bar{x}, \bar{k}} \alpha(\bar{x}, \bar{k}) \log \frac{\prod_{(i, j) \in G} g_{ij}(k_i, k_j) \prod_{i \in I} p_i(x_i, k_i)}{\prod_{i \in I} (g_i(k_i))^{h_i}}. \end{aligned}$$

Выполним некоторые эквивалентные преобразования максимизируемой функции (8.150)

$$\begin{aligned} &\sum_{\bar{x}, \bar{k}} \alpha(\bar{x}, \bar{k}) \log \frac{\prod_{(i, j) \in G} g_{ij}(k_i, k_j) \prod_{i \in I} p_i(x_i, k_i)}{\prod_{i \in I} (g_i(k_i))^{h_i}} \\ &= \sum_{\bar{x}, \bar{k}} \alpha(\bar{x}, \bar{k}) \log \prod_{(i, j) \in G} g_{ij}(k_i, k_j) \\ &\quad + \sum_{\bar{x}, \bar{k}} \alpha(\bar{x}, \bar{k}) \log \prod_{i \in I} p_i(x_i, k_i) - \sum_{\bar{x}, \bar{k}} \alpha(\bar{x}, \bar{k}) \log \prod_{i \in I} (g_i(k_i))^{h_i} \\ &= \sum_{(i, j) \in G} \sum_{k_i \in K} \sum_{k_j \in K} \alpha_{ij}(k_i, k_j) \log g_{ij}(k_i, k_j) \\ &\quad + \sum_{i \in I} \sum_{x_i \in X} \sum_{k_i \in K} \beta_i(x_i, k_i) \log p_i(x_i, k_i) - \sum_{i \in I} h_i \sum_{k_i \in K} \alpha_i(k_i) \log g_i(k_i), \end{aligned} \quad (8.151)$$

где

$$\begin{aligned}\alpha_{ij}(k_i, k_j) &= \sum_{(x_{i'}, i' \in I)} \sum_{(k_{i'}, i' \in I \setminus \{i, j\})} \alpha(\bar{x}, \bar{k}), \\ \beta_i(x_i, k_i) &= \sum_{(x_{i'}, i' \in I \setminus \{i\})} \sum_{(k_{i'}, i' \in I \setminus \{i\})} \alpha(\bar{x}, \bar{k}), \\ \alpha_i(k_i) &= \sum_{(x_{i'}, i' \in I)} \sum_{(k_{i'}, i' \in I \setminus \{i\})} \alpha(\bar{x}, \bar{k}).\end{aligned}$$

Последние три формулы приводятся лишь для того, чтобы показать правомерность вывода те (8.151). В действительности, числа α_{ij} , α_i и β_i вычисляются не по приведенным формулам, а значительно проще. Число $\alpha_{ij}(k, k')$ обозначает, насколько часто в эксперименте имело место событие, что i -ый скрытый параметр принимал значение k , а j -ый скрытый параметр - значение k' . Подобный смысл имеют числа α_i и β_i .

На основе тех же соображений, что и в лекции при доказательстве теоремы 8.1, можно доказать, что числа $g_{ij}(k_i, k_j)$, $p_i(x_i, k_i)$ и $g_i(k_i)$, максимизирующие (8.151), равны соответственно числам $\alpha_{ij}(k_i, k_j)$, $\beta_i(x_i, k_i)$ и $\alpha_i(k_i)$. Это значит, что требование (8.150) принимает вид

$$\begin{aligned}G^* = \operatorname{argmax}_G & \left(\sum_{(i,j) \in G} \sum_{k_i \in K} \sum_{k_j \in K} \alpha_{ij}(k_i, k_j) \log \alpha_{ij}(k_i, k_j) \right. \\ & \left. + \sum_{i \in I} \sum_{x_i \in X} \sum_{k_i \in K} \beta_i(x_i, k_i) \log \beta_i(x_i, k_i) - \sum_{i \in I} h_i \sum_{k_i \in K} \alpha_i(k_i) \log \alpha_i(k_i) \right).\end{aligned}\tag{8.152}$$

В этом выражении от множества G зависит первое слагаемое, так как оно представляет собой сумму по парам (i, j) , входящим в это множество. От G зависит и третье слагаемое, так как числа h_i зависят от G . Второе слагаемое от G не зависит и поэтому требование (8.152) равносильно требованию

$$\begin{aligned}G^* = \operatorname{argmax}_G & \left(\sum_{(i,j) \in G} \sum_{k_i \in K} \sum_{k_j \in K} \alpha_{ij}(k_i, k_j) \log \alpha_{ij}(k_i, k_j) \right. \\ & \left. - \sum_{i \in I} h_i \sum_{k_i \in K} \alpha_i(k_i) \log \alpha_i(k_i) \right).\end{aligned}$$

Если ввести в рассмотрение энтропии

$$H_{ij} = - \sum_{k_i \in K} \sum_{k_j \in K} \alpha_{ij}(k_i, k_j) \log \alpha_{ij}(k_i, k_j)\tag{8.153}$$

и

$$H_i = - \sum_{k_i \in K} \alpha_i(k_i) \log \alpha_i(k_i)\tag{8.154}$$

то получается, что

$$\begin{aligned} G^* &= \operatorname{argmax}_G \left(\sum_{i \in I} h_i H_i - \sum_{(i,j) \in G} H_{ij} \right) \\ &= \operatorname{argmax}_G \sum_{(i,j) \in G} (H_i + H_j - H_{ij}) . \end{aligned}$$

В конечном итоге получается следующая процедура построения множества G , то есть построения ациклической структуры зависимостей между частями сложного объекта.

1. По обучающей выборке (k^j , $j = 1, \dots, l$) (данные x^j не принимаются во внимание вообще) вычисляются числа $\alpha_{ij}(k_i, k_j)$ и $\alpha_i(k_i)$, $i \in I$, $j \in I$, $i \neq j$, $k_i \in K$, $k_j \in K$.
2. Для каждой пары (i, j) , $i \in I$, $j \in I$, $i \neq j$, вычисляется энтропия H_{ij} по формуле (8.153). Для каждого $i \in I$ вычисляется энтропия H_i по формуле (8.154).
3. Множество I отождествляется с множеством вершин полного графа, в котором каждая вершина соединена с каждой. Ребру, соединяющему вершины i и j , присваивается длина $H_i + H_j - H_{ij}$.
4. В полученном графе следует найти ациклический подграф (то есть, дерево), содержащий все вершины исходного графа и максимизирующий суммарную длину входящих в него ребер.

Теперь я спрашиваю, свел ли я задачу построения наилучшей структуры к задаче Кейли?

Да, и почти так, как это сделал Чау.

А можете ли вы объяснить мне основную идею Борувки решения этой задачи?

Да, с удовольствием. Основная идея решения задачи гениально проста, и можно только удивляться, почему так долго задача Кейли оставалась нерешенной и почему прекрасные идеи Борувки так долго оставались неизвестными.

Пусть M - множество ребер графа, то есть множество пар (i, j) , $i \in I$, $j \in I$, $i \neq j$. Пусть множество M упорядочено по убыванию длин ребер: если ребро (i', j') предшествует ребру (i'', j'') , то длина ребра (i', j') не меньше длины ребра (i'', j'') . Мы ищем связный подграф G^* , являющийся деревом и максимизирующий суммарную длину ребер.

Подграф G^* строится следующим образом. Просматриваются ребра из множества M в том порядке, как они записаны в множестве M , то есть по мере убывания их длины. Первое ребро из множества M , то есть ребро с наибольшей длиной, включается в граф G^* . Затем о каждом последующем ребре (i', j') принимается решение, включать ли его в G^* или нет. Если в

текущем графе G^* существует путь от вершины i' к вершине j' , то ребро (i', j') не включается в граф G^* . Если такой путь существует, то (i', j') включается в G^* .

Понятно, что на каждом шаге указанной процедуры граф G^* не будет содержать циклов, а в конце процедуры он будет состоять из $|I| - 1$ ребер и включать в себя все вершины из множества I .

Простота этого алгоритма невероятна. Интересно, какая же основная идея доказательства, что полученное в конечном итоге дерево оптимально.

Это можешь сделать уже сам. Можешь упростить себе задачу и рассмотреть только ситуацию, когда в множестве M нет двух ребер с одинаковой длиной.

Обоснование алгоритма Борувки состоит в справедливости двух довольно очевидных утверждений.

Утверждение 8.1 Пусть (i_0, j_0) - самое длинное ребро в множестве M , а G^* - граф, решающий задачу Кейли. В таком случае ребро (i_0, j_0) содержится в G^* . ▲

Доказательство. Я докажу, что если (i_0, j_0) не содержится в G^* , то G^* не является решением задачи Кейли. Поскольку G^* - связный граф, в нем содержится путь от i_0 до j_0 . Каждое ребро на этом пути короче, чем ребро (i_0, j_0) . Выберем любое ребро на этом пути и обозначим его (i', j') . Рассматриваемый путь совместно с ребром (i_0, j_0) образует цикл. Исключим из графа G^* ребро (i', j') и включим в него ребро (i_0, j_0) . Вновь полученный граф останется связным деревом, содержащим все вершины из I , но суммарная длина его ребер увеличится, потому что длина ребра (i', j') меньше длины ребра (i_0, j_0) . ■

Утверждение 8.2 Пусть G^* - граф, решающий задачу Кейли, а G' - его подграф. Пусть (i_0, j_0) ребро, удовлетворяющее следующим условиям:

1. В подграфе G' отсутствует путь из i_0 в j_0 .
2. Среди всех ребер, обладающих первым свойством, ребро (i_0, j_0) имеет наибольшую длину.

В таком случае ребро (i_0, j_0) содержится в G^* . ▲

Доказательство. Предположим, что ребро (i_0, j_0) не содержится в G^* . Поскольку G^* - связный граф, в нем существует путь от i_0 до j_0 . В силу первого условия имеется по крайней мере одно ребро на этом пути, которое не содержится в G' . В силу второго условия длина этого ребра меньше, чем длина ребра (i_0, j_0) . Дальнейшие соображения те же, что и при доказательстве утверждения 8.1. ■

Нам приятно видеть, что Ты вполне прилично усвоил содержание этой очень простой лекции. Благодарим Тебя за обсуждение.

Прошу прощения, но я не выяснил для себя еще один вопрос, который для меня важен. Я не уверен, что я понимаю ценность матричных представлений, которые проходят лейтмотивом в первой части лекции. Судя по настойчивости, с которой в лекции приводятся эти матричные представления, здесь есть нечто важное, что ускользывает от меня. Я совсем не уверен, что матричные представления являются неизбежным средством анализа рассмотренных задач. Я эти задачи понимаю и могу их исследовать и решать, не прибегая к матричным обозначениям вообще. Прав ли я, что здесь речь идет лишь о лаконичной записи определенных результатов, которые тем или иным путем были получены? Но если я прав, то так ли уж существенно, какой язык или какая математическая символика применяется для выражения тех или иных знаний? Не является ли значительно более существенным знание само по себе, то есть тот инвариант, который не зависит от формы, в которой оно представлено? Для чего мне может понадобиться новый формализм, если то, что он выражает, я могу выразить с помощью обычных и привычных для меня средств?

Твои вопросы можно понимать в узком и широком смысле. Матричное представление задач и алгоритмов мы использовали, как одно из многих возможных. Видишь ли, мы намерены опубликовать наши лекции, следовательно, пишем их не только для Тебя, хотя сотрудничество с Тобой нам очень приятно. Мы хотели бы, чтобы содержание наших лекций было понятно как можно большему кругу читателей. Некоторым более понятна графовая интерпретация задач, другим - статистическая, а для третьих, возможно, именно матричная интерпретация окажется наиболее выразительной. Что касается нас, авторов, то мы усматриваем основную ценность матричного представления в том, что оно дает возможность единообразного представления задач, которые до сих существовали как бы изолировано друг от друга, позволяет находить то общее, что эти, казалось бы, разрозненные задачи объединяет.

В отличие от Тебя мы считаем очень важным, на каком языке излагаются те или иные знания. Возможно, главным признаком незрелости сегодняшней науки о распознавании, а может быть, и причиной этой незрелости является именно отсутствие своего собственного языка, приспособленного для формулировки специфических проблем распознавания, а не проблем теории графов или математического программирования. Мы переходим, таким образом, к рассмотрению Твоих вопросов в широком смысле: так ли уж важен язык, в котором выражаются новые знания? Насколько мы поняли, Твой ответ на этот вопрос однозначно отрицательный. Мы долго думали над этим, и когда формировали лекцию, и после Твоих вопросов, и пришли к выводу, что мы с Тобой не согласны. Было бы хорошо, если бы мы смогли Тебя в чем-то убедить, и Твоя точка зрения приблизилась бы к нашей.

Не будем ходить слишком далеко и рассмотрим близкий к нам пример - алгоритм Борувки для решения задачи Кейли. Для того, чтобы опи-

сать этот алгоритм, нам потребовалось чуть больше половины страницы. Приведенное Тобой доказательство его правильности поместилось в одну страницу. Приблизительно так же лаконично излагается этот материал в современных учебниках по теории графов. А теперь возникает интересный вопрос: что же еще содержится в статье Боровки, где эти результаты изложены на 16 страницах? Мы прочитали эту статью и советуем и Тебе ее прочитать, потому что это очень поучительно. Во первых, Ты увидишь, что в этой статье не содержится больше ничего. В ней содержится только очень строгое описание алгоритма и ее доказательство, не разбавленное никакими лишними рассуждениями, которые имели бы цель увеличить объем статьи. Во-вторых, и это более важно, когда Ты начнешь читать статью Боровки, Твоим первым впечатлением будет, что мы ошиблись и дали Тебе ссылку совсем на другую статью, которая не имеет ничего общего с проблемой Кейли. Произойдет это потому, что во всей статье ни разу не встречается слово граф и используется совсем другая символика, за которой не так уж легко будет найти то, что Тебе сейчас уже известно. Возможно, именно из-за того, что статья написана на необычном языке (мы имеем в виду совсем не чешский язык), ее результаты столь длительное время оставались неизвестными и многократно повторялись другими авторами. Когда результаты Боровки стали все же известны, это имело уже скорее историческое, а не научное значение. Ситуация здесь подобна открытию Америки, как мы об этом упоминали в лекции 6.

В развитии любой научной дисциплины время от времени наступает этап, когда получение новых результатов как-бы приостанавливается. Происходит это по разным причинам, но иногда еще и потому, что представление новых результатов в рамках прежнего аппарата становится все более и более громоздким, пока наконец становится невозможным, потому что отсутствует символика, отсутствует, если хочешь, язык, позволяющий лаконично выражать знания. На этом этапе создание новых языковых средств для представления знаний является, возможно, более важным, чем получение новых знаний. Когда известные знания будут представлены в новой терминологии и люди к ней привыкнут, они станут удивляться, почему такие простые вещи раньше казались такими замысловатыми.

Твое неположительное отношение к матричной символике напомнило нам еще одно обстоятельство, которое неизбежно затрудняет введение в обиход новых языковых средств. Существование того или иного языка мощно определяет круг знаний, которые становятся общеизвестными. Это прежде всего знания, которые лаконично выражаются старыми языковыми средствами. Это создает очень неблагоприятные условия для принятия новых языковых средств, потому что именно общеизвестные знания выглядят в новой терминологии не совсем обычно и для них старая терминология вполне приемлема. Ты это прекрасно выразил словами "Зачем мне нужно что-то новое, если я и без этого понимаю все, что мне нужно".

В свое время нас очень интересовало, как это могло случиться, что еще совсем недавно, каких-нибудь тысячу лет назад, люди не умели складывать и умножать произвольные целые числа. Причина этого была не в

низком уровне образованности вообще. Причина здесь была более глубокая. Знаниями такого рода не обладал никто. Способность складывать и умножать некоторые большие числа считались несомненным признаком интеллектуальности, а формулировка общих правил выполнения этих операций для некоторых подклассов чисел признавалась достойным объектом научных исследований. С точки зрения сегодняшнего дня трудно понять такую ситуацию, когда известно, как складывать какие-то одни числа, и неизвестно, как складывать другие. Ведь речь идет об одних и тех же операциях. Было бы неправильно объяснять такую ситуацию тем, что наши предки были менее интеллектуальные, чем мы. Хотя человечество в целом сейчас более образовано, чем тысячу лет назад, нет никаких свидетельств, что наши прародители уступали нам по своим интеллектуальным способностям. Для этого достаточно вспомнить блестящие результаты европейской античной математики, откуда происходят понятия простого числа, наименьшего общего кратного, наибольшего общего делителя. Все это свидетельствует о глубоком понимании сущности целых чисел. Да и сам факт понимания таких операций, как сложение и умножение, свидетельствует о высокой математической культуре, несмотря на неумение выполнять эти операции для любых чисел.

Все это происходило потому, что не было единой системы представления чисел, которая нам сейчас известна, как запись чисел в арабской форме. Разные народы пользовались различными формами, более того, одно и то же количество могло записываться по-разному в зависимости от того, о количестве каких предметов шла речь. Из этих разнообразных форм до наших дней сохранилась запись количества в виде римских чисел. А ведь догадаться до алгоритма сложения и умножения римских чисел действительно не очень легко. В такой неразберихе вряд ли кому-то приходила мысль о возможности единообразного манипулирования числами вне зависимости от того, идет ли речь о количестве секунд или количестве овец. Ведь числа, которыми представлялись эти количества, казались совершенно разными. Не напоминает ли Тебе эта ситуация современное положение в распознавании образов?

Слегка напоминает, но я с нетерпением ожидаю, к чему вы приведете свои рассуждения, тем более, что они не совсем правильные. Люди умели манипулировать целыми числами значительно раньше упоминаемого вами периода. Уже много тысяч лет назад правила сложения и умножения чисел были известны в государствах Древнего Двуречья, а возможно, еще раньше в других цивилизациях.

Не будь на этот раз слишком строгим. Ведь мы сейчас занимаемся не исследованием истории математики, а чем-то совсем другим. Но если Ты уж не хочешь нам простить никаких вольностей, то скажем только, что еще тысячу лет назад определенные цивилизованные народы на обширных территориях такими знаниями не обладали.

Неразбериха с целыми числами имела место и в городе Самарканде, который в то время занимал центральное положение в среднеазиатской науке. Эта неразбериха продолжалась до тех пор, пока в Самарканд не пришел некто Мухамед ал-Хорезми из соседнего Хорезма. Он сообщил о новом виде записи целых чисел, использование которой позволяет выполнять умножение и сложение целых чисел. В силу этого метода операции с целыми числами перестали быть прерогативой отдельных интеллектуалов, а стали доступны любому мальчишке из улицы. Мухамед ибн Муса ал-Хорезми рассказал самаркандцам именно о том представлении целых чисел и операций с ними, которым до сих пор пользуются во всем мире. Этот результат входит в число немногих результатов, которые оказали мощное многовековое воздействие на развитие науки во всем мире. Однако это не единственный итог деятельности Мухамеда ибн Мусы. Он ввел в обиход неизвестный до тех пор способ определения количества, при котором интуиция, здравый смысл и сообразительность заменяется дисциплинированным выполнением ряда правил, которые настолько просты, что их может реализовать даже механическое устройство. В честь Мухамеда ал-Хорезми (а может быть, в честь города, откуда прибыл Мухамед) этот метод стали называть хорезмским методом, или методом ал-Хорезми. При последующем многовековом использовании этого термина он превратился в слово алгоритм.

Таким образом, Ты видишь, какие мощные, воистину тектонические сдвиги в мировой науке происходят от одного только введения новых обозначений для уже известных понятий. Поэтому мы никак не согласны с Тобой, что не имеет большого значения, в каком формализме, на каком языке формулируются те или иные знания, как новые, так и старые.

И все же хотелось бы добавить еще что-то. Личность Мухамеда ал-Хорезми настолько величественная, что даже попытка брать с него пример была бы непростительной нескромностью. И все же постарайся представить себя самого на его месте со всеми неизбежными подробностями, которые сопровождают рождение новых знаний. Если Ты это представишь себе достаточно реалистично, Ты непременно придешь к выводу, что жилось нашему любезному Мухамеду довольно отвратительно. Уж сколько глупостей пришлось ему выслушать на своей жизни!

Кому-то не нравился сам способ представления чисел. К примеру, запись 247 представлялась значительно менее наглядной, чем запись CCXLVII. Ведь непосредственно по записи CCXLVII видно, что речь идет о сумме $C + C + (L - X) + V + I + I$, а запись 247 ни о чем таком непосредственно не говорит. Чтобы найти, что обозначает эта запись, необходимо вычислить $2 \cdot 10^2 + 4 \cdot 10^1 + 7 \cdot 10^0$. Так что, господин коллега ал-Хорезми, вместо того, чтобы умножать числа только тогда, когда нам это нужно, надо будет умножать числа всякий раз, когда нам надо только узнать, о каком числе идет речь? Другому представлялся ненаглядным новый способ сложения чисел. Ведь по самой записи числа VII видно, что здесь представлена сумма чисел V и II. А во внешнем виде числа 7 никак не просматривается, что это число есть сумма 5 и 2. Еще кому-то не нравилось, что для записи

чисел в новой системе потребуются 10 символов 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, в то время как в старой системе требовались лишь VII символов I, V, X, L, C, D, M. Правда, с помощью этих символов невозможно выразить число, большее, чем MMMCMXCIX, но ведь для большинства практических задач это достаточно, а придумывать что-то для представления чисел, больших, чем MMMCMXCIX, это уже витание в облаках, наука для науки.

Наконец, правила умножения чисел представлялись всем очень громоздкими. Ведь нужно выучить наизусть и всегда помнить 100 правил, то есть, таблицу умножения. Но если уж произведения сотни пар чисел непременно нужно знать, то нет необходимости знать правила вычисления других произведений. На практике достаточно знать лишь указанную сотню произведений, а остальные определять на основании здравого смысла. Если бы бедняга Мухамед несмело спросил, сколько же дукаатов следует заплатить за CCLXXIV овец, если каждая из них стоит XLIX дукаатов, то тут уж возмущение оппонентов достигло бы предела. Как этот ал-Хорезми, в целом не глупый парень, может быть таким наивным и оторванным от реальной жизни. Видел ли кто-нибудь когда-либо, чтобы продавались CCLXXIV овец? На практике это либо CC, либо CCC, либо в худшем случае CCL овец. А где вы видели такую дурацкую цену за одну овцу, как XLIX дукаатов? На практике это всегда L дукаатов, и следовательно, все эти тонкие и довольно сложные соображения практически неприемлемы. На практике, если мне нужно умножить CC на L, я без всяких твоих теорий, коллега Мухамед, знаю, что мне надо будет заплатить C раз по C дукаатов, и дело с концом. И потом, любезный Мухамед, Ты же умный парень, пойми, наконец, что пора заняться чем-то полезным и практически реализуемым. Мы ведь не можем рассчитывать на то, что сможем переучить всех купцов в Самарканде и заставим их писать числа по новому, и только лишь для того, чтобы Тебе было легко складывать и умножать. Вся наша наука существует лишь от налогов, которые они платят, и от добровольных пожертвований, которые платят они же. Здесь не они существуют для нас, а мы для них. . .

В этом духе можно продолжать до бесконечности и не нужно даже ничего придумывать. Достаточно записать то, что каждый из нас слышит ежедневно.

Конечно же, я должен как-то среагировать на ваш рассказ. Он состоит из двух сильно отличающихся друг от друга частей. В первой части красочно и образно показываются примеры, доказывающие, что выбор формализма, в котором выражаются знания, может сыграть существенную роль в дальнейшей судьбе этих знаний. Несмотря на некоторые неточности в вашем рассказе, ваши доводы для меня убедительны. Конечно же, я был неправ, когда утверждал, что формализм, в котором выражены знания, не играет существенной роли.

Что касается второй части вашего рассказа, то это какая-то сатира, и мне почему-то представляется, что она адресована именно мне. Скажу прямо, что мое отношение к новым знаниям ближе к позиции тех болванов, которые оппонировали Мухамеду, чем к позиции Мухамеда. Не считаю это

своим достоинством, но и не настроен сейчас в этом оправдываться. Поэтому я еще раз утверждаю, что все результаты, которые в лекции были изложены с помощью матричных произведений, я мог бы получить и без них. И спрашиваю вас, можете ли вы привести пример задачи, которая решается в рамках предложенного вами формализма и не может быть решена без его использования?

На Твой вопрос невозможно ответить кратко, потому что сформулирован недостаточно точно. Но не будем сейчас помогать Тебе с его уточнением и ответим: Да, можем. Но Ты сам понимаешь, что речь должна идти о достаточно трудной задаче. Мы ее рассмотрим в следующей лекции.

Апрель 1998

8.13 Связь с TOOLBOX^{ом}

TOOLBOX по дискретным скрытым марковским моделям написал Я. Дупач, как свою дипломную работу весной 2000 года. К нему имеется доступ по адресу http://cmp.felk.cvut.cz/cmp/cmp_software.html. TOOLBOX построен на базе Matlab версии 5.3 и выше. Доступны исходные тексты алгоритмов.

В той части, которая касается материала данной лекции, реализованы алгоритмы распознавания, обучения и самообучения как для случая последовательностей, так и для ациклических структур.

8.14 Библиографические замечания

Марковские случайные последовательности были введены Марковым [Markov, 1916], как средство анализа 20 000 слов из романа в стихах "Евгений Онегин". Динамическое программирование используется в лекции для решения отдельных задач в духе Беллмана - автора динамического программирования [Bellman and Dreyfus, 1962]. В своем рассказе о применении динамического программирования для структурного распознавания мы опираемся на пионерские работы трех авторов: работы Ковалевского по распознаванию несегментированных текстовых строк и распознаванию изображений, составленных из линий [Kovalevski, 1967], [Kovalevski, 1969], работу Хазен [Chazen, 1968] по применению динамического программирования для определения наиболее вероятной последовательности скрытых состояний; работу Винцока [Vincjuk,] по применению динамического программирования для распознавания речевых сообщений.

Распознавание скрытых марковских последовательностей известно за рубежом под названием алгоритмов Витерби [Viterbi, 1967]. Наиболее известной работой по обучению в рамках марковских моделей является работа Баума и Уелша [Baum et al., 1970].

Формализм обобщенных матричных произведений использован в знаменитой монографии [Aho et al., 1975] для единообразного анализа вы-

числительных алгоритмов. Поиск наилучшей аппроксимации многомерной случайной величины с помощью ациклического графа описан в работе Чау [Chow, 1965; Chow and Liu, 1968]. Задача поиска наилучшего графа взаимосвязей компонент многомерной величины сведена к задаче Кейли [Cauley, 1889]. Решение задачи Кейли принадлежит математику Борувке [Boruvka, 1926] из города Брно. Позже это решение было многократно повторено другими авторами. Наиболее известной из них является работа Краскала [Kruskal, 1956].

Глава 9

Регулярные языки и соответствующие задачи распознавания

9.1 Регулярные языки

Предыдущая лекция была посвящена анализу следующей модели распознаваемого объекта.

Пусть X и K - конечные множества. Первое из них - это алфавит выходных символов, а второе - множество состояний автономного стохастического автомата. Автомат характеризуется функцией $p: K \times X \times K \rightarrow \mathbb{R}$, которая определяет следующее поведение автомата. Автомат проходит через последовательность состояний $k_0, k_1, \dots, k_i, \dots$ и генерирует последовательность символов $x_1, x_2, \dots, x_i, \dots$ на своем выходе. Начальное состояние k_0 автомата случайное и принимает то или иное значение в соответствии с вероятностями

$$\sum_{k \in K} \sum_{x \in X} p(k_0, x, k).$$

Автомат, находящийся в $(i - 1)$ -ый момент, $i = 1, 2, \dots$, в состоянии k_{i-1} , генерирует случайную пару (x_i, k_i) в соответствии с вероятностями

$$\frac{p(k_{i-1}, x_i, k_i)}{\sum_{x \in X} \sum_{k \in K} p(k_{i-1}, x, k)},$$

подает символ x_i на выход и переходит в состояние k_i . Функция p определяет распределение вероятностей на множестве выходных последовательностей. В генеральной совокупности последовательностей длины n вероятность последовательности x_1, x_2, \dots, x_n определяется суммой

$$\sum_{k_0} \sum_{k_1} \dots \sum_{k_n} \left(p(k_0, x_1, k_1) \prod_{i=2}^n \frac{p(k_{i-1}, x_i, k_i)}{\sum_{x \in X} \sum_{k \in K} p(k_{i-1}, x, k)} \right).$$

В то же время с помощью автомата можно и более грубо описать последовательности, которые могут появиться на его выходе, а именно, определить

лишь множество последовательностей, вероятность которых на выходе автомата не равна нулю. Если нас интересует только, может ли последовательность x_1, x_2, \dots, x_n появиться на выходе автомата вообще, и не интересуется, с какой вероятностью она может появиться, то не требуется такая детальная его характеристика, как функция $p: K \times X \times K \rightarrow \mathbb{R}$. Достаточно знать лишь для каждой тройки $k' \in K, x \in X, k'' \in K$, равна ли вероятность $p(k', x, k'')$ нулю или нет. Автомат может быть описан в этом случае менее детально с помощью функции P вида $K \times X \times K \rightarrow \{0, 1\}$, значение $P(k', x, k'')$ которой равно 0, если $p(k', x, k'') = 0$, и 1, если $p(k', x, k'') \neq 0$. Таким образом подмножество последовательностей определяется просто с помощью бинарной функции трех переменных. Естественно, некоторые подмножества последовательностей не могут быть определены таким простым способом. Те подмножества, которые допускают такое представление, известны под названием *регулярный язык*. Определим это понятие более точно.

Пусть X - конечное множество, которое будем называть *алфавитом*, а его элементы - символами. Конечную последовательность, составленную из символов алфавита X , будем называть *предложением* в алфавите X . Множество всех возможных последовательностей в алфавите X будет обозначаться X^* , а то или иное подмножество $L \subset X^*$ предложений будет называться *языком* в алфавите X .

Пусть K - конечное множество *состояний*. Пусть функция $\varphi: K \rightarrow \{0, 1\}$ определяет подмножество состояний, называемых *начальными состояниями* автомата, так, что если $\varphi(k) = 1$, то k есть одно из начальных состояний (которое может быть и единственным). Пусть функция $\psi: K \rightarrow \{0, 1\}$ определяет подобным образом множество *конечных состояний*. Пусть функция $P: K \times X \times K \rightarrow \{0, 1\}$, называемая функцией переходов, имеет следующий смысл. Если в i -1-ый момент автомат находился в состоянии k' , то в i -ый момент символ на выходе автомата может равняться x и автомат может находиться в состоянии k только при условии $P(k', x, k) = 1$.

Автомат определяется пятеркой $A = \langle X, K, \varphi, P, \psi \rangle$ введенных понятий. Они однозначно определяют множество последовательностей символов, которые могут появиться на выходе автомата. Это множество называется языком автомата A и обозначается $L(A)$. Более точно, предложение x_1, x_2, \dots, x_n принадлежит языку $L(\langle X, K, \varphi, P, \psi \rangle)$, если существует такая последовательность k_0, k_1, \dots, k_n , для которой:

1. $\varphi(k_0) = 1$;
2. $P(k_{i-1}, x_i, k_i) = 1, i = 1, 2, \dots, n$;
3. $\psi(k_n) = 1$.

Это определение можно записать в виде формулы

$$F(x_1, x_2, \dots, x_n) = \bigvee_{k_0 \in K} \bigvee_{k_1 \in K} \dots \bigvee_{k_n \in K} \left(\varphi(k_0) \wedge \left(\bigwedge_{i=1}^n P(k_{i-1}, x_i, k_i) \right) \wedge \psi(k_n) \right), \quad (9.1)$$

где $F(x_1, x_2, \dots, x_n)$ есть утверждение 'Последовательность $\bar{x} = (x_1, x_2, \dots, x_n)$ принадлежит языку $L(\langle X, K, \varphi, P, \psi \rangle)$ '.

Определение (9.1) можно записать в более краткой форме, если принять следующую интерпретацию функций φ , P и ψ . Функция φ будет пониматься как $|K|$ -мерный вектор-строка, k -ая компонента которого равна $\varphi(k)$, $k \in K$. Функция ψ есть $|K|$ -мерный вектор-столбец, k -ая компонента которого равна $\psi(k)$. Для заданной последовательности x_1, x_2, \dots, x_n , будем считать определенной матрицу P_i , $i = 1, 2, \dots, n$, в которой в k' -ой строке и k'' -ом столбце записано $P(k', x_i, k'')$. Используя эти обозначения, определение (9.1) можно понимать как *матричное произведение*

$$F(x_1, x_2, \dots, x_n) = \varphi \odot \left(\bigodot_{i=1}^n P_i \right) \odot \psi \quad (9.2)$$

в полукольце на множестве $\{0, 1\}$ с операциями \vee , соответствующей сложению, и \wedge , соответствующей умножению.

Кроме определения регулярного языка в виде (9.1), известен еще ряд других эквивалентных определений, которые мы приведем далее ради полноты. Для нашего изложения окажется наиболее удобным именно приведенное определение (9.1) и особенно (9.2), так как они немедленно указывают алгоритм распознавания принадлежности заданного предложения x_1, x_2, \dots, x_n заданному языку $L(\langle X, K, \varphi, P, \psi \rangle)$.

Язык определяется как множество предложений, для которых произведение (9.2) принимает значение 1. Алгоритм распознавания состоит просто в вычислении этого произведения. Сложность его, очевидно, имеет порядок $\mathcal{O}(|K|^2 n)$.

9.2 Другие способы представления регулярных языков

9.2.1 Регулярные языки и автоматы

Пусть X - алфавит входных символов (а не выходных, как было в предыдущем рассмотрении), которые подаются на вход конечного автомата. Пусть K - множество состояний этого автомата, а k_0 - одно из начальных состояний. Пусть $q: K \times X \rightarrow K$ - функция переходов, определяющая следующее поведение автомата. Если автомат в $(i-1)$ -ый момент находился в состоянии k , а в i -ый момент на его вход подан символ x , то в i -ый момент времени автомат окажется в состоянии $q(k, x)$. Следовательно, если зафиксировать начальное состояние k_0 , то предложение x_1, x_2, \dots, x_n , поданное на вход автомата, однозначно определяет последовательность $k_1 = q(k_0, x_1)$, $k_2 = q(k_1, x_2)$, \dots , $k_n = q(k_{n-1}, x_n)$ состояний, через которые пройдет автомат, в том числе и последнее состояние k_n .

Таким образом автомат реализует отображение множества X^* предложений в множество K состояний. Любому предложению $\bar{x} \in X^*$ ставится в соответствие то состояние, в котором оказывается автомат после подачи на его вход предложения \bar{x} . Обозначим это отображение $Q: X^* \rightarrow K$. Отображение Q однозначно определяется множествами X , K , начальным состоянием k_0 функцией $q: K \times X \rightarrow K$ переходов.

Пусть $K' \subset K$ определенное подмножество состояний. Подмножество предложений $\bar{x} \in X^*$, для которого $Q(\bar{x}) \in K'$, называется языком, принимаемым данным автоматом. Этот язык определяется автоматом, то есть пятеркой $\langle X, K, k_0 \in K, q: K \times X \rightarrow K, K' \subset K \rangle$.

Указанным способом можно определить не любые языки, а только регулярные, но зато все регулярные. Мы не будем доказывать это утверждение, так как в дальнейшем мы будем опираться главным образом на определения (9.1) (9.2).

Обратим внимание на важное обстоятельство. Мы дали два определения регулярного языка, и оба определения основаны на понятии автомата. Хотя эти два определения эквивалентны, речь здесь идет о двух совершенно разных автоматах. В первом случае речь идет о недетерминированном автомате, который генерирует предложения из определенного языка. Во втором случае речь идет о детерминированном автомате, который распознает, принадлежит ли поданное на его вход предложение определенному языку или нет. Оба определения общеизвестны и служат основой для обширных исследований. В данной работе мы будем основываться на определении регулярного языка с помощью генерирующего недетерминированного автомата, а не детерминированного распознающего автомата. Если об этом забыть, то могут возникнуть серьезные недоразумения при дальнейшем чтении.

9.2.2 Регулярные языки и грамматики

Рассмотрим автомат $A = \langle X, K, \varphi, P, \psi \rangle$ и соответствующий ему регулярный язык $L(A)$. Пятерка $\langle X, K, \varphi, P, \psi \rangle$ совсем необязательно должна пониматься как автомат, а понятия X, K, φ, P, ψ можно назвать совсем по-другому, чем это было сделано в подразделе 9.2.1. При такой иной интерпретации формальных понятий получается то, что обычно называется *регулярной грамматикой*. Укажем соответствие между понятиями, образующими автомат, и понятиями, образующими грамматику, и тем самым определим, что такое регулярная грамматика и порождаемый ею язык.

Назовем множества X и K *терминальным и нетерминальным алфавитами* соответственно. Функцию φ представим в виде подмножества $K^0 = \{k \in K \mid \varphi(k) = 1\}$, которое назовем *множеством аксиом*. Функции P и ψ выразим в виде подмножеств троек вида (k', x, k'') , $k' \in K$, $x \in X$, $k'' \in K$, и пар вида (k, x) , $k \in K$, $x \in X$. Эти тройки и пары обычно называются подстановками, грамматическими правилами и еще иначе. Мы будем их называть *правилами*.

Множество правил можно построить на основании автономного автомата, с помощью которого было дано первичное определение регулярного языка. Если для некоторой тройки (k', x, k'') выполняется $P(k', x, k'') = 1$, то тройка (k', x, k'') становится правилом и записывается в виде $k' \rightarrow xk''$. Если помимо равенства $P(k', x, k'') = 1$ выполняется еще $\psi(k'') = 1$, наряду с правилом $k' \rightarrow xk''$ вводится еще правило $k' \rightarrow x$. Обозначим R полученное таким способом множество правил. Таким образом, пятер-

ка $\langle X, K, \varphi, P, \psi \rangle$, которую мы раньше называли автоматом, представлена теперь регулярной грамматикой, которая определяется как четверка $\langle X, K, K^0, R \rangle$, где

- X - терминальный алфавит,
- K - нетерминальный алфавит,
- K^0 - множество аксиом,
- R - множество правил.

Регулярная грамматика определяет *язык*, то есть подмножество $L \subset X^*$, следующим образом.

1. Последовательность, состоящая из единственного символа, который является одной из аксиом, называется выведенной в данной грамматике.
2. Если последовательность $\bar{x}k'$, $\bar{x} \in X^*$, $k' \in K$, выведена, а множество правил содержит правило $k' \rightarrow x'k''$, то предложение $\bar{x}x'k''$ тоже считается выведенным.
3. Если последовательность $\bar{x}k'$, $\bar{x} \in X^*$, $k' \in K$, выведено, а множество правил содержит правило $k' \rightarrow x'$, то последовательность $\bar{x}x'$ принадлежит языку L , порождаемому данной грамматикой.

Обращаем внимание на то, что в приведенном определении различаются понятия "последовательность, выведенная в данной грамматике" и "предложение, порождаемое данной грамматикой". Первое - это последовательность символов, в которой все символы, кроме последнего, являются терминальными, а последний - обязательно нетерминальный. Второе - это последовательность, состоящая из одних только терминальных символов.

Если грамматика $\langle X, K, K^0, R \rangle$ получена из автомата $\langle X, K, \varphi, P, \psi \rangle$ указанным выше способом, то множество порождаемых ею предложений совпадает с множеством предложений, генерируемых автоматом. Мы не будем доказывать это утверждение, полагая, что оно почти очевидно.

Возможен и обратный переход от задания регулярного языка с помощью регулярной грамматики к заданию его же с помощью автономного конечного автомата. Автономный конечный автомат и регулярные грамматики - это два эквивалентные средства для задания множеств определенного типа - регулярных языков.

9.2.3 Регулярные языки и регулярные выражения

Введем следующие три операции на множестве языков.

1. *Итерация языка* L есть язык, обозначаемый L^* и определяемый следующим образом.
 - (а) Пустое предложение, то есть предложение нулевой длины, принадлежит L^* .
 - (б) Если предложение \bar{x} принадлежит L^* , предложение \bar{y} принадлежит L , то предложение $\bar{x}\bar{y}$ также принадлежит L^* .
2. *Конкатенация языков* L_1 и L_2 есть язык, обозначаемый L_1L_2 и содержащий все предложения вида $\bar{x}\bar{y}$, $\bar{x} \in L_1$, $\bar{y} \in L_2$, и только их.

3. Объединение языков L_1 и L_2 есть язык $L_1 \cup L_2$.

Пусть X - конечное множество символов. Некоторые языки в алфавите X можно записать с помощью *регулярных выражений* по следующим правилам.

1. Символ \emptyset есть одно из регулярных выражений и служит для обозначения *пустого множества предложений*.
2. Символ $\#$ есть одно из регулярных выражений и служит для обозначения языка, который состоит из единственного предложения, которое является пустым, то есть имеет нулевую длину.
3. Для каждого символа $x \in X$ запись x является регулярным выражением для записи языка, состоящего из единственного предложения, состоящего из единственного символа x .
4. Пусть α - регулярное выражение языка L . Тогда $(\alpha)^*$ есть регулярное выражение для итерации языка L .
5. Пусть α_1 и α_2 - регулярные выражения для языков L_1 и L_2 соответственно. Тогда запись $\alpha_1\alpha_2$ есть регулярное выражение для конкатенации L_1L_2 этих языков, а запись α_1, α_2 - для их объединения $L_1 \cup L_2$.

Например, запись $a(b,c)^*$ есть регулярное выражение для множества предложений, начинающихся символом a , вслед за которым идет любая последовательность (возможно, и пустая), составленная из символов b и c .

Известно следующее фундаментальное свойство регулярных выражений. Любой регулярный язык можно задать с помощью регулярного выражения. И наоборот, любое регулярное выражение определяет регулярный язык.

9.2.4 Пример регулярного языка представленного различными способами

Определим неформально, но как можно более однозначно определенное множество предложений, которое затем представим с помощью средств, введенных в предыдущих подразделах. Пусть алфавит X состоит из символов $a, b, c, +, \times, =$. Мы хотим выразить множество предложений, которые можно было бы понимать, как самые простые команды в некотором языке программирования. Предложение (то есть команда) состоит из двух частей, левой и правой, которые разделены символом $=$. Левая часть предложения должна быть идентификатором переменной, то есть непустой последовательностью, составленной из символов a, b или c . Правая часть предложения должна быть непустой последовательностью

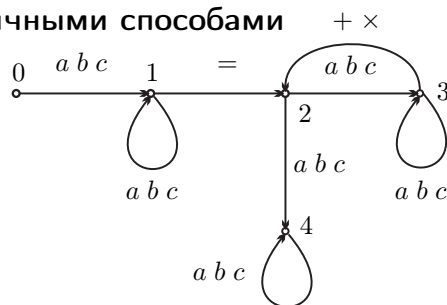


Figure 9.1 Nondeterministic autonomous automaton corresponding to the regular expression $(a, b, c)(a, b, c)^* = ((a, b, c)(a, b, c)^* (+, \times))^* (a, b, c)(a, b, c)^*$.

идентификаторов, разделенных символами $+$ или \times . Например, предложение $a = ab + c \times a + aba$ принадлежит языку, который мы хотим выразить. А предложения $bc =$, или $a + b$, или $a = a + \times b$ не принадлежат этому языку.

Регулярным выражением этого языка является запись

$$(a, b, c)(a, b, c)^* = ((a, b, c)(a, b, c)^*(+, \times))^*(a, b, c)(a, b, c)^* .$$

Этот же язык можно выразить с помощью недетерминированного автономного автомата, показанного на рис. 9.1. Выходной алфавит этого автомата состоит из символов $a, b, c, =, +, \times$. Множество состояний - это $\{0, 1, 2, 3, 4\}$ - множество вершин графа на рисунке. Начальным является состояние 0, а конечным - состояние 4. Функция P представлена на графе с помощью направленных ребер (стрелок) и символов, записанных на ребрах. Если стрелка начинается в вершине k' , заканчивается на вершине k'' и на стрелке написан символ x , это обозначает, что $P(k', x, k'') = 1$. Для всех других троек функция P принимает значение 0. Это значит, что в рассматриваемом примере функция P принимает значение 1 на следующих тройках:

$$\begin{array}{ll} (0, a, 1), (0, b, 1), (0, c, 1), & (3, a, 3), (3, b, 3), (3, c, 3), \\ (1, a, 1), (1, b, 1), (1, c, 1), & (3, +, 2), (3, \times, 2), \\ (1, =, 2), & (2, a, 4), (2, b, 4), (2, c, 4), \\ (2, a, 3), (2, b, 3), (2, c, 3), & (4, a, 4), (4, b, 4), (4, c, 4). \end{array}$$

Этот список вместе с графом можно понимать как следующую грамматику. Ее терминальный алфавит - это множество $\{a, b, c, =, +, \times\}$, нетерминальный алфавит - множество $\{0, 1, 2, 3, 4\}$, аксиома - это 0, а правила - это

$$\begin{array}{ll} 0 \rightarrow a1, 0 \rightarrow b1, 0 \rightarrow c1, & 3 \rightarrow +2, 3 \rightarrow \times 2, \\ 1 \rightarrow a1, 1 \rightarrow b1, 1 \rightarrow c1, & 2 \rightarrow a4, 2 \rightarrow b4, 2 \rightarrow c4, \\ 1 \rightarrow = 2, & 4 \rightarrow a4, 4 \rightarrow b4, 4 \rightarrow c4, \\ 2 \rightarrow a3, 2 \rightarrow b3, 2 \rightarrow c3, & 4 \rightarrow a, 4 \rightarrow b, 4 \rightarrow c. \\ 3 \rightarrow a3, 3 \rightarrow b3, 3 \rightarrow c3, & \end{array}$$

И наконец, автомат, распознающий предложения данного языка, показан с помощью графа на рис. 9.2. Алфавит входных символов - множество $\{a, b, c, =, +, \times\}$, а множество состояний K - это $\{0, 1, 2, 3, 4, 5\}$. Начальным является состояние 0, а конечным - состояние 3. Автомат распознает входное предложение в том смысле, что любое правильное предложение переводит автомат в состояние 3, а любое неправильное - в какое-то другое. Это достигается выбором функции переходов q , которая управляет сменой состояния автомата при подаче на его вход того или иного символа. Функция переходов представлена на рис. 9.2 стрелками, на которых записаны символы. Если стрелка начинается в вершине k' , заканчивается в вершине k'' и на ней записан символ x , это обозначает, что $q(k', x) = k''$.

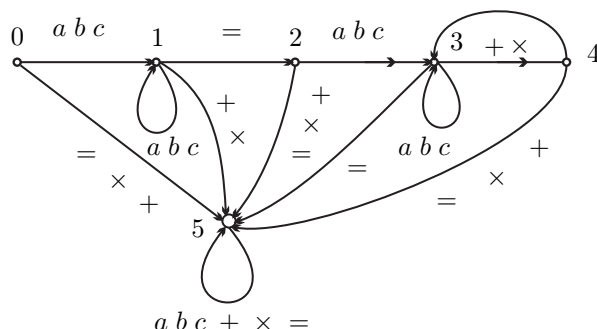


Figure 9.2 Deterministic automaton recognising the language from the example.

9.3 Модификации регулярных языков; точное и наилучшее соответствие

В разделе 9.1 мы определили понятие регулярных языков и средства для их задания. Задача распознавания формулируется на основе этих понятий, как построение алгоритма, который для любого заданного регулярного языка и любого предложения определяет, принадлежит ли данное предложение данному языку или нет. Эта задача оказывается не очень трудной, особенно, если регулярный язык выражен с помощью автомата, генерирующего все возможные предложения данного языка. В этом случае регулярный язык определяется как множество последовательностей, для которых матричное произведение (9.2) равно 1. Такое определение регулярного языка немедленно представляет и алгоритм распознавания.

Однако многие понятия только очень загрубленно могут быть представлены в виде подмножеств наблюдений, которые им соответствует. К примеру, такими являются понятия здоровый человек или хорошая погода. Формализация таких понятий требует чего-то большего, чем определение тех или иных подмножеств наблюдений. Они требуют задания той или иной функции на множестве наблюдений, значения которой обозначают степень уверенности, что наблюдаемый объект обладает интересующим нас свойством. Именно определение вещественнозначной функции на множестве последовательностей служит затем основой для формулировки задач распознавания, а не определение того или иного подмножества последовательностей.

Естественно было бы понимать функцию, о которой идет речь, как распределение вероятностей на множестве X^* всех возможных последовательностей. Такая модификация, или если хотите, такое обобщение регулярных языков было рассмотрено нами в предыдущей лекции, основанной на постулировании стохастического механизма, в данном случае автомата, генерирующего случайные последовательности. Однако это совсем не единственный выход за пределы бинарной модели распознаваемого объекта. В распознавании образов нашли распространение и другие подходы, в част-

ности, следующие два подхода.

Первый подход основан на прямом обобщении понятия регулярного языка. С помощью определенных формальных средств определяется не подмножество L допустимых последовательностей, а неотрицательно определенная функция $F: X^* \rightarrow \mathbb{R}$. Задача распознавания состоит в построении алгоритма, который для каждой поданной на его вход последовательности \bar{x} вычисляет число $F(\bar{x})$.

Второй подход состоит в определении функции $d: X^* \times X^* \rightarrow \mathbb{R}$ (обычно тоже неотрицательно определенной), которая для каждой пары последовательностей \bar{x}_1 и \bar{x}_2 определяет их отличие друг от друга. Само понятие регулярного языка никаким образом не модифицируется, но по другому формулируется задача распознавания. Распознавание теперь состоит не в определении, принадлежит ли заданная последовательность \bar{x} языку L , а в вычислении "расстояния" последовательности \bar{x} от языка L , то есть в вычислении числа

$$\min_{\bar{y} \in L} d(\bar{x}, \bar{y}).$$

Такого рода задачи называются *задачами на наилучшее соответствие* (*best matching problems*). Задачи же проверки отношения $\bar{x} \in L$ называются *задачами на точное соответствие* (*exact matching problems*).

В следующих подразделах приводятся примеры этих подходов.

9.3.1 Размытые автоматы и размытые языки

Определение размытых автоматов и языков основаны на понятиях размытых множеств, которые вводятся следующим образом.

Пусть X - множество, не обязательно алфавит символов. Подмножество $X' \subset X$ в обычном смысле этого слова может пониматься как функция $f: X \rightarrow \{0, 1\}$, принимающая значение 1 на элементах подмножества X' и значение 0 за пределами подмножества X' . Размытое подмножество множества X понимается как функция $f: X \rightarrow \mathbb{R}$, принимающая значения из интервала от 0 до 1. Пересечение размытых множеств, представленных функциями f_1 и f_2 , определяется как размытое множество, представленное функцией $f: X \rightarrow \mathbb{R}$, значение которой в точке x есть $f(x) = \min(f_1(x), f_2(x))$. Объединение этих множеств определяется как размытое множество, представленное функцией $f: X \rightarrow \mathbb{R}$, значение которой в точке x есть $f(x) = \max(f_1(x), f_2(x))$.

В разделе 9.1 был определен автомат как пятерка $\langle X, K, \varphi, P, \psi \rangle$, где X и K - два конечных множества, функции φ и ψ представляют два подмножества в K , а функция P представляет подмножество в $K \times X \times K$. Размытый автомат определяется такой же пятеркой с той только разницей, что функции φ , P и ψ представляют не множества в обычном смысле этого слова, а размытые множества.

Автомат $\langle X, K, \varphi, P, \psi \rangle$ (не размытый) определяет язык, как подмножество последовательностей x_1, x_2, \dots, x_n , $n = 1, 2, 3, \dots$, для которых

$$\bigvee_{k_0 \in K} \bigvee_{k_1 \in K} \cdots \bigvee_{k_n \in K} \left(\varphi(k_0) \wedge \left(\bigwedge_{i=1}^n P(k_{i-1}, x_i, k_i) \right) \wedge \psi(k_n) \right) = 1. \quad (9.3)$$

Размытый автомат $\langle X, K, \varphi, P, \psi \rangle$ определяет размытый язык, то есть размытое подмножество последовательностей, представленное той же функцией $X^* \rightarrow \mathbb{R}$, что и (9.3), но с операцией \bigvee , понимаемой как объединение размытых множеств, и операцией \bigwedge , понимаемой как пересечение размытых множеств. Таким образом, язык размытого автомата $\langle X, K, \varphi, P, \psi \rangle$ есть размытое множество подпоследовательностей, задаваемое функцией

$$\max_{k_0 \in K} \max_{k_1 \in K} \cdots \max_{k_n \in K} \min \left(\varphi(k_0), \min_i P(k_{i-1}, x_i, k_i), \psi(k_n) \right). \quad (9.4)$$

Пара операций (\max , \min) образует полукольцо на множестве вещественных чисел от 0 до 1, где \max служит сложением, а \min - умножением. Поэтому выражение (9.4) можно записать как матричное произведение

$$F(x_1, x_2, \dots, x_n) = \varphi \odot \left(\bigodot_{i=1}^n P_i \right) \odot \psi, \quad (9.5)$$

где φ - это вектор-строка, P_i , $i = 1, \dots, n$, - матрицы, ψ - вектор-столбец. Матричное произведение в формуле (9.5) должно выполняться в соответствующем полукольце, то есть, с операцией \max в качестве сложения и операцией \min в качестве умножения. При вычислении матричного произведения (9.5) следует иметь в виду, что матричное умножение не коммутативно.

Мы видим, что размывание понятий автомата и регулярного языка не привело к каким-то видимым трудностям при распознавании. Выражением (9.5) сформулирована задача распознавания как определения степени принадлежности заданной последовательности x_1, x_2, \dots, x_n заданному размытому множеству. Это же выражение указывает алгоритм ее вычисления. Сложность этих вычислений имеет, очевидно, порядок $\mathcal{O}(|K|^2 n)$.

9.3.2 Штрафные автоматы и соответствующие языки

Пусть X и K - два конечных множества, а $\varphi: K \rightarrow \mathbb{R}$, $P: K \times X \times K \rightarrow \mathbb{R}$, $\psi: K \rightarrow \mathbb{R}$ - три функции, определяющие поведение автомата. Автомат может начать свою работу в любом состоянии $k \in K$, но заплатить при этом штраф, равный $\varphi(k)$. Если автомат в $(i-1)$ -ый момент находился в состоянии k' , то в i -ый момент он может сгенерировать символ x и перейти в состояние k , заплатив за это штраф $P(k', x, k)$. И наконец, автомат может завершить генерирование последовательности в любой момент i , заплатив за прекращение работы штраф $\psi(k_i)$, зависящий от состояния k_i , в котором алгоритм прекратил свою работу.

Пятерка $\langle X, K, \varphi, P, \psi \rangle$ и число ε определяют язык, который назовем штрафным. В этот язык входят те и только последовательности, штраф за генерирование которых не превосходит ε . Множество языков, которые могут быть определены указанным способом, включают все регулярные языки, а также некоторые другие языки, не являющиеся регулярными.

Распознавание принадлежности последовательности x_1, x_2, \dots, x_n заданному штрафному языку сводится к вычислению числа

$$\min_{k_0} \min_{k_1} \cdots \min_{k_n} \left(\varphi(k_0) + \sum_i P(k_{i-1}, x_i, k_i) + \psi(k_n) \right), \quad (9.6)$$

и сравнению его с числом ε . Вычисление (9.6) можно представить как вычисление матричного произведения

$$\varphi \odot \left(\bigodot_{i=1}^n P_i \right) \odot \psi$$

в котором матрицы умножаются в полукольце с операциями \min в качестве сложения и $+$ как умножения.

Мы видим, что и в этом случае задача остается на уровне тривиальной, так как сама ее формулировка немедленно указывает алгоритм ее решения. Вычислительная сложность этого алгоритма имеет все тот же порядок $\mathcal{O}(|K|^2 n)$.

9.3.3 Простейшая задача на наилучшее соответствие

Пусть $d: X \times X \rightarrow \mathbb{R}$ - функция, значение которой $d(x', x'')$, $x' \in X$, $x'' \in X$, обозначает штраф за замену символа x' на x'' . На основании этой функции определяется и штраф за замену последовательности x'_1, x'_2, \dots, x'_n последовательностью $x''_1, x''_2, \dots, x''_n$. Он равен

$$\sum_{i=1}^n d(x'_i, x''_i).$$

В частном случае, когда переменные x_i принимают только два значения, и при надлежащем выборе функции d , указанная сумма есть известное хеммингово расстояние, известное в теории кодирования. Пусть $L \subset X^*$ - регулярный язык (не размытый и не штрафной), соответствующий автомату $\langle X, K, \varphi, P, \psi \rangle$. Простейшая задача на наилучшее соответствие состоит в том, чтобы данную последовательность x_1, x_2, \dots, x_n заменить последовательностью y_1, y_2, \dots, y_n из языка L так, чтобы штраф $\sum_{i=1}^n d(y_i, x_i)$ был минимален. Это значит, что должна быть решена минимизационная зада-

ча

$$\left. \begin{aligned} \min_{y_1} \min_{y_2} \cdots \min_{y_n} \sum_{i=1}^n d(y_i, x_i), \quad \text{при условиях:} \\ \varphi(k_0) = 1, \\ P(k_{i-1}, y_i, k_i) = 1, \quad i = 1, \dots, n, \\ \psi(k_n) = 1, \end{aligned} \right\} \quad (9.7)$$

для заданной последовательности x_1, x_2, \dots, x_n и заданных функций d , φ , P and ψ . Эту задачу можно преобразовать на основе следующих соображений. Пусть

$$M = \max_{y \in X} \max_{x \in X} d(y, x).$$

Введем новые функции φ' , P' и ψ' ,

$$\begin{aligned} \varphi'(k) &= Mn + 1, & \text{если} & \quad \varphi(k) = 0, \\ \varphi'(k) &= 0, & \text{если} & \quad \varphi(k) = 1, \\ \psi'(k) &= Mn + 1, & \text{если} & \quad \psi(k) = 0, \\ \psi'(k) &= 0, & \text{если} & \quad \psi(k) = 1, \\ P'(k', y, k'') &= Mn + 1, & \text{если} & \quad P(k', y, k'') = 0, \\ P'(k', y, k'') &= 0, & \text{если} & \quad P(k', y, k'') = 1, \end{aligned}$$

и, используя их, запишем задачу (9.7) как

$$\min_{k_0} \cdots \min_{k_n} \min_{y_1} \cdots \min_{y_n} \left(\varphi'(k_0) + \sum_{i=1}^n \left(P'(k_{i-1}, y_i, k_i) + d(y_i, x_i) \right) + \psi'(k_n) \right). \quad (9.8)$$

Для произвольной последовательности k_0, \dots, k_n , y_1, \dots, y_n , которая удовлетворяет условию задачи (9.7), минимальное значение суммы не будет превосходить Mn . Для любой последовательности k_0, \dots, k_n , которая не удовлетворяет условиям задачи (9.7), эта сумма будет не меньше, чем $Mn + 1$. Отсюда следует, что минимум в задаче (9.8) может достигаться только на последовательностях k_0, \dots, k_n , y_1, \dots, y_n , которые удовлетворяют условию (9.7). На последовательностях, которые удовлетворяют условиям (9.7), минимизируемые функции в (9.8) и (9.7) принимают одинаковые значения.

Число (9.8), очевидно, равно

$$\min_{k_0} \min_{k_1} \cdots \min_{k_n} \left(\varphi'(k_0) + \sum_{i=1}^n \min_{y_i} \left(P'(k_{i-1}, y_i, k_i) + d(y_i, x_i) \right) + \psi'(k_n) \right).$$

Если ввести обозначение

$$P''_i(k_{i-1}, x_i, k_i) = \min_{y \in X} \left(P'(k_{i-1}, y, k_i) + d(y, x_i) \right), \quad (9.9)$$

значение (9.8) можно записать как

$$\min_{k_0} \min_{k_1} \cdots \min_{k_n} \left(\varphi'(k_0) + \sum_{i=1}^n P_i''(k_{i-1}, x_i, k_i) + \psi'(k_n) \right).$$

Последнее выражение имеет тот же вид, что и выражение (9.6), которое следует вычислять при распознавании принадлежности штрафному языку. Это значит, что рассматриваемая простейшая задача на наилучшее соответствие сводится к матричному умножению

$$\varphi' \odot \left(\bigodot_{i=1}^n P_i'' \right) \odot \psi'. \quad (9.10)$$

В ранее рассмотренных задачах умножаемые векторы и матрицы получались непосредственно из исходных данных задачи. В рассматриваемой задаче матрицы P_i'' получаются в результате решения совокупности не очень трудных оптимизационных задач (9.9). Суммарная сложность вычисления матриц P_i'' имеет порядок $\mathcal{O}(|K|^2 |X| n)$. После того, как эти матрицы вычислены, следует вычислить произведение (9.10), вычислительная сложность которого имеет порядок $\mathcal{O}(|K|^2 n)$. Ясно, что для вычисления чисел (9.9) совсем необязательно, чтобы последовательность x_1, x_2, \dots, x_n была известной. Их можно вычислить заблаговременно для всех значений $k' \in K, x \in X, k'' \in K$ по формуле

$$P''(k', x, k'') = \min_{y \in X} \left(P'(k', y, k'') + d(y, x) \right). \quad (9.11)$$

В этом случае вычислительная сложность распознавания каждой последовательности снижается до $\mathcal{O}(|K|^2 n)$, а вычисления (9.9), имеющие сложность $\mathcal{O}(|K|^2 |X| n)$, исключаются из этапа распознавания вообще. Зато добавляются вычисления (9.11) со сложностью порядка $\mathcal{O}(|K|^2 |X|^2)$, которые выполняются один раз и результаты которых затем используются при распознавании любой последовательности x_1, x_2, \dots, x_n .

9.4 Заключение по первой части лекции и введение во вторую часть

Мы увидели, что разнообразные задачи распознавания последовательностей решаются алгоритмами, которые состоят в вычислении произведений определенным образом устроенных матриц. Различие между алгоритмами состоит лишь в том, что эти произведения выполняются в различных полукольцах.

1. При вычислении *вероятности данной последовательности на выходе данного автомата* требуется умножать матрицы в полукольце $(+, \times)$, то есть умножать матрицы в обычном смысле этого слова.
2. При поиске *наиболее вероятной последовательности состояний, через которые прошел автомат*, требуется, чтобы матрицы умножались в полукольце (\max, \times) .

3. При решении о принадлежности данной последовательности данному регулярному языку требуется, чтобы матрицы умножались в полукольце (\vee, \wedge) .
4. При вычислении степени принадлежности последовательности данному размытому языку следует умножать матрицы в полукольце (\max, \min) .
5. Решение о принадлежности последовательности данному штрафному языку сводится к умножению матриц в полукольце $(\min, +)$.
6. Решение простейшей задачи на наилучшее соответствие последовательности данному регулярному языку сводится к умножению матриц в полукольце $(\min, +)$, как и в предыдущем случае.

Понимание единообразия алгоритмов решения различных, казалось бы, задач является очень полезным. Прежде всего, эти алгоритмы просто-напросто легче запоминаются, когда они единообразно представлены. Затем, единообразие алгоритмов позволяет сэкономить усилия при их программировании. Не требуется писать программы для решения каждой задачи в отдельности, а можно написать одну общую программу, в которой при переходе от одной задачи к другой меняется только описание операций сложения и умножения в соответствующем полукольце. И последнее, но не менее важное. При понимании единообразия задач достигается субъективное чувство, что все эти задачи простые. К примеру, решить о принадлежности предложения данному языку кажется поначалу значительно проще, чем найти в данном языке предложение, наименее отличающееся от предъявленного предложения. Или, скажем это в более общем виде, задачи на точное соответствие представляются существенно более простыми, чем задачи на наилучшее соответствие. Но после того, как выясняется родственность алгоритмов для их решения, обе эти задачи начинают восприниматься как одинаково простые. Или, еще один пример того же рода, отыскание КРАТЧАЙШЕГО пути на графе перестает казаться существенно сложнее отыскания ЛЮБОГО пути на графе после того, как выяснено, что обе эти задачи решаются родственными алгоритмами.

Мы видели также, что представление исходной задачи в виде матричного произведения не представляло особого труда. Почти во всех задачах не требовалось выполнять какие-то специальные вычисления для построения матриц - сомножителей искомого произведения. Они уже содержались в исходных данных задачи, и их надо было только несколько иначе интерпретировать. Можно сказать, что задача уже в своем исходном представлении находилась в выигрышной позиции, в которой алгоритм ее решения становился очевидным.

Несколько иной была рассмотренная нами задача на наилучшее соответствие. Матрицы, которые следовало умножать, не содержались непосредственно в исходных данных задачи. Однако эти матрицы строились по исходным данным путем решения несложных вспомогательных оптимизационных задач. Таким образом, хотя задача в своей исходной формулировке не находилась в выигрышной позиции, она переводилась в такую

позицию за один ход .

Сейчас мы перейдем к анализу и решению задачи, которая переводится в выигрышную позицию в результате довольно сложной многоходовой комбинации. Эта более сложная задача известна как левенштейнская аппроксимация предъявленной последовательности последовательностью из заданного регулярного языка. Ей мы посвятим оставшуюся часть данной лекции.

9.5 Левенштейнская аппроксимация предложения предложением из регулярного языка

9.5.1 Предварительная формулировка задачи

Мы собираемся анализировать задачу, принадлежащую классу задач на наилучшее соответствие. Эти задачи имеют следующий общий вид. Пусть L - регулярный язык, определенный автоматом $\langle X, K, \varphi, P, \psi \rangle$, и пусть функция $d: X^* \times X^* \rightarrow \mathbb{R}$ определяет для каждой пары последовательностей $\bar{x}_1 \in X^*$, $\bar{x}_2 \in X^*$ число $d(\bar{x}_1, \bar{x}_2)$, называемое *отличием последовательности \bar{x}_2 от последовательности \bar{x}_1* . Отличие не обязательно является симметричной функцией и не обязательно образует метрику на множестве последовательностей. Задача состоит в построении алгоритма, который для каждой последовательности $\bar{x} \in X^*$ и каждого регулярного языка $L \subset X^*$, поданных на вход алгоритма, вычисляет число

$$D(\bar{x}) = \min_{\bar{y} \in L} d(\bar{y}, \bar{x}). \quad (9.12)$$

Число $D(\bar{x})$ будем называть *отличием последовательности \bar{x} от языка L* .

Мы исследуем частный случай задачи вида (9.12), когда функция d принадлежит классу так называемых левенштейнских функций.

9.5.2 Функции Левенштейна

Определим три операции редактирования, с помощью которых одна последовательность преобразуется в другую, и определим стоимость каждой операции. Операции будут обозначаться двумя первыми буквами английского слова, обозначающего эти операции.

INsert (вставка) преобразует последовательность $\bar{x}_1\bar{x}_2$, $\bar{x}_1 \in X^*$, $\bar{x}_2 \in X^*$, в последовательность $\bar{x}_1x\bar{x}_2$, $x \in X$. Стоимость этой операции определяется функцией $in: X \rightarrow \mathbb{R}$, значение $in(x)$, $x \in X$, которой есть стоимость вставки символа x в последовательность.

CHange (замена) преобразует последовательность $\bar{x}_1x\bar{x}_2$, $\bar{x}_1 \in X^*$, $x \in X$, $\bar{x}_2 \in X^*$, в последовательность $\bar{x}_1x'\bar{x}_2$, $x' \in X$. Стоимость этого преобразования определяется функцией $ch: X \times X \rightarrow \mathbb{R}$. Число $ch(x, x')$, $x \in X$, $x' \in X$, - это стоимость замены символа x на символ x' .

DElete (исключение) преобразует последовательность $\bar{x}_1x\bar{x}_2$, $\bar{x}_1 \in X^*$, $x \in X$, $\bar{x}_2 \in X^*$, в последовательность $\bar{x}_1\bar{x}_2$. Его стоимость определяется функцией $de: X \rightarrow \mathbb{R}$. Число $de(x)$, $x \in X$, - это стоимость исключения символа x из последовательности.

Стоимость последовательности операций определяется как сумма стоимостей операций, входящих в эту последовательность. Стоимость пустой последовательности операций определяется как 0.

Пусть \bar{x}_1 и \bar{x}_2 - две последовательности. Существует бесконечное количество последовательностей редакторских операций, которые преобразуют \bar{x}_1 в \bar{x}_2 . Стоимость самой дешевой из них назовем левенштейновым отличием последовательности \bar{x}_2 от последовательности \bar{x}_1 . *Левенштейново отличие* будем обозначать $d(\bar{x}_1, \bar{x}_2)$.

Левенштейново отличие определяется тройкой функций $in: X \rightarrow \mathbb{R}$, $ch: X \times X \rightarrow \mathbb{R}$ и $de: X \rightarrow \mathbb{R}$. Множество этих троек порождает класс функций Левенштейна вида $X^* \times X^* \rightarrow \mathbb{R}$, с которым мы будем иметь дело на этой лекции. Некоторые функции этого класса образуют метрику на множестве последовательностей, но, вообще говоря, левенштейновы функции не определяют расстояние. Более того, они не обязательно симметричны. Поэтому мы будем применять несимметричное название для их значений. Число $d(\bar{x}_1, \bar{x}_2)$ - это отличие последовательности \bar{x}_2 от последовательности \bar{x}_1 , а не отличие \bar{x}_1 от \bar{x}_2 . Мы введем единственное ограничение, что функции in , ch и de принимают только неотрицательные значения.

Приведенное определение левенштейновского отличия очень коварное. Оно провоцирует предположения о некоторых свойствах, кажущихся очевидными, но не имеющих места на самом деле, то есть не вытекающих из определения. Алгоритмы, основанные на таких предположениях, решают только некоторые задачи из указанного класса. В общем случае их работоспособность не гарантируется. Конечно, нет ничего плохого в алгоритмах, которые решают задачу не во всей ее общности, а только некоторые ее частные случаи. Но при этом следует достаточно точно описать подкласс задач, решаемый предлагаемым алгоритмом правильно, чтобы он принимал к решению только задачи, которые входят в область его компетенции. Если это не сделано, то мы имеем дело с псевдорешением. Хорошо известный и излюбленный алгоритм вычисления функций Левенштейна является выразительным примером такого псевдорешения. Мы приведем его здесь ради полноты изложения и покажем, где кроется ошибка в его применении.

9.5.3 Известный алгоритм вычисления левенштейновского отклонения

Пусть $\bar{x} = (x_1, x_2, \dots, x_i, \dots, x_m)$ и $\bar{y} = (y_1, y_2, \dots, y_j, \dots, y_n)$ - две последовательности из X^* . Общеизвестный алгоритм вычисления левенштейнова отличия $d(\bar{y}, \bar{x})$ последовательности \bar{x} от последовательности \bar{y} представляется с помощью графа на рис. 9.3.

На рис. 9.3 представлен прямоугольник, составленный из m строк и n столбцов. Слева от i -ой строки написан i -ый символ из последовательности \bar{x} . Сверху от j -ого столбца написан j -ый символ из последовательности \bar{y} . На рисунке показаны горизонтальные, вертикальные и наклонные прямолинейные отрезки, представляющие ребра графа. Множество ребер определяет множество допустимых путей из левого верхнего угла прямо-

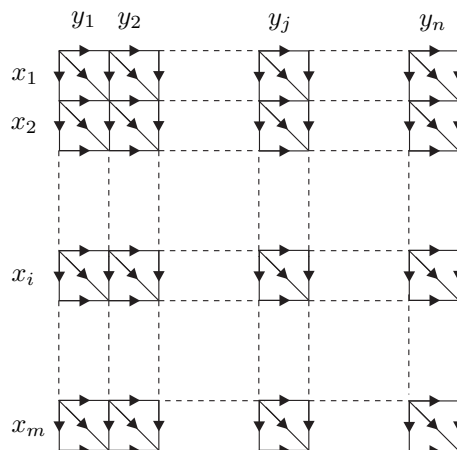


Figure 9.3 Graph of transitions illustrating the performance of an algorithm for calculating Levenstein dissimilarity.

угольника в правый нижний. Ребра ориентированы так, что движение по вертикальным отрезкам возможно только сверху вниз, по горизонтальным - слева направо, а по наклонным - с северо-востока на юго-запад.

Каждому пути на графе соответствует последовательность редакторских операций, которая преобразует последовательность \bar{y} в последовательность \bar{x} . Прохождение пути по ребру в i -ой строке соответствует вставке символа x_i в последовательность. Прохождение пути по ребру в j -ом столбце соответствует исключению символа y_j из последовательности. И наконец, прохождение пути по наклонному ребру в i -ой строке и j -ом столбце выражает замену символа y_j на символ x_i .

Присвоим каждому ребру длину, которая равна стоимости редакторской операции, которой это ребро соответствует. Длина каждого вертикального ребра в i -ой строке равна стоимости $in(x_i)$ вставки символа x_i . Длина каждого горизонтального ребра в j -ом столбце равна стоимости $de(y_j)$ исключения символа y_j . Наконец, длина наклонного ребра в i -ой строке и j -ом столбце равна стоимости $ch(y_j, x_i)$ замены символа y_j на символ x_i . При длинах ребер, определенных таким образом, длина каждого пути оказывается равной стоимости последовательности операций, которую этот путь представляет. Следовательно (ВНИМАНИЕ! СЛЕДУЕТ ОШИБКА!), поиск самой дешевой последовательности редакторских операций, преобразующих \bar{y} в \bar{x} , сведена к поиску кратчайшего пути на графе от верхнего левого угла к правому нижнему. Ошибочность приведенных рассуждений становится ясной по следующему контрпримеру.

Пример 9.1 Придирки к известному алгоритму вычисления левенштейнова отклонения. Пусть алфавит X есть $\{a, b, c, d\}$, последовательность \bar{y} состоит из единственного символа a , а последовательность

\bar{x} - из единственного символа b . Пусть требуется вычислить левенштейново отличие последовательности b от последовательности a . Граф, соответствующий кажущемуся решению задачи для этого случая, представлен на рис. 9.4. На этом графе имеются только три пути из левого верхнего угла в правый нижний. Эти три пути соответствуют следующим трем преобразованиям последовательности a в последовательность b .

1. Первая возможность.
 - Заменить a на b .
2. Вторая возможность.
 - Исключить a .
 - Вставить b .
3. Третья возможность.
 - Вставить b .
 - исключить a .

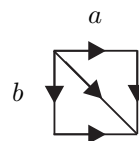


Figure 9.4 Counterexample. Computing Levenshtein dissimilarity.

В результате получается, что будет дан ответ, что левенштейново отличие b от a равно $\min(ch(a, b), de(a) + in(b))$. В действительности же это отличие может быть и меньшим, потому что граф 9.4 представляет только три варианта превращения последовательности a в последовательность b . В действительности здесь имеется и много других вариантов, например, такой:

- Заменить a на c .
- Исключить c .
- Вставить d .
- Заменить d на b .

Граф на рис. 9.4 представляет лишь очень малую часть всех возможных редакторских операций, которые превращают a в b . Следовательно, описанный алгоритм решает задачу правильно только в случае, когда есть априорная уверенность, что самая дешевая последовательность операций принадлежит именно этой малой части. В общем случае это, конечно же, не гарантируется. ▲

Обнаруженная ошибка в описанной процедуре вычисления левенштейнова отличия не приводит к каким-либо разрушительным последствиям. Как только она обнаружилась, ее легко исправить. Мы показали эту ошибку только как пример многочисленных ловушек, которые здесь подстерегают разработчика алгоритма, легкомысленно доверяющего так называемому здравому смыслу и своей интуиции. В этой задаче скрыто много ловушек, мало заметных даже при очень тщательном неформальном рассмотрении. Отметим, что нас интересует даже не левенштейново отличие одной заданной последовательности от другой заданной, а отличие заданной последовательности от бесконечного множества последовательностей, образующих регулярный язык. Даже на первый взгляд эта задача заметно труднее, и

ловушек здесь больше. Поэтому задача левенштейновой аппроксимации последовательности предложением из заданного языка должна решаться с соблюдением как можно более жесткой формальной строгости.

9.5.4 Свойства функций Левенштейна

Мы введем новое, эквивалентное определение левенштейновского отличия. Оно несколько сложнее, чем приведенное ранее, но оно нужно для дальнейшего формального анализа.

Пусть X - конечный алфавит, а X^* - множество всех возможных последовательностей конечной длины, составленных из символов этого алфавита. Мы отождествим множество X^* с множеством вершин определенного графа. Строго говоря, поступать так некорректно, так как в этом случае мы будем говорить о графе с бесконечным количеством вершин, а это уже не граф. Однако не будем сейчас обращать внимание на неконструктивность введенного графа, так как он вводится лишь ради наглядности последующих рассуждений, а не для того, чтобы производить с ним те или иные вычислительные операции.

В соответствии с тремя функциями $in: X \rightarrow \mathbb{R}$, $ch: X \times X \rightarrow \mathbb{R}$ и $de: X \rightarrow \mathbb{R}$ введем в графе ребра трех типов: in , ch и de . Пусть некоторые две вершины графа соответствуют двум последовательностям вида $\bar{x}_1\bar{x}_2$, $\bar{x}_1x\bar{x}_2$, где $\bar{x}_1 \in X^*$, $\bar{x}_2 \in X^*$, $x \in X$. Это значит, что эти последовательности могут быть получены друг из друга с помощью вставки или исключения символа x . Введем стрелку типа in от вершины $\bar{x}_1\bar{x}_2$ к вершине $\bar{x}_1x\bar{x}_2$ и определим, что ее длина равна $in(x)$. Подобным образом введем стрелку типа de от вершины $\bar{x}_1x\bar{x}_2$ к вершине $\bar{x}_1\bar{x}_2$ и присвоим ей длину $de(x)$.

Пусть некоторые две вершины графа соответствуют последовательностям $\bar{x}_1y\bar{x}_2$ и $\bar{x}_1x\bar{x}_2$, где $\bar{x}_1 \in X^*$, $\bar{x}_2 \in X^*$, $x \in X$, $y \in X$. Введем стрелку типа ch от вершины $\bar{x}_1y\bar{x}_2$ к вершине $\bar{x}_1x\bar{x}_2$ и присвоим ей длину $ch(y, x)$. Введем также стрелку в обратном направлении, от вершины $\bar{x}_1x\bar{x}_2$ к вершине $\bar{x}_1y\bar{x}_2$, и присвоим ей длину $ch(x, y)$.

Левенштейново отличие определяется как функция $d: X^* \times X^* \rightarrow \mathbb{R}$, значение которой $d(\bar{y}, \bar{x})$ для пары \bar{y}, \bar{x} есть длина кратчайшего пути в построенном графе от вершины, представляющей последовательность \bar{y} , до вершины, представляющей последовательность \bar{x} . Важное свойство левенштейнова отклонения констатирует следующая лемма.

Лемма 9.1 О порядке редакторских операций. *Для любых двух последовательностей $\bar{y} \in X^*$ и $\bar{x} \in X^*$ один из кратчайших путей имеет следующий вид. Он начинается последовательностью (возможно, пустой) стрелок типа in , за ней следует последовательность (возможно, пустая) стрелок типа ch , и завершается путь последовательностью (возможно, пустой) стрелок типа de .* ▲

Доказательство. Пусть $\bar{y}, \bar{x}_1, \bar{x}_2, \dots, \bar{x}_n, \bar{x}$ - кратчайший путь от вершины \bar{y} к вершине \bar{x} , который не обладает свойством, указанным в лемме. Нарушение этого свойства обнаруживается в какой-то тройке последова-

тельных вершин \bar{x}_{i-1} , \bar{x}_i , \bar{x}_{i+1} , то есть, в паре стрелок \bar{x}_{i-1}, \bar{x}_i и \bar{x}_i, \bar{x}_{i+1} . Это может произойти только в трех ситуациях, каждую из которых рассмотрим отдельно.

1. Стрелка $(\bar{x}_{i-1}, \bar{x}_i)$ имеет тип de , а стрелка $(\bar{x}_i, \bar{x}_{i+1})$ тип in . Это значит, что последовательности $\bar{x}_{i-1}, \bar{x}_i, \bar{x}_{i+1}$ имеют один из двух следующих видов:

$$\left. \begin{array}{l} \bar{x}_{i-1} = \bar{x}'y\bar{x}''\bar{x}''' , \\ \text{или} \quad \bar{x}_i = \bar{x}'\bar{x}''\bar{x}''' , \\ \bar{x}_{i+1} = \bar{x}'\bar{x}''x\bar{x}''' , \end{array} \right\} \quad \text{или} \quad \left. \begin{array}{l} \bar{x}_{i-1} = \bar{x}'\bar{x}''y\bar{x}''' , \\ \bar{x}_i = \bar{x}'\bar{x}''\bar{x}''' , \\ \bar{x}_{i+1} = \bar{x}'x\bar{x}''\bar{x}''' , \end{array} \right\}$$

где $\bar{x}' \in X^*$, $\bar{x}'' \in X^*$, $\bar{x}''' \in X^*$, $x \in X$, $y \in X$.

В первом случае последовательность $\bar{x}_i = \bar{x}'\bar{x}''\bar{x}'''$ можно заменить на последовательность $\bar{x}'_i = \bar{x}'y\bar{x}''x\bar{x}'''$. Во втором случае ее можно заменить на $\bar{x}'_i = \bar{x}'x\bar{x}''y\bar{x}'''$. В обоих случаях мы получим новый путь из вершины \bar{y} в вершину \bar{x} с той же длиной, что и прежний путь. Но в новом пути стрелка $(\bar{x}_{i-1}, \bar{x}'_i)$ будет иметь тип in , а стрелка $(\bar{x}'_i, \bar{x}_{i+1})$ - тип de .

2. Стрелка $(\bar{x}_{i-1}, \bar{x}_i)$ имеет тип ch , стрелка $(\bar{x}_i, \bar{x}_{i+1})$ - тип in . Это возможно в следующих двух случаях:

$$\left. \begin{array}{l} \bar{x}_{i-1} = \bar{x}'y\bar{x}''\bar{x}''' , \\ \text{или} \quad \bar{x}_i = \bar{x}'x\bar{x}''\bar{x}''' , \\ \bar{x}_{i+1} = \bar{x}'x\bar{x}''z\bar{x}''' , \end{array} \right\} \quad \text{или} \quad \left. \begin{array}{l} \bar{x}_{i-1} = \bar{x}'\bar{x}''y\bar{x}''' , \\ \bar{x}_i = \bar{x}'\bar{x}''x\bar{x}''' , \\ \bar{x}_{i+1} = \bar{x}'z\bar{x}''\bar{x}''' . \end{array} \right\}$$

В первом случае последовательность $\bar{x}_i = \bar{x}'x\bar{x}''\bar{x}'''$ следует заменить на $\bar{x}'_i = \bar{x}'y\bar{x}''z\bar{x}'''$. Во втором случае \bar{x}_i надо заменить на $\bar{x}'_i = \bar{x}'z\bar{x}''y\bar{x}'''$. В обоих случаях получается новый путь от \bar{y} до \bar{x} с той же длиной, что и прежний. Но в новом пути стрелка $(\bar{x}_{i-1}, \bar{x}'_i)$ будет иметь тип in , а стрелка $(\bar{x}'_i, \bar{x}_{i+1})$ - тип ch .

3. Стрелка $(\bar{x}_{i-1}, \bar{x}_i)$ имеет тип de , а стрелка $(\bar{x}_i, \bar{x}_{i+1})$ - тип ch . Это возможно в следующих двух случаях:

$$\left. \begin{array}{l} \bar{x}_{i-1} = \bar{x}'x\bar{x}''y\bar{x}''' , \\ \text{или} \quad \bar{x}_i = \bar{x}'\bar{x}''y\bar{x}''' , \\ \bar{x}_{i+1} = \bar{x}'\bar{x}''z\bar{x}''' , \end{array} \right\} \quad \text{или} \quad \left. \begin{array}{l} \bar{x}_{i-1} = \bar{x}'y\bar{x}''x\bar{x}''' , \\ \bar{x}_i = \bar{x}'y\bar{x}''\bar{x}''' , \\ \bar{x}_{i+1} = \bar{x}'z\bar{x}''\bar{x}''' . \end{array} \right\}$$

В первом случае следует заменить последовательность $\bar{x}_i = \bar{x}'\bar{x}''y\bar{x}'''$ на $\bar{x}'_i = \bar{x}'x\bar{x}''z\bar{x}'''$. Во втором случае следует заменить $\bar{x}_i = \bar{x}'y\bar{x}''\bar{x}'''$ на $\bar{x}'_i = \bar{x}'z\bar{x}''x\bar{x}'''$. Полученный новый путь от \bar{y} до \bar{x} будет той же длины, что и прежний путь, но стрелка $(\bar{x}_{i-1}, \bar{x}'_i)$ будет иметь тип ch , а стрелка $(\bar{x}'_i, \bar{x}_{i+1})$ - тип de .

Последовательно изменяя путь от \bar{y} до \bar{x} по указанным правилам, мы в конечном итоге найдем путь, в котором ни одна из указанных трех ситуаций не встретится. Лемма доказана. ■

В силу доказанной леммы мы видим, что один из кратчайших путей от последовательности \bar{y} к последовательности \bar{x} состоит из трех участков, каждый из которых может быть и пустым. Первый участок проходит только по стрелкам типа *in*, второй - по стрелкам *ch*, а третий - по стрелкам *de*.

Учитывая это свойство левенштейнова отличия, мы можем дать его иное, эквивалентное определение. Определим три частных отличия, которые обозначим d_{in} , d_{ch} и d_{de} . Число $d_{in}(\bar{y}, \bar{x})$ определяется как длина (кратчайшего) пути от \bar{y} до \bar{x} , проходящего только по стрелкам типа *in*. В этом определении слово "кратчайший" лишнее. Путь от \bar{y} до \bar{x} по стрелкам d_{in} существует только тогда, когда \bar{y} есть подпоследовательность в последовательности \bar{x} . Если такой путь существует, то все пути от \bar{y} до \bar{x} имеют одинаковую длину. Если такого пути нет, будем считать, что $d_{in}(\bar{y}, \bar{x}) = \infty$.

Число $d_{de}(\bar{y}, \bar{x})$ определяется как длина (кратчайшего) пути от \bar{y} до \bar{x} , проходящая только по стрелкам типа *de*. Если такого пути нет, считается, что $d_{de}(\bar{y}, \bar{x}) = \infty$. И здесь тоже требование "кратчайший" лишнее, так как все пути от \bar{y} до \bar{x} имеют одну и ту же длину.

Подобным образом определяется число $d_{ch}(\bar{y}, \bar{x})$ как длина кратчайшего (и здесь это требование не лишнее) пути от \bar{y} до \bar{x} , проходящего только по стрелкам типа *ch*. Если такого пути нет, считается, что $d_{ch}(\bar{y}, \bar{x}) = \infty$. Последняя ситуация получается при различных длинах последовательностей \bar{y} и \bar{x} . Если последовательности одинаковые, то есть $\bar{y} = \bar{x}$, считается, что $d_{in}(\bar{y}, \bar{x}) = d_{ch}(\bar{y}, \bar{x}) = d_{de}(\bar{y}, \bar{x}) = 0$.

Введенные нами стрелки графа соответствуют редакторским операциям. Сейчас мы введем дополнительные стрелки в граф, которые будут соответствовать последовательному применению операций какого-то одного типа. Их мы назовем длинными стрелками в отличие от введенных ранее стрелок, которые будем называть короткими. Пусть \bar{y} и \bar{x} - две последовательности и пусть последовательность \bar{x} получена из \bar{y} вставкой некоторых символов. Введем в граф длинную стрелку типа *in* от \bar{y} до \bar{x} и присвоим ей длину $d_{in}(\bar{y}, \bar{x})$. Одновременно введем в граф длинную стрелку типа *de* от \bar{x} до \bar{y} и присвоим ей длину $d_{de}(\bar{x}, \bar{y})$. Пусть \bar{y} и \bar{x} - две последовательности одинаковой длины. Введем в граф длинную стрелку типа *ch* от \bar{y} до \bar{x} и присвоим ей длину $d_{ch}(\bar{y}, \bar{x})$.

После введения этих понятий становится ясным, что длина кратчайшего пути от \bar{y} до \bar{x} по коротким стрелкам, то есть левенштейново отличие \bar{x} от \bar{y} , равна длине кратчайшего пути от \bar{y} до \bar{x} по длинным стрелкам. Один из этих кратчайших путей проходит не более, чем три длинные стрелки типа *in*, *ch* и *de*, идущие друг за другом в указанном порядке. Некоторые из этих стрелок могут и отсутствовать. Математическим представлением этого высказывания есть выражение

$$d(\bar{y}, \bar{x}) = \min_{\bar{z}_1 \in X^*} \min_{\bar{z}_2 \in X^*} \left(d_{in}(\bar{y}, \bar{z}_1) + d_{ch}(\bar{z}_1, \bar{z}_2) + d_{de}(\bar{z}_2, \bar{x}) \right). \quad (9.13)$$

Это представление не годится для конструктивного вычисления отличия $d(\bar{y}, \bar{x})$ последовательности \bar{x} от последовательности \bar{y} , но оно полезно, по-

тому что раскладывает понятие левенштейнова отличия на три частных понятия, которые более обозримы. Видно, что вычисление левенштейнова отличия \bar{x} от \bar{y} сводится к отысканию двух вспомогательных последовательностей \bar{z}_1 и \bar{z}_2 , от которых зависят три слагаемые, сумму которых следует оптимизировать. Каждое из этих слагаемых может быть разложено далее на более и более элементарные составляющие, и это мы сделаем позже, когда до этого дойдет очередь.

9.5.5 Формулировка задачи и ее обсуждение

Пусть X, K - два конечных множества, а $\varphi: K \rightarrow \{0, 1\}$, $P: K \times X \times K \rightarrow \{0, 1\}$, $\psi: K \rightarrow \{0, 1\}$ - три функции, определяющие регулярный язык $L \subset X^*$ как множество последовательностей x_1, x_2, \dots, x_n , для которых справедливо утверждение

$$\bigvee_{k_0 \in K} \bigvee_{k_1 \in K} \cdots \bigvee_{k_n \in K} \left(\varphi(k_0) \wedge \left(\bigwedge_{i=1}^n P(k_{i-1}, x_i, k_i) \right) \wedge \psi(k_n) \right).$$

Пусть $in: X \rightarrow \mathbb{R}$, $ch: X \times X \rightarrow \mathbb{R}$ и $de: X \rightarrow \mathbb{R}$ - три функции, определяющие левенштейново отличие $d: X^* \times X^* \rightarrow \mathbb{R}$.

Задача состоит в создании алгоритма, который для каждой последовательности $\bar{x} \in X^*$ и каждой шестерки функций $(\varphi, P, \psi, in, ch, de)$ вычисляет число

$$D(\bar{x}) = \min_{\bar{y} \in L} d(\bar{y}, \bar{x}). \quad (9.14)$$

Эта задача существенно отличается от всех рассмотренных нами ранее оптимизационных задач, связанных с распознаванием последовательностей. Предыдущие задачи состояли в поиске оптимума на множестве последовательностей. Грубо говоря, следовало находить какую-то наилучшую последовательность. При этом из самого характера решаемой задачи была известна длина искомой последовательности. И хотя область поиска при этом оставалась очень обширной, она включала в себя конечное количество последовательностей, правда, очень большое. Задача (9.14), которую мы только что сформулировали, требует поиска последовательности с не заданной заранее длиной. Известно лишь, что это последовательность из заданного регулярного языка. Область поиска в этом случае включает бесконечное количество последовательностей.

Выразим эту особенность еще другими словами. В предыдущих задачах мы имели дело с многомерными задачами оптимизации. Количество переменных, которые следовало выбрать оптимальным образом, было очень большим, но все же заранее заданным. Задача (9.14) более сложная, чем задачи многомерной оптимизации, потому что здесь заранее не задано количество переменных, по которым следует что-то оптимизировать, и это количество может быть сколь угодно большим.

В сформулированной задаче мы усматриваем еще одну особенность. В предыдущих задачах речь шла об оптимизации функций, которые были в

определенном смысле простыми. Это значит, что если уж искомая последовательность найдена, то вычисление ее качества не представляло никакого труда. Оптимизируемая функция была задана в виде явной формулы, по которой следовало просто выполнить те или иные вычисления. Говоря иными словами, следовало найти $\min_x f(x)$ в ситуации, когда значение $f(x)$ легко вычисляется для любого x . Задача (9.14) существенно иная. Функция $d(\bar{y}, \bar{x})$, которую следует минимизировать, однозначно определена, но не представлена в виде явной формулы. Она определена в виде формулировки оптимизационной задачи, которую следует решить, даже если требуется найти значение $d(\bar{y}, \bar{x})$ для фиксированной пары \bar{y}, \bar{x} . Функция, которую следует оптимизировать, задана в виде вспомогательной оптимизационной задачи. Основная же задача состоит в нахождении таких параметров, которые оптимизируют результат решения вспомогательной задачи.

Вспомогательная задача сама по себе далеко не тривиальна. Строго говоря, на данной стадии нашего повествования алгоритм вычисления числа $d(\bar{y}, \bar{x})$ нам еще не известен. Ведь мы увидели, что общепринятый алгоритм, претендующий на это решение, не является таковым. Однако этот алгоритм псевдорешения имеет сложность порядка $\mathcal{O}(n_y n_x)$, где n_y и n_x - длины последовательностей \bar{y} и \bar{x} . Алгоритм, действительно решающий задачу, вряд ли окажется проще. Таким образом, в задаче (9.14) требуется найти наименьшее число в бесконечном множестве чисел, и в этом множестве имеются и числа, вычисление которых потребовать сколь угодно длительного времени. Ведь в регулярном языке могут присутствовать и сколь угодно длинные последовательности.

Мы поднимаем решаемую задачу на своеобразный пьедестал не только потому, что она этого заслуживает, а чтобы заранее подготовиться к тому, что конструирование алгоритма не может быть слишком легким. Сформулированная задача не допускает легкомысленного к себе отношения. Чтобы упростить дальнейшее изложение, в следующем подразделе мы сформулируем без доказательства основной теоретический результат, устанавливающий конструктивную разрешимость задачи. Доказательство этого результата приводится в последующих подразделах. В конце лекции приводятся явные формулы для решения задачи. Однако мы настоятельно рекомендуем воздержаться от их применения без ясного понимания, что они обозначают.

9.5.6 Формулировка основного результата и его обсуждение

Пусть $\langle X, K, \varphi, P, \psi \rangle$ - штрафной автомат. Это значит, что X и K - два конечных множества, а $\varphi: K \rightarrow \mathbb{R}$, $P: K \times X \times K \rightarrow \mathbb{R}$, $\psi: K \rightarrow \mathbb{R}$ - три неотрицательно определенные функции. Указанная пятерка определяет функцию $F: X^* \rightarrow \mathbb{R}$, значение $F(\bar{x})$, $\bar{x} \in X^*$, которой обозначает

минимальную стоимость генерирования последовательности \bar{x} ,

$$F(\bar{x}) = \min_{k_0} \min_{k_1} \cdots \min_{k_n} \left(\varphi(k_0) + \sum_{i=1}^n P(k_{i-1}, x_i, k_i) + \psi(k_n) \right). \quad (9.15)$$

Функция F проста в том смысле, что ее вычисление имеет сложность порядка $\mathcal{O}(|K|^2 n)$, где n - длина последовательности \bar{x} . В этом же смысле просты все рассмотренные до сих пор задачи.

Основной результат последующего анализа состоит в том, что для любой функции $D: X^* \rightarrow \mathbb{R}$ вида (9.14) существует ее эквивалентное представление в виде (9.15). Вычисление числа D имеет, таким образом, тот же порядок сложности, что и все ранее рассмотренные задачи. Оставим пока в стороне вопрос о том, насколько сложен переход от представления функции в виде (9.14) к представлению в виде (9.15). Сформулируем точно заявленный результат.

Теорема 9.1 Эквивалентное представление левенштейнова отличия.

Пусть X и K - два конечных множества, $\varphi: K \rightarrow \{0, 1\}$, $P: K \times X \times K \rightarrow \{0, 1\}$, $\psi: K \rightarrow \{0, 1\}$ - три функции, определяющие регулярный язык L , как множество последовательностей x_1, x_2, \dots, x_n , для которых выполняется

$$\bigvee_{k_0 \in K} \bigvee_{k_1 \in K} \cdots \bigvee_{k_n \in X} \left(\varphi(k_0) \wedge \left(\bigwedge_{i=1}^n P(k_{i-1}, x_i, k_i) \right) \wedge \psi(k_n) \right) = 1. \quad (9.16)$$

Пусть $in: X \rightarrow \mathbb{R}$, $ch: X \times X \rightarrow \mathbb{R}$, $de: X \rightarrow \mathbb{R}$ - три неотрицательно определенные функции, которые определяют левенштейновы отклонения $d: X^* \times X^* \rightarrow \mathbb{R}$ и $D: X^* \rightarrow \mathbb{R}$,

$$D(\bar{x}) = \min_{\bar{y} \in L} d(\bar{y}, \bar{x}). \quad (9.17)$$

Для каждой шестерки функций $(\varphi, P, \psi, in, ch, de)$, определяющих функцию D в соответствии с (9.17), существует такая пара функций P', ψ' , что равенство

$$D(\bar{x}) = \min_{k_0} \min_{k_1} \cdots \min_{k_n} \left(\varphi(k_0) + \sum_{i=1}^n P'(k_{i-1}, x_i, k_i) + \psi'(k_n) \right) \quad (9.18)$$

выполняется для любой последовательности $\bar{x} = (x_1, x_2, \dots, x_n) \in X^*$. \blacktriangle

Сформулированная теорема утверждает, что вычисление числа $D(\bar{x})$, вопреки всей сложности его определения, имеет сложность порядка $\mathcal{O}(|K|^2 n)$, где n - длина последовательности \bar{x} . Это сложность того же порядка, что и при распознавании принадлежности последовательности \bar{x} регулярному языку (самому обыкновенному, не размытому и не штрафному). Таким образом, теорема 9.1 выражает весьма привлекательное свойство, которое, однако, настолько априори неправдоподобно, что над этим стоит задуматься...

Не меньшего внимания заслуживает и следующее. Предположим, что нас интересует не вычисление $D(\bar{x})$ в соответствии с (9.17), а обработка пары фиксированных последовательностей \bar{y}, \bar{x} с целью (а) определить, принадлежит ли \bar{y} языку L , и (б) вычислить число $d(\bar{y}, \bar{x})$. Ответ на первый вопрос состоит в проверке равенства (9.16), сложность которой имеет порядок $\mathcal{O}(|K|^2 n_y)$, где n_y - длина последовательности \bar{y} . Хотя мы и не сформулировали алгоритм вычисления числа $d(\bar{y}, \bar{x})$, мы можем утверждать, что сложность его не меньше, чем $\mathcal{O}(n_y n_x)$. Суммарная сложность, таким образом, имеет порядок $\mathcal{O}(|K|^2 n_y + n_y n_x)$. Теорема 9.1 фактически утверждает, что вычисление $\min_{\bar{y} \in L} d(\bar{y}, \bar{x})$ имеет сложность порядка $\mathcal{O}(|K|^2 n_x)$, которая не зависит от длины последовательности \bar{y} , на которой достигается минимум, и более того, сложность вычисления $\min_{\bar{y} \in L} d(\bar{y}, \bar{x})$ меньше, чем сложность вычисления числа $d(\bar{y}, \bar{x})$ для некоторых последовательностей \bar{y} из L .

Все эти замечательные свойства, вытекающие из теоремы 9.1, представляются неправдоподобными, они противоречат интуиции. Поэтому доказательство теоремы должно удовлетворять самым строгим требованиям формальной правильности.

Мы будем доказывать теорему 9.1 в рамках формализма обобщенных сверток. Он похож на матричные произведения, которые мы уже использовали на этой и предыдущих лекциях. Поскольку сейчас мы будем использовать их не для краткой записи уже доказанных и интуитивно понятных утверждений, а для доказательства утверждений, не согласующихся с интуицией, сформулируем основные понятия этого формализма точнее, чем это было сделано до сих пор.

9.5.7 Обобщенные свертки и их свойства

Пусть W - множество, на котором заданы две двухместные операции \oplus и \otimes , определяющие для каждой пары элементов $x \in W$ и $y \in W$ результаты, соответственно, $x \oplus y$ и $x \otimes y$, также принадлежащие множеству W . Будем считать, что операции \oplus и \otimes обладают следующими свойствами:

1. $x \oplus y = y \oplus x$.
2. $(x \oplus y) \oplus z = x \oplus (y \oplus z)$.
3. $x \otimes y = y \otimes x$.
4. $(x \otimes y) \otimes z = x \otimes (y \otimes z)$.
5. $x \otimes (y \oplus z) = (x \otimes y) \oplus (x \otimes z)$.
6. Множество W содержит нулевой элемент, обозначаемый 0^\oplus , такой, что равенства $x \oplus 0^\oplus = x$ и $x \otimes 0^\oplus = 0^\oplus$ выполняются для любого $x \in W$.
7. Множество W содержит единичный элемент, обозначаемый 1^\otimes , такой, что равенство $x \otimes 1^\otimes = x$ выполняется для любого $x \in W$.

Множество W с введенными операциями \oplus и \otimes , обладающие указанными свойствами, называется коммутативным полукольцом.

Ключевым моментом для нашей задачи является, что пара операций $\min, +$ образует полукольцо на множестве вещественных чисел, дополнен-

ном специальным "числом" ∞ . Этому числу приписываются следующие свойства:

$$\begin{aligned}x + \infty &= \infty, \\ \min(x, \infty) &= x.\end{aligned}$$

Действительно, для любой пары элементов из $R \cup \{\infty\}$ выполняется

1. $\min(x, y) = \min(y, x)$.
2. $\min(\min(x, y), z) = \min(x, \min(y, z))$.
3. $x + y = y + x$.
4. $(x + y) + z = x + (y + z)$.
5. $x + \min(y, z) = \min(x + y, x + z)$.
6. Множество $R \cup \{\infty\}$ содержит "число" ∞ , для которого при любом x выполняются равенства $\min(x, \infty) = x$ и $x + \infty = \infty$. "Число" ∞ является, следовательно, нулевым элементом относительно операции \oplus .
7. В множестве $R \cup \{\infty\}$ содержится число 0 , которое при любом x удовлетворяет равенству $x + 0 = x$. Число 0 является, следовательно, единицей относительно операции \otimes .

Мы будем иметь дело с функциями, определенными на конечных множествах и принимающих значения на коммутативных полукольцах. Обозначение $f[x, y]$ будет использоваться для функции в целом, то есть отображения конечного множества в полукольцо W . Внутри квадратных скобок будут записаны идентификаторы переменных, служащих аргументами этой функции. Обозначение $f(x, y)$ с круглыми скобками будет использоваться для значения функции $f[x, y]$, которое она принимает при определенных значениях аргументов. Так, $f(a, b, c)$ обозначает значение функции $f[x, y, z]$ при $x = a, y = b, z = c$.

Пусть X - это множество $\{x_1, x_2, \dots, x_n\}$. Выражение $\bigoplus_{x \in X} f(x)$ будет использоваться для краткой записи суммы $f(x_1) \oplus f(x_2) \oplus \dots \oplus f(x_n)$.

Пусть X, Y, Z - три конечных множества, (W, \oplus, \otimes) - коммутативное полукольцо, а $f_1[x, y]: X \times Y \rightarrow W, f_2[y, z]: Y \times Z \rightarrow W$ - две функции. Выражение

$$f_1[x, y] \bigotimes_y f_2[y, z]$$

будет использоваться для краткой записи функции $f[x, z]: X \times Z \rightarrow W$, значения которой определяются выражением

$$f(x, z) = \bigoplus_{y \in Y} (f_1(x, y) \otimes f_2(y, z)). \quad (9.19)$$

Функция $f_1[x, y] \bigotimes_y f_2[y, z]$ будет называться *сверткой функций* $f_1[x, y]$ и $f_2[y, z]$ по переменной y .

Идентификаторы x, y, z в определении (9.19) следует понимать не только как обозначение одной переменной, но и как обозначение группы переменных, в том числе и пустой. Группа переменных, скажем, x, y, z со значениями $x \in X, y \in Y, z \in Z$ может пониматься и как одна переменная

со значениями в $X \times Y \times Z$. Так, свертка $f_1[x, y, z] \otimes_z f_2[z, y, u]$ есть функция $f[x, y, u]$. В общем случае, свертка двух функций f_1 and f_2 зависит от всех переменных, от которых зависят функции f_1 и f_2 , за исключением переменных, по которым выполняется свертка. Учитывая, что идентификаторы x, y, z в определении (9.19) могут пониматься и как пустые группы переменных, становится осмысленной и свертка $f_1[x, y] \otimes_y f_2[y]$, которая является функцией одной переменной x , равно как и свертка $f_1[x] \otimes f_2[y]$, являющаяся функцией двух переменных x и y , и наконец, свертка $f_1[x] \otimes_x f_2[x]$, которая вообще не функция, а некоторый элемент из W .

Мы можем видеть, что свертки являются обобщением матричного произведения в линейной алгебре. Определение (9.19) можно понимать, как произведение матрицы f_1 размера $|X| \times |Y|$ на матрицу f_2 размера $|Y| \times |Z|$. Результатом умножения является матрица f размера $|X| \times |Z|$. Отметим однако, что сверточная запись иногда более удобна, чем матричная. Операция свертки в коммутативном полукольце всегда коммутативна, то есть

$$f_1[x, y] \otimes_y f_2[y, z] = f_2[y, z] \otimes_y f_1[x, y].$$

Умножение же матриц не обязательно коммутативно, даже если речь идет о их умножении в коммутативном полукольце. Происходит это потому, что при записи свертки в виде матричного произведения исчезает возможность прямого указания переменной, по которой выполняется свертка. Это всегда должна быть вторая переменная в первом сомножителе и первая переменная во втором сомножителе. При записи произведения матриц в виде свертки отсутствует эта жесткая регламентация. Именно поэтому операция свертки коммутативна в отличие от произведения матриц. Кроме того, запись операции над функциями в виде сверток удобнее матричного представления, когда речь идет о функциях, зависящих более чем от двух аргументов.

При всей общности введенного определения, свертки обладают свойствами, благодаря которым их можно эквивалентно преобразовывать, упрощать и т.п. Дополнительно к уже упоминавшейся коммутативности это

- ассоциативность

$$\left(f_1[x, y] \otimes_y f_2[y, z] \right) \otimes_z f_3[z, u] = f_1[x, y] \otimes_y \left(f_2[y, z] \otimes_z f_3[z, u] \right), \quad (9.20)$$

- дистрибутивность

$$\begin{aligned} f_1[x, y] \otimes_y \left(f_2[y, z] \oplus f_3[y, z] \right) \\ = \left(f_1[x, y] \otimes_y f_2[y, z] \right) \oplus \left(f_1[x, y] \otimes_y f_3[y, z] \right), \end{aligned}$$

- и еще одно свойство без названия, вытекающее из коммутативности и ассоциативности

$$\left(f_1[x, y, z] \otimes_y f_2[y] \right) \otimes_z f_3[z] = \left(f_1[x, y, z] \otimes_z f_3[z] \right) \otimes_y f_2[y]. \quad (9.21)$$

Пусть X - некоторое множество, а $\delta[x, y]$ - функция вида $X \times X \rightarrow W$, для которой $\delta(x, y) = 1^\otimes$ при $x = y$ и $\delta(x, y) = 0^\oplus$ при $x \neq y$. Назовем ее функцией Кронекера. Для функции Кронекера выполняется равенство

$$f[x, y] \otimes_y \delta[y, z] = f[x, z].$$

Это значит, что свертка с функцией Кронекера не меняет саму функцию, а меняет только обозначение переменной, от которой она зависит.

Когда свертывание функций выполняется в полукольце, в котором в качестве пары \oplus, \otimes используются операции $\min, +$, свертки приобретают дополнительные свойства. Эти свойства вытекают из *идемпотентности сложения*, которая обозначает, что $x \oplus x = x$ при любом x . Пусть $f[x, y]$ функция вида $X \times X \rightarrow (\mathbb{R} \cup \{\infty\})$. Для любой функции f определим функцию $f^0[x, y]$ как функцию Кронекера, а функцию $f^n[x, y]$, как свертку $f[x, z] \otimes_z f^{n-1}[z, y]$.

Лемма 9.2 Сходимость суммы. Пусть множество X состоит из k элементов. В таком случае сумма

$$f^0[x, y] \oplus f^1[x, y] \oplus f^2[x, y] \oplus \dots \oplus f^n[x, y]$$

стремится к $(\delta \oplus f)^{k-1}$ при $n \rightarrow \infty$,

$$\lim_{n \rightarrow \infty} \bigoplus_{i=0}^n f^i[x, y] = (\delta \oplus f)^{k-1}[x, y]. \quad \blacktriangle$$

Доказательство. Докажем сначала вспомогательное утверждение, что при любом n выполняется

$$\bigoplus_{i=0}^n f^i = (\delta \oplus f)^n. \quad (9.22)$$

При $n = 0$ и $n = 1$ равенство (9.22) очевидно. При $n = 0$ левая часть равенства - это δ , а правая часть - это $(\delta \oplus f)^0$, то есть тоже δ . При $n = 1$ как правая, так и левая часть равенства (9.22) - это $\delta \oplus f$. Докажем, что если равенство (9.22) справедливо для какого-то n , то оно справедливо и для $n+1$.

Учитывая идемпотентность сложения, то есть свойство $f \oplus f = f$, запишем следующую цепочку равенств,

$$\begin{aligned}
\bigoplus_{i=0}^{n+1} f^i[x, y] &= \left(\bigoplus_{i=0}^n f^i[x, y] \right) \oplus \left(\bigoplus_{i=1}^{n+1} f^i[x, y] \right) \\
&= \left(\bigoplus_{i=0}^n f^i[x, y] \right) \oplus \left(f[x, z] \otimes_z \left(\bigoplus_{i=0}^n f^i[z, y] \right) \right) \\
&= \left(\delta[x, z] \otimes_z \left(\bigoplus_{i=0}^n f^i[z, y] \right) \right) \oplus \left(f[x, z] \otimes_z \left(\bigoplus_{i=0}^n f^i[z, y] \right) \right) \\
&= \left(\delta[x, z] \oplus f[x, z] \right) \otimes_z \left(\bigoplus_{i=0}^n f^i[z, y] \right) \\
&= \left(\delta[x, z] \oplus f[x, z] \right) \otimes_z \left(\delta[z, y] \oplus f[z, y] \right)^n \\
&= \left(\delta[x, y] \oplus f[x, y] \right)^{n+1}.
\end{aligned}$$

Докажем теперь основное утверждение леммы 9.2. Отождествим множество X с множеством вершин полного направленного графа и припишем стрелке от вершины x к вершине y , $x \in X$, $y \in X$, длину $f(x, y)$. Число $f(x, y)$ есть длина пути от вершины x к вершине y , который состоит из одной стрелки. Этот путь единственный. Число $f(x, z) \otimes_z f(z, y)$ - это длина пути от x до y , кратчайшего среди путей, которые состоят из двух стрелок. И вообще, $f^n(x, y)$ - это длина пути от x до y , кратчайшего среди тех, длина которых состоит из n стрелок. Это утверждение справедливо и для $n = 0$. Кратчайший путь от x до y , не содержащий ни одной стрелки, равен, очевидно, $0 = 1^{\otimes}$ при $x = y$, и равен $\infty = 0^{\otimes}$ при $x \neq y$. Длина этого пути, таким образом, равна $\delta(x, y)$, то есть $f^0(x, y)$. Сумма $\bigoplus_{i=0}^n f^i[x, y]$ - это длина пути от x до y , кратчайшего среди путей, длина которых не превосходит n . Один из этих кратчайших путей проходит не более чем через $k - 1$ стрелок. Действительно, путь, проходящий более, чем через $k - 1$ стрелок, обязательно включает в себя цикл, который можно исключить и получить другой путь, тоже оптимальный, но состоящий из меньшего количества стрелок. Таким образом, при $n \geq k$ выполняется

$$\bigoplus_{i=0}^n f^i[x, y] = \bigoplus_{i=0}^{k-1} f^i[x, y].$$

Используя ранее доказанное равенство (9.22), получаем

$$\lim_{n \rightarrow \infty} \bigoplus_{i=0}^n f^i = (\delta \oplus f)^{k-1}.$$

■

Доказанная лемма указывает конструктивный способ вычисления бесконечного полинома $\bigoplus_{i=0}^{\infty} f^i$. Он будет обозначаться f^* , где f - та или иная функция вида $X \times X \rightarrow W$.

На основе леммы 9.2 можно вывести конструктивный способ вычисления свертки по множеству последовательностей, то есть по переменной, которая может принимать и бесконечное количество значений.

Лемма 9.3 Свертки по последовательностям. Пусть K и X - два конечных множества, $X^* = \bigcup_{i=0}^{\infty} X^i$, а $f: K \times X \times K \rightarrow W$, $\varphi: X \rightarrow W$ - две функции. Пусть $\#$ обозначает пустую последовательность (нулевой длины).

Пусть $F: K \times X^* \times K \rightarrow W$ - такая функция, что

$$F(k', \#, k'') = \delta(k', k''), \quad k' \in K, \quad k'' \in K, \quad (9.23)$$

$$F(k', \bar{x}x, k'') = F(k', \bar{x}, k'') \bigotimes_k f(k, x, k''), \quad k' \in K, \quad k'' \in K, \quad \bar{x} \in X^*, \quad x \in X.$$

Пусть $\Phi: X^* \rightarrow W$ - такая функция, что

$$\Phi(\#) = 1^{\otimes},$$

$$\Phi(\bar{x}x) = \Phi(\bar{x}) \otimes \varphi(x), \quad \bar{x} \in X^*, \quad x \in X. \quad (9.24)$$

В этом случае свертка

$$F[k', \bar{x}, k''] \bigotimes_{\bar{x}} \Phi[\bar{x}] \quad (9.25)$$

равна

$$\left(\delta[k', k''] \oplus \left(f[k', x, k''] \bigotimes_x \varphi(x) \right) \right)^{|K|-1}.$$

▲

Доказательство. Непосредственно из определения свертки следует, что свертка (9.25) есть лишь краткое обозначение функции двух переменных k' и k'' , значение которой для данной пары $(k', k'') \in K^2$ есть сумма

$$\bigoplus_{n=0}^{\infty} \bigoplus_{\bar{x} \in X^n} F(k', \bar{x}, k'') \otimes \Phi(\bar{x}), \quad (9.26)$$

где X^n - множество последовательностей x_1, x_2, \dots, x_n , $x_i \in X$, длины n . Это утверждение будет повторяться несколько раз, и поэтому выразим его в виде равенства

$$F[k', \bar{x}, k''] \bigotimes_{\bar{x}} \Phi[\bar{x}] = \bigoplus_{n=0}^{\infty} \bigoplus_{\bar{x} \in X^n} F(k', \bar{x}, k'') \otimes \Phi(\bar{x}). \quad (9.27)$$

Строго говоря, приведенное равенство некорректно, так как в левой части его записана функция $K \times K \rightarrow W$, а в правой части - значение, которое эта функция принимает для пары k', k'' . Несмотря на эту некорректность, мы используем обозначение (9.27), полагая, что после сделанной оговорки недоразумения уже не возникнут.

Сумма

$$\bigoplus_{\bar{x} \in X^0} F(k', \bar{x}, k'') \otimes \Phi(\bar{x})$$

равна, очевидно,

$$F(k', \#, k'') \otimes \Phi(\#) = \delta(k', k'') \otimes 1^{\otimes} = \delta(k', k'').$$

В соответствии с определениями (9.23), (9.24) и определением свертки суммы

$$\bigoplus_{\bar{x} \in X^1} F(k', \bar{x}, k'') \otimes \Phi(\bar{x})$$

приобретает вид

$$\bigoplus_{\bar{x} \in X^1} F(k', \bar{x}, k'') \otimes \Phi(\bar{x}) = f[k', x, k''] \bigotimes_x \varphi(x). \quad (9.28)$$

Докажем, что при любом $n = 1, 2, \dots$ выполняется равенство

$$\bigoplus_{\bar{x} \in X^n} F(k', \bar{x}, k'') \otimes \Phi(\bar{x}) = \left(f[k', x, k''] \bigotimes_x \varphi[x] \right)^n. \quad (9.29)$$

При $n = 1$ равенство справедливо, так как оно совпадает с равенством (9.28). Докажем, что если равенство (9.29) справедливо для некоторого n , то оно справедливо и для $n + 1$. Это доказательство приводит следующая цепочка равенств,

$$\begin{aligned} \bigoplus_{\bar{x} \in X^{n+1}} F(k', \bar{x}, k'') \otimes \Phi(\bar{x}) &= \bigoplus_{\bar{x} \in X^n} \bigoplus_{x \in X} F(k', \bar{x}x, k'') \otimes \Phi(\bar{x}x) \\ &= \bigoplus_{\bar{x} \in X^n} \bigoplus_{x \in X} \bigoplus_{k \in K} F(k', \bar{x}, k) \otimes f(k, x, k'') \otimes \Phi(\bar{x}) \otimes \varphi(x) \\ &= \bigoplus_{k \in K} \left(\bigoplus_{\bar{x} \in X^n} F(k', \bar{x}, k) \otimes \Phi(\bar{x}) \right) \otimes \left(\bigoplus_{x \in X} f(k, x, k'') \otimes \varphi(x) \right) \\ &= \left(f[k', x, k] \bigotimes_x \varphi[x] \right)^n \bigotimes_k \left(f[k, x, k''] \bigotimes_x \varphi[x] \right) \\ &= \left(f[k', x, k''] \bigotimes_x \varphi[x] \right)^{n+1}. \end{aligned}$$

Подставим (9.29) в (9.26) и получим, что свертка (9.26) равна

$$\bigoplus_{n=0}^{\infty} \left(f[k', x, k''] \bigotimes_x \varphi[x] \right)^n.$$

Далее на основании леммы 9.2 получаем, что свертка (9.26) равна

$$\left(\delta[k', k''] \oplus \left(f[k', x, k''] \bigotimes_x \varphi[x] \right) \right)^{|K|-1}.$$

Лемма доказана. \blacksquare

Мы видим, что чисто формальный анализ сверток позволяет формулировать правила их эквивалентных преобразований. Существенно, что два из этих правил, обоснованные в лемме 9.2 и лемме 9.3, позволяют заменять бесконечные свертки их конечным эквивалентом. Основная трудность в левенштейновой аппроксимации состоит именно в том, что требуется отыскать минимум на множестве всех возможных последовательностей произвольной длины. Чтобы воспользоваться полученными правилами замены бесконечных сверток конечными эквивалентами, сформулируем задачу левенштейновой аппроксимации в конволюционном (сверточном) виде. Исходная формулировка задачи была приведена в подразделе 9.5.5.

9.5.8 Конволюционная формулировка задачи и основного результата

Пусть X и K - два конечных множества, а $\varphi: K \rightarrow \{0, \infty\}$, $P: K \times X \times K \rightarrow \{0, \infty\}$, $\psi: K \rightarrow \{0, \infty\}$ - три функции, которые определяют язык $L \subset X^*$ так, что последовательность $\bar{x} = (x_1, x_2, \dots, x_n)$ принадлежит L тогда и только тогда, когда

$$F(\bar{x}) = \min_{k_0 \in K} \min_{k_1 \in K} \cdots \min_{k_n \in K} \left(\varphi(k_0) + \sum_{i=1}^n P(k_{i-1}, x_i, k_i) + \psi(k_n) \right) = 0. \quad (9.30)$$

Это условие эквивалентно существованию последовательности k_0, k_1, \dots, k_n , для которой справедливо

$$\left. \begin{aligned} \varphi(k_0) &= 0, \\ P(k_{i-1}, x_i, k_i) &= 0, \quad i = 1, 2, \dots, n, \\ \psi(k_n) &= 0, \end{aligned} \right\} \quad (9.31)$$

так как для любой последовательности k_0, k_1, \dots, k_n сумма

$$\varphi(k_0) + \sum_{i=1}^n P(k_{i-1}, x_i, k_i) + \psi(k_n)$$

равна либо 0, либо ∞ . Эта сумма равна 0, когда все условия (9.31) выполняются, и равна ∞ , если по крайней мере одно условие из семейства (9.31) не выполняется. Очевидно, что язык L , определенный таким образом, является регулярным и, наоборот, любой регулярный язык можно определить в виде (9.31).

Пусть функции $in: X \rightarrow \mathbb{R}$, $ch: X \times X \rightarrow \mathbb{R}$, $de: X \rightarrow \mathbb{R}$ определяют функцию Левенштейна $d: X^* \times X^* \rightarrow \mathbb{R}$. Решаемая задача (9.14) состоит в том, чтобы для каждой предъявленной последовательности $\bar{x} \in L$ вычислить число

$$D(\bar{x}) = \min_{\bar{y} \in L} d(\bar{y}, \bar{x}). \quad (9.32)$$

Число (9.30), зависящее от последовательности $\bar{x} \in X^*$, обозначим $F(\bar{x})$. Поскольку $F(\bar{x})$ принимает только значения 0 или ∞ , число $D(\bar{x})$, опреде-

ленное требованием (9.32), можно выразить в виде

$$D(\bar{x}) = \min_{\bar{y} \in X^*} \left(F(\bar{y}) + d(\bar{y}, \bar{x}) \right). \quad (9.33)$$

Это же можно выразить в виде свертки по всем возможным последовательностям $\bar{y} \in X^*$

$$D[\bar{x}] = F[\bar{y}] \bigotimes_{\bar{y}} d[\bar{y}, \bar{x}] \quad (9.34)$$

в полукольце, образованном операциями \min и $+$ на множестве неотрицательных вещественных чисел, дополненном "числом" ∞ .

В соответствии с (9.30) функцию $F: X^* \rightarrow \mathbb{R}$ можно выразить в виде свертки

$$\begin{aligned} F[\bar{y}] &= F[y_1, y_2, \dots, y_n] \quad (9.35) \\ &= \varphi[k_0] \bigotimes_{k_0} P[k_0, y_1, k_1] \bigotimes_{k_1} P[k_1, y_2, k_2] \bigotimes_{k_2} \dots \bigotimes_{k_{n-1}} P[k_{n-1}, y_n, k_n] \bigotimes_{k_n} \psi[k_n]. \end{aligned}$$

Далее мы будем опираться на тот очевидный факт, что функции $F: X^* \rightarrow \mathbb{R}$, которые могут быть представлены в виде (9.35), можно представить и в виде

$$F[\bar{y}] = \varphi[k] \bigotimes_k f[k, \bar{y}, k'] \bigotimes_{k'} \psi[k'],$$

где

$$f(k', \bar{y}, k'') = f(k', \#, k'') = \delta(k', k''), \quad k' \in K, \quad k'' \in K$$

для пустой последовательности \bar{y} и

$$f[k', \bar{y}, k''] = f[k', \bar{y}_1 y, k''] = f[k', \bar{y}_1, k] \bigotimes_k P[k, y, k'']$$

для непустой последовательности \bar{y} .

Основной результат, который ранее был заявлен в теореме 9.1, можно теперь выразить в следующей, конволюционной форме.

Теорема 9.2 **Основной результат в конволюционной форме.** *Для каждой шестерки функций $(\varphi, P, \psi, in, ch, de)$ существуют такие две функции $P': K \times X \times K \rightarrow \mathbb{R}$ и $\psi': K \rightarrow \mathbb{R}$, с помощью которых функция $D: X^* \rightarrow \mathbb{R}$, определенная выражением (9.34), представляется в виде*

$$D[\bar{x}] = \varphi[k] \bigotimes_k f'[k, \bar{x}, k'] \bigotimes_{k'} \psi'[k'], \quad (9.36)$$

где

$$\begin{aligned} f'(k, \#, k') &= \delta(k, k'), \quad k \in K, \quad k' \in K, \\ f'[k', \bar{x}, k''] &= f'[k', \bar{x}, k] \bigotimes_k P'[k, x, k'']. \end{aligned}$$



Теорема 9.2 утверждает, что левенштейново отличие последовательности $\bar{x} = (x_1, x_2, \dots, x_n)$ от регулярного языка L можно вычислить, как матричное произведение

$$\varphi P'_1 P'_2 \cdots P'_{n-1} P'_n \psi',$$

где векторы φ и ψ' представляют функции φ и ψ' одной переменной $k \in K$, а матрицы P'_i , $i = 1, \dots, n$, представляют функции $P'[k, x_i, k']$ двух переменных $k \in K$, $k' \in K$. Это значит, что это вычисление имеет сложность порядка $\mathcal{O}(|K|^2 n)$, где n - длина последовательности \bar{x} .

Доказательство заявленного результата приводится в следующем подразделе.

9.5.9 Доказательство основного результата

Представим доказанное ранее свойство (9.13) левенштейновых функций $d[\bar{y}, \bar{x}]$ в виде свертки

$$d[\bar{y}, \bar{x}] = d_{in}[\bar{y}, \bar{y}_1] \otimes_{\bar{y}_1} d_{ch}[\bar{y}_1, \bar{y}_2] \otimes_{\bar{y}_2} d_{de}[\bar{y}_2, \bar{x}].$$

Подставим полученное выражение для $d[\bar{y}, \bar{x}]$ в определение (9.34),

$$D[\bar{x}] = F[\bar{y}] \otimes_{\bar{y}} \left(d_{in}[\bar{y}, \bar{y}_1] \otimes_{\bar{y}_1} d_{ch}[\bar{y}_1, \bar{y}_2] \otimes_{\bar{y}_2} d_{de}[\bar{y}_2, \bar{x}] \right), \quad (9.37)$$

и в силу ассоциативности свертки запишем выражение (9.37) как

$$D[\bar{x}] = \left(\left(F[\bar{y}] \otimes_{\bar{y}} d_{in}[\bar{y}, \bar{y}_1] \right) \otimes_{\bar{y}_1} d_{ch}[\bar{y}_1, \bar{y}_2] \right) \otimes_{\bar{y}_2} d_{de}[\bar{y}_2, \bar{x}].$$

Доказательство теоремы 9.2 сводится к доказательству следующих трех лемм, которые раскладывают оптимальное редактирование одной последовательности в другую на три редактирования: с помощью одних лишь вставок символов, одних лишь замен и одних лишь исключений.

Лемма 9.4 Редактирование последовательностей с помощью вставок.

Пусть X, K - два конечных множества, а $\varphi: K \rightarrow \mathbb{R}$, $P: K \times X \times K \rightarrow \mathbb{R}$, $\psi: K \rightarrow \mathbb{R}$ - три функции, которые определяют функцию $F: X^* \rightarrow \mathbb{R}$ так, что

$$F[\bar{y}] = \varphi[k'] \otimes_{k'} f[k', \bar{y}, k''] \otimes_{k''} \psi[k''], \quad (9.38)$$

где

$$\left. \begin{aligned} f[k, \#, k''] &= \delta[k', k''], \\ f[k', \bar{x}x, k''] &= f[k', \bar{x}, k] \otimes_k P[k, x, k'']. \end{aligned} \right\}$$

Пусть функция $in: X \rightarrow \mathbb{R}$ определяет функцию $d_{in}: X^* \times X^* \rightarrow \mathbb{R}$ так, что значение $d_{in}(\bar{y}, \bar{y}_1)$ есть стоимость самой дешевой последовательности вставок, которая превращает последовательность \bar{y} в последовательность \bar{y}_1 .

В таком случае существует такая функция $P_1: K \times X \times K \rightarrow \mathbb{R}$, что функция

$$F_1[\bar{y}_1] = F[\bar{y}] \bigotimes_{\bar{y}} d_{in}[\bar{y}, \bar{y}_1] \quad (9.39)$$

идентична функции

$$\varphi[k'] \bigotimes_{k'} f_1[k', \bar{y}_1, k''] \bigotimes_{k''} \psi[k''],$$

где

$$f_1[k', \#, k''] = \delta[k', k''], \quad (9.40)$$

$$f_1[k', \bar{y}_1 y_1, k''] = f_1[k', \bar{y}_1, k] \bigotimes_k P_1[k, y_1, k'']. \quad (9.41)$$

▲

Лемма 9.5 Редактирование последовательностей с помощью замен.

Пусть X, K - два конечных множества, а $\varphi: K \rightarrow \mathbb{R}$, $P_1: K \times X \times K \rightarrow \mathbb{R}$, $\psi: K \rightarrow \mathbb{R}$ - три функции, определяющие функцию $F_1: X^* \rightarrow \mathbb{R}$ так, что

$$F_1[\bar{y}_1] = \varphi[k'] \bigotimes_{k'} f_1[k', \bar{y}_1, k''] \bigotimes_{k''} \psi[k''], \quad (9.42)$$

где

$$\left. \begin{aligned} f_1[k', \#, k''] &= \delta[k', k''], \\ f_1[k', \bar{y}_1 y_1, k''] &= f_1[k', \bar{y}_1, k] \bigotimes_k P_1[k, y_1, k'']. \end{aligned} \right\} \quad (9.43)$$

Пусть функция $ch: X \times X \rightarrow \mathbb{R}$ определяет функцию $d_{ch}: X^* \times X^* \rightarrow \mathbb{R}$, значение $d_{ch}[\bar{y}_1, \bar{y}_2]$ которой есть стоимость самой дешевой последовательности замен, превращающих последовательность \bar{y}_1 в последовательность \bar{y}_2 .

В таком случае существует такая функция $P_2: K \times X \times K \rightarrow \mathbb{R}$, что функция

$$F_2[\bar{y}_2] = F_1[\bar{y}_1] \bigotimes_{\bar{y}_1} d_{ch}[\bar{y}_1, \bar{y}_2] \quad (9.44)$$

идентична функции

$$\varphi[k'] \bigotimes_{k'} f_2[k', \bar{y}_2, k''] \bigotimes_{k''} \psi[k''], \quad (9.45)$$

где

$$f_2[k', \#, k''] = \delta[k', k''], \quad (9.46)$$

$$f_2[k', \bar{y}_2 y_2, k''] = f_2[k', \bar{y}_2, k] \bigotimes_k P_2[k, y_2, k'']. \quad (9.47)$$

▲

Лемма 9.6 Редактирование последовательностей с помощью исключений. Пусть X, K - два конечных множества, а $\varphi: K \rightarrow \mathbb{R}$, $P_2: K \times X \times K \rightarrow \mathbb{R}$, $\psi: K \rightarrow \mathbb{R}$ - три функции, определяющие функцию $F_2: X^* \rightarrow \mathbb{R}$ так, что

$$F_2[\bar{y}_2] = \varphi[k'] \bigotimes_{k'} f_2[k', \bar{y}_2, k''] \bigotimes_{k''} \psi[k''], \quad (9.48)$$

где

$$\left. \begin{aligned} f_2[k', \#, k''] &= \delta[k', k''], \\ f_2[k', \bar{y}_2 y_2, k''] &= f_2[k', \bar{y}_2, k] \bigotimes_k P_2[k, y_2, k'']. \end{aligned} \right\} \quad (9.49)$$

Пусть функция $de: X \rightarrow \mathbb{R}$ определяет функцию $d_{de}: X^* \times X^* \rightarrow \mathbb{R}$, значение $d_{de}[\bar{y}_2, \bar{x}]$ которой есть стоимость самой дешевой последовательности исключений, превращающей последовательность \bar{y}_2 в последовательность \bar{x} .

В таком случае существуют такие функции $P': K \times X \times K \rightarrow \mathbb{R}$ и $\psi': K \rightarrow \mathbb{R}$, что функция

$$D[\bar{x}] = F_2[\bar{y}_2] \bigotimes_{\bar{y}_2} d_{de}[\bar{y}_2, \bar{x}] \quad (9.50)$$

идентична функции

$$\varphi[k'] \bigotimes_{k'} f'[k', \bar{x}, k''] \bigotimes_{k''} \psi'[k''],$$

где

$$f'[k', \#, k''] = \delta[k', k''], \quad (9.51)$$

$$f'[k', \bar{x}x, k''] = f'[k', \bar{x}, k] \bigotimes_k P'[k, x, k'']. \quad (9.52)$$

▲

Доказательство. (леммы 9.4)

1. Для чисел $d_{in}(\bar{y}, \bar{y}_1)$ справедливо следующее. Если являются пустыми как \bar{y} , так и \bar{y}_1 , то $d_{in}(\#, \#) = 0 = 1^{\otimes}$. Если \bar{y} - пустая последовательность, а \bar{y}_1 - непустая, то, очевидно,

$$d_{in}(\#, \bar{y}'_1 y_1) = d_{in}(\#, \bar{y}'_1) + d_{in}(\#, y_1),$$

или то же самое, записанное в виде свертки,

$$d_{in}[\#, \bar{y}'_1 y_1] = d_{in}[\#, \bar{y}'_1] \otimes d_{in}[\#, y_1]. \quad (9.53)$$

Если последовательность \bar{y} состоит из единственного символа y , а последовательность \bar{y}_1 - из единственного символа y_1 , то $d_{in}(\bar{y}, \bar{y}_1) = \infty$ при $(y \neq y_1)$ и $d_{in}(\bar{y}, \bar{y}_1) = 0$ в противном случае. Запишем это утверждение кратко,

$$d_{in}[y, y_1] = \delta[y, y_1]. \quad (9.54)$$

Если как \bar{y} , так и \bar{y}_1 непустые, то значение $d_{in}(\bar{y}, \bar{y}_1)$ получается в результате следующих соображений. Оптимальное преобразование последовательности $\bar{y} = \bar{y}'y$ в последовательность $\bar{y}_1 = \bar{y}'_1y_1$ с помощью одних лишь вставок может быть лишь следующих двух видов: либо символ y_1 был вставлен, либо он не был вставлен. В первом случае стоимость самой дешевой последовательности вставок, очевидно, равна

$$d_{in}(\bar{y}'y, \bar{y}'_1) + d_{in}(\#, y_1).$$

Во втором случае она равна

$$d_{in}(\bar{y}', \bar{y}'_1) + d_{in}(y, y_1).$$

Число $d_{in}(\bar{y}'y, \bar{y}'_1y_1)$ есть меньшее из этих двух чисел,

$$d_{in}(\bar{y}'y, \bar{y}'_1y_1) = \min(d_{in}(\bar{y}'y, \bar{y}'_1) + d_{in}(\#, y_1), d_{in}(\bar{y}', \bar{y}'_1) + d_{in}(y, y_1))$$

или, в виде свертки,

$$d_{in}[\bar{y}'y, \bar{y}'_1y_1] = (d_{in}[\bar{y}'y, \bar{y}'_1] \otimes d_{in}[\#, y_1]) \oplus (d_{in}[\bar{y}', \bar{y}'_1] \otimes d_{in}[y, y_1]). \quad (9.55)$$

2. Исследуем теперь более детально функцию

$$F_1[\bar{y}_1] = F[\bar{y}] \bigotimes_{\bar{y}} d_{in}[\bar{y}, \bar{y}_1].$$

Используем выражение (9.38) для $F[\bar{y}]$ и получим

$$F_1[\bar{y}_1] = \varphi[k'] \bigotimes_{k'} f[k', \bar{y}, k''] \bigotimes_{k''} \psi[k''] \bigotimes_{\bar{y}} d_{in}[\bar{y}, \bar{y}_1]$$

или, на основании свойства (9.21),

$$F_1[\bar{y}_1] = \varphi[k'] \bigotimes_{k'} \left(f[k', \bar{y}, k''] \bigotimes_{\bar{y}} d_{in}[\bar{y}, \bar{y}_1] \right) \bigotimes_{k''} \psi[k'']$$

Таким образом мы доказали, что функция $F_1[\bar{y}_1]$ имеет вид

$$\varphi[k'] \bigotimes_{k'} f_1[k', \bar{y}_1, k''] \bigotimes_{k''} \psi[k''],$$

где

$$f_1[k', \bar{y}_1, k''] = f[k', \bar{y}, k''] \bigotimes_{\bar{y}} d_{in}[\bar{y}, \bar{y}_1]. \quad (9.56)$$

Теперь следует доказать, что функция $f_1[k', \bar{y}_1, k'']$, определенная в соответствии с (9.56), удовлетворяет условиям (9.40) и (9.41).

3. Докажем свойство (9.40). Если $\bar{y}_1 = \#$, то $d_{in}(\bar{y}, \bar{y}_1) = 0 = 1^{\otimes}$ только в случае $\bar{y} = \#$. Действительно, непустую последовательность невозможно превратить в пустую, вставляя в нее символы. Равенство (9.56) в этом случае приобретает вид

$$f_1[k', \#, k''] = f[k', \#, k''].$$

По условию леммы 9.4 выполняется равенство $f[k', \#, k''] = \delta[k', k'']$, в силу чего справедливо и равенство $f_1[k', \#, k''] = \delta[k', k'']$.

4. Запишем более детально выражение (9.56) для непустой последовательности \bar{y}_1 , которая, следовательно, имеет вид $\bar{y}_1 y_1$,

$$f_1(k', \bar{y}_1 y_1, k'') = \bigoplus_{\bar{y} \in X^*} f(k', \bar{y}, k'') \otimes d_{in}(\bar{y}, \bar{y}_1 y_1).$$

Слагаемые в правой части этого выражения включают в себя одно слагаемое $f(k', \bar{y}, k'') \otimes d_{in}(\bar{y}, \bar{y}_1 y_1)$, соответствующее пустой последовательности $\bar{y} = \#$, и все остальные слагаемые для непустых последовательностей вида $\bar{y} y$. Поэтому

$$\begin{aligned} f_1(k', \bar{y}_1 y_1, k'') &= f(k', \#, k'') \otimes d_{in}(\#, \bar{y}_1 y_1) \\ &\oplus \bigoplus_{\bar{y} \in X^*} \bigoplus_{y \in X} f(k', \bar{y} y, k'') \otimes d_{in}(\bar{y} y, \bar{y}_1 y_1). \end{aligned}$$

Используя выражение (9.53) для $d_{in}(\#, \bar{y}_1 y_1)$, выражение (9.39) для $f(k', \bar{y} y, k'')$ и выражение (9.55) для $d_{in}(\bar{y} y, \bar{y}_1 y_1)$, получаем

$$\begin{aligned} f_1(k', \bar{y}_1 y_1, k'') &= f(k', \#, k'') \otimes d_{in}(\#, \bar{y}_1) \otimes d_{in}(\#, y_1) \quad (9.57) \\ &\oplus \bigoplus_{\bar{y} \in X^* \setminus \{\#\}} f(k', \bar{y}, k'') \otimes d_{in}(\bar{y}, \bar{y}_1) \otimes d_{in}(\#, y_1) \\ &\oplus \bigoplus_{\bar{y} \in X^*} \bigoplus_{k \in K} \left(f(k', \bar{y}, k) \otimes d(\bar{y}, \bar{y}_1) \otimes \left(\bigoplus_{y \in X} P(k, y, k'') \otimes d_{in}(y, y_1) \right) \right). \end{aligned}$$

Запишем первую и вторую строки в последнем выражении в виде суммы по всем последовательностям \bar{y} , включая пустую,

$$\begin{aligned} f_1(k', \bar{y}_1 y_1, k'') &= \bigoplus_{\bar{y} \in X^*} f(k', \bar{y}, k'') \otimes d_{in}(\bar{y}, \bar{y}_1) \otimes d_{in}(\#, y_1) \quad (9.58) \\ &\oplus \bigoplus_{\bar{y} \in X^*} \bigoplus_{k \in K} \left(f(k', \bar{y}, k) \otimes d(\bar{y}, \bar{y}_1) \otimes \left(\bigoplus_{y \in X} P(k, y, k'') \otimes d_{in}(y, y_1) \right) \right). \end{aligned}$$

5. В силу дистрибутивности умножения относительно сложения первую строку в (9.58) можно записать как

$$d_{in}(\#, y_1) \otimes \left(\bigoplus_{\bar{y} \in X^*} f(k', \bar{y}, k'') \otimes d_{in}(\bar{y}, \bar{y}_1) \right),$$

где сумма в круглых скобках есть $f_1(k', \bar{y}_1, k'')$ по определению (9.56). Таким образом, первая строка в выражении (9.58) есть

$$f_1(k', \bar{y}_1, k'') \otimes d_{in}(\#, \bar{y}_1). \quad (9.59)$$

В силу свойства функции $\delta[k, k']$ Кронекера число (9.59) можно записать в виде свертки

$$\bigoplus_{k \in K} f_1(k', \bar{y}_1, k) \otimes \delta(k, k'') \otimes d_{in}(\#, y_1). \quad (9.60)$$

6. Рассмотрим вторую строку в (9.58). На основе равенства (9.54) можно записать

$$\bigoplus_{y \in X} P(k, y, k'') \otimes d_{in}(y, y_1) = \bigoplus_{y \in X} P(k, y, k'') \otimes \delta(y, y_1) = P(k', y_1, k''),$$

и тогда вторая строка в (9.58) приобретает вид

$$\bigoplus_{k \in K} P(k, y_1, k'') \otimes \left(\bigoplus_{\bar{y} \in X^*} f(k', \bar{y}, k) \otimes d_{in}(\bar{y}, \bar{y}_1) \right).$$

По определению (9.56) сумма в круглых скобках равна

$$\left(\bigoplus_{\bar{y} \in X^*} f(k', \bar{y}, k) \otimes d_{in}(\bar{y}, \bar{y}_1) \right) = f_1(k', \bar{y}_1, k),$$

и вторую строку в (9.58) можно записать как

$$\bigoplus_{k \in K} f_1(k', \bar{y}_1, k) \otimes P(k, y_1, k''). \quad (9.61)$$

7. Подставив (9.60) вместо первой строки в (9.58) и подставив (9.61) вместо второй строки, получим

$$f_1(k', \bar{y}_1 y_1, k'') = \bigoplus_{k \in K} f_1(k', \bar{y}_1, k) \otimes \left(\delta(k, k'') \otimes d_{in}(\#, y_1) \oplus P(k, y_1, k'') \right).$$

Таким образом, мы доказали, что

$$f_1[k', \bar{y}_1 y_1, k''] = f_1[k', \bar{y}_1, k] \bigotimes_{k \in K} P_1[k, y_1, k''],$$

где

$$P_1[k, y_1, k''] = P[k, y_1, k''] \oplus (\delta[k, k''] \otimes d_{in}(\#, y_1)),$$

и следовательно, доказали, что функция f_1 , определенная выражением (9.56), удовлетворяет условиям (9.41). ■

Отметим ради полноты, что $d_{in}(\#, y_1)$ равно $in[y_1]$, и таким образом, искомая функция P_1 определяется в явном виде

$$P_1[k', y, k''] = P[k', y, k''] \oplus (\delta[k', k''] \otimes in[y])$$

на основании известных функций P и in .

Доказательство. (леммы 9.5)

1. Так же, как и при доказательстве леммы 9.4, подставим выражение (9.42) для функции $F_1[\bar{y}_1]$ в (9.44) и получим выражение для $F_2[\bar{y}_2]$

$$F_2[\bar{y}_2] = \varphi[k'] \otimes_{k'} f_1[k', \bar{y}_1, k''] \otimes_{k''} \psi[k''] \otimes_{\bar{y}_1} d_{ch}[\bar{y}_1, \bar{y}_2].$$

В силу свойства (9.21) можно записать

$$F_2[\bar{y}_2] = \varphi[k'] \otimes_{k'} \left(f_1[k', \bar{y}_1, k''] \otimes_{\bar{y}_1} d_{ch}[\bar{y}_1, \bar{y}_2] \right) \otimes_{k''} \psi[k''].$$

Это выражение доказывает, что функцию F_2 можно представить в требуемой форме (9.45). Функция $f_2[k', \bar{y}_2, k'']$ есть свертка

$$f_2[k', \bar{y}_2, k''] = f_1[k', \bar{y}_1, k''] \otimes_{\bar{y}_1} d_{ch}[\bar{y}_1, \bar{y}_2]. \quad (9.62)$$

Докажем сейчас, что функция $f_2[k', \bar{y}_2, k'']$, определенная таким образом, удовлетворяет условиям (9.46) и (9.47).

2. Докажем свойство (9.46). При $\bar{y}_2 = \#$ свертка (9.62) принимает вид

$$f_2[k', \#, k''] = f_1[k', \#, k''] \otimes d_{ch}[\#, \#],$$

потому что непустую последовательность \bar{y}_1 невозможно превратить в пустую, применяя только замены символов. Число $f_1(k', \#, k'')$ есть $\delta(k', k'')$ по условию леммы. Число $d_{ch}(\#, \#)$ есть $0 = 1^{\otimes}$. Следовательно, $f_2(k', \#, k'')$ равно $\delta(k', k'')$, и функция f_2 удовлетворяет условию (9.46).

3. Пусть последовательность \bar{y}_2 не пустая и имеет вид $\bar{y}_2 y_2$. Запишем детально число $f_2(k', \bar{y}_2 y_2, k'')$, определенное выражением (9.62),

$$f_2(k', \bar{y}_2 y_2, k'') = \bigoplus_{\bar{y}_1 \in X^*} f_1(k', \bar{y}_1, k'') \otimes d_{ch}(\bar{y}_1, \bar{y}_2 y_2). \quad (9.63)$$

В предыдущем выражении должны приниматься во внимание только последовательности \bar{y}_1 той же длины, что и последовательность $\bar{y}_2 y_2$, так как замены символов не меняют длину последовательностей. Для последовательностей $\bar{y}_1 y_1$ и $\bar{y}_2 y_2$ одинаковой длины выполняется

$$d_{ch}(\bar{y}_1 y_1, \bar{y}_2 y_2) = d_{ch}(\bar{y}_1, \bar{y}_2) \otimes d_{ch}(y_1, y_2). \quad (9.64)$$

Для последовательностей вида $\bar{y}_1 y_1$ по условию (9.43) выполняется

$$f_1(k', \bar{y}_1 y_1, k'') = \bigoplus_{k \in K} f_1(k', \bar{y}_1, k) \otimes P_1(k, y_1, k''). \quad (9.65)$$

Подставим (9.64) и (9.65) в (9.63) и получим

$$f_2(k', \bar{y}_2 y_2, k'') = \bigoplus_{\bar{y}_1 \in X^*} \bigoplus_{y_1 \in X} \left(\bigoplus_{k \in K} f_1(k', \bar{y}_1, k) \otimes P_1(k, y_1, k'') \right) \otimes d_{ch}(\bar{y}_1, \bar{y}_2) \otimes d_{ch}(y_1, y_2).$$

Изменив в предыдущей сумме порядок суммирования, получим

$$\bigoplus_{k \in K} \left(\bigoplus_{\bar{y}_1 \in X^*} f_1(k', \bar{y}_1, k) \otimes d_{ch}(\bar{y}_1, \bar{y}_2) \right) \otimes \left(\bigoplus_{y_1 \in X} P_1(k, y_1, k'') \otimes d_{ch}(y_1, y_2) \right).$$

По определению (9.62) сумма $\bigoplus_{\bar{y}_1 \in X^*} f_1(k', \bar{y}_1, k) \otimes d_{ch}(\bar{y}_1, \bar{y}_2)$ равна $f_2(k', \bar{y}_2, k)$. Введем обозначение

$$P_2(k, y_2, k'') = \bigoplus_{y_1 \in X} P_1(k, y_1, k'') \otimes d_{ch}(y_1, y_2) \quad (9.66)$$

и представим $f_2(k', \bar{y}_2 y_2, k'')$ в виде

$$f_2(k', \bar{y}_2 y_2, k'') = \bigoplus_{k \in K} f_2(k', \bar{y}_2, k) \otimes P_2(k, y_2, k'').$$

Это значит, что функция f_2 удовлетворяет условию (9.47). Функция P_2 определяется выражением (9.66). ■

Ради полноты рассмотрения выразим функцию P_2 непосредственно через известные функции. Число $d_{ch}(y_1, y_2)$ есть стоимость самой дешевой последовательности замен, которая превращает символ y_1 в символ y_2 . Это число можно вычислить как длину кратчайшего пути между двумя вершинами графа, состоящего из $|X|$, соответствующих символам $x \in X$. Длина ребра в этом графе от вершины y_1 до вершины y_2 равна $ch(y_1, y_2)$. Длина $d_{ch} : X \times X \rightarrow \mathbb{R}$ кратчайшего пути определяется как сумма $\bigoplus_{i=0}^{\infty} ch^i$. Ранее мы обозначили эту сумму как ch^* доказали, что она равна $(\delta \oplus ch)^{|X|-1}$.

Следовательно, функция P_2 , которая раньше определялась выражением (9.66), приобретает вид

$$P_2[k, y_2, k''] = P_1[k, y_1, k''] \bigotimes_{y_1} ch^*[y_1, y_2],$$

который представляет в явном виде, как следует вычислять функцию P_2 на основании известных функций P_1 и ch .

Доказательство. (леммы 9.6) Подвергнем анализу функцию $D[\bar{x}]$, определяемую выражением (9.50). Пусть \bar{x} - последовательность x_1, x_2, \dots, x_n . Значение функции D для этой последовательности равно

$$D(\bar{x}) = F_2[\bar{y}_2] \bigotimes_{\bar{y}_2} d_{de}[\bar{y}_2, \bar{x}], \quad (9.67)$$

где $d_{de}[\bar{y}_2, \bar{x}]$ понимается теперь как функция одной переменной \bar{y}_2 , так как в дальнейших рассуждениях последовательность \bar{x} будет фиксирована. В приведенную сумму должны входить не все слагаемые \bar{y}_2 , а только те, которые имеют вид

$$\bar{y}_2 = \bar{x}_1 x_1 \bar{x}_2 x_2 \bar{x}_3 \cdots \bar{x}_{n-1} x_{n-1} \bar{x}_n x_n \bar{x}_{n+1}, \quad (9.68)$$

где $\bar{x}_1, \bar{x}_2, \dots, \bar{x}_{n+1}$ - это последовательности. Говоря иными словами, в сумме (9.67) должны учитываться только те последовательности \bar{y}_2 , которые можно превратить в \bar{x} исключением некоторых символов, то есть только те последовательности, которые включают в себя \bar{x} как подпоследовательность. В соответствии с условием (9.49), для последовательности вида (9.68) функция $F_2[\bar{y}_2]$ равна

$$\begin{aligned} F_2[\bar{y}_2] &= \varphi[k_0] \bigotimes_{k_0} f_2[k_0, \bar{x}_1, k'_0] \bigotimes_{k'_0} P_2[k'_0, x_1, k_1] \bigotimes_{k_1} f_2[k_1, \bar{x}_2, k'_1] \\ &\quad \bigotimes_{k'_1} P_2[k'_1, x_2, k_2] \bigotimes_{k_2} \cdots \bigotimes_{k_{n-1}} f_2[k_{n-1}, \bar{x}_n, k'_{n-1}] \\ &\quad \bigotimes_{k'_{n-1}} P_2[k'_{n-1}, x_n, k_n] \bigotimes_{k_n} f_2[k_n, \bar{x}_{n+1}, k'_n] \bigotimes_{k'_n} \psi[k'_n]. \end{aligned} \quad (9.69)$$

Число $d_{de}[\bar{y}_2, \bar{x}]$, следовательно, равно

$$d_{de}[\bar{x}_1, \#] \otimes d_{de}[\bar{x}_2, \#] \otimes \cdots \otimes d_{de}[\bar{x}_n, \#] \otimes d_{de}[\bar{x}_{n+1}, \#]. \quad (9.70)$$

Подставим (9.70) в (9.67) и, учитывая, что свертка выполняется только последовательностям вида (9.68), получим

$$\begin{aligned} F_2[\bar{y}_2] \bigotimes_{\bar{y}_2} d_{de}[\bar{y}_2, \bar{x}] &= F_2[\bar{x}_1 x_1 \bar{x}_2 x_2 \bar{x}_3 \cdots \bar{x}_n x_n \bar{x}_{n+1}] \bigotimes_{\bar{x}_1} d_{de}(\bar{x}_1, \#) \\ &\quad \bigotimes_{\bar{x}_2} d_{de}(\bar{x}_2, \#) \bigotimes_{\bar{x}_3} \cdots \bigotimes_{\bar{x}_{n+1}} d_{de}(\bar{x}_{n+1}, \#). \end{aligned}$$

Если в это выражение вместо $F_2[\bar{y}_2]$ подставить подробное его представление (9.69), получим, что

$$\begin{aligned} D[\bar{x}] &= \varphi[k_0] \bigotimes_{k_0} P'[k_0, x_1, k_1] \bigotimes_{k_1} P'[k_1, x_2, k_2] \bigotimes_{k_2} \cdots \\ &\quad \bigotimes_{k_{n-1}} P'[k_{n-1}, x_n, k_n] \bigotimes_{k_n} \psi'[k_n], \end{aligned}$$

где

$$P'[k', x, k''] = f_2[k', \bar{x}, k] \bigotimes_{\bar{x}} d_{de}[\bar{x}, \#] \bigotimes_k P_2[k, x, k''], \quad (9.71)$$

$$\psi'[k] = f_2[k', \bar{x}, k''] \bigotimes_{\bar{x}} d_{de}[\bar{x}, \#] \bigotimes_{k''} \psi[k'']. \quad (9.72)$$

Этим доказана лемма 9.6. \blacksquare

Выражения (9.71) и (9.72), однако, не указывают способ конструктивного построения функций P' и ψ' , так как они включают свертку $f_2[k', \bar{x}, k''] \bigotimes_{\bar{x}} d_{de}[\bar{x}, \#]$ по бесконечному множеству всех возможных последовательностей \bar{x} . Способ их конструктивного построения указывает лемма 9.3, которая утверждает, что эта бесконечная свертка равна

$$\left(\delta[k', k''] \oplus \left(P_2[k', x, k''] \bigotimes_x de[x] \right) \right)^{|K|-1}.$$

Таким образом, функции P' и ψ' строятся с помощью явных выражений

$$P'[k', x, k''] = \left(\delta[k', k] \oplus \left(P_2[k', x, k] \bigotimes_x de[x] \right) \right)^{|K|-1} \bigotimes_k P_2[k, x, k''],$$

$$\psi'[k] = \left(\delta[k, k'] \oplus \left(P_2[k, x, k'] \bigotimes_x de[x] \right) \right)^{|K|-1} \bigotimes_{k'} \psi[k']$$

на основании известных функций P_2 , ψ и de .

Доказанные три леммы 9.4, 9.5 и 9.6 доказывают теорему 9.2, а следовательно, и эквивалентную ей 9.1, так как указывают, как следует строить функции P' и ψ' , существование которых требуется доказать в теоремах. Функции P' и ψ' следует строить на основе пяти функций P, ψ, in, ch, de по следующим правилам.

Algorithm 9.1 Конструктивное построение функций P' и ψ' .

1. Построить функцию $P_1: K \times X \times K \rightarrow \mathbb{R}$,

$$P_1[k', y, k''] = P[k', y, k''] \oplus (\delta[k', k''] \otimes in[y]). \quad (9.73)$$

Сложность этого шага имеет порядок $\mathcal{O}(|K| |X| |K|)$.

2. Функция $ch^*: X \times X \rightarrow \mathbb{R}$ вычисляется, например, следующим способом. Сначала принимается, что функция ch^* равна $\delta \oplus ch$. Затем несколько раз выполняется следующий оператор, но не более, чем $\log |X|$ раз,

$$ch^*[x, y] ::= ch^*[x, z] \bigotimes_z ch^*[z, y]. \quad (9.74)$$

Этот оператор выполняется до тех пор, пока функция ch^* не перестанет меняться. Сложность одного выполнения оператора (9.74) имеет порядок $\mathcal{O}(|X|^3)$, а сложность построения функции ch^* - не больше, чем $\mathcal{O}(|X|^3 \log |X|)$.

3. Построить функцию $P_2: K \times X \times K \rightarrow \mathbb{R}$,

$$P_2[k', x, k''] = P_1[k', y, k''] \otimes_y ch^*[y, x]. \quad (9.75)$$

Сложность этого построения имеет порядок $\mathcal{O}(|K|^2 |X|^2)$.

4. Построить вспомогательную функцию $q: K \times K \rightarrow \mathbb{R}$,

$$q[k', k''] = \delta[k', k''] \oplus (P_2[k', x, k''] \otimes_x de[x]). \quad (9.76)$$

Сложность этого построения имеет порядок $\mathcal{O}(|K|^2 |X|)$.

5. Построить вспомогательную функцию $q^*: K \times K \rightarrow \mathbb{R}$, например, следующим образом. Сначала принимается, что функция q^* совпадает с q . Затем многократно, но не более, чем $\log |K|$ раз выполняется оператор

$$q^*[k', k''] ::= q^*[k', k] \otimes_k q^*[k, k''], \quad (9.77)$$

пока q^* не перестанет меняться. Вычислительная сложность этого этапа имеет порядок $\mathcal{O}(|K|^3 \log |K|)$.

6. Построить функцию $P': K \times X \times K \rightarrow \mathbb{R}$,

$$P'[k', x, k''] = q^*[k', k] \otimes_k P_2[k, x, k'']. \quad (9.78)$$

Сложность этого вычисления имеет порядок $\mathcal{O}(|K|^3 |X|)$.

7. Построить функцию $\psi': K \rightarrow \mathbb{R}$,

$$\psi'[k] = q^*[k, k'] \otimes_{k'} \psi[k']. \quad (9.79)$$

Сложность этого вычисления имеет порядок $\mathcal{O}(|K|^2)$.

Указанные вычисления не зависят от последовательности \bar{x} , для которой должно быть вычислено отличие от заданного регулярного языка. Поэтому, если речь идет о вычислении отличия многих последовательностей от одного и того же языка и при одной той же левенштейновой функции, указанные вычисления можно выполнить один раз, а затем использовать одну и ту же пару функций P' и ψ' для анализа всех последовательностей, предъявленных для анализа.

Если функции P' и ψ' уже построены, левенштейново отличие $D(\bar{x})$ последовательности $\bar{x} = (x_1, x_2, \dots, x_n)$ от заданного языка вычисляется как свертка

$$D(\bar{x}) = \varphi[k_0] \otimes_{k_0} P'[k_0, x_1, k_1] \otimes_{k_1} \cdots \otimes_{k_{n-1}} P'[k_{n-1}, x_n, k_n] \otimes_{k_n} \psi'[k_n].$$

Одна из возможных процедур для ее вычисления состоит в построении последовательности функций f_0, f_1, \dots, f_n по формулам

$$\left. \begin{aligned} f_0[k_0] &= \varphi[k_0], \\ f_i[k_i] &= f_{i-1}[k_{i-1}] \otimes_{k_{i-1}} P'[k_{i-1}, x_i, k_i], \quad i = 1, 2, \dots, n, \\ D(\bar{x}) &= f_n[k_n] \otimes_{k_n} \psi'[k_n]. \end{aligned} \right\} \quad (9.80)$$

Отметим, что это далеко не единственная возможная процедура.

9.5.10 Интерпретация основного результата

С самого начала мы сформулировали задачу левенштейновой аппроксимации, не прибегая к понятию сверток. Затем мы перевели исходную формулировку на язык сверток и на этом языке вывели алгоритм ее решения, сформулированный также на языке сверток. Теперь осталось сделать последний шаг и перевести полученное решение в язык тех операций, на котором была дана исходная формулировка задачи. Мы повторим исходную формулировку задачи, а затем укажем процедуру ее решения, не прибегая к понятию свертки. Естественно, мы уже не будем обосновывать эту процедуру, так как это было сделано на языке сверток. Мы ограничимся только словесным объяснением тех вычислений, в которых состоит решение задачи.

Исходная формулировка задачи состоит в следующем. Пусть даны два конечных множества X и K и три функции $\varphi: K \rightarrow \{0, \infty\}$, $P: K \times X \times K \rightarrow \{0, \infty\}$, $\psi: K \rightarrow \{0, \infty\}$, которые определяют множество $L \in X^*$ последовательностей, для которых существует такая последовательность k_0, k_1, \dots, k_n , что

$$\begin{aligned} \varphi(k_0) &= 0, \\ P(k_{i-1}, x_i, k_i) &= 0, \quad i = 1, \dots, n, \\ \psi(k_n) &= 0. \end{aligned}$$

Пусть даны также три функции $in: X \rightarrow \mathbb{R}$, $ch: X \times X \rightarrow \mathbb{R}$, $de: X \rightarrow \mathbb{R}$, которые определяют функцию $d: X^* \times X^* \rightarrow \mathbb{R}$, значения которой $d(\bar{y}, \bar{x})$ есть левенштейново отличие последовательности \bar{x} от последовательности \bar{y} . Задача состоит в построении алгоритма, который для любой поданной на его вход последовательности $\bar{x} \in X^*$ вычисляет число

$$D(\bar{x}) = \min_{\bar{y} \in L} d(\bar{y}, \bar{x}), \quad (9.81)$$

Алгоритм вычисления числа $D(\bar{x})$ состоит из двух частей. Первая часть - подготовительная и имеет сложность, которая не превосходит $(\mathcal{O}(|K| \log |K|), \mathcal{O}(|X^*| \log |X|), \mathcal{O}(|K|^3 |X|), \mathcal{O}(|K|^2 |X|))$. Вычисления в этой части не зависят от входной последовательности \bar{x} и для данного множества L и данной левенштейновой функции выполняются только один

раз. Вторая часть вычислений зависит от входной последовательности и сложность ее имеет порядок $\mathcal{O}(|K|^2 n)$, где n - длина \bar{x} .

Вычисления по первой части представлены ранее в виде сверточных выражений (9.73) - (9.79). Переведем их на обычный язык и снабдим их словесными комментариями.

1. Построить функцию $P_1: K \times X \times K \rightarrow \mathbb{R}$ (см. формулу (9.73)),

$$P_1(k', y, k'') = \begin{cases} \min(P(k', y, k''), in(y)), & \text{если } k' = k'' , \\ P(k', y, k'') , & \text{если } k' \neq k'' . \end{cases}$$

Числа $P_1(k', y, k'')$ обозначают стоимость самого дешевого добавления символа y в конец строки при условии, что до этого добавления автомат находился в состоянии k' , а после этого будет находиться в состоянии k'' . При $k' \neq k''$ символ y может быть дописан к строке только так, что этот символ сгенерирует автомат. Стоимость дописывания в этом случае будет равна $P(k', y, k'')$. При $k' = k'' = k$ имеются две возможности дописывания символа y в конец строки. Первая возможность состоит в том, что символ y генерируется автоматом. Стоимость в этом случае равна $P(k, y, k)$. Вторая возможность состоит в том, что работа автомата прерывается и символ y появляется в конце строки в результате редакторской операции вставки со стоимостью $in(y)$. Из этих двух возможностей выбирается более дешевая.

2. Построить функцию $ch^*: X \times X \rightarrow \mathbb{R}$, например, следующим образом (см. формулу (9.74)). Сначала числам $ch^*(x, y)$ присваиваются значения $ch^*(x, y) = 0$ при $x = y$ и $ch^*(x, y) = ch(x, y)$ при $x \neq y$. Затем эти числа многократно преобразуются оператором

$$ch^*(x, y) ::= \min_{z \in X} (ch^*(x, z) + ch^*(z, y)) .$$

Число $ch^*(x, y)$ есть стоимость самой дешевой последовательности замен, возможно, и пустой, превращающей символ x в символ y . Стоимость пустой последовательности замен есть 0 при $x = y$ и ∞ при $x \neq y$. Последовательность, состоящая из одной замены, имеет стоимость $ch(x, y)$. Самая дешевая последовательность двух замен имеет стоимость

$$\min_{x'} (ch(x, x') + ch(x', y)) ,$$

а самая дешевая последовательность из трех и более замен имеет стоимость

$$\min_{x_1, x_2, \dots, x_n} \left(ch(x, x_1) + \sum_{i=2}^n ch(x_{i-1}, x_i) + ch(x_n, y) \right)$$

3. Построить функцию $P_2: K \times X \times K \rightarrow \mathbb{R}$ (см. формулу (9.75)),

$$P_2(k', x, k'') = \min_{y \in X} (P_1(k', y, k'') + ch^*(y, x)) .$$

Число $P_2(k', x, k'')$ - это стоимость самого дешевого способа дописывания символа x в конец последовательности при условии, что до этого дописывания автомат находился в состоянии k' , а после дописывания - в состоянии k'' . Это число подобно числу $P_1(k', x, k'')$ и в то же время существенно от него отличается. Добавленный символ мог быть сгенерирован автоматом или вставлен редактором. После этого с этим символом выполняется (или не выполняется) сколь угодно длинная последовательность замен, результатом которой является символ x . Как видим, число $P_2(k', x, k'')$ есть результат оптимального выбора из довольно обширного множества.

4. Построить вспомогательную функцию $q: K \times K \rightarrow \mathbb{R}$,

$$q(k', k'') = \begin{cases} 0, & \text{если } k' = k'', \\ \min_{x \in K} (P_2(k', x, k'') + de(x)), & \text{если } k' \neq k''. \end{cases}$$

Число $q(k', k'')$ есть стоимость самого дешевого процесса, в результате которого автомат переходит из состояния k' в состояние k'' , причем такого, что после окончания процесса последовательность символов оказывается такой же, как до его начала, хотя в течение самого процесса к этой последовательности и дописывался какой-то символ. Этот процесс состоит из следующих частей:

- Добавление символа в конец последовательности, сгенерировав его автоматом или выполнив редакторскую вставку;
 - Последовательность (возможно, пустая) замен добавленного символа;
 - Исключение символа из последовательности.
5. Построить вспомогательную функцию $q^*: K \times K \rightarrow \mathbb{R}$ (см. формулу (9.77)). Сначала выполнить присвоение

$$q^*(k', k'') = \begin{cases} 0, & \text{если } k' = k'', \\ q(k', k''), & \text{если } k' \neq k''. \end{cases}$$

Затем многократно выполнить оператор

$$q^*(k', k'') ::= \min_{k \in K} (q^*(k', k) + q^*(k, k'')).$$

Число $q^*(k', k'')$ подобно числу $q(k', k'')$. Различие между этими числами в том, что число $q(k', k'')$ есть стоимость самого дешевого способа перехода из состояния, при котором разрешено один и только один раз дописать символ в последовательность, выполнить с ним ряд замен и исключить. Число $q^*(k', k'')$ есть стоимость самого дешевого способа перехода из состояния k' в состояние k'' , при котором в последовательность дописывается какое-угодно количество символов, включая нулевое, выполняются с ними какие-угодно замены (или не выполняются их) и в конечном итоге все эти символы исключаются.

6. Построить функцию $P' : K \times X \times K \rightarrow \mathbb{R}$ (см. формулу (9.78)),

$$P'(k', x, k'') = \min_{k \in K} (q^*(k', k) + P_2(k, x, k'')) .$$

Число $P'(k', x, k'')$ есть стоимость самого дешевого процесса, в результате которого автомат переходит из состояния k' в состояние k'' , а последовательность символов наращивается символом x . Во время этого процесса автомат генерирует какую-то последовательность символов, с каждым из них выполняются какие-то замены, и в конечном итоге все они исключаются. Затем в последовательность дописывается один символ, либо позволяя автомату сгенерировать его, либо вставляя его редактором, с этим символом выполняется ряд замен, и в конечном итоге он остается в последовательности.

7. Вычислить числа $\psi'(k)$ (см. формулу (9.79)),

$$\psi'(k) = \min_{k' \in K} (q^*(k, k') + \psi(k')) .$$

Число $\psi'(k)$ есть стоимость самого дешевого процесса следующего класса. Автомат, находящийся в состоянии k , генерирует какую-то последовательность символов и оказывается в состоянии k' , в котором он прекращает работу. Затем каждый сгенерированный символ произвольное количество раз заменяется другим символом и в конечном итоге исключается.

Вычислительная процедура (9.80) имеет следующий вид, если ее перевести с языка сверток на обычный язык. Для поданной на вход алгоритма последовательности $\bar{x} = (x_0, x_1, \dots, x_n)$ вычисляются числа $f_i(k)$, $k \in K$, $i = 0, 1, \dots, n$, и число $D(\bar{x})$ по формулам

$$\begin{aligned} f_0(k_0) &= \varphi(k_0), \quad k_0 \in K, \\ f_i(k_i) &= \min_{k_{i-1} \in K} (f_{i-1}(k_{i-1}) + P'(k_{i-1}, x_i, k_i)), \quad k_i \in K, \quad i = 1, 2, \dots, n, \\ D(\bar{x}) &= \min_{k_n \in K} (f_n(k_n) + \psi'(k_n)) . \end{aligned}$$

Мы видим, что даже если пытаться всего лишь объяснить уже обоснованного алгоритм, но при этом стремиться обойтись лишь естественными предложениями русского языка, то это объяснение становится слишком громоздким и ненаглядным. Более того, мы не уверены, что приведенное нами объяснение является однозначно понимаемым. Что касается убедительно обоснованного вывода, а не объяснения уже обоснованного алгоритма, то нам представляется почти безнадежным сделать это с помощью одних лишь так называемых разумных соображений, выраженных предложениями естественного языка.

9.6 Обсуждение

Я заметил существенное различие между лекцией 8 и лекцией 9, хотя в обеих лекциях идет речь о распознавании последовательностей. В предыдущей лекции 8 активно используются результаты общей статистической

теории распознавания. Данная лекция совсем иная. Проблема структурного распознавания внезапно рассматривается совсем с другой стороны и как бы с самого начала. Действительно, на протяжении всей лекции слово "вероятность" не упоминалось ни разу и поэтому эта лекция как бы оторвана от предыдущих лекций. Фактически именно этой лекцией можно было бы начинать рассказ о структурном распознавании. Возможно, я опять упустил что-то важное, но я сейчас не вижу, как связана данная лекция с содержанием предыдущих.

Вряд ли Ты упустил что-то важное. Твое недоумение происходит оттого, что Ты ожидаешь от сегодняшнего распознавания несколько больше, чем оно может дать. Распознавание образов еще не представляет собой хорошо структурированную дисциплину с ясной иерархией хорошо подогнанных друг к другу задач. Да Ты сам уже говорил нечто подобное. Распознавание образов, к сожалению, представляет собой нечто иное, чем, скажем, линейное программирование, рамки которого четко очерчены одной общей задачей, выраженной к тому же в лаконичной и прозрачной форме. В распознавании образов существует пока что много изолированных друг от друга островков. Разрозненность методов имеется даже в тех сравнительно узких разделах, которые по определенным неформальным признакам объединены под одним названием, как например, структурное распознавание. На некоторые из этих методов удается посмотреть с единой точки зрения и таким образом их сблизить или заполнить новыми методами те места, где раньше зияла пустота. Однако есть много подходов и направлений, которые развиваются независимо друг от друга и не обогащая друг друга.

Две предыдущие лекции покоятся на двух различных направлениях, развитых в общей теории распознавания. Лекция 8 основана на статистической теории распознавания, как мы ее изложили в первых шести лекциях. Нам очень приятно, что Ты видишь эту непосредственную связь. Лекция 9 основана на так называемом *методе ближайшего соседа*, которого мы в данном курсе не касались вообще. Естественно, что Ты этой связи не видишь. Мы не видим достаточно тесной связи между статистической теорией распознавания и методом ближайшего соседа, хотя и допускаем, что такая связь есть. Исследователи этого метода, скорее всего, видят эту связь, и мы сейчас не намерены устанавливать или отрицать ее. Мы лишь утверждаем, что материал данной лекции является одной из возможных реализаций метода ближайшего соседа.

Метод ближайшего соседа основан не на задании той или иной статистической модели объекта, а на совершенно других понятиях. Первым из них является определенное подмножество L в множестве X всех возможных наблюдений. Подмножество L можно понимать, как множество некоторых идеальных, неискаженных наблюдений. Существенным неформальным свойством этих идеальных наблюдений является то, что характеристики распознаваемого объекта, которые следует определить при распознавании, легко вычисляются, если результат наблюдения принадлежит L , и

очень трудно вычисляется в противном случае. В распознавании образов наиболее часто предполагается, что множество L разбито на конечное количество классов так, что для каждого наблюдения из L легко определяется, к какому классу он принадлежит.

Вторым важным понятием является функция $d: X \times X \rightarrow \mathbb{R}$. Значения $d(x, y)$ этой функции формализуют содержательное представление о том, насколько наблюдения x и y сходны друг с другом.

На основе этих двух понятий определяется, то есть постулируется процедура распознавания. Она состоит в том, что для любого наблюдения $x \in X$ указывается идеальное наблюдение $y^* \in L$, наиболее сходное с x . Затем характеристики найденного идеального наблюдения y^* приписываются и наблюдению x . В частности, принимается, что x принадлежит тому же классу, что и y^* . На наш взгляд, в настоящее время еще далеко не достигнута ясность о взаимосвязи этого подхода со статистическим подходом.

Лекция 9 указывает, как следует реализовать метод ближайшего соседа в случае, когда L есть регулярный язык, а $d: X \times X \rightarrow \mathbb{R}$ есть функция Левенштейна. Таким образом, лекции 8 и 9 не связаны друг с другом в том смысле, что они опираются на два различных подхода в распознавании образов, взаимосвязь между которыми нам сейчас не совсем ясна. Однако, если лекции отличаются друг от друга по формулировке решаемых задач, то этого нельзя сказать об алгоритмах решения этих задач. Ведь это фактически одни и те же вычислительные процедуры. Именно алгоритмы образуют сейчас перешеек между содержанием этих двух лекций. И, возможно, именно сходство алгоритмов решения задач послужит ключом к пониманию, чем же родственны сами задачи.

Да, эти алгоритмы очень похожи. Но их объединяет не только то, что является их достоинством. Я бы хотел обсудить одну общую их черту, и не очень приятную.

Допустим, что на основании последовательности $x_1, x_2, \dots, x_i, \dots, x_n$ наблюдений мне нужно определить последовательность k_0, k_1, \dots, k_n состояний. Все описанные в лекциях алгоритмы сходны в том, что решение о всей последовательности k_0, k_1, \dots, k_n целиком принимается на основании всей последовательности x_1, \dots, x_n . Даже если я понимаю правильность такой процедуры, мне представляется неестественным, что решение о первом элементе k_0 в искомой последовательности я смогу принять только после того, как станет известен последний элемент x_n в последовательности наблюдений. Дело тут не только в неестественности. Неприятные вопросы здесь возникают при первом же столкновении с реальным применением рекомендуемых алгоритмов. Когда я (или алгоритм) читаю книгу, то рано или поздно я должен решить, какой была первая буква в этой книге. Алгоритмы предписывают, что это решение следует принимать на основании всей последовательности наблюдений. Но как определить, где же заканчивается последовательность наблюдений. В конце слова? Или в конце строки? Или в конце страницы?

... А может, в конце книги? Не можем удержаться от того, чтобы не довести дело до абсурда. Ведь если учесть, что содержания различных книг не могут считаться независимыми друг от друга, то выходит, что следует сначала прочесть все книги, и только после этого можно решать, какой же была первая буква в первой прочитанной книге. А ведь что-то в этом есть. Ведь действительно, только прочитав много книг, казалось бы, не связанных друг с другом, можно правильно понять, что же было написано в первой прочитанной книге. Об этом можно было бы писать стихами!

И формальное содержание всех этих стихов было бы следующим. Пусть x_1, x_2, \dots, x_n - последовательность наблюдений, а $k_0^, k_1^*, \dots, k_n^*$ - последовательность состояний, определенная как*

$$(k_i^* \mid i = 0, \dots, n) = \operatorname{argmax}_{k_0} \max_{k_1} \dots \max_{k_n} \left(\sum_{i=1}^n f_i(k_{i-1}, x_i, k_i) \right). \quad (9.82)$$

Необходимо указать, при каких условиях можно однозначно определить первый элемент k_0^ последовательности $k_0^*, k_1^*, \dots, k_n^*$, если известна только начальная часть $x_1, x_2, \dots, x_l, l < n$, последовательности x_1, x_2, \dots, x_n .*

Этот вопрос поставил перед собой В.А.Ковалевский сразу же после разработки первого известного алгоритма структурного анализа последовательностей [Kovalevski, 1967], и он же на него ответил. Покажем смысл этого ответа с помощью графа на рис. 9.5, который мы уже рассматривали (см. рис. 8.2). Как и прежде, граф образован вершинами α и β и группой из $|K|n$ вершин. Каждая вершина из этой группы представлена точкой (k, i) , $k \in K, i = 0, \dots, n$, с горизонтальной координатой i и вертикальной координатой k . Граф представляет ситуацию, когда множество K состоит из трех состояний A, B, C .

Твой вопрос можно сейчас задать таким образом. Для каждого ребра графа определена его длина и, следовательно, определен и кратчайший путь на графе от вершины α до вершины β . Однако эти длины известны не все, а только для ребер, находящихся левее координаты l , где $l < n$. Поэтому и кратчайший путь от α до β неизвестен. Однако Тебя он и не интересует. Тебя интересует только, через какую вершину в самом левом столбце этот кратчайший путь проходит. Эти вершины обозначены на рис. 9.5 белыми кружочками.

Ты можешь ответить на этот вопрос на основании следующих соображений. Хоть Ты и не знаешь весь кратчайший путь от α до β , Ты определенно знаешь, что он проходит через одну из вершин с горизонтальной координатой l . На рис. 9.5 эти вершины (A, l) , (B, l) и (C, l) обозначены черными кружочками. Ты можешь найти кратчайший путь от вершины α до каждого из черных кружочков, то есть три пути: от вершины α до (A, l) , до вершины (B, l) и до вершины (C, l) . Чтобы найти эти три пути, достаточно знать длины тех ребер, которые Тебе известны. Повидимому, Тебе ясно, что каков бы ни был кратчайший путь от α до β , то есть, какими бы ни были неизвестные Тебе длины ребер, начальный участок этого кратчайшего

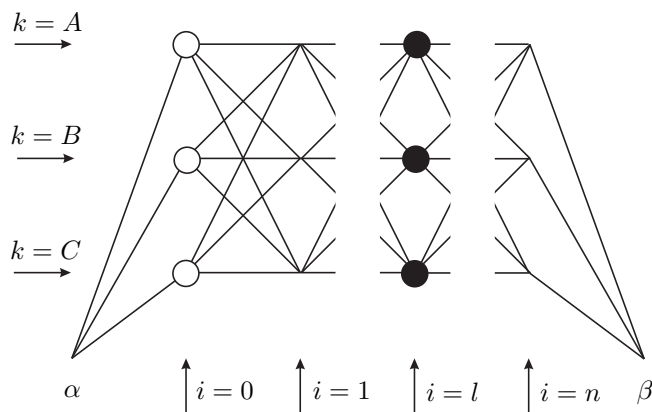


Figure 9.5 Structural analysis of a sequence with not all lengths of graph edges known.

пути обязательно совпадет с одним из трех построенных Тобой участков. Для каждого из найденных Тобой трех участков Ты можешь посмотреть, через какой белый кружочек на рис. 9.5 он проходит. Если все три участка проходят через один и тот же белый кружочек, то и кратчайший путь от α до β также проходит через этот кружочек.

А если это условие не выполняется?

Значит, информация о длинах ребер, которой Ты располагаешь, еще недостаточна для однозначного решения о скрытом состоянии k_0 , и Тебе следует продолжить наблюдения. Понятны ли Тебе эти неформальные объяснения?

Думаю, что да!

Тогда запиши, пожалуйста, точный ответ на свой вопрос.

Я введу обозначение

$$F_l(k_l) = \max_{k_0} \max_{k_1} \cdots \max_{k_{l-1}} \sum_{i=1}^l f_i(k_{i-1}, x_i, k_i) \quad (9.83)$$

и обозначу $k_{0l}(k_l)$ первый элемент в последовательности $k_0^*, k_1^*, \dots, k_{l-1}^*$, которая максимизирует (9.83). Числа $F_l(k_l)$ и состояния $k_{0l}(k_l)$ следует вы-

числять по следующим рекуррентным правилам,

$$\begin{aligned} F_1(k'') &= \max_{k_0} f_1(k_0, x_1, k''), \\ k_{01}(k'') &= \operatorname{argmax}_k f_1(k, x_1, k''), \\ F_l(k'') &= \max_{k \in K} \left(F_{l-1}(k) + f_l(k, x_{l-1}, k'') \right), \\ k_{0,l}(k'') &= k_{0,l-1} \left(\operatorname{argmax}_{k \in K} \left(F_{l-1}(k) + f_l(k, x_{l-1}, k'') \right) \right). \end{aligned}$$

Если для некоторого l все значения $k_{0l}(k'')$, $k'' \in K$, окажутся равными друг другу, скажем, равными k^* , то начальная часть x_1, x_2, \dots, x_l последовательности наблюдений достаточна для однозначного определения состояния k_0^* в последовательности $k_0^*, k_1^*, \dots, k_n^*$, которая максимизирует сумму

$$\sum_{i=1}^n f_i(k_{i-1}, x_i, k_i).$$

И этим состоянием является состояние k^* .

Представь себе сейчас, что Ты не застал момент начала наблюдений, которые начались в момент $-T$ в прошлом и будут продолжаться до момента T в будущем. В этом интервале наблюдаемый объект генерирует символы

$$x_{-T+1}, x_{-T+2}, \dots, x_{-1}, x_0, x_1, \dots, x_{T-1}, x_T, \quad (9.84)$$

на основании которых можно определить наиболее вероятную последовательность состояний объекта

$$k_{-T}^*, k_{-T+1}^*, \dots, k_{-1}^*, k_0^*, k_1^*, \dots, k_{T-1}^*, k_T^*. \quad (9.85)$$

Но Тебе известна только часть наблюдений из последовательности (9.84) от x_{-l+1} до x_l . Тебе предстоит решить, располагаешь ли Ты достаточной информацией для однозначного решения о состоянии k_0^* в последовательности (9.85), и при положительном ответе указать состояние k_0^* .

Введу в рассмотрение числа $F_{ij}(k_i, k_j)$, $-l \leq i < j \leq l$, $k_i \in K$, $k_j \in K$,

$$F_{ij}(k_i, k_j) = \max_{k_{i+1}} \max_{k_{i+2}} \dots \max_{k_{j-2}} \max_{k_{j-1}} \sum_{t=i+1}^j f_t(k_{t-1}, x_t, k_t). \quad (9.86)$$

Обозначу $k_{ij}(k_i, k_j)$ элемент k_0 в последовательности $k_{-l+1}, k_{-l+2}, \dots, k_{l-2}, k_{l-1}$, которая максимизирует (9.86). Ответ на поставленный мне вопрос следующий: если состояние $k_{-l,l}(k_{-l}, k_l)$ одно и то же для любой пары $k_{-l} \in K, k_l \in K$, и равно, скажем, k^* , то последовательность $x_{-l+1}, x_{-l+2}, \dots, x_{l-1}, x_l$ достаточна для однозначного определения элемента

k_0 в последовательности $k_{-T}, k_{-T+1}, \dots, k_{T-1}, k_T$, которая максимизирует сумму $\sum_{i=-T+1}^T f_i(k_{i-1}, x_i, k_i)$. И этот элемент является как раз k^* .

Следует ли мне выписать рекуррентные формулы для вычисления функций F_{ij} и k_{ij} ?

Пожалуй, нет. Для Тебя это, повидимому, настолько легкая задача, что на нее жаль Твоего времени.

Спасибо. Мне тоже хочется сойти с этой тропинки, тем более, что мне кажется, что с самого начала я пошел не по той тропинке, что нужно. Ведь о чем, собственно, идет речь? Некий автомат прогенерировал очень длинную последовательность сигналов и в это время прошел через последовательность состояний $k_0, k_1, k_2, \dots, k_n$. На основании последовательности x_1, x_2, \dots, x_n можно принять оптимальное в каком-то смысле решение о состоянии k_0 , в котором находился автомат до начала генерирования последовательности x_1, x_2, \dots, x_n . Теперь представим себе, что в моем распоряжении имеется лишь начальный участок x_1, x_2, \dots, x_l , $l < n$, последовательности, сгенерированной автоматом. Спрашивается, можно ли на основании этих имеющихся данных принять решение о состоянии k_0 с тем же качеством, как и при решении, принятом на основании всей последовательности x_1, x_2, \dots, x_n , сгенерированной автоматом.

Этот вопрос я слишком поспешно заменил вопросом о том, можно ли, располагая лишь частью x_1, x_2, \dots, x_l , $l < n$, наблюдений, указать начальное состояние k_0^* в последовательности $k_0^*, k_1^*, k_2^*, \dots, k_n^*$, максимизирующей апостериорную вероятность при известной всей последовательности x_1, x_2, \dots, x_n наблюдений. Так вот эта замена одного вопроса другим была слишком поспешной.

Если мы интересуемся только первым элементом последовательности состояний и хотим его определить с наименьшей вероятностью ошибки, то этим элементом совсем не обязательно является первый элемент в наиболее вероятной последовательности. Эта мысль неоднократно повторялась в ваших лекциях, и сейчас я с ней полностью согласен. Даже при известной последовательности x_1, x_2, \dots, x_n не следует принимать решение в пользу первого элемента в наиболее вероятной последовательности состояний. В этом случае следует вычислить $|K|$ чисел $p(k_0, x_1, x_2, \dots, x_n)$, $k_0 \in K$, и принять решение в пользу того значения k_0 , которому соответствует наибольшее из этих чисел. И теперь я интересуюсь, что мне следует делать, если я располагаю не всей последовательностью x_1, x_2, \dots, x_n , а только ее частью x_1, x_2, \dots, x_l .

Конечно, я могу вычислить $|K|$ чисел $p(k_0, x_1, x_2, \dots, x_l)$, $k_0 \in K$, и на основании этих чисел принять решение о k_0 . Хотя это и будет наилучшее использование имеющейся информации, качество такого решения будет, вообще говоря, хуже, чем качество решения, принятого на основе всей последовательности x_1, x_2, \dots, x_n . Но моя задача состоит не только в том, чтобы принять наилучшее решение по имеющимся данным x_1, x_2, \dots, x_l , но еще и в том, чтобы качество этого решения несколько не уступало решению,

которое бы я принял с учетом всех последующих данных x_{l+1}, x_{l+2}, \dots . Иными словами говоря, надо уметь отвечать на вопрос, можно ли пренебречь информацией о состоянии k_0 , которая содержится в наблюдениях $x_{l+1}, x_{l+2}, \dots, x_i, \dots$, или нет. При положительном ответе на него я бы мог прекратить наблюдения и принять решение о состоянии k_0 с гарантией, что это решение невозможно улучшить при сколь-угодно длительном дальнейшем наблюдении объекта. При отрицательном ответе на этот вопрос я должен был бы продолжать наблюдения.

На вопрос, поставленный таким образом, я надеялся найти ответ в рамках последовательного анализа Вальда. Пока что я там не нашел удовлетворительного ответа, правда, я еще не уделил этому достаточно много времени. Я начинаю подозревать, что вопрос либо чертовски трудный, либо ответ на него всегда отрицательный. И вот почему.

Обозначу вероятность $p(k_0|x_1, x_2, \dots, x_l)$ как $p_l(k_0)$ и сформулирую вопрос: существует ли такое число l , что при любом $i > l$ вероятности $p_i(k_0)$ будут равны вероятностям $p_l(k_0)$? Только при положительном ответе на этот вопрос я могу утверждать, что все наблюдения $x_i, i > l$, не несут дополнительную информацию о состоянии k_0 . Это известный факт из теории информации. Но на поставленный вопрос можно ответить положительно только в том тривиальном случае, когда наблюдения $x_i, i > l$, и состояние k_0 статистически независимы. Но в марковской модели, с которой мы сейчас имеем дело, все параметры взаимно зависимы. Это мне стало ясно еще при рассмотрении механической модели в лекции 8. Отсюда следует, что даже при очень большом i каждое наблюдение x_i сообщает определенную, возможно, очень малую, но ненулевую информацию о состоянии k_0 . Но ведь это ужасно! Выходит, что все-таки я не смогу принять решение о первой букве в книге, пока не прочитаю всю книгу. Тут впору кричать караул!

Парень, Ты прекрасно умеешь задавать вопросы, этого у Тебя не отнять. Как это бывало уже и прежде, ты находишься очень близко от интересного решения. Рассмотрим его вместе.

На основании наблюдений x_1, x_2, \dots, x_l можно вычислить не только $|K|$ чисел $p(k_0, x_1, x_2, \dots, x_l)$, $k_0 \in K$, но еще и $|K|$ групп чисел $(p(k_0, x_1, x_2, \dots, x_l, k_l), k_0 \in K)$, $k_l \in K$, где k_l -ая группа соответствует состоянию k_l в момент l . К сожалению, эти числа не определяют однозначно совокупность чисел $p(k_0, x_1, x_2, \dots, x_i, \dots)$, соответствующих бесконечно длинному наблюдению. Но к счастью, они определяют довольно простое множество, которое гарантированно содержит в себе эту совокупность. Для краткости введем обозначение q_l для $|K|$ -мерного вектора, координатами которого являются вероятности $p(k_0, x_1, x_2, \dots, x_l)$. Подобным образом обозначим $q_l(k_l)$, $k_l \in K$, $|K|$ -мерный вектор, k_0 -ой координатой которого является вероятность $p(k_0, x_1, x_2, \dots, x_l, k_l)$. Нас интересуют вероятности $p(k_0, x_1, x_2, \dots, x_n)$, $n > l$, (то есть, вектор q_n), которые невозможно вычислить, потому что наблюдения $x_{l+1}, x_{l+2}, \dots, x_n$ неизвестны. Однако из

марковской модели следует, что эти вероятности равны

$$p(k_0, x_1, x_2, \dots, x_n) = \sum_{k_l \in K} p(k_0, x_1, x_2, \dots, x_l, k_l) p(x_{l+1}, x_{l+2}, \dots, x_n | k_l).$$

Запишем это в векторной форме

$$q_n = \sum_{k_l \in K} q_l(k_l) p(x_{l+1}, x_{l+2}, \dots, x_n | k_l). \quad (9.87)$$

Это равенство утверждает, что вектор q_n (то есть совокупность вероятностей $p(k_0, x_1, x_2, \dots, x_n)$, $k_0 \in K$) принадлежит выпуклому конусу, ребрами которого являются векторы $q_l(k_l)$ (то есть совокупности $(p(k_0, x_1, x_2, \dots, x_l, k_l)$, $k_0 \in K$), $k_l \in K$). Обозначим этот конус Q_l . Теперь можем поставить следующий вопрос: в каком случае можно сказать, чему равен $\operatorname{argmax}_{k_0 \in K} p(k_0, x_1, x_2, \dots, x_n)$, если известно только, что совокупность чисел $p(k_0, x_1, x_2, \dots, x_n)$ принадлежит конусу Q_l ? Ответ здесь совершенно очевиден. Если для любой совокупности $p'(k_0)$, $k_0 \in K$, принадлежащей конусу Q_l , состояние $\operatorname{argmax}_{k_0 \in K} p'(k_0)$ является одним и тем же состоянием k_0^* , то $\operatorname{argmax}_{k_0 \in K} p(k_0, x_1, x_2, \dots, x_n)$ есть также состояние k_0^* .

Для того, чтобы проверить сформулированное условие, конечно же, нет нужды проверять все векторы в конусе Q_l . Достаточно проверить только ребра этого конуса, то есть совокупности $(p(k_0, x_1, x_2, \dots, x_l, k_l)$, $k_0 \in K$), $k_l \in K$. Мы уверены, что Ты сможешь сам доказать следующее утверждение.

Пусть существует такое значение $k_0^* \in K$, что для любых $k_0 \in K$ и $k_l \in K$ выполняется неравенство

$$p(k_0^*, x_1, x_2, \dots, x_l, k_l) \geq p(k_0, x_1, x_2, \dots, x_l, k_l).$$

Тогда при любых наблюдениях $x_{l+1}, x_{l+2}, \dots, x_n$ и любом состоянии k_0 выполняется неравенство

$$p(k_0^*, x_1, x_2, \dots, x_n) \geq p(k_0, x_1, x_2, \dots, x_n).$$

Да, это легко доказывается.

Ваши объяснения относятся к случаю, когда речь идет об определении наиболее вероятного состояния k_0 . Интересно узнать, обобщаются ли приведенные соображения на общий случай минимизации байесовского риска при любой, но наперед заданной матрице штрафов.

Конечно, да. Взгляни на этот общий случай уже без нашей помощи. Вспомни теорему о выпуклости классов в пространстве вероятностей, доказанную на первой лекции. Наконец выстрелило ружье из первого акта драмы под названием распознавание образов. Мы о нем уже почти забыли.

Даже если я понял все ваши объяснения, я не понял, где же ошибся, когда пришел к выводу, что при любом i наблюдение x_i содержит в себе ненулевую информацию о состоянии k_0 .

Да совсем Ты не ошибся! При любом i наблюдение x_i уточняет знания о состоянии k_0 , то есть влияет на апостериорные вероятности его значений. Отсюда действительно следует, что существует такая функция потерь, что решение о состоянии k_0 без учета наблюдения x_i будет отличаться от решения, принятого с учетом x_i . И это справедливо для любого i . Но нас же интересует ситуация, когда функция потерь задана. Она может быть любой, но она Тебе известна еще до того, как Ты начал распознавать. В этом случае могут возникнуть условия, когда наблюдения x_i , начиная с некоторого i , перестают быть полезными. Несмотря на то, что они несут информацию о состоянии k_0 , всех последующих после момента i наблюдений уже недостаточно, чтобы изменить решение о состоянии k_0 .

Есть еще одно препятствие, которое мешает мне быть уверенным, что я разобрался в этих вопросах. Имеют ли рассмотренные нами вопросы какое-то отношение к вальдовскому последовательному анализу? Ведь там тоже ставится вопрос о разумном прекращении наблюдений. Почему Вам удалось ответить на эти вопросы, ни разу не сославшись на Вальда?

Вопросы, которые мы обсудили, не принадлежат ‘континенту’, который открыл Вальд. Этот континент очень обширный и поэтому наш ответ будет довольно поверхностным. Очень грубо говоря, вальдовские процедуры отвечают на вопрос, как долго следует исследовать объект, чтобы можно было оценить его состояние с заранее заданным качеством. Вальдом было доказано, что при определенных условиях и определенной модели распознаваемого объекта предложенные им процедуры сходятся в том смысле, что обязательно завершаются принятием решения, и качество этого решения не хуже определенного заранее заданного.

Мы настойчиво обращаем Твое внимание, что такая сходимость имеет место при определенных предположениях об объекте. Зачастую, с неаккуратной ссылкой на Вальда, этот результат представляют в вульгарной формулировке, что при достаточно длительном наблюдении объекта его состояние можно оценить с любой степенью точности. Это очень небрежное манипулирование результатами Вальда. Только что мы с Тобой убедились, что при анализе марковских последовательностей вполне может наступить ситуация, когда решение на основании выполненных наблюдений не будет достаточно качественным, но никакое дальнейшее наблюдение его не улучшит просто потому, что это решение не изменится.

Мы с Тобой тоже рассмотрели задачу о целесообразном прекращении наблюдений. Но в нашей задаче разрешалось прекращать наблюдения не при достижении достаточного качества принимаемого решения, как у Вальда, а в ситуации, когда это качество, каким бы оно ни было, не может улучшиться при сколь угодно длительном продолжении наблюдений. Таким образом, мы имеем два различных условия прекращения наблюдений.

К сожалению, нельзя гарантировать, что при достаточно длительных наблюдениях каждое из этих условий выполнится. Для одних марковских объектов никогда не выполнится вальдово условие прекращения наблюдений, для других никогда не выполнится сформулированное нами условие.

И вот теперь возникает действительно интересный вопрос. Можно ли утверждать, что при достаточно длительных наблюдениях одно из этих условий обязательно выполнится? Можно ли доказать, что процесс наблюдения обязательно закончится либо потому, что окажется возможным принять решение с заранее указанным качеством, либо потому, что уже достигнутое качество не может быть улучшено?

Правильно ли я понял, что этот вопрос открыт?

Мы, по крайней мере, не знаем ответа на этот вопрос.

Я познакомился ближе, хотя все еще очень поверхностно, с методом ближайшего соседа и увидел, что понимание его методов может оказаться плодотворным для рассматриваемых нами статистических задач распознавания последовательностей. Мы видели, что ряд задач как в этой, так и в предыдущей лекциях сводится к оптимизации функции $F: K^{n+1} \times X^n \rightarrow \mathbb{R}$ со значениями $F(\bar{k}, \bar{x})$ по последовательности \bar{k} при заданной последовательности \bar{x} . При этом множество последовательностей \bar{k} было задано так, что оно соответствовало множеству путей на графе, а возникающие оптимизационные задачи сводились к поиску кратчайшего пути. На эти задачи можно смотреть с точки зрения методов ближайшего соседа, отождествив множество L идеальных наблюдений с множеством путей на графе, и понимая число $F(\bar{k}, \bar{x})$, как сходство реального наблюдения \bar{x} с идеальным наблюдением \bar{k} . Правда, при этом оказывается, что множество L идеальных наблюдений не обязательно является подмножеством всех возможных наблюдений, но для того, что я хочу сказать, это не имеет большого значения.

Существенным является, что сформулированный вами метод ближайшего соседа является лишь частным случаем того, что сейчас разработано в этой области. В более общем случае ищут не единственного соседа, а группу ближайших соседей. Либо находят все идеальные объекты, отличающиеся от наблюдения не более, чем на заранее заданную величину, либо находят заранее заданное количество, скажем, d идеальных объектов, наиболее сходных с реальным наблюдением. Решение о скрытых свойствах наблюдаемого объекта принимается затем не на основе единственного идеального объекта, наиболее похожего на наблюдаемый объект, а на основе целой группы идеальных объектов. Назову такой метод методом d ближайших соседей. Применительно к рассматриваемым нами моделям этот метод обозначает, что распознавание должно осуществляться не через поиск единственного, а именно, кратчайшего пути на графе, а через поиск совокупности путей, состоящей из d кратчайших путей.

А теперь встает вопрос, когда в технологическом процессе обработки марковских последовательностей может потребоваться именно такая обработка множества путей на графе. Я этот вопрос вам даже не задаю, потому что сам вижу, что она нужна почти что сплошь и рядом.

Прежде всего, статистическая задача распознавания изначально может ставиться не как поиск состояния, с наибольшей вероятностью совпадающего с действительным состоянием, а как поиск подмножества, состоящего из d состояний, при котором минимизируется вероятность того, что действительное состояние окажется за пределами отобранного подмножества. Очевидно, что решение такой задачи состоит в отборе d состояний с наибольшими апостериорными вероятностями, или, применительно к нашей модели, в отыскании d кратчайших путей на графе.

В ваших лекциях уже фактически возникала необходимость отыскания не единственного наилучшего пути на графе. При выводе алгоритмов настройки распознающих алгоритмов требовалось не просто находить наилучший путь на графе, а строго наилучший. Это значит, что помимо отыскания наилучшего пути надо было убедиться, что он единственный, то есть, что все остальные хуже. Это можно сделать лишь через поиск не одного, а двух наилучших путей с последующей проверкой, что второй путь хуже первого.

И наконец, я укажу чуть ли не самый интересный пример. Совсем нередкой является ситуация, когда решаемая задача может быть точно формализована, но решение этой задачи исключительно трудное. Например, множество последовательностей \bar{k} , соответствующих каждой заданной последовательности \bar{x} , задается графом такой исключительной сложности, что поиск кратчайшего пути на этом графе практически невозможен. В то же время для каждой заданной последовательности \bar{k} легко определяется, согласуется ли она с наблюдением \bar{x} или нет. Ситуация эта подобна той, когда отношение последовательностей \bar{x} и \bar{k} задано в виде равенства $\varphi(\bar{x}, \bar{k}) = 0$, где φ - легко вычисляемая функция. В этом случае для любой заданной пары (\bar{x}, \bar{k}) легко проверяется, является ли она допустимой или нет. В то же время может оказаться исключительно трудным определить, существует ли последовательность \bar{k}^* , образующая с заданной последовательностью \bar{x}^* допустимую пару так, что $\varphi(\bar{x}^*, \bar{k}^*) = 0$.

Предположим, что мне встретилась именно такая ситуация, когда задача распознавания сводится к вычислению

$$\bar{k}^* = \operatorname{argmax}_{\bar{k}} F(\bar{k}, \bar{x})$$

для заданного \bar{x} при дополнительном условии

$$\varphi(\bar{x}, \bar{k}) = 0.$$

В этой ситуации я бы применил следующий технологический прием ее решения. Требование $\varphi(\bar{x}, \bar{k}) = 0$ я бы заменил более слабым требованием $\varphi^*(\bar{x}, \bar{k}) = 0$ так, что из $\varphi(\bar{x}, \bar{k}) = 0$ следует $\varphi^*(\bar{x}, \bar{k}) = 0$, и к тому же так,

чтобы задача отыскания

$$\bar{k}^* = \operatorname{argmax}_{\bar{k}} F(\bar{k}, \bar{x})$$

при условии

$$\varphi^*(\bar{x}, \bar{k}) = 0$$

сводилась к поиску кратчайшего пути на графе. Исходная же задача состоит, таким образом, в том, чтобы среди путей на графе, удовлетворяющих условию $\varphi(\bar{x}, \bar{k}) = 0$, найти кратчайший. В такой ситуации, которая совсем не редкость в распознавании, я бы поступил следующим образом. С самого начала я бы нашел кратчайший путь \bar{k}^0 на графе и проверил бы, выполняется ли условие $\varphi(\bar{x}, \bar{k}^0) = 0$. Если оно выполняется, то решена исходная задача. Если оно не выполняется, то можно принять решение "отказ от распознавания". Это, конечно же, самый простой выход. Более разумно было бы найти путь \bar{k}^1 , кратчайший среди тех путей на графе, которые не совпадают с \bar{k}^0 . Подобным образом можно было бы последовательно находить пути $\bar{k}^0, \bar{k}^1, \dots, \bar{k}^i, \dots$, и так далее. Первый путь \bar{k}^i в этой последовательности, для которого выполняется условие $\varphi(\bar{x}, \bar{k}^i) = 0$, и будет решением исходной задачи. Таким образом, задавшись некоторым числом, скажем 10000, можно было бы применять решение "отказ от распознавания" лишь в том случае, если ни один из 10000 наилучших путей на графе не удовлетворяет условию $\varphi(\bar{x}, \bar{k}) = 0$.

Указанный прием может оказаться исключительно полезным в том случае, когда требование $\varphi(\bar{x}, \bar{k}^0) = 0$ в принципе не выражается формально, а выражено, например, словами "путь на графе, который нравится наблюдателю". В таком случае алгоритм представляет наблюдателю один путь за другим, начиная с кратчайшего, до тех пор, пока наблюдатель не прекратит этот процесс. Последний выданный алгоритмом путь - это и есть кратчайший из тех путей на графе, которые нравятся наблюдателю.

К таким выводам я прихожу, когда смотрю на наши задачи с точки зрения метода d ближайших соседей. Я надеюсь, что эти выводы представляют интерес.

Все, что Ты сказал, очень интересно. Но Ты не задал вопрос, и мы не знаем, что Тебе отвечать.

Для реализации всех указанных технологий надо располагать эффективным алгоритмом поиска d кратчайших путей на графе, а не единственного кратчайшего пути. При этом желательно, чтобы сложность алгоритма росла не слишком быстро с ростом d . Я не знаю такого алгоритма. Можете ли вы помочь мне найти его?

Мы не думаем, что это сложная задача. Полагаем, что существует алгоритм, сложность которого растет линейно с ростом d так, что порядок этой сложности оказывается не хуже, чем $\mathcal{O}(|K|^2 dn)$.

В это с трудом верится.

Пусть имеется граф G , определяющий множество L путей от вершины α до вершины β . Допустим, что я уже нашел в L путь \bar{k}^* с наименьшей длиной и мне нужно найти сейчас путь \bar{k}^1 из множества $L \setminus \{\bar{k}^*\}$, имеющий наименьшую длину. Сложность заключается в том, что множество $L \setminus \{\bar{k}^*\}$ невозможно представить в виде множества путей на том или ином подграфе графа G . Из графа G невозможно исключить ни одного ребра, так как через каждое ребро проходит некоторый путь из множества $L \setminus \{\bar{k}^*\}$. Множество $L \setminus \{\bar{k}^*\}$ состоит не только из тех путей, которые не пересекаются с \bar{k}^* , оно включает в себя и те пути, которые отличаются от пути \bar{k}^* хотя бы одним ребром.

Правда, несмотря на все эти сложности, можно построить такой новый граф G^1 , что множество путей на нем будет соответствовать именно множеству $L \setminus \{\bar{k}^*\}$. Но в этом графе будет в два раза больше вершин, чем в исходном графе. Это значит, что поиск двух кратчайших путей на графе окажется в три раза сложнее, чем поиск единственного кратчайшего пути.

При росте количества путей, которые следует найти, сложность, как мне кажется, лавинообразно нарастает. Если мне нужно найти, скажем, 10-ый путь, то он должен отличаться по крайней мере одним ребром от каждого из 9 уже найденных путей. Мне трудно сообразить, как представить такое множество путей, и к тому же так, чтобы кратчайший путь в этом множестве легко было найти.

Мы очень хорошо поняли Твои трудности и покажем, как их преодолеть. Но это будет последняя возможность, которую Ты нам предоставляешь, чтобы убедить Тебя в плодотворности алгебраических формулировок задач в виде сверток в тех или иных полукольцах. Судя по всему, мы Тебя в этом еще не убедили. Если бы Ты уделил этому материалу то внимание, которое он заслуживает, сейчас бы у Тебя не возникли трудности.

Действительно, я еще не включил методы обобщенных сверток в круг тех средств, которыми мог бы активно пользоваться. Разобранный вами пример с левенштейновой аппроксимацией не кажется мне убедительным. Ведь вы намеренно усложнили задачу, сняв все ограничения на функции in , de и ch . По мне это неоправданно общее рассмотрение задачи. Достаточно ввести вполне естественное ограничение на функции in , de и ch , чтобы задачи левенштейновой аппроксимации стали разумно решаемыми, или даже тривиальными. Я имею в виду, например, ограничения типа неравенств треугольника. Мне известны работы Вагнера [Wagner and Fischer, 1974; Wagner and Seiferas, 1978], которые выполняют левенштейнову аппроксимацию не только в рамках регулярных языков, но и в более общем случае контекстно-свободных языков.

А в задаче нахождения d кратчайших путей на графе Ты не видишь неестественного нагромождения трудностей?

Не вижу. Эта задача в своей изначальной формулировке кажется мне трудной. Но я уже догадываюсь, что вы знаете алгоритм решения и он окажется неожиданным для меня.

А теперь посмотри, как можно записать задачу о поиске d кратчайших путей на графе в виде обобщенной свертки. Ты увидишь, что после ее формулировки в таком виде она оказывается настолько простой, что о ней не стоит и говорить. Правда, от Тебя поначалу потребуется некоторое терпение, потому что ради краткости мы изложим эти идеи по образцу наименее перевариваемых математических текстов, в которых сначала вводятся какие-то определения без каких-либо объяснений, зачем они понадобятся.

Введем следующие определения, которые нам пригодятся для формулировки задачи о d кратчайших путях на графе.

1. Пусть R - множество неотрицательных вещественных чисел, дополненное особым "числом" ∞ . Предполагается, что для этого числа и для любого числа $a \in R$ справедливо, что $\infty + a = \infty$ и $\min(\infty, a) = a$.
2. Пусть R^d - множество упорядоченных массивов вида (a_1, a_2, \dots, a_d) , где $a_i \in R$, $i = 1, 2, \dots, d$, $a_1 \leq a_2 \leq \dots \leq a_d$.
3. На множестве $R^d \times R^d$ определим функцию двух переменных, которая принимает значения из множества R^d . Эту функцию назовем *сложением массивов*. Для любых двух массивов $a \in R^d$ и $b \in R^d$ эта функция следующим образом определяет сумму $c = a \oplus b$.
Под вычислением суммы $a = (a_1, a_2, \dots, a_d)$ и $b = (b_1, b_2, \dots, b_d)$ имеется в виду

- построение массива $(a_1, a_2, \dots, a_d, b_1, b_2, \dots, b_d)$ длиной $2d$;
- упорядочение этого массива по возрастанию;
- взятие первых d чисел полученного массива в качестве суммы $a \oplus b$.

Сложение массивов, определенное таким способом, есть ассоциативная и коммутативная операция с нулевым элементом 0^\oplus , которым является массив $(\infty, \infty, \dots, \infty)$.

4. Определим функцию вида $R^d \times R^d$ двух переменных, которая принимает значения из множества R^d , и назовем ее *умножением массивов*. Для каждой пары массивов $a = (a_1, a_2, \dots, a_d)$ и $b = (b_1, b_2, \dots, b_d)$ их произведение $a \otimes b$ определяется так, что
 - строится массив $(a_i + b_j, i = 1, 2, \dots, d; j = 1, 2, \dots, d)$ длиной d^2 ;
 - полученный массив упорядочивается по возрастанию;
 - первые d чисел полученного упорядоченного массива объявляются массивом $a \otimes b$.

Определенное таким способом умножение есть ассоциативная и коммутативная операция, с нулевым элементом 1^\otimes , которым является массив $(0, \infty, \dots, \infty)$.

5. Операция умножения дистрибутивна относительно введенного ранее сложения, то есть, $a \otimes (b \oplus c) = (a \otimes b) \oplus (a \otimes c)$. Кроме того, произведение любого массива на определенный ранее нулевой массив есть нулевой массив. Поэтому введенные операции сложения и умножения

образуют полукольцо на множестве R^d упорядоченных массивов длины d .

6. Пусть X - конечное множество, а f - функция $X \rightarrow R$. Для каждой функции f введем обозначение f' для следующей функции вида $X \rightarrow R^d$, принимающей значения из множества R^d . Для $x \in X$ значение $f'(x)$ - это массив длиной d , первым элементом которого есть число $f(x)$, а остальные $d - 1$ числа равны ∞ . Из этого определения следует, что произведение $\bigotimes_{x \in X} f'(x)$ - это массив длиной d , первым элементом которого есть число $\sum_{x \in X} f(x)$, а остальные $d - 1$ чисел равны ∞ . Из этого же определения следует, что сумма $\bigoplus_{x \in X} f'(x)$ - это упорядоченный массив, составленный из d самых малых чисел из совокупности $(f(x), x \in X)$.

Сформулируем теперь *исходную задачу* поиска d кратчайших путей на графе.

Пусть X и K - конечные множества, а p_i , $i = 1, 2, \dots, n$, - это n функций вида $K \times X \times K \rightarrow \mathbb{R}$. Для любых двух последовательностей $\bar{x} = (x_1, x_2, \dots, x_n)$, $x_i \in X$, и $\bar{k} = (k_0, k_1, \dots, k_n)$, $k_i \in K$, определено число

$$F(\bar{x}, \bar{k}) = \sum_{i=1}^n p_i(k_{i-1}, x_i, k_i) \quad (9.88)$$

Задача состоит в построении алгоритма, который для любой последовательности $\bar{x} \in X^n$ определяет d самых малых чисел из совокупности $(F(\bar{x}, \bar{k}), \bar{k} \in K^{n+1})$, которая состоит $|K|^{n+1}$ чисел.

Наконец, сформулируем эту задачу в виде *обобщенных сверток* и немедленно увидим *ее решение*.

С использованием введенных операций сложения и умножения массивов сформулированная задача состоит в вычислении массива Q , определяемого формулой

$$Q(\bar{x}) = \bigoplus_{\bar{k} \in K^{n+1}} F(\bar{x}, \bar{k}) = \bigoplus_{\bar{k} \in K^{n+1}} \bigotimes_{i=1}^n p'_i(k_{i-1}, x_i, k_i). \quad (9.89)$$

В этой формуле использованы обозначения F' и p'_i , а не F и p_i , как это было в формуле (9.88). Значения функций F' и p'_i (со штрихами) - это массивы длиной d , а не числа, которые являются значениями функций F и p_i (без штрихов).

В определении (9.89) ничего не изменится, если каждое слагаемое умножить на множители $\varphi'(k_0)$ и $\psi'(k_n)$, равные единице, то есть равные массивам $(0, \infty, \dots, \infty)$. Таким образом, получаем

$$Q(\bar{x}) = \bigoplus_{\bar{k} \in K^{n+1}} \varphi'(k_0) \otimes \left(\bigotimes_{i=1}^n p'_i(k_{i-1}, x_i, k_i) \right) \otimes \psi'(k_n). \quad (9.90)$$

Выражение (9.90) можно представить более подробно в виде

$$Q(\bar{x}) = \bigoplus_{k_0 \in K} \left(\bigoplus_{k_1 \in K} \left(\bigoplus_{k_2 \in K} \cdots \left(\bigoplus_{k_n \in K} \varphi'(k_0) \otimes \left(\bigotimes_{i=1}^n p'_i(k_{i-1}, x_i, k_i) \right) \otimes \psi'(k_n) \right) \cdots \right) \right). \quad (9.91)$$

В силу дистрибутивности умножения массивов относительно их сложения, каждый сомножитель в формуле (9.91) можно вынести за знаки суммирования по тем переменным, от которых этот сомножитель не зависит. В итоге получаем

$$Q(\bar{x}) = \bigoplus_{k_0} \varphi'(k_0) \otimes \left(\bigoplus_{k_1} p'_1(k_0, x_1, k_1) \otimes \cdots \left(\bigoplus_{k_n} p'_n(k_{n-1}, x_n, k_n) \otimes \psi'(k_n) \right) \cdots \right).$$

Мне кажется, что я уже все понял. Теперь следует последовательно выполнять суммирование, начиная с самых внутренних скобок. В результате каждого такого суммирования получаются $|K|$ массивов длины d . Сначала я вычислю массивы $f_{n-1}(k_{n-1})$ длиной d для каждого $k_{n-1} \in K$ по формуле

$$f_{n-1}(k_{n-1}) = \bigoplus_{k_n \in K} \left(p'_n(k_{n-1}, x_n, k_n) \otimes \psi'(k_n) \right). \quad (9.92)$$

Затем для каждого $k_{n-2} \in K$ я вычислю массивы $f_{n-2}(k_{n-2})$,

$$f_{n-2}(k_{n-2}) = \bigoplus_{k_{n-1} \in K} \left(p'_{n-1}(k_{n-2}, x_{n-1}, k_{n-1}) \otimes f_{n-1}(k_{n-1}) \right)$$

и буду продолжать эти вычисления для $i = n-3, n-4, \dots, 2, 1, 0$ по формуле

$$f_i(k_i) = \bigoplus_{k_{i+1} \in K} \left(p'_{i+1}(k_i, x_{i+1}, k_{i+1}) \otimes f_{i+1}(k_{i+1}) \right), \quad k_i \in K. \quad (9.93)$$

Искомый массив $Q(\bar{x})$, состоящий из d кратчайших путей на графе, записанных в возрастающем порядке, я получу по формуле

$$Q(\bar{x}) = \bigoplus_{k_0 \in K} \left(\varphi(k_0) \otimes f_0(k_0) \right) = \bigoplus_{k_0 \in K} f_0(k_0). \quad (9.94)$$

Ты не поставил завершающую точку в своих формулах, которой Ты бы нас очень обрадовал. Вычисления по формулам (9.92), (9.93) и (9.94) можно представить как вычисление матричного произведения

$$\varphi P_1 P_2 \cdots P_{n-1} P_n \psi \quad (9.95)$$

в полукольце на множестве упорядоченных массивов длины d с введенными операциями сложения и умножения массивов. В формуле (9.95) символы φ

и ψ представляют соответственно вектор-строку и вектор-столбец размерности $|K|$. Каждая компонента этого вектора - это массив $(0, \infty, \infty, \dots, \infty)$ длины d . Элементами матриц $P_i, i = 1, \dots, n$, также являются массивы длины d . В k' -ой строке и k'' -ом столбце матрицы P_i записан массив, первым элементом которого является число $p_i(k', x_i, k'')$, а все остальные элементы равны ∞ .

Представленное решение, конечно, очень убедительно показывает плодотворность алгебраических представлений наших задач. Я говорю не столько о конечном результате (9.95), сколько о том, как лаконичен и прозрачен переход от формальной постановки задачи (9.89) к вычислительной процедуре (9.92), (9.93), (9.94). Особенно поражает, что для этого перехода не требуется никакая сообразительность и никакие интуитивные догадки. Алгебраическая форма задачи непосредственно указывает направление ее решения. Искомый объект (в данном случае, совокупность длин кратчайших путей) определен формулой (9.95). Чисто формальные и гарантированное эквивалентные преобразования этой формулы приводят к формуле, которая представляет вычислительную процедуру решения задачи.

Совсем иначе обстоит дело с задачей в ее графовой постановке. Задача определена не формулой, а словесно. Лишь поначалу кажется, что графовое представление, дополненное некоторыми предложениями естественно-го языка, обеспечивает наглядность. Однако это не так, потому что такая формулировка задачи не поддержана никакими инструментальными средствами, с помощью которых одно словесное требование можно было бы эквивалентно заменить другим словесным утверждением. Эта эквивалентность основана лишь на так называемых разумных соображениях.

Понимая сейчас алгебраическую постановку задачи, я вижу, что наглядность графowego представления на самом деле дезориентирует. Действительно, граф что-то представляет, но совсем не то, что должно быть найдено в процессе решения задачи. Искомый объект, то есть совокупность d кратчайших путей на графе, вообще не представляется с помощью графа. При наложении путей из этой совокупности друг на друга получается граф, который представляет не только эти пути, но еще и много других.

Мне кажется, что сейчас, когда я понял исключительную простоту алгоритма отыскания d кратчайших путей на графе, я мог бы справиться и с более общими задачами. Я имею в виду, например, распознавание совокупности объектов, которые образуют не последовательность, а более сложную структуру, например, ациклический граф. Мне бы хотелось обобщить показанный вами результат на случай левенштейновой аппроксимации, когда требуется найти не только наилучшую аппроксимацию, но d наилучших аппроксимаций.

Не сомневаемся в том, что Тебе удастся обобщить процедуру (9.95) на случай ациклических графов. Но будь очень внимателен при рассмотрении левенштейновой аппроксимации. Нам удалось минимизировать левенштейнову функцию на регулярном языке в силу идемпотентности операции мини-

мизации, используемой в качестве сложения в соответствующем полукольце. Только благодаря этому свойству нам удалось представить некоторые бесконечные свертки в виде конечных выражений. Вспомни леммы 9.2 и 9.3 из лекции. Сложение массивов d в данном нами определении не идемпотентно. Следовательно, Тебе придется придумать что-нибудь, что сыграло бы ту же роль, что и леммы 9.2 и 9.3.

Мне бы хотелось рассмотреть вычислительную процедуру (9.95) во всех деталях, не на уровне сложения и умножения массивов, а на уровне операций с числами, то есть, с элементами этих массивов.

Мы с удовольствием сделаем это вместе с Тобой, потому что процедура (9.95) очень поучительна с вычислительной точки зрения. Давай начнем с Твоего рассмотрения процедуры.

Задача состоит в вычислении массива длиной d , определенного матричным произведением

$$\varphi P_1 P_2 \cdots P_n \psi. \quad (9.96)$$

Следует вычислить последовательность $|K|$ -мерных векторов-строк f_1, f_2, \dots, f_n , где $f_1 = \varphi \cdot P_1$, а $f_i = f_{i-1} \cdot P_i$, $i = 2, \dots, n$. Затем следует вычислить "скалярное произведение" $f_n \cdot \psi$. Вычисление (9.96) состоит в n -кратном умножении $|K|$ -мерного вектора-строки на матрицу размера $|K| \times |K|$. Я оценю сложность этого умножения, принимая во внимание, что компоненты вектора f_{i-1} и матрицы P_i - это массивы длиной d , а не числа. Компонента $f_i(k)$ вектора $f_i = f_{i-1} \cdot P_i$ - это массив длиной d , определенный как сумма

$$f_i(k) = \bigoplus_{k' \in K} f_{i-1}(k') \otimes P_i(k', k). \quad (9.97)$$

Чтобы построить массив $f_i(k)$ для всех $k \in K$, надо построить $|K|^2$ вспомогательных массивов

$$c(k', k) = f_{i-1}(k') \otimes P_i(k', k), \quad k' \in K, \quad (9.98)$$

а затем вычислить их сумму

$$f_i(k) = \bigoplus_{k' \in K} c(k', k). \quad (9.99)$$

Эти вычисления следует выполнить для всех $k \in K$. Таким образом, вычисление произведения $f_i = f_{i-1} \otimes P_i$ требует $|K|^2$ умножений массивов длины d и $|K|^2$ сложений этих массивов. Я проанализирую вычислительную сложность этих операций с массивами.

Пусть $a = (a_1, a_2, \dots, a_d)$ и $b = (b_1, b_2, \dots, b_d)$ - два массива. Чтобы вычислить их произведение, надо построить массив $(a_i + b_j, 1 \leq i \leq d,$

$1 \leq j \leq d$) и выбрать d наименьших чисел из этого массива. Но в рассматриваемом нами случае один из этих массивов, а именно, массив $P_i(k, k')$ имеет специфический вид. Все числа этого массива, кроме первого, равны ∞ . Таким образом, $b = (b_1, \infty, \infty, \dots, \infty)$. Произведение массива $a = (a_1, a_2, \dots, a_d)$ на массив такого частного вида есть просто массив $(a_1 + b_1, a_2 + b_1, \dots, a_d + b_1)$. Очевидно, что сложность построения такого массива имеет порядок $\mathcal{O}(d)$.

Пусть $a = (a_1, a_2, \dots, a_d)$ и $b = (b_1, b_2, \dots, b_d)$ - два массива. Их суммирование обозначает выбор наименьших d чисел из массива $(a_1, a_2, \dots, a_d, b_1, b_2, \dots, b_d)$. Поскольку массивы a и b упорядочены по возрастанию, их сложение имеет сложность порядка $\mathcal{O}(d)$, так как его можно выполнить, например, следующей программой.

```

k = i = j = 1;
while ( k ≤ d ) {
  if ( a_i ≤ b_j )
    { c_k = a_i; i = i + 1; }
  else
    { c_k = b_j; j = j + 1; }
  k = k + 1;
}

```

Таким образом, сложность вычисления произведения $f_{i-1} \cdot P_i$ имеет порядок $\mathcal{O}(|K|^2 d)$, а сложность вычисления произведения (9.96) имеет порядок $\mathcal{O}(|K|^2 d n)$. Это результат, который вы предсказывали до начала нашего рассмотрения задачи. Этот результат обозначает, что поиск d кратчайших путей на графе всего лишь d раз сложнее, чем поиск единственного кратчайшего пути на графе.

Ты проскочил мимо важного свойства сложения массивов. Ты прав, что сложение $a_1 \oplus a_2$ имеет сложность порядка $\mathcal{O}(d)$. Но отсюда не следует, что сложность вычисления суммы $a_1 \oplus a_2 \oplus \dots \oplus a_m$ имеет порядок $\mathcal{O}((m-1)d)$. В действительности она намного меньше и равна $\mathcal{O}((m-1) + (d-1) \log m)$. Если Ты это поймешь, то сможешь устроить умножение $f_{i-1} \cdot P_i$ не за время $\mathcal{O}(|K|^2 d)$, а за меньшее время, равное $\mathcal{O}(|K|^2 + d |K| \log |K|)$. Мы советуем Тебе продумать программу умножения вектора f_{i-1} на матрицу P_i более тщательно.

Чтобы не иметь дело с ненужными подробностями, я введу упрощенное обозначение для вычисления массива $f_i(k)$ для какого-то одного фиксированного значения $k \in K$ в соответствии с формулой (9.97). Поскольку значение k я зафиксировал, я могу его опустить в формуле (9.97) и получить

$$f' = \bigoplus_{k \in K} f(k) \otimes P'(k), \quad (9.100)$$

где f' , $f(k)$, $P'(k)$, $k \in K$, - это все массивы длиной d . Массивы $P'(k)$, $k \in K$, имеют частный вид. В них только первый элемент не равен ∞ ,

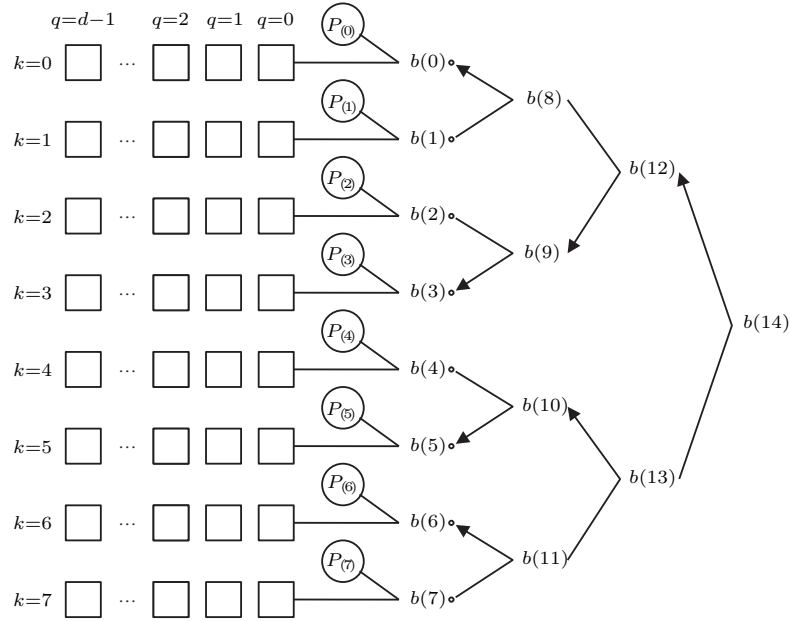


Figure 9.6 Arrangement of data for effective computation of the product of vector f_{i-1} with matrix P_i . The squares contain numbers $f(k, q)$.

а все остальные равны ∞ . Первый элемент массива $P'(k)$ обозначу $P(k)$. Вычисления по формуле (9.100) обозначают следующее. Имеются $d \times |K|$ чисел $f(k, q)$, $k = 0, 1, \dots, |K| - 1$, $q = 0, 1, \dots, d - 1$, а также $|K|$ чисел $P(k)$, $k = 0, 1, \dots, |K| - 1$. Совокупность чисел $f(k, q)$ частично упорядочена в том смысле, что для каждой тройки k, q_1, q_2 , $q_1 < q_2$, выполняется неравенство $f(k, q_1) \leq f(k, q_2)$. Числа $P(k)$ - произвольные.

На рис. 9.6 массив $f(k, q)$, $k = 0, 1, \dots, |K| - 1$, $q = 0, 1, \dots, d - 1$, представлен в виде прямоугольного поля, составленного из квадратиков. В каждом квадратике записано число $f(k, q)$. Совокупность чисел $P(k)$, $k = 0, 1, \dots, |K| - 1$, представлена на рис. 9.6 в виде столбца, составленного из кружочков. В кружочках записаны числа $P(k)$. Рисунок представляет случай, когда $|K| = 8$, а d произвольно.

Мне необходимо оценить сложность построения массива чисел, составленного из d наименьших чисел из совокупности $(f(k, q) + P(k), k = 0, 1, \dots, |K| - 1, q = 0, 1, 2, \dots, d - 1)$.

Очевидно, что первое число в этом массиве равно $\min_k (f(k, 0) + P(k))$. Вычислить его не трудно, и для этого требуется время порядка $\mathcal{O}(|K|)$. Однако, я буду вычислять это число не общепринятым очевидным способом, а иначе, с помощью структуры данных, которая представлена деревом в правой части рис. 9.6. Количество вершин в этом дереве равно $2|K| - 1$.

В вершинах этого дерева записываются числа $b(j)$, которые вычисляются следующим сегментом программы:

```

for ( k = 0; k < |K|; k ++ )
  { b(k) = f(k, 0) + P(k); q*(k) = 1; }
k = 0; j = |K|;
while ( j ≠ 2|K| - 1 ) {
  if ( b(k) ≤ b(k+1) )
    { b(j) = b(k); ind(j) = k; }
  else
    { b(j) = b(k+1); ind(j) = k+1; }
  k = k + 2; j = j + 1;
}

```

(9.101)

После окончания работы этого программного сегмента в каждой вершине дерева записывается меньшее из двух чисел, записанных в двух вершинах, соседних к данной вершине и расположенных слева от нее. К примеру, $b(9) = \min(b(2), b(3))$ и $b(12) = \min(b(8), b(9))$. При этом числа $\text{ind}(j)$, $j = |K|, \dots, 2(|K| - 1)$, указывают, какое именно из двух чисел было переписано с левой вершины в правую. Один из возможных наборов значений указателей показан на рис. 9.6 с помощью стрелок. Так, к примеру, стрелка из девятой вершины направлена на третью вершину, и это обозначает, что $\text{ind}(9) = 3$ и что $b(3) \leq b(2)$. Очевидно, что при такой организации данных число $b(2(|K| - 1))$ в корне дерева есть наименьшее из чисел $b(j)$, $j = 0, \dots, |K| - 1$, записанных в листках дерева, а совокупность стрелок позволяет выйти на листок, в котором записано это наименьшее число. К примеру, совокупность стрелок, представленных на рис. 9.6, соответствует случаю, когда $b(14) = b(3)$. Программный сегмент (9.101) вычисляет также числа $q^*(k)$, которые указывают, сколько чисел было уже взято из совокупности чисел $f(k, q)$, $q = 0, \dots, d - 1$, для записи их в соответствующие листья дерева.

Приведенная программа поиска наименьшего числа из совокупности чисел $(f(k, 0) + P(k), k = 0, \dots, |K| - 1)$, лишь на первый взгляд представляется неоправданно усложненной. В действительности же ее сложность имеет порядок $\mathcal{O}(|K| - 1)$, и это та же самая сложность, что и при общеизвестном способе поиска наименьшего числа в совокупности чисел $f(k, 0) + P(k)$, $k = 0, \dots, |K| - 1$. Существенное же преимущество программы (9.101) состоит в том, что помимо поиска наименьшего числа она создает дополнительные данные, которые позволяют находить следующее наименьшее число не за $(|K| - 1)$ операций, а за существенно меньшее количество операций, равное $\log |K|$. Я покажу, как это происходит.

Предположим, что первое число, записанное в выходной массив f' , было взято из k^* -ой строки массива $f(k, q)$. Это значит, что число $f(k^*, 0) + P(k^*) = f'(0)$ - наименьшее число в совокупности $(f(k, 0) + P(k), k = 0, 1, \dots, |K| - 1)$, которая состоит из $|K|$ чисел. Следующее число $f'(1)$, которое надо будет записать в выходной массив, будет либо число $f(k^*, 1) + P(k^*)$, либо наименьшее из чисел $f(k, 0) + P(k)$,

$k \neq k^*$. Таким образом, числа $f'(0)$ и $f'(1)$ - это наименьшие числа в двух группах чисел, отличающихся лишь одним числом. Поэтому при вычислении числа $f'(1)$ нет необходимости проверять все числа в новой группе, а достаточно вычислять лишь те значения $b(j)$, $\text{ind}(j)$, $j = |K|, \dots, 2(|K| - 1)$, которые могли измениться. Их количество не превышает $\log |K|$, и они записаны в вершинах дерева на пути от листка k^* к корню.

Проиллюстрирую эти рассуждения на рассматриваемом примере. Допустим, что с помощью программы (9.101) были заполнены все данные в дереве, то есть, числа $b(j)$ и стрелки $\text{ind}(j)$ для каждой вершины. Предположим, что эти данные таковы, как показано на рис. 9.6. По совокупности представленных стрелок видно, что наименьшим среди чисел $f(k, 0) + P(k)$, $k = 0, \dots, |K| - 1$, оказалось число $f(3, 0) + P(3)$. Следовательно, число $b(3)$, которое раньше равнялось $f(3, 0) + P(3)$, следует теперь заменить числом $f(3, 1) + P(3)$. Соответственно следует изменить и некоторые данные в вершинах дерева, то есть некоторые из чисел $b(j)$ и $\text{ind}(j)$. Но это придется сделать только в $3 = \log 8$ вершинах с номерами 9, 12 и 14. Число $b(9)$ изменится на $\min(b(2), b(3))$, в вершину 12 будет записано число $\min(b(9), b(8))$, и наконец, число $\min(b(12), b(13))$ будет записано в корень дерева. Соответствующим образом будут определены новые значения указателей $\text{ind}(j)$ в вершинах $j = 9, 12, 14$.

Опишу теперь общий случай. Предположим, что программа уже записала q чисел в выходной массив f' , и это были q наименьших чисел в совокупности чисел $(f(k, q) + P(k), k = 0, 1, \dots, |K| - 1, q = 0, 1, \dots, d - 1)$. Предположим также, что алгоритм располагает следующими данными. Это числа $q^*(k)$, $k = 0, \dots, |K| - 1$, которые указывают, что из k -ой строки массива $f(k, q)$ уже изъяты $q^*(k)$ наименьших чисел и записаны в соответствующие листки дерева. Таким образом, в данный момент в листьях дерева записаны числа $b(k)$, которые равны числам $f(k, q^*(k) - 1) + P(k)$, $k = 0, 1, \dots, |K| - 1$, соответственно. Допустим также, что числа $b(j)$ и указатели $\text{ind}(j)$, $j = |K|, |K| + 1, \dots, 2(|K| - 1)$, во всех остальных вершинах дерева сформированы так, что они соответствуют числам, записанным в листьях.

Следующее число, наименьшее в оставшейся части совокупности $(f(k, l) + P(k), k = 0, 1, \dots, |K| - 1, l = q^*(k), q^*(k) + 1, \dots, d - 1)$, то есть число $f'(q)$ в выходном массиве, находится по алгоритму 9.2.

Algorithm 9.2 Поиск очередного наименьшего числа.

1. Необходимо определить, из какого листка было взято число $f'(q - 1)$, которое было последним записано в выходной массив,

$$\left. \begin{aligned} k^* &= \text{ind}(2(|K| - 1)); \\ \text{while } (k^* \geq |K|) &k^* = \text{ind}(k^*); \end{aligned} \right\} \quad (9.102)$$

После выполнения этого фрагмента программы число k^* дает ответ на этот вопрос. А именно, число $f'(q - 1)$, последним записанное в выходной массив f' , есть число $f(k^*, q^*(k^*) - 1) + P(k^*)$.

2. В k^* -ый листок дерева записывается очередное число из k^* -ой строки массива $f(k, q)$, одновременно добавляя к нему число $P(k^*)$.

$$\left. \begin{aligned} b(k^*) &= f(k^*, q^*(k^*)) + P(k^*); \\ q^*(k^*) &= q^*(k^*) + 1; \end{aligned} \right\} \quad (9.103)$$

3. Данные во всех остальных вершинах дерева приводятся в соответствие с новым содержимым в k^* -ом листке дерева.

$$\left. \begin{aligned} \text{while } (k^* \neq 2(|K| - 1)) \{ \\ \quad j1 = k^* \text{ xor } 1; \quad j2 = |K| + k^*/2; \\ \quad \text{if } (b(k^*) \leq b(j1)) \\ \quad \quad \{ b(j2) = b(k^*); \text{ind}(j2) = k^*; \} \\ \quad \text{else} \\ \quad \quad \{ b(j2) = b(j1); \text{ind}(j2) = j1; \} \\ \quad k^* = j2; \\ \} \end{aligned} \right\} \quad (9.104)$$

4. Число из корня дерева записывается в качестве q -ого числа выходного массива f' , и число q увеличивается на 1.

Операция $k^* \text{ xor } 1$ в алгоритме 9.2 обозначает инверсию младшего разряда в двоичном представлении неотрицательного числа k^* . Под операцией $k^*/2$ имеется в виду целочисленное деление, при котором не учитывается младший разряд в двоичном представлении k^* . Число $j2$ - это номер вершины, соединенной с вершиной номер k^* ребром и лежащей на пути от вершины k^* к корню дерева. Число $j1$ - это номер второй вершины, соединенной с вершиной $j2$ ребром и такой, что путь от $j1$ к корню ведет через $j2$. Смысл индексов k^* , $j1$ и $j2$ показан на рис. 9.7.

При каждом выполнении алгоритма 9.2 в выходной массив дописывается новое число. Одновременно с этим формируются данные для поиска следующего числа, которое надо будет записать в выходной массив. Сложность однократного выполнения алгоритма 9.2 определяется количеством вхождений в циклы в фрагментах (9.102) и (9.104). Это количество всегда равно $\log |K|$. Это и есть порядок сложности формирования одного (но не первого) числа в выходном массиве f' . Формирование всего выходного массива длиной d , то есть вычисление (9.97), состоит из вычисления первого элемента выходного массива, имеющего сложность $|K|$, и вычисления $(d - 1)$ остальных элементов. Общая сложность вычисления (9.97) имеет, следовательно, порядок $\mathcal{O}(|K| + (d - 1) \log |K|)$. Вычисления (9.97) надо выполнить для каждого значения $k \in K$, и следовательно,

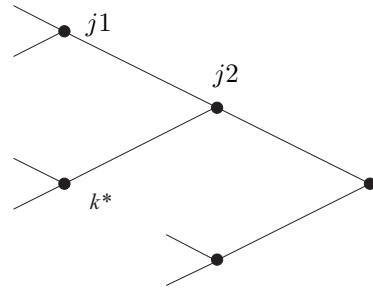


Figure 9.7 Meaning of indices j^* , $j1$ and $j2$ in Algorithm 9.2.

сложность операции $f_i = f_{i-1} \cdot P_i$ имеет порядок $\mathcal{O}(|K|(|K| + (d-1) \log |K|))$. Сложность построения массива (9.96) длиной d имеет порядок

$$\mathcal{O}(|K|^2 n + |K|(\log |K|)n(d-1)). \quad (9.105)$$

Это очень интересный и неожиданный для меня результат. Я ожидал, что поиск d наилучших путей на графе окажется существенно сложнее, чем поиск единственного наилучшего пути. В действительности же все оказалось наоборот. Труднее всего найти именно первый наилучший путь. Построение каждого из оставшихся d путей существенно проще.

Коль скоро Ты так хорошо понимаешь преимущества тщательно продуманной программы перед программой, написанной без больших усилий, советуем Тебе продолжить исследование программы. Если Ты очень пристально посмотришь на выполняемые вычисления, Ты увидишь, что вычислительную сложность можно сделать лучше, чем (9.105), или, более точно, ее порядок можно снизить до уровня

$$\mathcal{O}(|K|^2 n + (\log |K|) n (d-1)). \quad (9.106)$$

Отметим, что число $(\log |K|) n (d-1)$ - это количество бит, необходимое просто для хранения $(d-1)$ кратчайших путей. Это значит, что алгоритм решения задачи со сложностью (9.106) будет алгоритмом того же порядка сложности, что и простое переписывание искомым путей из одной памяти в другую.

Если бы я не имел опыта общения с Вами, мне бы показалось, что дальнейшее снижение вычислительной сложности невозможно. Действительно, матричное произведение

$$\varphi P_1 P_2 \cdots P_n \psi \quad (9.107)$$

невозможно вычислить, минуя n -кратное умножение вектора f на матрицу P , не так ли? Сложность этого умножения уж никак может быть меньше, чем $\mathcal{O}(n C)$, где C - сложность вычисления произведения $f P$. Результатом умножения f на P являются $|K|$ массивов длины d . Следовательно, $|K| d$ чисел нужно получить и запомнить. В силу этого, в выражении для сложности вычисления произведения $f P$ должна как-то присутствовать величина $\mathcal{O}(|K| d)$, а в выражении для сложности вычисления произведения (9.107) должна содержаться какая-то составляющая, равная $\mathcal{O}(|K| dn)$. Но ничего подобного я не нахожу в приведенной вами оценке сложности (9.106).

Мы покажем Тебе изъян в Твоих доводах, но сначала вернемся к уже рассмотренным процедурам и покажем, как доводы, подобные Твоим, могли бы привести к неоправданному пессимистическому выводу.

И Ты, и мы ясно понимаем, что суммирование массивов a_1 и a_2 имеет сложность $\mathcal{O}(d)$. Эту сложность невозможно снизить, так как речь

идет о вычислении d чисел. Отсюда можно было бы сделать вывод, что сложность вычисления суммы a_1, a_2, \dots, a_n не может быть меньше, чем $\mathcal{O}((n-1)d)$, потому что (ВНИМАНИЕ! СЛЕДУЕТ ОШИБКА!) сумму n массивов невозможно вычислить иначе, как вычислив сначала сумму двух массивов, затем добавив к полученному массиву третий, затем четвертый и так далее, пока, наконец, не будет выполнено $n-1$ сложений.

Но ведь только что Ты написал прекрасный алгоритм, вычисляющий сумму $\bigoplus_{k=1}^n a_k$ не за $\mathcal{O}((n-1)d)$ операций, а с существенно меньшими затратами, равными $\mathcal{O}((n-1) + (d-1) \log n)$. Даже если Ты просмотришь свой алгоритм самым дотошным способом, ты не найдешь в нем ничего, что можно было бы интерпретировать, как вычисление суммы некоторого подмножества слагаемых. Это значит, что Тебе удалось организовать вычисление суммы определенного множества массивов, минуя вычисление суммы какого-либо подмножества этих массивов.

Вернемся теперь к вычислению массива длины d

$$\varphi P_1 P_2 \cdots P_n \psi, \quad (9.108)$$

опираясь на процедуру

$$\left. \begin{aligned} f_0 &= \varphi \\ f_i &= f_{i-1} P_i, \quad i = 1, 2, \dots, n, \end{aligned} \right\} \quad (9.109)$$

которую мы уже достаточно хорошо понимаем. Произведение (9.108) есть $f_n \psi$, то есть массив длиной d

$$\bigoplus_{k \in K} f_n(k) \otimes \psi(k), \quad (9.110)$$

где каждый из массивов $f_n(k)$ и $\psi(k)$, $k \in K$, имеет длину d . Ты исследовал эти вычисления и пришел к правильному выводу, что их сложность имеет порядок $\mathcal{O}(|K| + (d-1) \log |K|)$. Посмотри еще раз внимательно на свой алгоритм и отметь, что для вычисления (9.110) совсем не обязательно знать полностью все $|K|$ массивов $f_n(k)$, $k \in K$. Совокупность этих массивов состоит из $|K|d$ чисел, но только d из них используются для вычисления (9.110). Алгоритм даже не обращается к памяти, где хранятся остальные числа. Следовательно, их можно было бы и не вычислять.

Ты можешь продолжить эти рассуждения. Для вычисления вектора f_n также не требуется располагать исчерпывающей информацией о векторе f_{n-1} . Для получения же частичной информации о векторе f_n , которая необходима для дальнейших вычислений, требуется, повидимому, еще меньше данных о векторе f_{n-1} . Процедура (9.109) является в этом смысле чересчур расточительной. На каждом шаге $i = 1, 2, \dots, n$ вычисляется масса данных на случай, если они понадобятся. Но большая их часть не понадобится. Если Ты спроектируешь алгоритм так, что вычисляются только те данные, которые гарантированно понадобятся, Ты увидишь, что можно

вычислить произведение (9.108), минув вычисления частичных произведений φP_1 , $\varphi P_1 P_2$, $\varphi P_1 P_2 P_3$, и т.д. Постарайся спланировать вычисления так, чтобы данные, которые не понадобятся, не вычислялись вообще.

Я вернусь к графовой интерпретации задачи. Я буду говорить о графе, вершинами которого являются вершины α и β (см. рис. 9.5) и еще $K(n+1)$ вершин вида (k, i) , $k = 0, 1, \dots, K-1$, $i = 0, 1, \dots, n$. Здесь символ K обозначает уже не множество значений переменной k , а их количество, так как под символом k я буду понимать целое число и использовать его в качестве индекса в определенных массивах.

В графе имеются следующие ориентированные ребра (стрелки). В вершине α начинаются K стрелок, заканчивающиеся на вершинах вида $(k, 0)$. Длина стрелки $(\alpha, (k, 0))$ равна $\varphi(k)$. Из каждой вершины вида $(k', i-1)$ выходят K стрелок, которые заканчиваются на вершинах вида (k'', i) . Длина каждой такой стрелки равна $P_i(k', k'')$. Из каждой вершины вида (k, n) выходит единственная стрелка, заканчивающаяся на вершине β и имеющая длину $\psi(k)$.

Меня интересует сложность вычисления длин d кратчайших путей от вершины α до вершины β .

Для анализа задачи и алгоритма ее решения я введу следующее обозначение. Пусть γ обозначает вершину вида (k, i) или вершину β . Символом $C(\gamma)$ я обозначу множество путей от вершины α до вершины γ . Я упорядочу это множество путей по возрастанию их длины и обозначу $c(\gamma, q)$ путь под номером q в этом упорядоченном множестве. Я буду считать, что пути в множестве $C(\gamma)$ пронумерованы, начиная номером 0. Таким образом, кратчайший путь от вершины α до вершины γ есть $c(\gamma, 0)$. Множества $C(\gamma)$ и пути $c(\gamma, q)$ не будут представлены явно в программе, которую я собираюсь построить. Эти объекты вводятся мною не с целью их обработки с помощью программы, а с целью объяснения тех величин, которые в программе создаются и трансформируются.

Данные, которые явно присутствуют в программе, разделены на группы, соответствующие вершинам γ . Смысл этих данных один и тот же как для вершин вида (k, i) , так и вершины β . Я опишу достаточно подробно данные и правила их преобразования только для вершин вида (k, i) . Затем менее подробно я определю данные для вершины β и надеюсь, что это будет ясно без подробных объяснений.

Наиболее важной группой данных является совокупность чисел $f(k, i, q)$, каждое из которых обозначает длину пути $c(k, i, q)$, то есть q -ого пути от вершины α до вершины (k, i) . Число $f(\beta, q)$ обозначает длину q -ого пути от α до β . Для фиксированного i совокупность чисел $f(k, i, q)$ есть как раз вектор-строка f_i в алгебраическом представлении задачи, а совокупность чисел $f(\beta, 0), f(\beta, 1), \dots, f(\beta, d-1)$ есть массив, который представляет произведение (9.109). В ситуации $q \geq |C(k, i)|$ я считаю, что $f(k, i, q) = \infty$.

Для каждой вершины (k, i) и каждого номера q определены указатели $k'(k, i, q)$ и $q'(k, i, q)$, имеющие следующий смысл: путь $c(k, i, q)$ состоит из пути $c(k'(k, i, q), i-1, q'(k, i, q))$ в конец которого добавлена вершина (k, i) .

Посредством указателей $k'(k, i, q)$ и $q'(k, i, q)$ можно построить любой путь $c(k, i, q)$. Например, q -ый путь от α до β есть

$$\alpha, (k_1, 1), (k_2, 2), \dots, (k_i, i), \dots, (k_n, n), \beta,$$

где

$$k_n = k'(\beta, q), q_n = q'(\beta, q), \quad k_{i-1} = k'(k_i, i, q_i), \quad q_{i-1} = q'(k_i, i, q_i).$$

Алгоритм, который я опишу, последовательно формирует указанные данные так, что на каждом шаге данные $f(k, i, q)$, $k'(k, i, q)$, $q'(k, i, q)$ сформированы не для всех возможных троек (k, i, q) индексов, а только для некоторых. Какая часть данных уже сформирована, указывают числа $q^*(k, i)$. Для каждой вершины (k, i) они указывают, что данные $f(k, i, q)$, $k'(k, i, q)$, $q'(k, i, q)$ уже сформированы для $q^*(k, i)$ кратчайших путей от вершины α до (k, i) .

Кроме указанных данных, для каждой вершины (k, i) формируются дополнительные данные, которые буду называть деревом вершины (k, i) . Эти данные имеют ту же структуру и тот же смысл, как это было представлено на рис. 9.6. Вершины дерева (k, i) соответствуют тройкам (k, i, j) , $j = 0, 1, \dots, 2(K-1)$, причем в каждой вершине записаны число $b(k, i, j)$ и указатель $\text{ind}(k, i, j)$. Корень дерева (k, i) соответствует тройке $(k, i, 2(K-1))$. Число $b(k, i, 2(K-1))$ в корне есть длина $f(k, i, q^*(k, i))$ длинейшего из всех уже сформированных путей, то есть $q^*(k, i)$ -ый кратчайший путь в множестве $C(k, i)$ путей от α до (k, i) . Листья дерева (k, i) соответствуют тройкам (k, i, j) , $j = 0, \dots, K-1$. В них записаны числа $b(k, i, 0), b(k, i, 1), \dots, b(k, i, K-1)$ и указатели $\text{ind}(k, i, 0), \text{ind}(k, i, 1), \dots, \text{ind}(k, i, K-1)$. Число $b(k, i, k')$ равно $f(k', i-1, q) + P_i(k', k)$, где номер q записан в указателе $\text{ind}(k, i, k')$.

Числа $b(k, i, j)$ и указатели $\text{ind}(k, i, j)$ во всех прочих вершинах имеют тот же смысл, который я указал ранее на примере рис. 9.6. Только указатели $\text{ind}(k, i, j)$ в листьях дерева имеют несколько иной смысл.

Структура данных конструируемого алгоритма представляется слишком громоздкой. Однако так это кажется только на первый взгляд. Данные разбиты на две группы. Первая состоит из чисел $f(k, i, q)$, $k'(k, i, q)$ и $q'(k, i, q)$. Память, требуемая для их хранения, зависит от чисел $q^*(k, i)$, потому что она пропорциональна сумме $\sum_{k,i} q^*(k, i)$. Позже я покажу, что сумма $\sum_{k=0}^{K-1} q^*(k, i)$ не превосходит число $q^*(\beta)$. Число $q^*(\beta)$ показывает, сколько крайчайших путей из вершины α до вершины β уже построено. Следовательно, общий объем памяти для хранения чисел $f(k, i, q)$, $k'(k, i, q)$ и $q'(k, i, q)$ не превышает сумму $\sum_{i=0}^n q^*(\beta) = (n+1)q^*(\beta)$. Это объем памяти того же порядка, что и память, необходимая просто для хранения конечных результатов работы программы, то есть информации об определенной совокупности кратчайших путей на графе.

Другая группа данных сформирована в виде совокупности деревьев, соответствующих множеству вершин графа. Общий объем этих данных имеет порядок $\mathcal{O}(K^2 n)$. Тот же порядок имеет объем памяти для хранения совокупности чисел $P_i(k', k)$, представляющих исходные данные задачи. Таким

образом, для реализации алгоритма потребуется память того же порядка, что и суммарная память для хранения исходных данных алгоритма и его результатов. Существенно снизить эту память вряд ли возможно: ведь исходные данные алгоритма и его результаты где-то хранить надо.

Совсем другое дело, что эти данные весьма разнородны и их трудно охватить единым взглядом. Но это уже забота программиста, написать программу так, чтобы все время поддерживалась правильная структура данных и на каждом шаге работы алгоритма эти данные имели тот смысл, который они должны иметь.

Только наблюдая эти разнообразные данные, я смог по настоящему оценить изящество алгебраической конструкции, использованной до этого. Именно она предоставляла мне возможность охватить на макроуровне весь алгоритм в целом, увидеть его принципиально важные черты, а не ковыряться в бесчисленных незначительных мелочах. Несмотря на то, что эти мелочи игнорировались, задачей удалось овладеть и построить алгоритм, который в конечном итоге оказался не таким уж плохим. Сейчас, когда я хочу выжать из алгоритма все, что можно, мне приходится влезать в те мелочи, которые на макроуровне не просматривались. Однако я бы никогда не справился с этой кучей мелочей, если бы не располагал твердой базой в виде решения задачи на алгебраическом макроуровне.

Наконец наше терпение вознаграждено, и Ты понял преимущества и плодотворность алгебраических представлений задач, с которыми мы имеем дело. Хотя, может быть, сейчас последует это злополучное "однако"?

Нет, на этот раз не последует.

Я опишу вспомогательный алгоритм, который будет наиболее существенной составной частью алгоритма. Я назову этот алгоритм $NEXT(k, i)$ и определю его следующим образом. Алгоритм меняет данные таким образом, что число $q^*(k, i)$ увеличивается на 1, а затем все остальные данные приводятся в соответствие с этим изменением. До начала работы алгоритма $NEXT(k, i)$ все данные были сформированы так, что они содержали всю информацию о q кратчайших путях от вершины α до вершины (k, i) . После окончания работы алгоритма $NEXT(k, i)$ все данные изменены так, что теперь они предоставляют информацию о $(q + 1)$ кратчайших путях от α до (k, i) .

Прототипом для алгоритма $NEXT(k, i)$ послужит тройка фрагментов (9.102), (9.103) и (9.104) программы для построения массива $\bigoplus_{k \in K} f(k) \otimes P(k)$, где $f(k)$ и $P(k)$ - массивы. Эти фрагменты надо модифицировать так, чтобы в них учитывалось, что алгоритм $NEXT(k, i)$ должен трансформировать данные в дереве, соответствующем именно вершине (k, i) . Алгоритм должен также предусматривать действия, которые он должен выполнить, когда те или иные необходимые данные еще не вычислены.

Algorithm 9.3 $NEXT(k, i)$

1. Прежде всего необходимо определить, какое число из массива $f(k', i-1, q)$ использовалось при формировании числа $f(k, i, q^*(k, i) - 1)$ - длины $q^*(k, i)$ -ого пути от α до (k, i) . В программе-прототипе для этого служила команда (9.102). Сейчас нет необходимости в таких вычислениях, потому что необходимая информация хранится в указателях $k'(k, i, q^*(k, i) - 1)$ и $q'(k, i, q^*(k, i) - 1)$. Вместо команды (9.102) выполняется просто

$$k^* = k'(k, i, q^*(k, i) - 1); \quad q^* = q'(k, i, q^*(k, i) - 1).$$

Указатели k^* и q^* обозначают, что число $f(k, i, q^*(k, i))$ было вычислено как сумма чисел $f(k^*, i-1, q^*)$ и $P_i(k^*, k)$.

2. Сейчас следует получить длину $f(k^*, i-1, q^*)$ очередного, $(q^* + 1)$ -ого пути из множества $C(k^*, i-1)$, добавить к нему $P_i(k^*, k)$ и поместить полученную сумму в k^* -ый листок (k, i) -ого дерева.

$$\left. \begin{array}{l} \text{if } (q^*(k^*, i-1) = q^* + 1) \text{ NEXТ}(k^*, i-1); \\ b(k, i, k^*) = f(k^*, i-1, q^* + 1) + P_i(k^*, k); \end{array} \right\} \quad (9.111)$$

Указанный фрагмент программы построен по прототипу (9.103) за исключением первой команды в (9.111), которая отсутствовала в прототипе (9.103). Число $f(k^*, i-1, q^* + 1)$, необходимое для выполнения второй команды в (9.111), возможно, еще не получено. Это происходит, когда $q^* + 2 > q^*(k^*, i-1)$. Однако до выполнения команды (9.111) число $f(k^*, i-1, q^*)$ уже было вычислено. Это значит, что $q^*(k^*, i-1) \geq q^* + 1$. Таким образом, если число $f(k^*, i-1, q^* + 1)$ еще не вычислено, то $q^*(k^*, i-1)$ в точности равно $q^* + 1$. Чтобы вычислить число $f(k^*, i-1, q^* + 1)$, достаточно один раз запустить программу NEXТ($k^*, i-1$), что и предписывается первой строчкой в (9.111).

3. Данные в (k, i) -ом дереве следует привести в соответствие с изменившимся числом $b(k, i, k^*)$ в листке дерева. Это выполняется с помощью программы, которая почти не отличается от прототипа (9.104).

$$\left. \begin{array}{l} \text{while } (k^* \neq 2(K-1)) \{ \\ \quad j1 = k^* \text{ xor } 1; \quad j2 = K + k^*/2; \\ \quad \text{if } (b(k, i, k^*) \leq b(k, i, j1)) \\ \quad \quad \{ b(k, i, j2) = b(k, i, k^*); \text{ ind}(k, i, j2) = k^*; \} \\ \quad \text{else} \\ \quad \quad \{ b(k, i, j2) = b(k, i, j1); \text{ ind}(k, i, j2) = j1; \} \\ \quad \quad k^* = j2; \\ \} \end{array} \right\} \quad (9.112)$$

4. Число $f(k, i, q^*(k, i) + 1)$ и указатели $k'(k, i, q^*(k, i) + 1)$ и $q'(k, i, q^*(k, i) + 1)$, характеризующие только что определенный путь $s(k, i, q^*(k, i) + 1)$, необходимо сохранить в отведенной для них памяти. Число $f(k, i, q^*(k, i) + 1)$ равно числу $b(k, i, 2(K-1))$, уже сформированному для корня k, i -ого дерева. Указатели $k'(k, i, q^*(k, i) + 1)$ и $q'(k, i, q^*(k, i) + 1)$ формируются с помощью программы, постоянной по прототипу (9.102).

$$\left. \begin{array}{l} q^*(k, i) = q^*(k, i) + 1; \\ j^* = 2(K-1); \\ f(k, i, q^*(k, i)) = b(k, i, j^*); \\ \text{while } (j^* \geq K) \quad j^* = \text{ind}(k, i, j^*); \\ k'(k, i, q^*(k, i)) = j^*; \quad q'(k, i, q^*(k, i)) = \text{ind}(k, i, j^*); \end{array} \right\} \quad (9.113)$$

Важное свойство алгоритма $\text{NEXT}(k, i)$ состоит в том, что при его выполнении алгоритм $\text{NEXT}(k', i - 1)$ вызывается не более одного раза (а может быть, и не вызывается вообще). При этом, если алгоритм $\text{NEXT}(k', i - 1)$ вызывается, то он вызывается для единственного значения k' . Это значит, что вычисления по алгоритму $\text{NEXT}(k, i)$ состоят из вычислений, сложность которых имеет порядок $\mathcal{O}(\log K)$ (в силу цикла `while` в командах (9.112), (9.113)), и вычислений по алгоритму $\text{NEXT}(k', i - 1)$ для некоторого единственного значения k' . Вычисления по алгоритму $\text{NEXT}(k', i - 1)$ состоят из вычислений, сложность которых имеет порядок $\mathcal{O}(\log K)$, и возможно, вычислений по алгоритму $\text{NEXT}(k'', i - 2)$ для некоторого единственного значения k'' . Так как функция $\text{NEXT}(k, 0)$ не вызывается ни разу, суммарные вычисления по алгоритму $\text{NEXT}(k, i)$ имеют сложность, которая не превышает $\mathcal{O}(i \log K)$.

Незначительно модифицируя алгоритм $\text{NEXT}(k, i)$, можно переписать его в виде, при котором он будет определен не только для вершин вида (k, i) , но и для вершины β . Для этого в программах (9.111), (9.112) и (9.113) следует везде вместо (k, i) написать β , а вместо $i - 1$ написать n . Сложность алгоритма $\text{NEXT}(\beta)$ имеет порядок, который не превышает $\mathcal{O}(n \log K)$.

До первого вызова программы $\text{NEXT}(\beta)$ все данные должны быть инициализированы так, как это указывает следующая программа

Algorithm 9.4 Инициализация данных до запуска программы $\text{NEXT}(\beta)$.

1. Для вершин вида $(k, 0)$ числа $q^*(k, 0)$ равны 1, что обозначает, что имеются сведения о двух кратчайших путях от α до $(k, 0)$. Длина первого равна $f(k, 0, 0) = \varphi(k)$, длина второго равна $f(k, 0, 1) = \infty$, что обозначает, что второй путь от α до $(k, 0)$ отсутствует вообще. Все остальные данные для вершин вида $(k, 0)$ не определены.
 2. Для всех прочих вершин устанавливается $q^*(\gamma) = 0$, что обозначает, что имеются сведения только о единственном кратчайшем пути от α до γ , длина которого запомнена как $f(\gamma, 0)$. Числа $f(\gamma, q)$ для $q > 0$ не определены.
 3. Указатель $k'(k, i, 0)$ равен $\text{argmax}_{k'} (f(k', i - 1, 0) + P_i(k', k))$, а указатель $q'(k, i, 0)$ равен 0.
 4. Указатель $k'(\beta, 0)$ равен $\text{argmax}_{k'} f(k', n, 0)$, а указатель $q'(\beta, 0)$ равен 0.
 5. Числа $b(k, i, k')$ в листках (k, i) -ого дерева устанавливаются равными $f(k', i - 1, 0)$. Указатели $\text{ind}(k, i, k')$ устанавливаются равными 0 для всех $k' = 0, 1, \dots, K - 1$. Информация во всех остальных вершинах дерева устанавливается в соответствии с данными, записанными в листках.
 6. Инициализация данных в дереве, соответствующем вершине (β) , выполняется подобно инициализации деревьев, соответствующих вершинам вида (k, i) .
-

Указанная инициализация состоит фактически из всех тех действий, которые надо выполнить при отыскании единственного кратчайшего пути от α до β . Однако, одновременно с поиском этого кратчайшего пути сохраняются все промежуточные данные в таком виде, который обеспечивает

эффективное выполнение программы NEXТ. Сложность инициализации имеет порядок $\mathcal{O}(K^2 n)$.

Таким образом я прихожу к результату, что поиск d кратчайших путей от α до β состоит из инициализации, сложность которой имеет порядок $\mathcal{O}(K^2 n)$, и $(d - 1)$ -кратного запуска программы NEXТ(β), порядок сложности которой не превышает $\mathcal{O}(n \log K)$. Общая сложность решения задачи не превышает

$$\mathcal{O}(K^2 n + (d - 1) n \log K).$$

Мне осталось только закрыть вопрос, который я лишь бегло затронул раньше. Объем памяти, необходимый для хранения данных $f(k, i, q)$, $k'(k, i, q)$ и $q'(k, i, q)$, имеет порядок $\sum_k \sum_i (q^*(k, i) + 1)$. Располагая теперь конкретным алгоритмом формирования чисел $q^*(k, i)$, можно утверждать, что непосредственно после инициализации сумма $\sum_k \sum_i (q^*(k, i) + 1)$ несущественно отличается от Kn . Далее, при каждом применении программы NEXТ(β) эта сумма увеличивается не более, чем на n , потому что для каждого i число $q^*(k, i)$ увеличивается только для какого-то одного значения k , причем увеличивается не более, чем на 1. Отсюда следует, что после d -кратного выполнения программы NEXТ(β) суммарный объем памяти для хранения данных $f(k, i, q)$, $k'(k, i, q)$ и $q'(k, i, q)$ не будет превышать $\mathcal{O}(Kn + nd)$.

Пожалуй, это все о поиске d кратчайших путей на графе, а следовательно, о поиске d наилучших аппроксимаций заданной последовательности последовательностями из заданного регулярного языка. Естественно, описанные процедуры можно обобщить на случай, когда речь идет не только о последовательностях, а о функциях, определенных на вершинах ациклических графов.

Алгоритм поиска d кратчайших путей на графе постепенно приобретает приличный вид. Ты бы мог уже его опубликовать, потому что он имеет ценность сам по себе, а не только применительно к нашим задачам распознавания. В целом мы видим, что Ты вполне прилично справился с материалом двух последних лекций.

Анализ последней задачи был для меня очень поучительным, потому что показал, насколько сильно отличается тщательно продуманный алгоритм от алгоритма, который первый приходит в голову и поначалу представляется единственно возможным.

Позвольте мне сказать откровенно, что в одном месте вашей лекции я увидел явное пренебрежение к оценке вычислительной сложности. Речь идет об оценке сложности вычисления матричного полинома $\bigoplus_{i=0}^{\infty} A^i$ в полукольце с идемпотентным сложением. Без каких-либо сомнений вы утверждали, что для матрицы A размера $k \times k$ сложность вычисления этого полинома имеет порядок $\mathcal{O}(k^3 \log k)$. Но ведь это сложность самого простого алгоритма, который первый приходит в голову. Уже долгие годы известен и пользуется популярностью алгоритм Флойда [Floyd, 1962], решающий задачу за время порядка $\mathcal{O}(k^3)$. Как мне тут связать одно с другим?

Только так, что простишь нас за пренебрежение этим вопросом в том месте лекции, о котором Ты говоришь. Нам не хотелось отвлекаться и прерывать целеустремленное движение к основной цели лекции.

Спасибо, что Вы оцениваете мое понимание материала как вполне удовлетворительное. Не стану имитировать скромность и соглашусь, что в рассмотренном разделе чувствую себя достаточно свободно. Больше меня тревожит то, что я вижу ощутимый разрыв между тем, что я узнал, и теми трудностями, с которыми я встечаюсь на первых же шагах решения практической задачи распознавания текстов. Если Вы мне позволите высказаться без обиняков, как Вы мне и раньше позволяли, то мне представляется, что теоретические результаты последних двух лекций не имеют ничего общего с практическими задачами, которые мне непременно нужно решить.

Попробуй сформулировать свою мысль более конкретно.

Это займет некоторое время, потому что мой вопрос невозможно выразить кратко.

После лекции 7 мы тщательно обсудили вопросы распознавания стандартных алфавитно-цифровых символов. В конце нашего обсуждения вы предостерегли меня, что при переходе от распознавания отдельных символов к распознаванию текста меня ожидают дополнительные серьезные трудности. Вы обещали, что ключи к преодолению этих трудностей я найду в лекциях 8 и 9. Ваши предсказания сбылись, но только частично. С трудностями я действительно встретился, но материал последних двух лекций не помогает их преодолеть. А я ведь в нем вполне удовлетворительно разбираюсь, судя по вашей оценке. У меня складывается впечатление, что в лекциях говорится о чем-то одном, правда, очень интересном, а мои практические трудности - это что-то совсем другое.

Прекрати толочь воду в ступе. Ты уже повторяешься.

В лекциях говорится о том, каким образом на основании последовательности x_1, x_2, \dots, x_n изображений найти последовательность k_1, k_2, \dots, k_n имен букв, которые на этих изображениях представлены, используя при этом априорные сведения о взаимной зависимости соседних букв в строке.

Эта модельная ситуация отличается от того, что я имею в конкретной прикладной задаче. Входные данные для распознавания совсем не имеют вид последовательности x_1, x_2, \dots, x_n , где x_i - изображение i -ой буквы с именем k_i . Входная информация представлена в виде единого несегментированного изображения x , на котором представлена вся текстовая строка. В этих входных не содержится прямых указаний, где заканчивается одна буква и начинается другая. Чтобы использовать рекомендации лекции, мне нужно сначала как-то разделить изображение строки на части, соответствующие отдельным буквам. Но это чуть ли не основная трудность во

всем технологическом цикле распознавания текста, и с этой трудностью я остаюсь опять один на один.

Предположим, однако, что я с этой трудностью как-то справился. Что же полезного я могу извлечь теперь из материала двух последних лекций? Только то, что могу использовать знания о взаимной зависимости соседних букв в строке для улучшения результатов распознавания. Однако структурные зависимости отдельных знаков в естественном (и даже в искусственном) языке значительно сложнее, чем те, которые можно выразить с помощью таких примитивных средств, как регулярные языки и их стохастические обобщения. Использование этих простейших моделей языка может привести к заметному улучшению только в том случае, если результаты изолированного распознавания были очень плохими. Однако в этом случае даже улучшенные результаты не окажутся достаточно хорошими.

На основании идей, о которых вы рассказали после лекции 7, я написал программу, которая вполне прилично распознавала изолированные изображения букв. При распознавании же текстовых строк ошибки появлялись совсем не потому, что не учитывались статистические зависимости между буквами, а в основном из-за плохой сегментации изображения слитной строки на отдельные буквы. Для сегментации я использовал алгоритм, который я разработал на основании собственных разумных соображений, а не на основании тех или иных теоретических рекомендаций.

Все это в целом мне напоминает ситуацию, когда в рекламе какого-то кафе мне обещают, что мне дадут любую еду, какую бы я ни пожелал. На проверку же оказывается, что это действительно так, но только при условии, что я сам себе эту еду приготовлю. Обещание вроде бы формально выполнено, но в действительности все это - сплошной обман.

Таким образом, я вижу существенный пробел между изложенной теорией и моими практическими задачами. Теория рекомендует, как следует улучшить качество распознавания на основе взаимной зависимости знаков в строке. Но мне его улучшать не надо. При распознавании изолированных изображений я достигаю такое высокое качество распознавания, что учет зависимости между знаками дает едва-едва заметное улучшение. Основная трудность состоит в том, чтобы исходное изображение x строки преобразовать в последовательность x_1, x_2, \dots, x_n изображений, каждое из которых соответствует одной букве. По этому поводу теория молчит, потому что предполагается, что последовательность x_1, x_2, \dots, x_n уже задана, и считается, что это уже мои проблемы, откуда брать эту последовательность.

Вижу ли я этот пробел правильно? Можете ли вы мне помочь именно с проблемой сегментации текстовой строки? Именно эта часть практической задачи мне представляется ключевой, а совсем не учет статистической зависимости между знаками.

Здесь действительно можно усмотреть определенный зазор между теорией и практикой, но только в силу того, что теория понимается несколько сужено. В Твоем случае это сужение произошло тогда, когда Ты принял, что алфавит K состояний должен непременно состоять из наименований

букв. При такой интерпретации множества состояний немедленно получается, что исходные для распознавания данные должны быть представлены в виде последовательности x_1, x_2, \dots, x_n , где x_i есть часть изображения, занятая i -ой буквой. Но для применения теоретических рекомендаций совсем необязательно именно таким образом понимать последовательности k_1, k_2, \dots, k_n и x_1, x_2, \dots, x_n .

Исходные данные представлены в виде двумерного массива $(x(i, j), 1 \leq i \leq n, 1 \leq j \leq m)$, который состоит из n столбцов и m строк. Как обычно, сигнал $x(i, j)$ понимается как яркость изображения в точке с координатами (i, j) . Эти данные можно понимать как последовательность x_1, x_2, \dots, x_n , где x_i - одномерный массив длиной m . Попросту говоря, это i -ый столбец исходного двумерного массива.

Но в этом случае получается, что множество X значений сигнала x_i необозримо велико.

Да, это так. Но пока что пусть это Тебя не тревожит. Сейчас важно установить лишь, что этот многомерный сигнал x_i зависит лишь от двух величин, принимающих вполне обозримое количество значений. Первой величиной является число q , которое обозначает, что изображение буквы, которой принадлежит i -ый столбец, началось в столбце $i - q$. Второй величиной является имя k буквы, которой принадлежит i -ый столбец. Для каждого имени $k \in K$ определена ширина $Q(k)$ - количество столбцов в изображении буквы с именем k . Пара (k, q) , от которой зависит столбец в двумерном массиве $(x(i, j), 1 \leq i \leq n, j \leq m)$, принимает значения из множества $\{(k, q) \mid k \in K, q = 0, 1, \dots, Q(k) - 1\}$. Это и есть множество состояний автомата, генерирующего изображению строки.

Кажется, я уже понимаю. Главная идея состоит в том, что состояния автомата совсем необязательно должны соответствовать тому параметру, который должен определяться в результате распознавания, а может представлять нечто более подробное. Я помню, что мы уже обсуждали это после лекции 7.

А если Тебе уже все понятно, то продолжай дальше сам.

Автомат проходит $(k_1, q_1), (k_2, q_2), \dots, (k_n, q_n)$ состояний и генерирует последовательность e_1, e_2, \dots, e_n столбцов некоторого (идеального) изображения. Каждый столбец состоит из m элементов. Последовательность e_1, e_2, \dots, e_n формирует двумерный массив $(e(i, j), i = 1, 2, \dots, n, j = 1, 2, \dots, m)$, который понимается как изображение. Чтобы это изображение было идеальной, неискаженной помехами текстовой строкой, должны выполняться определенные ограничения на состояние (k_i, q_i) в i -ый момент в зависимости от состояния (k_{i-1}, q_{i-1}) в предыдущий момент. Равным образом, должно быть указано, как сигнал e_i зависит от состояния (k_i, q_i) . Ради простоты я сейчас игнорирую взаимную зависимость имен

букв в строке и рассматриваю лишь механизм генерирования идеальных изображений произвольных текстовых строк. Он может быть, например, следующим.

1. Множество начальных состояний автомата - это множество $\{(k, 0) \mid k \in K\}$, которое выражает тот вполне понятный факт, что генерирование текстовой строки начинается с того, что генерируется самый левый (нулевой) столбец изображения некоторой буквы.
2. Множеством конечных состояний является $\{(k, Q(k) - 1) \mid k \in K\}$. Это значит, что генерирование изображения строки может закончиться только в те моменты времени, когда полностью сгенерировано изображение некоторой буквы.
3. Если автомат находится в состоянии (k, q) , $k \in K$, $q \neq Q(k) - 1$, (то есть, когда изображение буквы с именем k еще не полностью сгенерировано), следующим может быть только состояние $(k, q+1)$ (то есть, должно продолжаться генерирование изображения текущей буквы с тем же именем k).
4. Если автомат находится в состоянии $(k, Q(k) - 1)$, $k \in K$, (то есть, когда закончено генерирование изображения буквы с именем k), автомат может либо прекратить генерирование изображения строки, либо перейти в состояние $(k', 0)$, $k' \in K$ (то есть начать генерирование изображения очередной буквы).
5. Если автомат в i -ый момент находится в состоянии (k_i, q_i) , то к ранее построенному массиву $e(i', j)$, $i' = 1, 2, \dots, i - 1$, $j = 1, 2, \dots, m$, он дописывает столбец e_i , который зависит от имени k_i текущей буквы и номера q_i столбца в изображении этой буквы.

Обозначим $E(k, q)$ множество всех изображений q -ого столбца в букве с именем k . Множество $E(k, q)$ не обязательно должно быть очень обширным, так здесь речь идет об идеальных, не искаженных помехами изображениях. В простейших случаях это множество может состоять всего лишь из одного изображения. Разнообразие реальных изображений q -ого столбца буквы k можно представить как расширение множества $E(k, q)$, так и введением функции $d(k, q, x)$, определяющей, в какой степени реальное изображение x представляет q -ый столбец буквы k .

Введенные понятия служат основой для формулировки задачи, решением которой является как сегментация массива $(x(i, j), 1 \leq i \leq n, 1 \leq j \leq m)$ на изображения, соответствующие отдельным буквам, так и распознавание этих букв. Задача формулируется, как отыскание последовательности $(k_1^*, q_1^*), (k_2^*, q_2^*),$

$$((k_i^*), i = 1, 2, \dots, n) = \underset{((k_i, q_i), i=1, 2, \dots, n)}{\operatorname{argmax}} \sum_{i=1}^n d(k_i, q_i, x_i),$$

при условиях

$$\begin{aligned} (k_1, q_1) &\in \{(k, 0) \mid k \in K\}; \\ (k_n, q_n) &\in \{(k, Q(k) - 1) \mid k \in K\}; \\ k_i &= k_{i-1}, \quad q_i = q_{i-1} + 1, && \text{если } q_{i-1} < Q(k_{i-1}) - 1; \\ k_i &= K, \quad q_i = 0, && \text{если } q_{i-1} = Q(k_{i-1}) - 1. \end{aligned}$$

Полученная последовательность (k_i^*, q_i^*) , $i = 1, \dots, n$, определяет последовательность i_1, i_2, \dots, i_M индексов, для которых $q_{i_m}^* = 0$. Индекс i_m есть горизонтальная координата, в которой начинается m -ая буква в строке, $k_{i_m}^*$ есть имя этой буквы, а M - количество букв в распознанной строке.

Пожалуй, для начала годится, но только для начала. Насколько мы Тебя знаем, через некоторое время Ты сможешь предложить более продуманный алгоритм.

Июль 1998

9.7 Связь с TOOLBOX"ом

TOOLDBOX, реализующий описанные в данной лекции алгоритмы распознавания, разработал П.Соукуп в процессе дипломной работы весной 2001 года. TOOLBOX доступен по адресу http://cmp.felk.cvut.cz/cmp/cmp_software.html, равно как и исходные тексты в языке C.

9.8 Библиографические замечания

Лекция основана на формальных грамматиках и языках в смысле Хомского [Chomsky, 1957; Chomsky et al., 1971]. Размытые модификации языков и грамматик основаны на понятиях размытых множеств [Zadeh, 1965; Zimmermann et al., 1984].

Обширная группа работ по структурному распознаванию основана на вычислении левенштейнова отличия [Levenstein, 1965] распознаваемого объекта от того или иного заданного эталона. Известны попытки обобщения левенштейновых метрик на случай, когда речь идет о вычислении расстояния не между последовательностями, а между объектами с более сложной структурой, например, графами [Levenstein, 1965; Bunke, 1996]. Кашьяп (Kashyap) и Ооммен (Oommen) [Kashyap and Oommen, 1984; Oommen, 1987] дали статистическую интерпретацию левенштейновых метрик, что привело к новым оригинальным постановкам задач.

Наиболее известные алгоритмы минимизации левенштейновых функций на регулярных и контекстно-свободных языках принадлежат Вагнеру [Wagner and Fischer, 1974; Wagner and Seiferas, 1978]. Эти алгоритмы решают задачу только в случае, когда левенштейновы функции образуют

метрику на множестве последовательностей. Эффективные алгоритмы вычисления левенштейнова расстояния от последовательности до регулярно-го языка описаны в работе [Amengual and Vidal, 1996]. Решение задачи минимизации левенштейновых функций на регулярных языках во всей ее общности описано в данной лекции.

Замечательный алгоритм Флойда для вычисления кратчайших расстояний между вершинами неориентированного графа, о котором упоминает Иржи Пеха в дискуссии, описан в статье [Floyd, 1962].

Иржи Пеха не единственный, кого заинтересовала задача поиска d кратчайших путей на графе. После опубликования чешской версии данных лекций в 1999 году мы узнали о работе [Jimenez and Marzal, 2000], в которой описан алгоритм отыскания d наиболее дешевых выводов данного предложения в данном контекстно-свободном языке. Описанные результаты очень близки к результатам, которые получил Иржи Пеха. Повидимому, еще многих других привлекает эти интересные и далеко не тривиальные задачи.

Алгебраические конструкции для формулировки и решения оптимизационных задач структурного распознавания разработал Шлезингер [Schlesinger, 1989; Schlesinger, 1994; Schlesinger, 1997].

Глава 10

Контекстно-свободные языки, их двумерные обобщения и соответствующие задачи распознавания

10.1 Вводные замечания

Время от времени научная терминология словно издевается над доверчивым читателем, будто намеренно сбивая его с толку. Это происходит, когда тому или иному точному понятию присваивается название, которое используется и в повседневной жизни, но для не очень точных, размытых понятий. Так, теория катастроф не имеет ничего общего с тем, что происходит в повседневной жизни, когда говорят о каком-то неожиданном бедствии. Точно так же математическая теория игр совсем не является формализацией игры на футбольном поле или шахматной доске.

Понятие "контекстно-свободный язык" принадлежит к числу таких зловредных понятий. Если не знать точное определение этого понятия, то набор слов в его названии может сильно дезориентировать. Может показаться, что речь идет о языке, в котором слова выстраиваются друг за другом без учета контекста, образуя некоторую абракадабру из взаимно независимых фрагментов. Это, конечно, совсем не так. Понятие контекстно-свободного языка имеет точное математическое определение, которое мы приведем. Однако догадаться до него невозможно именно из-за выразительного названия, которое в обыденной жизни обозначает нечто совсем другое.

В этой лекции описывается формализм для конструктивного задания множеств изображений и распределений вероятностей на этих множествах. На основе этих данных формулируются и решаются задачи распознавания, подобные рассмотренным в предыдущих лекциях 8 и 9. Отличается же данная лекция от двух предыдущих тем, что объекты распознавания - это не обязательно одномерные последовательности. Это могут быть двумерные или многомерные массивы или иметь еще более общую структуру. Мы увидим, что предложенный формализм является естественным обобщением

контекстно-свободных грамматик и языков в иерархии Н.Хомского, которые в свою очередь являются обобщением ранее рассмотренных регулярных грамматик и языков.

10.2 Неформальное объяснение двумерных грамматик и языков

Представим себе воображаемый диалог человека с искусственно созданным устройством, например, компьютером. Ожидается, что в результате этого диалога будет определено множество изображений, которым можно присвоить определенное имя, скажем, имя Ш. Поскольку буква Ш будет использоваться далее в математических выражениях, то из типографских соображений она будет обозначаться латинскими буквами *SH*. Неформальное понимание, что имеется в виду под буквой Ш, иллюстрируется на рис. 10.1.

Диалог начинается предъявлением некоторого изображения компьютеру, которому задается вопрос, можно ли этому изображению присвоить имя Ш. Компьютер проверяет, содержится ли в его библиотеке программа, отвечающая на такой вопрос. Если да, то он отвечает на вопрос и диалог пользователя с компьютером заканчивается. В противном случае компьютер сообщает пользователю об отсутствии нужной программы, что может интерпретироваться как вопрос "А что такое буква Ш", обращенный к пользователю. Пользователь может реагировать на эту реплику двумя принципиально различными способами. Он может ввести в компьютер недостающую программу. Но такой ответ пользователя компьютеру существенно отличается от ответа, который бы адресовался человеку. В последнем случае ответом была бы не процедура распознавания буквы Ш, то есть не набор инструкций, подлежащих выполнению, а некоторое иное, непроцедурное определение, какого вида изображения представляют букву Ш. Такой диалог возможен и с компьютером, конечно, при условии, что формат определений заранее оговорен так, что определение понимается одинаково пользователем и компьютером. Предположим, что разрешено пользоваться определениями лишь следующих трех видов.

1. Изображение может иметь имя s , если его можно разделить горизонтальной линией на две части так, что верхняя часть может иметь имя s_u , а нижняя - имя s_d .
2. Изображение может иметь имя s , если его можно разделить вертикальной линией на две части так, что левая часть может иметь имя s_l , а правая - имя s_r .
3. Изображение может иметь имя s , если оно может иметь имя s' .

Указанный формат определений следует понимать как метаправила, то есть правила для формулировки правил распознавания имени изображения. В то же время указанные метаправила можно понимать как правила формулировки правил генерирования изображений с наперед заданным именем.

Так, правило первого формата обозначает следующее: Чтобы ответить на вопрос, может ли предъявленное изображение иметь имя s , его необходимо всеми возможными способами разбить на две части с помощью горизонтальной линии и при каждом разбиении определить, может ли верхняя часть иметь имя s_u , а нижняя часть - имя s_d . Если хотя бы при одном разбиении оба ответа на два последние вопроса положительны, то предъявленному изображению можно присвоить имя s .

Это же правило можно понимать как правило генерирования изображений с именем s . Чтобы нарисовать изображение с именем s , надо нарисовать два изображения с одинаковым количеством столбцов. На первом изображении нарисовать любое изображение с именем s_u , а на втором - любое изображение с именем s_d . Затем эти два изображения следует объединить в одно изображение так, чтобы первое изображение формировало его верхнюю часть, а второе - нижнюю, то есть, присоединить первое изображение сверху второго. Аналогично понимаются правила второго и третьего формата.

Покажем теперь, как с помощью правил указанного формата определить множество изображений, которым можно присвоить имя SH .

С помощью правил второго и третьего формата можно дать следующее определение изображений букв с именем SH : изображение может иметь имя SH , если оно может иметь имя $SH1$, или если его можно разделить на две части вертикальной линией так, что левая часть может иметь имя $SH1$, а правая - имя WR (white rectangle, то есть белый прямоугольник). Первая часть этого определения иллюстрируется левой картинкой на рис. 10.1, а вторая часть - правой картинкой на рис. 10.1. Если из этого определения исключить все лишние слова, а оставить только то, что отличается одно определение от другого, то приведенное определение можно записать в следующей краткой форме:

$$\left. \begin{array}{l} SH ::= SH1 \mid WR ; \\ SH ::= SH1 . \end{array} \right\} \quad (10.1)$$

В этом определении символ $SH1$ понимается как имя изображения буквы Ш, которая прижата к правому краю прямоугольника (поля зрения), на котором она нарисована. Если компьютер понимает символы WR и $SH1$, то есть, если он умеет распознать белый прямоугольник и букву Ш, прижатую к правому краю поля зрения, то приведенное определение достаточно для автоматического построения программы, которая распознает изображения и таких букв Ш, которые не прижаты к правому краю поля зрения. Если же компьютер этими программами не располагает, то он продолжит диалог с пользователем с целью получить определение понятий $SH1$ и WR . Пользователь определяет эти понятия, возможно, вводя новые понятия. Диалог продолжится до тех пор, пока все понятия не окажутся либо определенными пользователем, либо изначально понятными компьютеру. Мы покажем одно из возможных определений множества изображений с именем Ш, полагая, что компьютеру изначально понятны только два по-

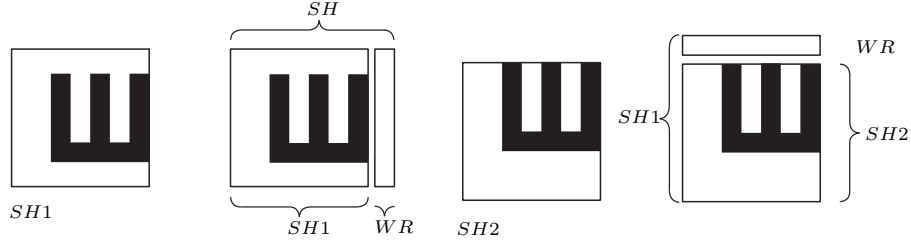


Figure 10.1 Generating letter SH according to rule (10.1), i.e., separating the image margin at the right side of the letter.

Figure 10.2 Illustration of the (10.2), i.e., separation of the image margin above the letter.

нения: WP (white pixel, белый пиксел) и BP (black pixel, черный пиксел). Для этого потребуются вспомогательные понятия WR (white rectangle, белый прямоугольник), BR (black rectangle, черный прямоугольник), U (имя изображений, напоминающих латинскую букву U), I (имя изображений, напоминающих латинскую букву I), L (имя изображений, напоминающих букву L). Определения этих символов иллюстрируются соответствующими картинками.

$$\left. \begin{aligned} SH1 &::= \frac{WR}{SH2}; \\ SH1 &::= SH2. \end{aligned} \right\} \quad (10.2)$$

$$\left. \begin{aligned} SH3 &::= SH4; \\ SH3 &::= \frac{SH4}{WR}. \end{aligned} \right\} \quad (10.4)$$

$$\left. \begin{aligned} SH2 &::= SH3; \\ SH2 &::= WR | SH3. \end{aligned} \right\} \quad (10.3)$$

$$SH4 ::= U | BR. \quad (10.5)$$

$$U ::= L | L; \quad (10.6)$$

$$L ::= \frac{I}{BR}; \quad (10.7)$$

$$I ::= BR | WR. \quad (10.8)$$

$$\left. \begin{aligned} BR &::= BR | BR; \\ BR &::= \frac{BR}{BR}; \\ BR &::= BP. \end{aligned} \right\} \quad (10.9)$$

$$\left. \begin{aligned} WR &::= WR | WR; \\ WR &::= \frac{WR}{WR}; \\ WR &::= WP. \end{aligned} \right\} \quad (10.10)$$

Правила (10.1) - (10.10) определяют множество изображений с именем SH . Совокупность данных, которые компьютер получает в результате указанного воображаемого диалога, можно представить в виде шестерки

$$G = \langle X, K, k_0, P_v, P_s, P_r \rangle, \quad (10.11)$$

которая является примером того, что будет несколько позже определено как двумерная контекстно-свободная грамматика. В приведенной шестерке X - это так называемый терминальный алфавит, то есть множество

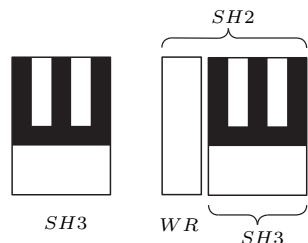


Figure 10.3 Illustration of the rule (10.3), i.e., separation of the margin at the right side of the letter.

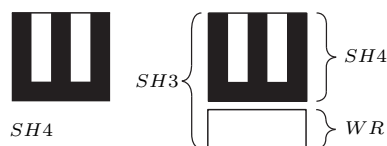


Figure 10.4 Illustration of the rule (10.4), i.e., separation of the margin below the letter.

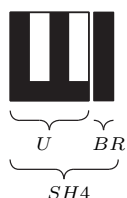


Figure 10.5 Illustration of the rule (10.5), i.e., separation of the black rectangle (BR) at the right side.



Figure 10.6 Illustration of the rule (10.6), i.e., decomposition of the remaining part of the letter into two shapes resembling letter L.



Figure 10.7 Illustration of the rule (10.7), i.e., decomposition of the shape resembling letter L into two parts: the black rectangle at the bottom and a shape resembling letter I, where the black rectangle is at the left and the white rectangle is at the right.



Figure 10.8 Illustration of the rule (10.8), i.e., decomposition of the shape resembling letter I into a black rectangle and a white rectangle.

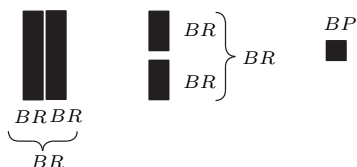


Figure 10.9 Illustration of the rule (10.9), i.e., the black rectangle can be created by concatenating of these black rectangles only.

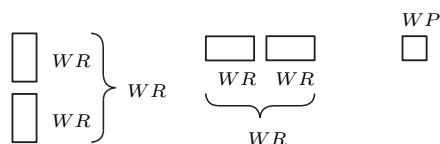


Figure 10.10 Illustration of the rule (10.10), i.e., the white rectangle can be composed of these white rectangles.

символов, которые предполагаются понятными без их определения. В рассмотренном примере это символы WP (white pixel, белый пиксел) и BP (black pixel, черный пиксел).

Множество K - это так называемый нетерминальный алфавит. Он состоит из конечного множества имен. В нашем примере это имена BR , WR , I , L , U , SH_4 , SH_3 , SH_2 , SH_1 и SH . Одно из этих имен, обозначенное в грамматике G как k_0 , называется аксиомой. Это имя, которое не используется при определении какого-то другого понятия в грамматике. В нашем примере это имя SH .

Следующими тремя компонентами грамматики являются три отношения P_h , P_v , P_r . Первые два отношения - это подмножества троек нетерминальных имен, $P_h \subset K \times K \times K$, $P_v \subset K \times K \times K$. Отношение P_r - это подмножество $P_r \subset K \times (K \cup X)$ пар. Индексы h , v , r в обозначениях P_h , P_v и P_r являются первыми буквами английских слов горизонтальный, вертикальный и переименование (renaming). Отношения P_h , P_v и P_r соответствуют правилам генерирования изображений. В нашем примере - это правила (10.1) - (10.10). Следовательно, в рассмотренном примере это следующие отношения:

$$\left. \begin{aligned} P_h &= \{(SH, SH_1, WR), (SH_2, WR, SH_3), (SH_4, U, BR), (U, L, L), \\ &\quad (I, BR, WR), (BR, BR, BR), (WR, WR, WR)\}, \\ P_v &= \{(SH_1, WR, SH_2), (SH_3, SH_4, WR), \\ &\quad (L, I, WR), (WR, WR, WR), (WR, WR, WR)\}, \\ P_r &= \{(SH, SH_1), (SH_1, SH_2), (SH_2, SH_3), (SH_3, SH_4), \\ &\quad (BR, BP), (WR, WP)\}. \end{aligned} \right\} (10.12)$$

С помощью отношений (10.12) или, что тоже самое, правил (10.1) - (10.10) можно сгенерировать изображения, напоминающих букву Ш, например, изображение, представленное на рис. 10.11(a). Применяя эти правила, нельзя сгенерировать некоторые изображения, которые буквой Ш не являются, такие, как, например, на рис. 10.11(b). Однако с помощью указанных правил можно сгенерировать и такие изображения, которые уж никак не являются буквой Ш, как, например, на рис. 10.12(a). С другой стороны, некоторые изображения, которые лишь незначительными помехами отличаются от идеального изображения буквы Ш, невозможно сгенерировать, используя лишь приведенные в примере правила, см. например, рис. 10.12(b). Таким образом, грамматика, рассмотренная в примере, приведена без каких-либо претензий на ее практическое использования для распознавания реальных изображений буквы Ш. Мы рассмотрели этот пример лишь с целью иллюстрации тех формальных понятий, которые будут определены в следующем разделе.



Figure 10.11 (a) An example of the image with a letter resembling *SH* which can be created by the introduced rules. (b) An example of the image not resembling *SH* which cannot be created by the introduced rules.

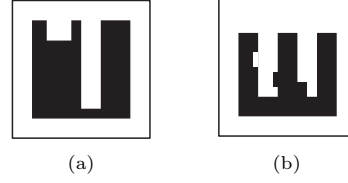


Figure 10.12 (a) An example of the image with a letter not resembling *SH* which can be created by introduced rules. (b) An example of the letter resembling *SH* which cannot be created by introduced rules.

10.3 Двумерные контекстно-свободные грамматики и языки

Пусть X - конечный алфавит символов, называемых терминальными символами. Пусть для любых двух положительных целых чисел m (число строк) и n (число столбцов) определено множество $T(m, n)$ как прямоугольник с размерами m и n в двумерной целочисленной решетке, $T(m, n) = \{(i, j) \mid 1 \leq i \leq m, 1 \leq j \leq n\}$. Изображение x определяется парой чисел m, n и функцией вида $T(m, n) \rightarrow X$. Или, кратко, $x = \langle m, n, T(m, n) \rightarrow X \rangle$. Значение изображения в точке $(i, j) \in T(m, n)$ будет обозначаться $x(i, j)$. Множество всех возможных изображений будет обозначаться X^* , а любое подмножество $L \subset X^*$ будет называться двумерным языком в алфавите X .

Определим операции горизонтальной и вертикальной конкатенации изображений. Пусть $x_1 = \langle m, n_1, T(m, n_1) \rightarrow X \rangle$ и $x_2 = \langle m, n_2, T(m, n_2) \rightarrow X \rangle$ - два изображения с одинаковым количеством m строк. Горизонтальная конкатенация изображений x_1 и x_2 - это изображение $x = \langle m, n_1 + n_2, T(m, n_1 + n_2) \rightarrow X \rangle$, обозначаемое далее $x = x_1 \mid x_2$, для которого выполняется

$$x(i, j) = \begin{cases} x_1(i, j), & \text{если } 1 \leq j \leq n_1, & i = 1, 2, \dots, m, \\ x_2(i, j - n_1), & \text{если } n_1 < j \leq n_1 + n_2, & i = 1, 2, \dots, m. \end{cases}$$

Пусть $x_1 = \langle m_1, n, T(m_1, n) \rightarrow X \rangle$ и $x_2 = \langle m_2, n, T(m_2, n) \rightarrow X \rangle$ - два изображения с одинаковым количеством n столбцов. Вертикальной конкатенацией изображений x_1 и x_2 является изображение $x = \langle m_1 + m_2, n, T(m_1 + m_2, n) \rightarrow X \rangle$, обозначаемое далее $x = \frac{x_1}{x_2}$, для которого выполняется

$$x(i, j) = \begin{cases} x_1(i, j), & \text{если } 1 \leq i \leq m_1, & j = 1, 2, \dots, n, \\ x_2(i - m_1, j), & \text{если } m_1 < i \leq m_1 + m_2, & j = 1, 2, \dots, n. \end{cases}$$

Пусть K - конечное множество, называемое нетерминальным алфавитом. Его элементы будем называть нетерминальными символами, или именами, или метками. Пусть P_h, P_v, P_r - три отношения вида $P_h \subset K \times K \times K$,

$P_v \subset K \times K \times K$, $P_r \subset K \times (K \cup X)$. Эти три отношения n определяют $|K|$ двумерных языков L_k , $k \in K$ и n , следующим образом.

Определение 10.1 Двумерные языки.

1. Для $k \in K$ изображение $x = \langle 1, 1, T(1, 1) \rightarrow X \rangle$ принадлежит языку L_k , если $(k, x(1, 1)) \in P_r$.
2. Для $k \in K$ изображение x принадлежит языку L_k , если существует нетерминальный символ k' , такой, что $x \in L_{k'}$ и $(k, k') \in P_r$.
3. Для $k \in K$ изображение x принадлежит языку L_k , если существуют такие изображения x_t , x_b и такие символы $k_t \in K$, $k_b \in K$, что $x = x_t/x_b$, $x_t \in L_{k_t}$, $x_b \in L_{k_b}$ и $(k, k_t, k_b) \in P_v$.
4. Для $k \in K$ изображение x принадлежит языку L_k , если существуют такие изображения x_l , x_r и такие символы $k_l \in K$, $k_r \in K$, что $x = x_l \mid x_r$, $x_l \in L_{k_l}$, $x_r \in L_{k_r}$ и $(k, k_l, k_r) \in P_h$.

▲

Совокупность данных, которые определяют языки L_k , $k \in K$, образуют шестерку

$$G = \langle X, K, k_0, P_h, P_v, P_r \rangle,$$

которую мы назовем двумерной контекстно-свободной грамматикой. В этой грамматике:

X – конечный терминальный алфавит,

K – конечный нетерминальный алфавит,

k_0 – один из нетерминальных символов, называемый аксиомой, $k_0 \in K$,

$P_h \subset K \times K \times K$ – множество правил горизонтальной конкатенации,

$P_v \subset K \times K \times K$ – множество правил вертикальной конкатенации,

$P_r \subset K \times (K \cup X)$ – множество правил переименования.

Язык L_{k_0} , где k_0 – аксиома грамматики G , называется языком грамматики G , который мы обозначим $L(G)$. Изображения из языка $L(G)$ будут называться допустимыми в грамматике G .

Грамматика, в которой одно из множеств P_h или P_v пустое, есть общеизвестная, то есть одномерная грамматика, представленная в канонической форме Н.Хомского. Язык такой одномерной грамматики содержит только "изображения", которые состоят либо из одной строки, либо из одного столбца. Фактически это уже не изображения, а последовательности. Хорошо известно, да и нетрудно показать, что любой регулярный язык можно определить с помощью контекстно-свободной грамматики. Это значит, что класс контекстно-свободных языков включает в себя все регулярные языки, анализу которых мы посвятили предыдущую лекцию. В данной лекции мы рассмотрим задачи распознавания в рамках контекстно-свободных языков, подобные решенным в предыдущих лекциях. Мы начнем это рассмотрение с задачи на точное соответствие заданного изображения заданной контекстно-свободной грамматике.

10.4 Задача на точное соответствие. Обобщенный алгоритм Кока-Янгера-Касами

Пусть $G = \langle X, K, k_0, P_h, P_v, P_r \rangle$ – двумерная контекстно-свободная грамматика, а $x = \langle m, n, T(m, n) \rightarrow X \rangle$ – изображение. Задача на точное соответствие состоит в построении алгоритма, который для любой грамматики G и любого изображения x определяет, выполняется ли $x \in L(G)$ или нет.

Алгоритм, решающий эту задачу, основан на следующих простых соображениях. Пусть $R(i_t, i_b, j_l, j_r)$, $1 \leq i_t \leq i_b \leq m$, $1 \leq j_l \leq j_r \leq n$, – прямоугольник, являющийся подмножеством прямоугольника $T(m, n)$; $R(i_t, i_b, j_l, j_r) = \{(i, j) \mid i_t \leq i \leq i_b, j_l \leq j \leq j_r\}$. Множество всех таких прямоугольников обозначим \mathcal{R} . Это множество частично упорядочено, поскольку на нем задано транзитивное отношение "прямоугольник R' является подмножеством прямоугольника R'' ". Следовательно, множество \mathcal{R} можно упорядочить в виде одномерной последовательности так, что если $R' \subset R''$ и $R' \neq R''$, то R' появляется в последовательности раньше, чем R'' .

Пусть $x(R)$, $R \in \mathcal{R}$, – сужение изображения x на множество R . Обозначим $f(i_t, i_b, j_l, j_r, k)$, $k \in K$, число, принимающее значение 1, если $x(R(i_t, i_b, j_l, j_r)) \in L_k$, и значение 0 в противном случае. Равенство $f(1, m, 1, n, k_0) = 1$, таким образом, эквивалентно утверждению $x \in L(G)$. Следовательно, при заданных грамматике и изображении следует вычислить единственный бит $f(1, m, 1, n, k_0)$.

Здесь и далее мы будем использовать обозначения P_h, P_v, P_r в двух смыслах. В одном контексте эти идентификаторы будут обозначать, как и раньше, отношения, то есть подмножества $P_h \subset K \times K \times K$, $P_v \subset K \times K \times K$ and $P_r \subset K \times (K \cup X)$. В других контекстах эти же обозначения будут пониматься как функции $P_h: K \times K \times K \rightarrow \{0, 1\}$, $P_v: K \times K \times K \rightarrow \{0, 1\}$, $P_r: K \times (K \cup X) \rightarrow \{0, 1\}$. Таким образом, записи $(k_1, k_2, k_3) \in P_h$ и $P_h(k_1, k_2, k_3) = 1$, $(k_1, k_2, k_3) \in P_v$ и $P_v(k_1, k_2, k_3) = 1$, и подобные будут считаться равноценными. Используя эти новые обозначения, мы можем записать следующее рекуррентное выражение для значений $f(i_t, i_b, j_l, j_r, k)$, $1 \leq i_t \leq i_b \leq m$, $1 \leq j_l \leq j_r \leq n$, $k \in K$,

$$\begin{aligned}
 & f(i_t, i_b, j_l, j_r, k) \\
 &= \left[\bigvee_{j=j_l}^{j_r-1} \bigvee_{k_l} \bigvee_{k_r} \left(f(i_t, i_b, j_l, j, k_l) \wedge P_h(k, k_l, k_r) \wedge f(i_t, i_b, j+1, j_r, k_r) \right) \right] \\
 & \bigvee \left[\bigvee_{i=i_t}^{i_b-1} \bigvee_{k_t} \bigvee_{k_b} \left(f(i_t, i, j_l, j_r, k_t) \wedge P_v(k, k_t, k_b) \wedge f(i+1, i_b, j_l, j_r, k_b) \right) \right] \\
 & \bigvee \left[\bigvee_{k' \in K \cup X} \left(f(i_t, i_b, j_l, j_r, k') \wedge P_r(k, k') \right) \right], \tag{10.13}
 \end{aligned}$$

которые следуют непосредственно из определения 10.1 языков L_k . Выра-

жения (10.13), а следовательно, и определение 10.1 языков L_k , $k \in K$, являются основой для следующего алгоритма, который распознает, принадлежит ли изображение $x = \langle m, n, T(m, n) \rightarrow X \rangle$ языку $L(G)$.

Algorithm 10.1 Двумерное обобщение алгоритма Кока-Янгера-Касами

1. Для каждой пары (i, j) , $1 \leq i \leq m$, $1 \leq j \leq n$, устанавливаются начальные значения функции f ,

$$f(i, i, j, j, x(i, j)) := 1.$$

Остальным значениям функции f присваиваются нулевые значения.

2. Последовательно анализируются все прямоугольники $R \in \mathcal{R}$, то есть, четверки $\langle i_1, i_2, j_1, j_2 \rangle$, причем так, что если $R' \subset R''$, то R' анализируется раньше, чем R'' . Для каждого прямоугольника $\langle i_1, i_2, j_1, j_2 \rangle$ и каждого символа $k \in K$ выполняются следующие действия.

- (a) Проверяется условие

$$\bigvee_{j_1 \leq j < j_2} \bigvee_{\substack{k_r \in K \\ k_l \in K}} (f(i_1, i_2, j_1, j, k_l) \wedge P_h(k, k_l, k_r) \wedge f(i_1, i_2, j + 1, j_2, k_r)). \quad (10.14)$$

Если оно выполняется, то $f(i_1, i_2, j_1, j_2, k) := 1$.

- (b) Если условие (10.14) не выполняется, то проверяется условие

$$\bigvee_{i_1 < i < i_2} \bigvee_{\substack{k_b \in K \\ k_t \in K}} (f(i, i_2, j_1, j_2, k_t) \wedge P_v(k, k_t, k_b) \wedge f(i + 1, i_2, j_1, j_2, k_b)). \quad (10.15)$$

Если условие (10.15) выполняется, то $f(i_1, i_2, j_1, j_2, k)$.

- (c) Если условие (10.15) не выполняется, то проверяется условие

$$\bigvee_{k' \in K} f(i_1, i_2, j_1, j_2, k) \wedge P_t(k, k'). \quad (10.16)$$

Если условие (10.16) выполняется, то $f(i_1, i_2, j_1, j_2, k) := 1$.

3. Решением задачи является бит $f(1, m, 1, n, k_0)$.
-

Приведенный алгоритм является прямым двумерным обобщением известного алгоритма Кока-Янгера-Касами, распознающего принадлежность одномерной последовательности контекстно свободному (одномерному) языку. Алгоритм Кока-Янгера-Касами является частным случаем приведенного алгоритма 10.1, когда одно из отношений P_h или P_v пустое.

Хотя алгоритм 10.1 и является двумерным обобщением известного "одномерного" алгоритма, из этого совсем не следует, что вычислительная сложность распознавания двумерных изображений больше, чем одномерных последовательностей.

Сложность вычисления по формулам (10.14), (10.15) и (10.16) зависит от размеров изображения и имеет порядок $\mathcal{O}(m^2 n^2 (m + n))$. Отсюда немедленно следует известный результат, что сложность распознавания последовательности длиной l с помощью алгоритма Кока-Янгера-Касами имеет порядок $\mathcal{O}(l^3)$.

Теперь мы приходим к парадоксальному выводу, что распознавание (двумерного) изображения, состоящего из $(m \times n)$ пикселей, на порядок проще, чем распознавание (одномерной) последовательности, состоящей из такого же количества mn элементов. В первом случае это сложность порядка $\mathcal{O}(m^2 n^2 (m + n))$, а во втором - порядка $\mathcal{O}((mn)^3)$. Этот результат можно объяснить, повидимому, тем, что при обобщении класса контекстно-свободных языков на двумерный случай мы включили в этот класс обширное множество других языков, но все они оказались проще, чем языки, присутствующие в этом классе до обобщения. В расширенном классе языков все прежние языки продолжают оставаться самыми сложными. Это, конечно же, очень приятная неожиданность. Ведь, казалось бы, переход от распознавания одномерных последовательностей к распознаванию двумерных изображений должен был бы привести к резкому увеличению сложности.

Нам остается только восхищаться тщательностью, с которой природа, или эволюция (или кто-то еще) подошла к конструированию носителей информации. Для случаев, когда речь идет об очень больших объемах, будто специально было придумано изображение, при котором трудности, связанные с большими объемами данных, хоть немного компенсируются его удобной структурой.

10.5 Общая структурная конструкция

К данному моменту мы рассмотрели два механизма для определения того или иного подмножества распознаваемых объектов. В двух предыдущих лекциях мы рассмотрели механизм определения подмножеств последовательностей. В данной лекции мы представили более общий механизм определения подмножеств двумерных массивов. Сейчас мы опишем общую конструкцию, которая будет включать уже рассмотренные два механизма как частные случаи. Она будет называться *структурной конструкцией*. Мы ее введем не только ради лаконичного представления полученных результатов. Мы увидим, что в рамках этой конструкции окажется возможным определять множества, которые не являются ни регулярными языками, ни контекстно-свободными. Кроме того, мы увидим, что в этой конструкции окажется заполненным разрыв между регулярными языками и контекстно-свободными, который сейчас бросается в глаза. Укажем наиболее выразительные характеристики этого разрыва.

1. Переход от регулярных языков к контекстно-свободным происходит слишком большим скачком, а не постепенно. Контекстно-свободные языки – это чересчур сильное обобщение регулярных языков. Это заметно хотя бы по тому, как резко возрастает вычислительная сложность задач распознавания.

Мы рассмотрели ряд задач распознавания в рамках регулярных языков и их разумных модификаций и убедились, что их вычислительная сложность линейно зависит от длины распознаваемой последовательности. Пока что мы рассмотрели лишь одну, причем простейшую за-

дачу в рамках контекстно-свободных языков. Это была задача на точное соответствие. Но вычислительная сложность уже этой простейшей задачи имеет порядок третьей степени длины распознаваемой последовательности. Это существенный скачок сложности, наблюдая который невозможно отказаться от предположения, что существуют промежуточные классы языков, более сложных, чем регулярные, но не столь богатые, как контекстно-свободные.

2. Даже невооруженным глазом виден существенный зазор между формализмом контекстно-свободных грамматик, как одномерных, так и двумерных, и реальными задачами распознавания. На этот зазор мы обратили внимание в разделе 10.2, когда иллюстрировали основные понятия контекстно-свободных грамматик на примере русской буквы Ш. Единственной характеристикой фрагмента изображения, которая определяет его зависимость от остальной части изображения, является имя этого фрагмента. Это имя должно принимать значение из конечного алфавита значений. В практических применениях этот алфавит должен быть не просто конечным, он должен быть достаточно малым, так как от этого зависит сложность алгоритма распознавания. С помощью таких слабых средств довольно трудно выразить зависимости между отдельными частями изображений в реальных задачах. В частности, такого рода средства плохо приспособлены для представления геометрических ограничений, например, на размеры того или иного фрагмента, на расстояния между фрагментами и т.п. В силу этого напрашивается дополнение формализма контекстно-свободных грамматик такими средствами, которые позволяли бы естественно вводить ограничения геометрического, числового характера.
3. В рамках контекстно-свободных грамматик мы рассмотрели лишь простейшую задачу распознавания, а именно задачу на точное соответствие. Естественно ввести в формализм контекстно-свободных грамматик дополнительные средства, позволяющие формулировать задачи на наилучшее соответствие и вводить размытые и стохастические модификации контекстно-свободных грамматик.

Мы введем систему понятий, на основе которой можно будет, сохраняя достоинства контекстно-свободных грамматик, избежать их отрыва от регулярных грамматик с одной стороны и от разумных практических задач с другой. Ряд вводимых далее формальных определений будет время от времени прерываться иллюстративными примерами. Как и любая формальная конструкция, вводимая система понятий применима не только для задач распознавания изображений. Однако для названия вводимых понятий мы будем пользоваться терминами, которые происходят из распознавания изображений.

10.5.1 Структурная конструкция для определения множеств распознаваемых объектов

Пусть T – множество, которое будет называть *полем наблюдения* или, когда это будет более естественно *полем зрения*, а V – конечный алфавит символов. Элемент поля зрения T будет называться *пикселом* и обозначаться t . Пусть 2^T – множество всех возможных подмножеств поля зрения T , а $\mathcal{T} \subset 2^T$ – подмножество подмножеств в T . Подмножество \mathcal{T} будет называться *структурой поля зрения*, а его элементы – *фрагментами поля зрения* или просто фрагментами. Это значит, что если $T' \in \mathcal{T}$, то $T' \subset T$, но не наоборот: структура \mathcal{T} содержит не все подмножества поля зрения. Далее мы будем иметь дело только с такими структурами, что для любого $t \in T$ выполняется $\{t\} \in \mathcal{T}$. Попросту говоря, мы предполагаем, что каждый пиксел образует один из элементов структуры.

Изображение или в более общем случае *наблюдение* x считается заданным на поле зрения T , если задан фрагмент $T_0 \in \mathcal{T}$ и функция $v: T_0 \rightarrow V$. Таким образом, изображение представляется парой (T_0, v) , $T_0 \in \mathcal{T}$.

Пример 10.1 Конечная последовательность букв. Пусть T – множество положительных целых чисел, а структура \mathcal{T} содержит все интервалы вида $\{t \mid 1 \leq t \leq n\}$, $n \in T$ и все множества вида $\{t\}$, $t \in T$. Пусть множество V состоит из букв A, B, C . В этом случае наблюдение – это последовательность конечной длины, составленная из букв A, B, C . ▲

Пример 10.2 Бинарное изображение. Пусть T – множество пар положительных целых чисел, а структура \mathcal{T} состоит из множеств вида $\{(i, j) \mid m_1 \leq i \leq m_2, n_1 \leq j \leq n_2\}$, $1 \leq m_1 \leq m_2$, $1 \leq n_1 \leq n_2$. Если $V = \{0, 1\}$, то наблюдение может пониматься как бинарное изображение с размерами $(m_2 - m_1 + 1) \times (n_2 - n_1 + 1)$. ▲

Пример 10.3 Бинарное изображение как функция времени на интервале от t_1 до t_2 . Пусть T – множество троек положительных целых чисел, а структура \mathcal{T} состоит из множеств вида $\{(i, j, t) \mid m_1 \leq i \leq m_2, n_1 \leq j \leq n_2, t_1 \leq t \leq t_2\}$, $1 \leq m_1 \leq m_2$, $1 \leq n_1 \leq n_2$, $1 \leq t_1 \leq t_2$. При $V = \{0, 1\}$ наблюдение можно понимать как переменное во времени изображение, заданное на временном интервале от t_1 до t_2 . ▲

Пример 10.4 Поле наблюдения с более сложной структурой. Возможны и более сложные структуры поля наблюдения T . Это может быть, например, множество вершин ациклического графа, или декартово произведение таких графов, или многое другое. ▲

Примечание 10.1 Алфавит V символов соответствует терминальным и нетерминальным алфавитам в формальных грамматиках. Для дальнейшего изложения нам нет нужды различать эти два алфавита. ▲

Множество $\mathcal{T} \times V$ будет называться множеством *структурных элементов* и обозначаться S . Отдельный структурный элемент будет обозначаться s с разнообразными индексами. Структурный элемент – это фрагмент из структуры \mathcal{T} , которому присвоен символ из V .

Пример 10.5 Структурный элемент. Если про какой-то пиксел сказано, что он черный, то тем самым задан структурный элемент. Если про какое-то подмножество пикселов сказано, что они образуют прямую линию, то тоже задан структурный элемент. Если про какой-то прямоугольник в поле зрения сказано, что на нем представлена буква А, то тоже задан задан структурный элемент. ▲

Важными являются следующие четыре понятия: сегментация фрагмента, иерархическая сегментация фрагмента, карта и иерархическая карта структурного элемента.

Сегментация фрагмента $T_0 \in \mathcal{T}$ – это подмножество $R \subset \mathcal{T}$ фрагментов, которое содержит фрагмент T_0 и совокупность попарно непересекающихся фрагментов, объединение которых равно T_0 . Говоря иными словами, если $T_0 = \bigcup_{i=1}^m T_i$, $T_i \in \mathcal{T}$, и $T_i \cap T_j = \emptyset$ для любых $i > 0$, $j > 0$, $i \neq j$, то $R = \{T_i \mid i = 0, 1, \dots, m\}$ есть сегментация фрагмента T_0 .

Функция $m: R \rightarrow V$, определенная на сегментации R фрагмента T_0 , есть карта структурного элемента $(T_0, m(T_0))$. Карта – это подмножество фрагментов, каждому из которых присвоен символ из V . Иными словами говоря, карта – это подмножество структурных элементов.

Выделим два важных частных случая карт. Карта, состоящая из трех поименованных фрагментов, называется *правилом*. Для этого частного случая карт вводится обозначение π . Те или иные подмножества правил будут обозначаться Π с разнообразными аргументами и индексами.

Другим важным частным случаем карты является *поименованное изображение*. Карта этого вида представляет собой фрагмент T_0 , символа v , присвоенного всему фрагменту в целом, и указание символа для каждого пиксела из фрагмента T_0 . Формально говоря, поименованное изображение – это сегментация $R = \{T_0\} \cup (\bigcup_{t \in T_0} \{\{t\}\})$ фрагмента T_0 на отдельные пикселы и функция $x: R \rightarrow V$. Поименованное изображение – это подмножество структурных элементов вида $\{(T_0, v_0)\} \cup \{(\{t\}, x(t)) \mid t \in T_0\}$.

Иерархическая сегментация определяется следующим рекуррентным способом.

1. Сегментация R , состоящая из трех фрагментов, есть иерархическая сегментация.
2. Пусть R – иерархическая сегментация, а $T' \in R$ – фрагмент, который не содержит в себе никакой другой фрагмент из R . Пусть $T'' \in \mathcal{T}$ и $T''' \in \mathcal{T}$ – два непересекающихся фрагмента, объединение которых равно T' . В таком случае $R \cup \{T'', T'''\}$ есть также иерархическая сегментация.

Иерархическая карта определяется как иерархическая сегментация R , на которой задана функция $R \rightarrow V$, указывающая имя каждого фрагмента из R . Иерархическая карта будет обозначаться H , а различные подмножества иерархических карт будут обозначаться \mathcal{H} с различными аргументами. Как карта, так и иерархическая карта будут пониматься как подмножества структурных элементов определенного вида.

Из приведенного определения немедленно следует, что каждой иерархической карте H соответствует одно и только одно подмножество правил. Обозначим его $\Pi(H)$.

Для заданного поименованного изображения m обозначим $\mathcal{H}(m)$ множество всех иерархических карт, содержащих m .

Пусть W – коммутативное полукольцо с операциями \oplus и \otimes , обозначающими сложение и умножение соответственно. Обозначим P функцию, которая указывает величину $P(\pi) \in W$ для каждого правила π .

Пример 10.6 Функция P является средством, с помощью которого регламентируется генерирование изображения на заданном фрагменте T_0 , причем такого изображения, которому можно присвоить заранее заданное имя v_0 . Изображение генерируется в процессе $|T_0| - 1$ шагов. Результатом каждого, скажем, i -ого шага является сегментация фрагмента T_0 на $i + 1$ поименованных фрагмента. На каждом шаге выбирается тот или иной фрагмент из уже построенной карты, который состоит более чем из одного пиксела и не содержит в себе ни одного фрагмента из текущей карты. Выбранный фрагмент разбивается на два меньших фрагмента. Это разбиение является результатом применения определенного правила, функция P управляет этим процессом так, как это показано в дальнейших примерах. ▲

Каждая иерархическая карта H характеризуется величиной $g(H)$, определяемой как

$$g(H) = \bigotimes_{\pi \in \Pi(H)} P(\pi). \quad (10.17)$$

Каждое поименованное изображение m характеризуется величиной $G(m)$, определяемой как

$$G(m) = \bigoplus_{H \in \mathcal{H}(m)} g(H) = \bigoplus_{H \in \mathcal{H}(m)} \bigotimes_{\pi \in \Pi(H)} P(\pi). \quad (10.18)$$

Основная задача структурного распознавания состоит в построении алгоритма, который для любого поименованного изображения m вычисляет его характеристику $G(m)$.

Пример 10.7 Пусть $W = \{0, 1\}$, операция \oplus – это дизъюнкция, а \otimes – конъюнкция. В этом случае P определяет подмножество правил, которые считаются допустимыми в решаемой задаче распознавания. Это подмножество правил определяет множество изображений, которые можно сгенерировать на фрагменте T_0 , причем таких, которым можно присвоить имя v_0 . Величина (10.18) указывает, входит ли предъявленное изображение в это множество или нет. ▲

Пример 10.8 Пусть W – полностью упорядоченное множество, операция \oplus понимается как \max , а \otimes – как \min . Функция P в этом случае определяет для каждого правила π степень его допустимости, которая выражается тем или иным элементом из W . Считается, что степень

допустимости процесса генерирования изображения не хуже, чем θ , если степень допустимости каждого примененного правила была не хуже, чем θ . Считается также, что степень допустимости изображения не хуже, чем θ , если существует хотя бы один процесс, который генерирует это изображение и который не хуже, чем θ . Выражение (10.18) в этом случае, насколько допустимо присоить предъявленному изображению имя v_0 . ▲

Пример 10.9 Пусть W - множество неотрицательных чисел, операция \oplus обозначает \min , операция \otimes обозначает суммирование чисел в обычном смысле этого слова. В этом случае функция P может пониматься как штраф за применение того или иного правила, число (10.17) обозначает суммарную стоимость того или иного процесса генерирования изображения, а (10.18) – стоимость самого дешевого процесса генерирования предъявленного изображения. ▲

Пример 10.10 Пусть W - множество неотрицательных чисел, а операции \oplus и \otimes понимаются как сложение и умножение чисел в обычном смысле этого слова. Если при этом функция P для любого s удовлетворяет условию $\sum_{s_1, s_2} P(s_1, s_2 | s) = 1$, то функцию P можно понимать как задание определенного случайного механизма генерирования изображений. В этом случае величина (10.18) есть вероятность предъявленного изображения в генеральной совокупности изображений с именем v_0 на фрагменте T_0 . ▲

10.5.2 Формулировка основной задачи структурного распознавания изображений

Пусть дано:

множество T ,

множество V ,

структура $\mathcal{T} \subset 2^T$,

функция $P: (T \times V) \times (T \times V) \times (T \times V) \rightarrow W$,

операции $\oplus: W \times W \rightarrow W$ и $\otimes: W \times W \rightarrow W$, нулевой элемент $0^\oplus \in W$ и единичный элемент $1^\otimes \in W$,

фрагмент $T_0 \in \mathcal{T}$, изображение $x: T_0 \rightarrow V$ и символ $v_0 \in V$.

По этим исходным данным следует вычислить величину

$$G(T_0, v_0, x) = \bigoplus_{H \in \mathcal{H}(T_0, v_0, x)} \bigotimes_{(s_1, s_2, s_3) \in \Pi(H)} P(s_1, s_2, s_3). \quad (10.19)$$

в этой формуле $\mathcal{H}(T_0, v_0, x)$ – это множество иерархических карт, содержащих структурные элементы (T_0, v_0) и $(\{t\}, x(t))$, $t \in T_0$, а $\Pi(H)$ – множество правил (s_1, s_2, s_3) в иерархической карте H .

10.5.3 Вычислительная процедура решения основной задачи

Для заданного наблюдения $x: T_0 \rightarrow V$ на фрагменте T_0 введем в рассмотрение следующую совокупность величин,

$$G(T', v', x(T')) = \bigoplus_{H \in \mathcal{H}(T', v', x(T'))} \bigotimes_{(s_1, s_2, s_3) \in \Pi(H)} P(s_1, s_2, s_3) \quad (10.20)$$

которые следует вычислить для каждого структурного элемента (T', v') , $T' \subset T_0$, $T' \in \mathcal{T}$, $v' \in V$. В формуле (10.20) $x(T')$ обозначает сужение анализируемого изображения x на фрагмент T' .

Если мы вычислим все величины (10.20), то будет решена и задача (10.19), так как величина (10.19) есть одна из величин из совокупности (10.20).

При решении каждой конкретной задачи изображение $x: T_0 \rightarrow V$ фиксировано, равно, как и его сужения на все возможные фрагменты. В силу этого эти данные можно исключить из формулы (10.20) с целью ее сокращения. С учетом ранее введенного обозначения s для структурного элемента, то есть пары (T', v') , $T' \subset T_0$, $T' \in \mathcal{T}$, $v' \in V$, и обозначения π для правила, то есть для тройки (s_1, s_2, s_3) , формула (10.20) приобретает более краткий вид

$$G(s) = \bigoplus_{H \in \mathcal{H}(s)} \bigotimes_{\pi \in \Pi(H)} P(\pi). \quad (10.21)$$

Величина $G(s)$, $s = (T', v')$, характеризует ситуацию, когда сужению распознаваемого изображения $x: T_0 \rightarrow V$ на фрагмент $T' \subset T_0$ присвоена метка v' . Множество $\mathcal{H}(s)$, $s = (T', v')$, есть множество всех иерархических карт H , то есть возможных процессов генерирования той части изображения, которая наблюдается на фрагменте T' , при условии, что этой части можно приписать имя v' . Множество $\Pi(H)$ есть множество всех правил, которые участвовали в процессе построения иерархической карты H .

Совокупность величин $G(s)$, $s = (t', v')$, $T' \subset T_0$, $T' \in \mathcal{T}$, $v' \in V$, не может быть произвольной. Эти величины удовлетворяют определенным отношениям, которые делают возможным их конструктивное вычисление. Проанализируем эти отношения.

Иерархическая карта $\mathcal{H}(s)$, конечно же, содержит s . Пусть пара s_1, s_2 образует правило со структурным элементом s . Обозначим $\mathcal{H}(s, s_1, s_2)$ подмножество иерархических карт из множества $\mathcal{H}(s)$, которые, помимо структурного элемента s содержат еще и s_1 и s_2 . Очевидно отношение $\mathcal{H}(s) = \bigcup_{s_1} \bigcup_{s_2} \mathcal{H}(s, s_1, s_2)$, в силу чего (10.21) можно записать в виде

$$G(s) = \bigoplus_{s_1} \bigoplus_{s_2} \bigoplus_{H \in \mathcal{H}(s, s_1, s_2)} \bigotimes_{\pi \in \Pi(H)} P(\pi). \quad (10.22)$$

Обозначим $G(s, s_1, s_2)$ сумму

$$\bigoplus_{H \in \mathcal{H}(s, s_1, s_2)} \bigotimes_{\pi \in \Pi(H)} P(\pi). \quad (10.23)$$

Правило s, s_1, s_2 присутствует в каждой иерархической карте H из множества $\mathcal{H}(s, s_1, s_2)$, и поэтому сомножитель $P(s, s_1, s_2)$ присутствует в каждом произведении $\bigotimes_{\pi \in \Pi(H)} P(\pi)$. Его можно вынести за знак суммирования так, что

$$G(s, s_1, s_2) = P(s, s_1, s_2) \otimes \left(\bigoplus_{H \in \mathcal{H}(s, s_1, s_2)} \bigotimes_{\pi \in \Pi(H) \setminus \{(s, s_1, s_2)\}} P(\pi) \right). \quad (10.24)$$

Множество $\Pi(H) \setminus \{(s, s_1, s_2)\}$ при любой иерархической карте $H \in \mathcal{H}(s, s_1, s_2)$ можно представить как объединение двух непересекающихся множеств. Первым является иерархическая карта H_1 из множества $\mathcal{H}(s_1)$, а вторым – иерархическая карта H_2 из множества $\mathcal{H}(s_2)$. В силу этого произведение $\bigotimes_{\pi \in \Pi(H) \setminus \{(s, s_1, s_2)\}} P(\pi)$ можно представить в виде $(\bigotimes_{\pi \in \Pi(H_1)} P(\pi)) \otimes (\bigotimes_{\pi \in \Pi(H_2)} P(\pi))$, а суммирование по всем картам из множества $\mathcal{H}(s, s_1, s_2)$ обозначает суммирование по всем картам из декартового произведения $\mathcal{H}(s_1) \times \mathcal{H}(s_2)$. В результате этого формула (10.24) приобретает вид

$$\begin{aligned} G(s, s_1, s_2) & \quad (10.25) \\ &= P(s, s_1, s_2) \otimes \left(\bigoplus_{H_1 \in \mathcal{H}(s_1)} \bigoplus_{H_2 \in \mathcal{H}(s_2)} \left(\bigotimes_{\pi \in \Pi(H_1)} P(\pi) \right) \otimes \left(\bigotimes_{\pi \in \Pi(H_2)} P(\pi) \right) \right). \end{aligned}$$

Далее, на основании очевидной формулы $\bigoplus_{i \in I} \bigoplus_{j \in J} (\beta_i \otimes \gamma_j) = (\bigoplus_{i \in I} \beta_i) \otimes (\bigoplus_{j \in J} \gamma_j)$ получаем

$$\begin{aligned} G(s, s_1, s_2) & \quad (10.26) \\ &= P(s, s_1, s_2) \otimes \left(\bigoplus_{H \in \mathcal{H}(s_1)} \bigotimes_{\pi \in \Pi(H)} P(\pi) \right) \otimes \left(\bigoplus_{H \in \mathcal{H}(s_2)} \bigotimes_{\pi \in \Pi(H)} P(\pi) \right). \end{aligned}$$

Это выражение состоит из трех сомножителей. По определению (10.21) второй сомножитель есть не что иное как $G(s_1)$, а третий сомножитель есть $G(s_2)$. Таким образом формула (10.26) записывается в следующей краткой форме

$$G(s, s_1, s_2) = G(s_1) \otimes P(s, s_1, s_2) \otimes G(s_2),$$

а это и есть величина (10.23). Подставим ее в (10.22) и получим рекуррентное выражение

$$G(s) = \bigoplus_{s_1, s_2} \left(G(s_1) \otimes P(s, s_1, s_2) \otimes G(s_2) \right), \quad (10.27)$$

которое является рабочей формулой для решения основной задачи. Покажем это.

С самого начала определяются величины $G(s)$ для структурных элементов вида $s = (\{t\}, v)$, $t \in T_0$, $v \in V$, так, что

$$G(\{t\}, v) = \begin{cases} 0^\oplus, & \text{если } x(t) \neq v, \\ 1^\otimes, & \text{если } x(t) = v, \end{cases} \quad (10.28)$$

где $x: T_0 \rightarrow V$ - распознаваемое изображение. Затем структурные элементы упорядочиваются в одномерную последовательность так, что если $s' = (T', v')$ предшествует элементу $s'' = (T'', v'')$, то $T' \neq T''$ и $T' \subset T''$. При $T' = T''$ структурные элементы записываются в последовательность в произвольном порядке, который фиксируется на протяжении всей последующей процедуры. Структурные элементы обрабатываются в принятой последовательности, и для каждого из них, скажем, для структурного элемента s , вычисляется величина $G(s)$ по формуле (10.27). К этому моменту времени уже имеются все данные, которые нужны для этого вычисления, потому что для этого нужны величины $G(s')$ только для тех величин s' , которые в принятой последовательности предшествуют s' . После просмотра всех структурных элементов вычислены величины $G(s)$ для всех структурных элементов, в том числе и для структурного элемента (T_0, v_0) . Полное количество операций пропорционально количеству правил (s, s_1, s_2) , $s = (T', v')$, $s_1 = (T'_1, v'_1)$, $s_2 = (T'_2, v'_2)$, $T' \subset T_0$, для которых $P(s, s_1, s_2) \neq 0^\oplus$.

Пример 10.11 Регулярные языки и структурная конструкция. В случае регулярных языков, равно как и их стохастических и размытых модификаций, множество T - это множество положительных целых чисел. Структура \mathcal{T} состоит из фрагментов вида $\{t \mid 1 \leq t \leq n\}$, $n \in T$, и фрагментов вида $\{t\}$, $t \in T$. Если речь идет о распознавании последовательности длины n , то для множества $T_0 = \{1, 2, \dots, n\}$ существует только $n-1$ троек фрагментов (T_1, T_2, T_3) , $T_i \subset T_0$, $i = 1, 2, 3$, таких, что T_1 есть объединение T_2 и T_3 . Действительно, тройка (T_1, T_2, T_3) может быть только вида $(T_1, T_1 \setminus \{t\}, \{t\})$, где t есть последний пиксел фрагмента T_1 . Поскольку структурный элемент s_3 может быть только вида $(x(t), \{t\})$ (где $x(t)$ - это t -ый символ в распознаваемой последовательности), то каждой тройке фрагментов соответствует не более $|V|^2$ правил $\pi = (s_1, s_2, s_3)$. Таким образом, сложность процедуры структурного анализа в этом случае имеет порядок $\mathcal{O}(|V|^2(n-1))$, то есть тот же порядок, что и в рассмотренных ранее алгоритмах, решающих задачи распознавания только лишь для регулярных языков. ▲

Пример 10.12 Контекстно-свободные языки и структурная конструкция. Для контекстно-свободных языков, как и для регулярных, множество T - это множество положительных целых чисел. Однако структура \mathcal{T} этого множества значительно сложнее. Она состоит из интервалов вида $\{t \mid i \leq t \leq j\}$, $i \in T$, $j \in T$, $i \leq j$. При распознавании последовательности длиной n множество T_0 есть $\{1, 2, \dots, n\}$, а количество сегментаций вида $\{T_1, T_2, T_3\}$, $T_i \subset T_0$, имеет порядок n^3 . Следова-

тельно, сложность описанного общего алгоритма решения задачи структурного анализа в этом случае имеет сложность тоже порядка n^3 , то есть сложность того же порядка, что и сложность известных алгоритмов Кока-Янгера-Касами, но применимых только для случая контекстно-свободных языков. ▲

Сейчас мы можем понять причину того резкого скачка сложности алгоритмов распознавания последовательностей, который имеет место при переходе от регулярных языков к контекстно-свободным. Это существенное различие состоит в различии структур множества T , которая в теории формальных грамматик явно не упоминается, но в неявном виде она присутствует. Структура регулярных языков такова, что любой фрагмент поля зрения единственным образом представляется в виде объединения двух фрагментов, в силу чего каждой последовательности соответствует единственная иерархическая сегментация, которая к тому же легко строится.

Ситуация совсем иная для контекстно-свободных языков. Здесь каждый фрагмент длиной n можно разбить на два других фрагмента $(n - 1)$ способами. В силу этого каждой последовательности соответствует огромное количество иерархических сегментаций.

Сейчас, когда вскрыт основной фактор, определяющий сложность структурного анализа последовательностей, очевидным становится существование языков, которые, строго говоря, не являются регулярными, но такими, что сложность структурного анализа имеет тот же порядок, что и для регулярных языков.

Пример 10.13 Нерегулярные языки с малым количеством иерархических сегментаций. Пусть структура T определена так же, как и в примере 10.12 для контекстно-свободных грамматик. Однако функция P такова, что для каждого фрагмента T_1 величина $P((T_1, v_1), (T_2, v_2), (T_3, v_3))$ не равна нулю только для какого-то одного его разбиения на два фрагмента T_2 и T_3 . Например, это может иметь место в случае, когда на фрагменты T_2 и T_3 наложено ограничение, что их длины должны отличаться не более чем на 1. Язык, определенный на такой структуре и функцией P такого вида, не будет регулярным. Тем не менее, сложность распознавания принадлежности последовательности такому языку будет линейно зависеть от длины последовательности, как это имеет место и для регулярных языков.

Подобным образом можно сконструировать языки, которые не являются контекстно-свободными, но сложность задачи распознавания для таких языков окажется меньше, чем сложность задачи распознавания для контекстно-свободных языков. Примеры таких языков мы рассмотрели в начале этой лекции. ▲

10.6 Обсуждение

In the lecture you introduced a structural construction the particular cases of which are regular and context-free grammars. Compared with the grammars,

the construction has additional tools at hand, by which not only regular and context-free languages, but also other sets of various forms can be defined. In the first place I would like to ask you what tools have made such generalisation possible. Then I will ask other questions.

The first step to generalisation is that the function $P: K \times K \times K \rightarrow \{0, 1\}$ expressing rules in common grammars is replaced by a function of a more general form $P: K \times K \times K \rightarrow W$ for an appropriate semi-ring W . The result is that not only languages can be defined, but even functions which are defined on a set of sequences.

You have already presented this generalisation and used it in previous lectures. I am interested in further steps in generalisation which appeared in this lecture.

It is important that we have created the construction without applying the concept of a sequence. In this way we have achieved that the sets of more diverse mathematical objects can be defined. These objects are not obligatory sequences, but they can be sequences too. Together with the generalisation, which was our first step, we obtained a construction not only for the definition of admissible observed sets, but also for the definition of certain functions on observed sets. Thus the construction has assumed a quite strong and general power.

I would like to be more at home with the problem. I do not understand properly why it is necessary to formalise observation in any other way than as a sequence. Of course, I understand that an image is something different than a sentence. However, I do not know why one should formalise these two representations of information in different manner. Well, even when in terms of the general structural construction I say that an observation is an ensemble of structural elements s_1, s_2, \dots, s_n , I still write it down as a sequence. Why could not the expression I have just written be called a sequence?

You have come across the same difficulties that pattern recognition encountered in the 1960s, when formalisation of an observation by a point in a multi-dimensional space seemed to be universally applicable. It was found even at that time that it was necessary to make a break with that charming idea. Actually, it is not important how the observation s_1, s_2, \dots, s_n is called. An observation can be called a vector, a sequence, a set, etc.. It is essential what operations on this object are considered understandable in a particular application. If years ago it was found that the formalisation of observation by means of the multi-dimensional vector is not convenient for the purpose then it resulted in something more serious than in replacing the word vector by another word. It meant that operations and concepts resulting from formalisation by means of vectors (such as vector addition, vector scalar product, hyperplane, convex subset, etc.) did not correspond to some applied problems. In some applications

nothing understandable corresponds to concepts related to vectors. Therefore in the formulation of such tasks and their solutions vectors should not be used. Finally, when nothing but the word vector itself remained from vector formulation then it was evident that another formalisation of observation should be introduced as well.

The case with sequences is similar. The question is not whether the observation s_1, s_2, \dots, s_n is called a sequence. It is of importance if the operations on sequences (such as concatenation, iteration, deleting a connected subsequence, etc.) state something that is understandable even in the concepts of the applied problem you are to master. And vice versa, if all properties of observation in your application can be easily expressed by means of operations that are natural for sequences. From the point of view of structural construction, a sequence is nothing else but an auxiliary concept representing structures in an illustrative way in that precise meaning in which structures were used at the lecture. For some structures the concept of sequence is beneficial, for others it is not. When saying that the observation in a particular application is not a sequence then we mean that the structure of the observation is expressed clumsily, not illustratively, through operations pertaining for sequences. You yourself can admit that some sets T and some structures \mathcal{T} lose their entire illustrative character when they are represented as a set of integers.

Everything depends on representation, i.e., how the elements of a set T are numbered.

Not everything. There are sets T having a completely clear structure \mathcal{T} , which loses its lucidity with any mapping of the set T to a set of integers.

For example?

An example is a two-dimensional integer-number lattice, i.e., a structure which is defined as a set of rectangles of finite dimensions.

I understand that the correct definition of the set T and its corresponding structure \mathcal{T} is the most important step in representing an applied problem by means of a structural construction.

We would still add to it the selection of an alphabet V for the labelling of structural elements.

Let me put the alphabet aside, for a while. Let me even put aside the structure \mathcal{T} , because I agree with you that the words by which I will determine what set T is referred to will immediately delimit the structure \mathcal{T} natural for a set.

Now I am coming to the main question. The entire structural construction is a tool for defining some sets. But at least one element in the construction is again a set. It is the set T , with respect to it nothing at all is stated, neither how it should be defined nor what its form could be. Briefly speaking, nothing

is said about it and therefore I can consider it as an arbitrary set. Thus, the entire structural construction stops being constructive. To define a set (here a set of admissible observations) I must have defined another set. And this set is just the set T .

Indeed, you have revealed the weakest point in the proposed structural construction. The definition of a set of admissible observations cannot start with the words 'let T be a set' because it is a too general sentence. At least, we should say 'let T be a finite set' or otherwise strongly limit the set T , so that further doing with respect to construction might become correct. We did not do it for different reasons, and so understand the sentence in the following informal sense 'let T be a set which is easy to define and quite obviously results from the applied problem being solved'.

But still I do not understand why the form of the set T could not be reasonably limited for the whole construction to become correct and in spite of it to contain all possible sets that can practically occur. Well, the variety of the sets T which are of practical interest is not very large. It can be a completely ordered set, e.g., a set of integers, or it can be a Cartesian product of a finite number of such sets. What else do I dare to ask?

And situations should be added in which the observation is a function defined on the vertices of an acyclic graph and also on the Cartesian product of a finite number of such graphs.

And would that really be all for the present?

Hardly all. Objects in images are commonly represented by contours. It is a function the domain of definition of which is a closed curve (a cycle), i.e., something which is not covered by the two previous cases. We do not intend to exclude beforehand the analysis of such objects.

Let us sum up what has been discussed by admitting that something is still missing in the proposed structural construction to make it constructive. But in actual application, the construction can be made precise to such an extent to become constructive. It is possible with the sets T at least which have been just mentioned. We do not intend just now to deal with describing the form of all observed fields which can occur in practice. The first reason is that we simply do not know the form. Every time we tried to limit the form of sets T , after some time we encountered a new practical problem in which the limitation had not been satisfied. But the main idea remained valid in the proposed construction. Secondly (and this is essential), in a general view of the task, not confused by useless details, the real simplicity of fundamental concepts of structural pattern recognition is being revealed.

These are really simple. Further procedure in structural construction seems to me so unsophisticated that I have been in fear so far that I might again not

understand something important. I would like to make sure that I understand the simplicity in a correct way. I will quote one of the possible implementations of the computational procedure for solving the basic problem.

Do it, but only after a while. We have not answered your first question yet. You asked us by means of what additional tools it was possible to use the extended potentiality of structural construction when compared with formal grammars.

You have certainly noticed that rules in the structural construction have another form than those in formal grammars. In the structural construction the rule is a triplet of structural elements (s_1, s_2, s_3) , where each element is a labelled fragment, i.e., a pair of the form (T, v) , $T' \in \mathcal{T}$, $v' \in V$. Thus a rule is the six-tuplet $\langle T_1, v_1, T_2, v_2, T_3, v_3 \rangle$. A rule in formal grammars has a simpler form. It is a triplet of labels and each grammar is characterised by a subset of triplets that is determined by the function $V \times V \times V \rightarrow \{0, 1\}$ which will be denoted PV . If the language of a classical formal grammar is expressed by means of a general structural construction then the function $P: \mathcal{T} \times V \times \mathcal{T} \times V \times \mathcal{T} \times V \rightarrow \{0, 1\}$ will have the form

$$P(T_1, v_1, T_2, v_2, T_3, v_3) = PT(T_1, T_2, T_3) \times PV(v_1, v_2, v_3). \quad (10.29)$$

Formal grammars are, therefore, a particular case of the structural construction in which the function P has the form of (10.29). Moreover, one single function $PT: \mathcal{T} \times \mathcal{T} \times \mathcal{T} \rightarrow \{0, 1\}$ must be used for any regular grammars and another single function for any context-free grammar.

With grammars the applicability of a rule does not depend on fragments, whereas with the structural construction it does. The triplet of labels v_1, v_2, v_3 can be admissible for one selected triplet of fragments T_1, T_2, T_3 and can be inadmissible for some other triple. Do I understand it correctly?

Yes, you do!

I will show how I understand the computational procedure solving the basic problem. I will present it for the case in which the function P assumes only two values, 0 and 1, and these values can be subject to logical addition and multiplication. During your lectures I got used to all other cases, seemingly more complicated, being in fact exactly as simple as this lucid one.

The structural recognition task is understood by me in such a way that a set of objects s_1, s_2, \dots, s_n (I have nearly said a sequence!) is given. It is to be found whether this set is admissible. In other words, it is to be checked whether the objects presented can be understood as parts of a composed object. I imagine the following procedure for solving the task. The set S^i , $i = 0, 1, 2, \dots$ of objects is being created step by step which are regarded as examined. At the beginning the set S^0 is represented by the set $\{s_1, s_2, \dots, s_n\}$ which was presented for analysis. Let a set S^{i-1} be created after the step $(i-1)$. Then a triplet of objects (s', s'', s''') is sought for which $P(s', s'', s''') = 1$, $s' \notin S^{i-1}$, $s'' \in S^{i-1}$,

$s''' \in S^{i-1}$, holds, and the set $S^i = S^{i-1} \cup \{s'\}$ is being created. Simply speaking, in each step the set of already found partial objects is increased by one more object the existence of which was proved in that step. This procedure of creating the set S continues until it is possible with respect to the function P .

I regard the gradual growing of the set presented as the most essential part of structural recognition. I would even say that it is its property. After the set S is created in the manner described some details are to be completed for its interpretation which I do not consider important. If I did not make any mistake somewhere then the simplicity is all too much remarkable. Moreover, I would say that all pattern recognition algorithms which we have discussed so far since Lecture 8 on Markovian sequences have been successfully packed into a simple procedure.

Do not wonder at it. A general view of the class of tasks (if possible at all) allows to see the properties of the tasks which are difficult to observe when the tasks are analysed apart. Well, usually if you intend to know, e.g., a large building, you had better move away from it a little than come nearer to it. It is similar to the situation we have already spoken about. As long as people have counted one type of objects by pairs, another by dozens, the third by tens, and the fourth by three scores, the manipulation with quantities seemed to be very complicated. Only since the time when unified representation of quantities was proposed, counting has been accessible to every child.

You have grasped well that part in structural recognition which does not change in nearly every application. You have used correct expressions, except for using the concept of object instead of the concept of structural element.

It seems to me that in this way the main idea was pointed out more illustratively.

It may be so, but not to get confused let us go back to the terminology introduced in the lecture.

The procedure you presented can be formulated more concretely to add to the subject matter already grasped an elucidation concerning the computational complexity of the algorithm, and to take into consideration factors that influence the complexity. Moreover, we will more precisely state which part of the algorithm is changed when one goes on from one application to the another.

We will create the universal algorithm. This means that its input are observations $\{s_1, s_2, \dots, s_n\}$ and the function P determining which structural analysis is to be performed on the particular observation. The function takes its values from the set W . Two operations \oplus and \otimes of the form $W \times W \rightarrow W$ are given which form a semi-ring on the set W . These operations are also defined in input data.

We will order structural elements in such a way that the structural element $s' = (T', v')$ precedes the structural element $s'' = (T'', v'')$ if $T' \subset T''$ and $T' \neq T''$. The ordering will be denoted $s' \prec s''$. If $T' = T''$ then the elements

s' and s'' will be arbitrarily ordered, either as $s' \prec s''$ or $s'' \prec s'$. The order defined will be regarded as fixed. We will order the rules π , i.e., triplets of structural elements (s_1, s_2, s_3) so that the rule $\pi' = (s'_1, s'_2, s'_3)$ precedes the rule $\pi'' = (s''_1, s''_2, s''_3)$, $\pi' \prec \pi''$, if $s'_1 \prec s''_1$. The algorithm consists of the following operations.

Algorithm 10.2 Structural construction.

1. For each element s , the $G(s) = 0^\oplus$ is defined, where the quantity 0^\oplus is taken from the input data.
2. For each element s_i from the input data $\{s_1, s_2, \dots, s_m\}$ the $G(s_i) = 1^\otimes$ is defined, where the quantity 1^\otimes is also taken from the input data.
3. The rules π are examined in a beforehand given order, and for each $\pi = (s_1, s_2, s_3)$ the quantity $G(s_1)$ is modified by the operator

$$G(s_1) := G(s_1) \oplus (G(s_2) \otimes P(s_1, s_2, s_3) \otimes G(s_3)),$$

where the operations \oplus and \otimes are determined from the input data.

From the expression for the algorithm there immediately follows that its computational time linearly depends on the number of the rules π for which $P(\pi) \neq 0^\oplus$, since each such rule is applied only once.

I can now see from the description that in every actual application of the structural construction I have to do a lot of work. Its result can be considered as a program which creates a sequence of triplets of structural elements (s_1, s_2, s_3) . These triplets together with the quantities $P(s_1, s_2, s_3)$ are then provided to the program which already is an invariant, i.e., it does not depend on the selected application. It is painful and often annoying work. It seems to me that structural analysis of data is simple only under the condition that somebody has already done all the unpleasant work beforehand. And this 'somebody' may be I, myself!

This is usual in applied informatics. We will remind you once more that no formalisation, including formal methods of pattern recognition, is a magical means for lazybones like The Magic Table fairy tales. No matter how well elaborated and lucid the formalism may be, it does not relieve the researcher of the pains of representing an informally conceived task in the particular formalism.

I seem to be closer to the lazybones dreaming of that magic means. Is it not possible to formalise this painful work in a narrower domain at least, for example, for one-dimensional context-free grammars? I am speaking about a system such that for the given set of sequences would be either capable of creating a context-free grammar for generating that set of sequences, or would assert with certainty that such a grammar does not exist. I have noticed several articles that aim at solving tasks of such a form.

Yes, of course. But realise please that we all (by which not only the three of us but the whole pattern recognition community are meant) are only at the beginning of a long path.

And what might the first steps along this path be like?

Keeping in line with our course, it is quite natural that first of all the learning task should be formulated correctly. This means that in formulating the task the insolvability of one task and the uselessness of others should be taken into account. The analysis of the learning task for regular languages we were dealing with in Lecture 8 can be regarded as the zero step in the due direction.

And now, have another considered look at the structural construction which we have proposed in this lecture. Different generalisations of regular and context-free languages which can be defined by means of construction also contain a stochastic generalisation of context-free languages. This means that by means of structural construction not only a certain context-free language can be defined but also the corresponding probability distribution on such a language. Spare some good thought for this as it is by no means trivial. Stochastic modification of context-free languages is not as simple as that in the case of regular languages. The probability distribution on a set of rules applied in grammars hardly ever determines the probability distribution on a set of sequences. It is a known problem, which within structural construction can be overcome thanks to the rule not being considered as a triplet of labels, as it is in grammars, but as a triplet of labelled fragments. Think it out yourself because it is worth considering. Now, however, it is essential that by means of structural construction varied probability distributions on a set of observations can be defined. A particular case is the probability distribution of a certain form on the context-free language. We regard it as a basis when formulating a task of a statistical estimation of a stochastic context-free grammar with respect to the observation of a finite set of random sequences. The first step in solving a learning task for structural recognition could be that for context-free grammars all the results should be repeated which were demonstrated for regular grammars in Lecture 8.

I am nearly sure that I have understood you in the right way, but still I would like to be certain about it. Structural construction is based on one type of concept and the formulation of statistical learning tasks is based on other concepts. In my opinion their mutual correspondence is as follows. The observed parameter of an object is an image, and the hidden parameter is a hierarchical map. An unknown parameter that determines the joint probability of the image and the hierarchical map is the probability distribution on a set of rules, where each rule is understood as a triplet of labelled structural elements and not as a triplet of labels.

You have understood it in the correct way.

It seems to me that now I could be able to create an understandable formulation of a learning task in structural pattern recognition and to find a practically applicable algorithm for its solution.

We have had no doubts about it, but in spite of that we are glad to hear it from you. We would like to thank you for your patience and the ideas you have contributed.

January 12, 1999.

10.7 Bibliographical notes

The subject matter explained in this lecture on context-free grammars is to a great extent original. The Cocke–Younger–Kasami algorithm is described in [Aho and Ullman, 1971] as well as in the original publications [Kasami, 1965; Younger, 1967].

The design of two-dimensional context-free grammars is owed to Schlesinger [Schlesinger, 1989].

Список литературы

- Aho, A., Hopcroft, J., and Ullman, J. (1975). *The design and analysis of computer algorithms*. Addison-Wesley, Reading, Mass.
- Aho, A. and Ullman, J. (1971). *The theory of parsing, translation, and compiling*, volume 1 – *Parsing*. Prentice-Hall, Englewood Cliff, New Jersey.
- Ajzerman, M., Braverman, E., and Rozoner, L. (1970). *Metod potencialnykh funkcij v teorii obucenia mashin*; in Russian (*The method of potential functions in machine learning theory*). Nauka, Moskva.
- Amengual, J. and Vidal, E. (1996). Two different approaches for cost-efficient Viterbi parsing with error correction. In *Proceedings of the 5th International Workshop Advances in Structural and Syntactical Pattern Recognition, Leipzig*, pages 30–39, Heidelberg, Germany. Springer-Verlag, Lecture Notes in Computer Science 1121.
- Anderson, T. (1958). *An introduction to multivariate statistical analysis*. John Wiley, New York, USA.
- Anderson, T. and Bahadur, R. (1962). Classification into two multivariate normal distributions with different covariance matrices. *Annals of Mathematical Statistics*, 33:420–431.
- Ball, G. and Hall, D. (1967). A clustering technique for summarizing multivariate data. *Behavioral Science*, 12:153–155.
- Baum, L., Petrie, T., Soules, G., and Weiss, M. (1970). A maximization technique occurring in the statistical analysis of probabilistic functions of Markov chains. *Annals of Mathematical Statistics*, 41:164–167.
- Bayes, T. (1763). An essay towards solving a problem in the doctrine of chance. *Philosophical Transactions of the Royal Society*, London. Reprinted in *Biometrika*, 45:298–315, 1958.
- Bellman, R. and Dreyfus, S. E. (1962). *Applied dynamic programming*. Princeton University Press, Princeton, New Jersey.
- Bishop, C. (1996). *Neural networks for pattern recognition*. Oxford University Press.

- Boruvka, O. (1926). O jistém problému minimálním; in Czech, (On a minimal problem). *Práce moravské přírodovědecké společnosti*, III(3):37–58.
- Boser, B., Guyon, I., and Vapnik, V. (1992). A training algorithm for optimal margin classifiers. In D. Haussler, editor, *5th Annual ACM Workshop on COLT*, pages 144–152, Pittsburgh, PA. ACM Press.
- Bunke, H. (1996). Structural and syntactic pattern recognition. In Chen, C., Pau, L., and Wang, P., editors, *Handbook of Pattern Recognition and Computer Vision*, chapter 1.5, pages 163–209. World Scientific, Singapore.
- Cayley, A. (1889). A theory on trees. *Quarterly Journal of Pure and Applied Mathematics*, 23:376–378.
- Chazen, E. (1968). *Metody optimalnych statisticheskikh reshenij i zadachi optimalnogo upravlenija*; in Russian (*Methods of optimal statistical solutions in optimal control tasks*). Sovetskoe radio, Moskva.
- Chen, C., Pau, L., and Wang, P., editors (1993). *Handbook of Pattern Recognition and Computer Vision*, Singapore. World Scientific.
- Chomsky, N. (1957). *Syntactic structures*. Mouton, The Hague.
- Chomsky, N., Allen, J., and Van Buren, P. (1971). *Chomsky: Selected Readings*. Oxford University Press, London-New York.
- Chow, C. (1965). Statistical independence and threshold functions. *IEEE Transactions on Computers*, 14:247–252.
- Chow, C. and Liu, C. (1968). Approximating discrete probability distributions with dependence trees. *IEEE Transactions on Information Theory*, 14:462–467.
- Cooper, D. and Cooper, P. (1964). Nonsupervised adaptive signal detection and pattern recognition. *Information and Control*, 7:416–444.
- Demster, A., Laird, N., and Rubin, D. (1977). Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistic Society*, B39:1–38.
- Devijver, P. and Kittler, J. (1982). *Pattern recognition: A statistical approach*. Prentice-Hall, Englewood Cliffs, NJ.
- Devroye, L., Györfi, L., and Lugosi, G. (1996). *A probabilistic theory of pattern recognition*. Springer-Verlag, New York.
- Duda, R. and Hart, P. (1973). *Pattern classification and scene analysis*. John Wiley and Sons, New York.
- Duda, R. O., Hart, P. E., and Stork, D. G. (2001). *Pattern Classification*. John Wiley & Sons, New York, USA.
- Fisher, R. (1936). The use of multiple measurements in taxonomic problems. *Annals of Eugenics*, 7, Part II:179–188.
- Floyd, R. (1962). Algorithm 97, Shortest path. *Communications of the ACM*, 5(6):345.
- Franc, V. and Hlaváč, V. (2001). A simple learning algorithm for maximal margin classifier. In Leonardis, A. and Bischof, H., editors, *Proceedings of the Workshop on Kernel & Subspace Methods for Computer Vision, Wien, Austria, August 25, 2001, adjoint to the International Conference on Artificial Neural Networks 2001*, pages 1–11. Technische Universität Wien, Austria.

- Fukunaga, K. (1990). *Introduction to statistical pattern recognition*. Academic Press, Boston, 2nd edition.
- Glushkov, V. (1962a). K voprosu o samoobuchenii v perceptrone; in russian (On the question of self-learning in the perceptron). *Zhurnal matematiki i matematicheskoy fiziki*, (6):1102–1110.
- Glushkov, V. (1962b). Teoria obuchenia odnogo klassa diskretnykh perceptronov; in Russian (Theory of learning of one class of discrete perceptrons). *Zhurnal matematiki i matematicheskoy fiziki*, (2):317–335.
- Grim, J. (1986). On numerical evaluation of maximum likelihood estimates for finite mixture of distributions. *Kybernetika*, 18:173–190.
- Halmos, P. R. (1971). *Kak pisaty matematicheskie texty*; in Russian (*How to write mathematical texts*). Number 5. Uspechi matematicheskikh nauk, Moskva.
- Jakubovich, V. A. (1966). Rekurrentnyje konechno schodjashchijesja algoritmy reshenija sistem neravenstv; in Russian (Recurrent finite converging algorithms solving a system of inequalities). *Doklady Akademii nauk*, 166(6):1308–1311.
- Jakubovich, V. A. (1969). Ob odnoj zadache samoobucheniya celesoobraznomu povedeniju; in Russian (On self-learning task related to goal driven behaviour). *Doklady Akademii nauk*, 189(3):495–498.
- Jimenez, V. and Marzal, A. (2000). Computation of the n best parse trees for weighted and stochastic context-free grammars. In Ferri, F., editor, *Proceedings of the Joint IAPR Workshops Structural and Syntactic Pattern Recognition and Statistical Pattern Recognition, Alicante, Spain, August 30–September 1, 2000*, volume Lecture Notes in Computer Science 1876, pages 183–192, Berlin. Springer-Verlag.
- Kasami, T. (1965). An efficient recognition and syntax analysis algorithm for context-free languages. Scientific report AFCLR-65-758, Air Force Cambridge Research Laboratory, Bedford, Mass., USA.
- Kashyap, L. and Oommen, B. (1984). String correction using probabilistic methods. *Pattern Recognition Letters*, 2(3):147–154.
- Kovalevski, V. (1965). Zadacha rozpoznavanja obrazov s točki zrenia matematicheskoy statistiki; in Russian (Pattern recognition tasks from the standpoint of mathematical statistics). In *Chitajushchije avtomaty (Reading automata)*, pages 3–41. Naukova Dumka, Kiev.
- Kovalevski, V. (1967). Optimalnyj algoritm rozpoznavanja nekotorykh posledovatelnostij izobrazhenij; in russian (Optimal algorithm recognizing some sequences of images). *Kybernetika*, (4):75–80.
- Kovalevski, V. (1969). Sequential optimization in pattern recognition and pattern description. In *Proceedings of the IFIP Congress, Amsterdam, 1968*, New York. Academic Press. Earlier version in Russian: Posledovatelnaja optimizacija v zadachach rozpoznavanija i opisaniya izobrazhenij, in proceedings: Raspoznavaniye obrazov i konstruirovaniye chitajushchich avtomatov, Institut Kibernetiki Academy of Sciences USSR, Kiev 1967, pages 3–26.
- Kozinec, B. (1973). Rekurentnyj algoritm razdelenia vypuklych oboloček dvuch mnozhestv; in Russian (Recurrent algorithm separating convex hulls

- of two sets). In Vapnik, V., editor, *Algoritmy obuchenia raspoznavania (Learning algorithms in pattern recognition)*, pages 43–50. Sovetskoye radio, Moskva.
- Kruskal, J. (1956). On the shortest spanning subtree of a graph and the traveling salesman problem. *Proceedings of the American Mathematical Society*, 7:48–50.
- Kuhn, H. and Tucker, A. (1950). Nonlinear programming. In *Proceedings of the Second Berkeley Symposium on Mathematical Statistics and Probability*, pages 481–492, Berkeley, Calif.
- Lehmann, E. (1959). *Testing statistical hypotheses*. John Wiley, New York.
- Levenstein, V. (1965). Dvojichnyje kody s ispravlenijem vypadenij, vstavok i zameshchenij simvolov; in Russian (Binary coded correcting deletions, insertions and replaces of symbols). *Doklady Akademii nauk SSSR*, 163(4):840–850.
- Linnik, J. (1966). *Statisticheskie zadachi s meshajushchimi parametrami*; in Russian (*Statistical tasks with intervening parameters*). Nauka, Moskva.
- Markov, A. (1916). Ob odnom primenenii statisticeskogo metoda; in Russian (An application of statistical method). *Izvestia imperialisticeskoj akademii nauk*, 6(4):239–242.
- Minsky, M. and Papert, S. (1969). *Perceptrons: An introduction to computational geometry*. MIT Press, Cambridge, Mass., USA. 2nd edition in 1988.
- Nadler, M. and Smith, E. (1993). *Pattern recognition engineering*. John Wiley and sons, New York, USA.
- Neyman, J. (1962). Two breakthroughs in the theory of statistical decision making. *Review de l'Inst. Intern. de Stat.*, 30(1):11–27.
- Neyman, J. and Pearson, E. (1928). On the use and interpretation of certain test criteria for purposes of statistical inference. *Biometrika*, 20A:175–240.
- Neyman, J. and Pearson, E. (1933). On the problem of the most efficient tests of statistical hypotheses. *Phil. Trans. Royal Soc. London*, 231:289–337.
- Nilsson, N. (1965). *Learning machine: Foundation of trainable pattern recognition classifying systems*. McGraw-Hill, New York.
- Novikoff, A. (1962). On convergence proofs for perceptrons. In *Proceedings of the Symposium on Mathematical Theory of Automata*, volume 12, pages 615–622, Brooklyn, New York. Polytechnic Institute of Brooklyn.
- Oommen, B. (1987). Recognition of noisy subsequences using constrained edit distances. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 9:676–685.
- Pavel, M. (1993). *Fundamentals of pattern recognition*. Marcel Dekker, Inc., New York, USA.
- Raudys, S. and Pikelis, V. (1980). On dimensionality, sample size, classification error, and complexity of classification algorithm in pattern recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 1:7–13.
- Robbins, H. (1951). Asymptotically subminimax solutions of compound statistical decision problems. In *Proceedings of the second Berkeley symposium on mathematical statistics and probability*, pages 131–148, Los

- Angeles. University of California Press.
- Robbins, H. (1956). An empirical Bayes approach to statistics. In *Proceedings of the Third Berkeley Symposium on Mathematical Statistics and Probability*, pages 157–163, Los Angeles. University of California Press.
- Rosenblatt, F. (1957). The perceptron: The perceiving and recognizing automaton. Technical Report 85-460-1, Cornell University, Aeronautical Lab., USA. Project PARA.
- Rosenblatt, F. (1959). *Two theorems of statistical separability in the perceptron*. H.M. Stat. Office, London.
- Rosenblatt, F. (1962). *Principles of neurodynamics: Perceptron and theory of brain mechanisms*. Spartan Books, Washington, D.C.
- Schlesinger, M. (1965). O samoproizvolnom razlichenii obrazov; in Russian (On automatic separation of patterns). In *Chitajushchie avtomaty (Reading Automata)*, pages 38–45. Naukova Dumka, Kiev.
- Schlesinger, M. (1968). Vzaimosvjaz obuchenija i samoobuchenija v raspoznavaniji obrazov; in Russian (Relation between learning and self-learning in pattern recognition). *Kibernetika*, (2):81–88.
- Schlesinger, M. (1972a). Issledovanie odnogo klassa razpoznajushich ustrojstv reshajushchich zadach proverki slozhnych hipotez; in Russian (Study of one class of recognition devices solving tasks of analysis of complex hypotheses). *Avtomatika*, (2):38–42.
- Schlesinger, M. (1972b). Sintez linejnogo reshajushhego pravila dla odnogo klassa zadach; in Russian (Synthesis of linear discrimination rule for one class of tasks). *Izdatelstvo Akademii nauk*, (5):157–160.
- Schlesinger, M. (1979a). Dopolnitelnyje sledstvia teorii dvoystvennosti v nebayesovskich zadachach raspoznavania; in Russian (Additional consequences of the duality theory in non-Bayesian recognition tasks). In *Raspoznavanie graficheskich i zvukovykh signalov (Recognition of graphic and sound signals)*, pages 36–47, Kiev. Institut Kibernetiki AV USSR.
- Schlesinger, M. (1979b). Teoria dvoystvennosti v nebayesovskich zadachach raspoznavania; in Russian (Theory of duality in non-Bayesian recognition tasks). In *Raspoznavanie graficheskich i zvukovykh signalov (Recognition of graphic and sound signals)*, pages 21–35, Kiev. Institut Kibernetiki AV USSR.
- Schlesinger, M. (1989). *Matematicheskie sredstva obrabotki izobrazenij*; in Russian (*Mathematical tools for image processing*). Naukova Dumka, Kiev.
- Schlesinger, M. (1994). Systeme von Funktionsoperationen angewendet auf eine Aufgabe der besten Ebereinstimmung; in German (Systems of function operations applied to one task of the best tuning). *Wissenschaftliche Beitrage zur Informatik*, Fakultat fuer Informatik Technische Universitaet Dresden, Germany, 7(3):62–79.
- Schlesinger, M. (1997). Algebraic method for solution of some best matching problems. In *Advances in Computer Vision, Proceedings of the Dagstuhl Seminar, Saarland, Germany, March 1997*, pages 201–210, Wien. Springer Verlag.
- Schlesinger, M. and Gimmel'farb, G. (1987). Vozmozhnosti sovremennogo

- raspoznavanija obrazov v prikladnom raspoznavaniji izobrazhenij; in Russian (Possibilities of current pattern recognition in applied image analysis). *Upravljajuchshije sistemy i mashiny*, (6):21–28.
- Schlesinger, M., Kalmykov, V., and Suchorukov, A. (1981). Sravnitelnyj analiz algoritmov sinteza linejnogo reshajushchego pravila dlja proverki slozhnyh gipotez; in Russian (Comparative analysis of algorithms synthesising linear decision rule for analysis of complex hypotheses). *Automatika*, (1):3–9.
- Schlesinger, M. and Svjatogor, L. (1967). O postrojenii etalonov dlja korelacionnyh chitajushchich automatov; in Russian (On creating etalons for correlation-based reading automata). In *III. vsjesojuznaja konferencija po informacionno poiskovym sistemam i automatizovannoj obrabotke nauchno-technicheskoj informacii* (3rd all-Soviet Union conference on information-searching systems and automatic processing of scientific and technological information), volume 3, pages 129–139, Moskva. Vsesojuznyj institut nauchnoj i technicheskoj informacii.
- Shor, N. (1979). *Metody minimizacii nedifferenciruemyh funkcij i ich prilozhenia*; in Russian (*Methods of non-differentiable functions minimizations and their applications*). Naukova Dumka, Kiev, Ukraine.
- Shor, N. (1998). *Nondifferentiable optimization and polynomial problems*. Kluwer Academic Publisher, Dordrecht, The Netherlands.
- Theodoridis, S. and Koutroubas, K. (1999). *Pattern Recognition*. Academic Press, San Diego, USA.
- Vapnik, V. (1995). *The nature of statistical learning theory*. Springer-Verlag, New York.
- Vapnik, V. (1998). *Statistical learning theory*. Adaptive and Learning Systems. Wiley, New York, New York, USA.
- Vapnik, V. and Chervonenkis, A. (1974). *Teoria raspoznavania obrazov, statisticheskie problemy obuchenia*; in Russian (*Pattern recognition theory, statistical learning problems*). Nauka, Moskva.
- Vidyasagar, M. (1996). *Theory of Learning and Generalization. With Application to Neural Networks and Control Systems*. Springer.
- Vincjuk, T. Rospoznavanie ustnoj rechi metodami dinamicheskogo programmirovania; in Russian (Speech recognition by means of dynamic programming).
- Viterbi, A. (1967). Convolutional codes and their performance in communications systems. *IEEE Transactions on Communications Technology*, 13(2):260–269.
- Waerden, B. v. d. (1957). *Mathematische Statistik*; in German (*Mathematical statistics*). Springer-Verlag, Berlin-Goettingen-Heidelberg.
- Wagner, R. and Fischer, M. (1974). The string-to-string correction problem. *Journal of ACM*, 21(1):168–173.
- Wagner, R. and Seiferas, J. (1978). Correcting counter-automaton-recognizable languages. *SIAM Journal of Computing*, 7(3):357–375.
- Wald, A. (1947). *Sequential analysis*. John Wiley, New York.
- Wald, A. (1950). Basic ideas of a general theory of statistical decision rules,. In *Proceedings of the International Congress of Mathematicians*, volume I. Russian translation, A. Wald: Posledovatelnyj analiz, Gosudarstvenoe

- izdatelstvo fiziko-matematicheskoy literatury, Moskva 1960, paper in Appendix., pages 308-325.
- Wald, A. and Wolfowitz, J. (1948). Optimum character of the sequential ratio test. *Ann. Math. Stat.*, 19(3):326–339.
- Wu, C. (1983). On the convergence properties of the EM algorithm. *Annals of Statistics*, 11:95–103.
- Young, T., editor (1994). *Handbook of Pattern Recognition and Image Processing: Computer Vision*, volume 2, San Diego, USA. Academic Press.
- Younger, D. (1967). Recognition of context-free languages in time n^3 . *Information and Control*, 10:189–208.
- Zadeh, L. (1965). Fuzzy sets. *Information and Control*, 8:338–353.
- Zagorujko, N. (1999). *Prikladnyje metoda nalaliza danykh i znaniy*; in Russian (*Applied methods of data and knowledge analysis*). Izdatelstvo Instituta Matematiki, Novosibirsk, Russia.
- Zimmermann, H., Zadeh, L., and Gaines, B. (1984). *Fuzzy sets and decision analysis*. North Holland, Amsterdam-New York.
- Zuchovickij, S. and Avdejeva, L. (1967). *Linejnoje i vypukloje programmirovaniye*; in Russian (*Linear and convex programming*). Nauka, Moskva.

Предметный указатель

Algorithm

- ε -оптимальное разделение конечных множеств точек, 191
 - ε -solution, generalised Anderson task, 204
 - constructive calculation of Levenstein dissimilarity, 465
 - Kozinec, 186
 - max. likelihood estimate for Markovian object, 375
 - minimax estimate for Markovian object, 378
 - recognition, of stochastic automaton, 334
 - unsupervised learning, 254
- automaton
- penalised, 432
 - stochastic, finite, 332

Bayes, 14

- Bayesian task, 14
- bilinear function, 51
- Boruvka, 410

Cartesian hull, 109

- change, 437
- Chervonenkis, 135
- classifier

- by integral of probability, 116
 - Fisher, 194
 - nearest neighbour, 115
- cluster analysis, 248
- conditional independence of features, 94
- cone, convex, 18
- convex cone, 18
- convolution, 447
- convolution generalised, 447

Decision, 14

- decision function, 14
- delete, 437
- deterministic learning, 145
- deterministic strategy, 56

False alarm, 44

- feature
- conditional independence, 94
 - of the object, 14
- fragment of the field, 521
- function
- bilinear, 51
 - decision, 14
 - discriminant linear, 161
 - Levenstein, 437
 - penalty, 14
 - penalty, quadratic, 31
 - quadratic discriminant, 95

- Generalised**
– convolution, 447
grammar
– context-free, 509
- Hull Cartesian**, 109
- Image**
– labelled, 522
insert, 437
- Kozinec**, 186
Kuhn, 52
- Labelled image**, 522
language
– context-free, 509
– regular, 423
learning
– deterministic, 145
– Markovian sequence, 368
Levenstein
– dissimilarity, 437, 438, 441
– function, 437
likelihood, 17
likelihood ratio, 17
linear discriminant function, 161
linear separability, 19
Linnik, 49
- Map**
– hierarchical, 522
matrix
– multiplication, generalised, 349
multi-set, training, 124
multiplication
– matrix, generalised, 349
- Nearest neighbour classifier**, 115
Neyman, 43
non-Bayesian task, 39, 233
Novikoff, 188
- Object**
– statistical model, 93
observation, 14
optimisation
– task, 46
overlooked danger, 44
- Parameter**, 14
– hidden, 14
– observable, 14
partial risk, 22
Pearson, 43
penalty function, 14
perceptron, 188
pixel, 521
problem
– exact matching, 431
– structural recognition of images, 524
product
– matrix, generalised, 349
- Quadratic discriminant function**, 95
quality of the strategy, 49, 50
- Ratio**, likelihood, 17
recognition
– from incomplete data, 339
– of Markovian model, 339
– of the stochastic automaton, 333
rectangle, 109
regular language, 423
replace, 437
risk, 14
– Bayesian, 14
– partial, 22
rule of a grammar, 522
- Segmentation**
– hierarchical, 522
– of the fragment, 522
separability
– linear, 19
– nonlinear, 96
sequence, 327
set
– training, 126, 128
state, 14
– of the object, 14
statistical theory of learning, 135

- stochastic automaton
 - recognition, 333
- straightening, feature space, 96
- strategy, 14
 - deterministic, 56
 - quality, 49
 - randomised, 56
- structural analysis, 303
- structural element, 521
- structure
 - element, 521
 - of observed field, 521
- T**ask
 - Anderson, 165
 - Bayesian, 14
 - clustering, 248
 - generalised Neyman, 46
 - generalized Neyman–Pearson, 61
 - learning, 254
 - linear programming, dual, 51
 - linear programming, primal, 51
 - Linnik, 49
 - minimax, 46, 62, 369
 - Neyman–Pearson, 43, 57
 - non-Bayesian, 39
 - recognition, 254
 - Wald, 64
- teacher, 147
- theorem
 - ε -solution, generalised Anderson task, 200
 - duality, first, 52
 - duality, second, 55
 - Kuhn–Tucker, 52
 - Novikoff, 188
- theory
 - statistical, of learning, 135
- training multi-set, 124
- Tucker, 52
- U**nsupervised learning, 256
- V**apnik, 135
- W**ald, 47