В. И. Донской

Алгоритмические модели обучения классификации: обоснование, сравнение, выбор

Симферополь
«ДИАЙПИ»
2014

УДК 519.7
ББК 22.12, 32.81
Д676
Донской В. И.
Д676 Алгоритмические модели обучения классификации: обоснование, сравнение, выбор. – Симферополь: ДИАЙПИ, 2014. – 228 с.
ISBN 978–966–491–534–9

В книге рассматриваются теоретические аспекты машинного обучения классификации. В центре изложения – обучаемость как способность применяемых алгоритмов обеспечивать эмпирическое обобщение. С обучаемостью непосредственно связаны вопросы сложности выборок, точности и надежности классификаторов. Большое внимание уделено алгоритмическим методам анализа процессов обучения и синтеза решающих правил, включая колмогоровский подход, связанный с алгоритмическим сжатием информации. Описаны принципы выбора моделей обучения и семейств классифицирующих алгоритмов в зависимости от постановок и свойств решаемых задач.

Книга предназначается для специалистов, занимающихся теорией машинного обучения; она будет полезной для аспирантов, разработчиков интеллектуализированного программного обеспечения и студентов старших курсов математических специальностей, специализирующихся в указанной области.

Рецензент
Заведующий кафедрой информационных систем управления
факультета прикладной математики и информатики
Белорусского госуниверситета
профессор *Краснопрошин В. В.*

Donskoy V. I.
Algorithmic learning classification models:
justification, comparison, choice. – Simferopol:
DIP, 2014. – 228 p.

The theoretical aspects of machine learning classification are examined in the book. In the center of exposition is learnability as ability of used algorithms to provide empiric generalization. To the learnability are directly related the questions of sample complexity, accuracy, and reliability of classifiers. Much attention is paid to the algorithmic methods of learning processes analysis and decision rules synthesis, including the Kolmogorov' approach related to the algorithmic data compression. Principles of choice of learning models and families of classifying algorithms are described subject to the initial problem statement and properties of the decided tasks.

ISBN 978–966–491–534–9

Предисловие

« ... Что же касается моей неосведомленности, молодой человек, то неужели вы думаете, что я зря сорок лет занимался своими необычными делами и не научился распознавать людей с первого взгляда?»

Р.Хаггард

Теории машинного обучения, классификации, распознаванию образов посвящено такое обширное множество книг и статей, что трудно даже представить библиографию, упоминающую большинство работ в этой области. За шестьдесят с лишним лет развития теории и разработки обучающихся программ и автоматов интерес к проблеме не только не угас, но постоянно усиливается. Это связано с интенсивным внедрением методов машинного обучения и распознавания в самые разные информационные технологии, приборы, устройства. Но прежде всего – в создание интеллектуализированных операционных систем новых поколений.

Идеи, связанные с моделированием мыслительной деятельности человека и реализации этих моделей в компьютерных программах, послужили стимулом к разработке новых разделов теории. Получили развитие такие направления, как индуктивная математика, решение задач в условиях неопределённости, неполноты и противоречивости начальной информации. Расширилось представление о возможностях компьютерных информационных систем.

Прагматизм, проявляющийся в опережающей теорию разработке прикладных систем обучения и классификации, оказался оправданным, и специалисты-теоретики иногда сами удивляются успешности компьютерных программ интеллектуализированного поиска, узнавания, классификации. В итоге большинство книг в этой области главным образом ориентировано на изложение работоспособных алгоритмов машинного обучения для различных семейств классификаторов – нейронных сетей, решающих деревьев, машин опорных векторов и других. Изобретение новых алгорит-

мов, модификация известных, совмещение подходов с целью повышения точности и надежности – также востребованное и широко отражаемое в литературе направление.

В отличие от большинства известных публикаций*, в этой книге основное внимание уделено скорее не конкретным алгоритмам обучения и семействам классификаторов, а теоретическим вопросам обучаемости, исследованию алгоритмических подходов к обучению, выбору семейств решающих правил, адекватных поставленной задаче и математическим особенностям описания начальной информации*. Тем не менее, с целью сравнения и анализа возможностей их применения, основные модели машинного обучения в книге представлены.

Содержание и отбор материала, конечно, отражает предпочтения автора, включает ряд вопросов, которые разрабатывались им лично. Поэтому несколько шире, чем другие классы моделей, представлены бинарные решающие деревья, особенно выбор критериев ветвления при их синтезе, и модели сжатия на основе колмогоровской сложности.

В книге отражена попытка создания типологической схемы для задач машинного обучения классификации и поиска закономерностей. Эта схема предназначена для обоснования выбора подходящего в каждом конкретном случае класса моделей решения поставленной задачи. Предложена классификация основных подходов к разработке процедур машинного обучения.

Несмотря на небольшой объём книги, в ней представлены практически все направления теории машинного обучения классификации – от линейной параметрической адаптации до комбинаторной теории переобучения. Но основными представляются главы 2 и 7: «Машинное обучение и обучаемость» и «Эмпирическое обобщение и классификация: классы задач, классы моделей и применимость теорий». Хотелось бы, чтобы специалисты обязательно с ними познакомились. В этих главах и в главе 4, посвящённой колмогоровской сложности в машинном обучении, содержится ряд новых, не публиковавшихся ранее результатов.

Заканчивая предисловие, хочу выразить глубочайшую признательность и благодарность своим коллегам – специалистам научной школы академика РАН Ю. И. Журавлева, к которой я имею честь принадлежать, и в первую очередь – самому Юрию Ивановичу, – за поддержку, благожелательность, многолетнее сотрудничество, возможность участия в научных конференциях «Математические методы распознавания образов», семинарах ВЦ РАН. Неоценимой на решающих этапах моей научной деятельности была поддержка чл.-корр. РАН К. В. Рудакова, помощь П. П. Кольцова, Д. В. Кочеткова, В. В. Рязанова, В. В. Краснопрошина.

Большую роль в отборе и формировании материала книги сыграли неоднократные обсуждения проблематики теории машинного обучения с К.В. Воронцовым.

Становлению и развитию исследований в области кибернетики, созданию научного коллектива в Таврическом национальном университете, где я проработал сорок лет – от программиста вычислительного центра до профессора, заведующего кафедрой информатики, – способствовали учёные Института кибернетики НАНУ им. В. М. Глушкова. Хочется с благодарностью вспомнить академика В. С. Михалевича, чл.-корр. А. А. Стогния, выразить признательность академику И. В. Сергиенко, чл.-корр. А.М. Гупалу, В.П. Гладуну, П. С. Кнопову, В. И. Норкину и другим учёным, поддержавшим проводившиеся на базе ТНУ научные конференции, открытие журнала «Таврический вестник информатики и математики» и оказавшим помощь в подготовке научных кадров.

В. И. Донской
Март 2014г.

1. Эмпирическая индукция и классификация

«Посредством логики доказывают,
посредством интуиции – изобретают»
А. Пуанкаре

В процессе познания окружающего мира, явлений и законов природы, человечество сформировало ряд приёмов и методов рассуждений и построения выводов. В широком смысле можно говорить, что эти приёмы и методы применяются для решения задач, с которыми сталкивается человек в любой сфере своей деятельности. Например, дедукция (от лат. *deductio* – выведение) представляет собой получение частных выводов на основе знания некоторых общих положений. Можно сказать, это дедукция – вывод от общего к частному.

Получение новых знаний при помощи дедукции используется достаточно широко. Большое значение дедуктивный метод имеет в математике, которая представляется, главным образом, дедуктивной наукой. Хотя с оговоркой: как заметил В.А. Стеклов, «При помощи логики никто ничего не открывает; силлогизм может только приводить к признанию той или другой, уже заранее известной истины, но как орудие изобретения бессилен. Математик иногда наперёд высказывает весьма сложное положение, совершенно не очевидное и затем начинает доказывать его. В изобретении чуть ли не каждого шага доказательства играет роль не логика, а *интуиция, которая идёт поверх всякой логики*».

Простейшая дедуктивная (аксиоматическая) теория определяется конечным набором аксиом (заведомо истинных базовых фактов) и конечной совокупностью правил вывода, при помощи которых из аксиом и уже выведенных фактов (лемм и теорем) можно получать новые факты. Легко понять, что в таком упрощенном представлении дедукция заключена в рамки исходного теоретического построения (аксиомы-правила), и этим построением уже всё, что можно получить, предопределено, хотя процесс получения результатов вывода может быть очень сложным и даже в некоторых случаях нереализуемым алгоритмически. В этом смысле множество результатов дедуктивного вывода с зафиксированным базисом замкнуто, и каждый получаемый результат может считаться новым лишь относительно.

Древнегреческий философ Аристотель (364 – 322 гг. до н.э.) первым разработал теорию дедуктивных умозаключений (силлогизмов), в которых заключение получается из посылок по логическим правилам. Эта теория легла в основу современного понятия математического доказательства. Французский математик и философ Рене Декарт (1596 – 1650) развил дедуктивный метод познания, расширяя его как метод построения дедуктив-

ных (математических) рассуждений над результатами воспроизводимых опытов.

Использование опыта (эмпирики) для поиска решений в естествознании полагали важнейшим научным приёмом такие выдающиеся учёные, как Роджер Бэкон (1214 – 1295) и Леонардо да Винчи (1452 – 1519). Но основателем метода эмпирической индукции (от лат *inductio* – наведение, побуждение; *in* – в, и *duco* – веду) все же по праву считают Фрэнсиса Бэкона (1561 – 1626). Работы Ф. Бэкона явились основанием эмпирико-индуктивного метода научного познания. Индукция как метод, согласно его теории, предполагает проведение эксперимента, наблюдение результатов и порождение гипотез. Этот подход Ф. Бэкон изложил в трактате «Новый органон» [6], вышедшем в свет в 1620 году.

Основные идеи Ф. Бэкона состояли в следующем [19].

Не следует полагаться на сформулированные аксиомы и формальные базовые понятия, какими бы привлекательными и справедливыми они не казались. Законы природы нужно «расшифровывать» из фактов опыта. Следует искать правильный метод анализа и обобщения опытных данных; здесь логика Аристотеля не подходит в силу её абстрактности, оторванности от реальных процессов и явлений.

Ф. Бэкон пытался сформулировать принцип научной индукции [5]. Прежде всего, эмпирические наблюдения систематизировались в виде *таблиц открытия*: *Присутствия*, *Отсутствия* и *Степеней*. Если изучается некоторое свойство, то собирается некоторое достаточное число случаев, когда это свойство *присутствовало*, и множество случаев, когда это свойство *отсутствовало*. Затем выделяется множество случаев, когда наблюдалось *изменение интенсивности* (*степени*) изучаемого свойства. Эти данные составляют три упомянутых таблицы, сравнение которых позволяет выделить факторы, сопутствующие свойству, усиливающие изучаемое свойство, а также факторы, исключающие его. В итоге получается некоторый «остаток» – «*форма*» исследуемого свойства. Аналогии и исключения использовались как важные приемы в составе метода эмпирической индукции и применялись для заполнения таблиц открытия.

В соответствии с теорией Ф. Бэкона, используя эмпирические данные, можно выявить «форму» или, говоря современным языком, закономерность, при помощи которой можно узнать и объяснить: обладает наблюдаемый объект некоторым свойством или нет. Математическое понятие, соответствующее вычислению некоторого свойства S, – это предикат $S : \mathbf{X} \to \{\text{да, нет}\}$ или, что эквивалентно, $S : \mathbf{X} \to \{1,0\}$ – функция, заданная на множестве изучаемых объектов X и принимающая только два значения. Можно сказать, что зная описание предиката S, можно распознавать: обладает объект интересующим исследователя свойством или нет, или, говоря шире, классифицировать объекты по выполнению и невыпол-

нению некоторого свойства. В этом случае предикат S называют *классификатором*.

Нахождение классификатора по набору эмпирических данных составляет центральную задачу (в современной терминологии) *теории машинного обучения и распознавания*. Построение математической теории классификации объектов и явлений стало важнейшей теоретической задачей. Основополагающие, пионерские работы, посвященные становлению этой теории, принадлежат А.Ш. Блоху [2], М.М. Бонгарду [3], Э.М. Браверману [1], В.Н. Вапнику [7], А. Гловацкому [25], Ю.И. Журавлеву [16], Л. Кэналу [27], Н. Нильссону [28], А. Новикову [29], Ф. Розенблатту [30], К.-С. Фу [23,24], Е. Ханту [26] и ряду других ученых.

Легко понять, что имея некоторый, пусть даже огромный, но *не пополняющийся набор исходных данных или аксиом*, можно выявить все свойства, какие только возможно, применяя имеющиеся приёмы вывода решений и построения классификаторов. Но как разорвать замкнутый круг сложившихся представлений, совершить принципиально новое открытие, построить новую теорию? Для этого нужны *новые эмпирические данные*, и постоянно собирая их, человек познаёт окружающий мир.

Французскому физику-теоретику Луи де Бройлю (1892 – 1987) принадлежит следующее высказывание [4]: «Разрывая с помощью иррациональных скачков … жёсткий круг, в который нас заключает дедуктивное рассуждение, *индукция, основанная на воображении и интуиции*, позволяет осуществить великие завоевания мысли; она лежит в основе всех истинных достижений науки».

Если вдуматься в процесс индуктивного обобщения (это – синоним эмпирической индукции), то его можно разделить на две фазы. Первая – *построение классификатора* S, а вторая – его *применение* к произвольному объекту $\widetilde{x} \in \mathbf{X}$. Если классификатор S является алгоритмом, то он представим посредством последовательности правил-команд, применяемых изначально к исходным данным – описанию объекта $\widetilde{x}$. Понятно, что применение алгоритмического классификатора представляет собой дедуктивный вывод, в то время как его построение реализуется индуктивным методом. Поэтому использование эмпирической индукции без дополняющей её дедукции не представляется оправданным, по крайней мере, с точки зрения современного представления о вычислимости.

По мере развития исследований в области физиологии человека удалось установить, что человеческий мозг состоит из двух полушарий, имеющих различную функциональную направленность. Правое полушарие, главным образом, реализует мыслительные процессы на основе эмпирической индукции, а левое – путем дедуктивных выводов. При этом полушария связаны между собой, и между ними происходит обмен информацией.

Основные функции, реализуемые левым полушарием, относятся к области логики, анализа, обеспечивают понимание речи, выполнение арифметических и других логически выстраиваемых операций.

Правое полушарие «реализует» интуицию, воображение, озарение, восприятие и опознание.

Можно упрощенно представить левое полушарие – как универсальный компьютер, реализующий логический анализ, а правое – как пока недостаточно изученную систему, реализующую эвристический синтез. Оба полушария одновременно вовлекаются в мыслительные процессы, обмениваются информацией и частично воспроизводят функции друг друга [20]. Таким образом, можно говорить о двойственном, дуальном процессе принятия решений головным мозгом человека.

Попытка построения простейших дуальных компьютерных моделей принятия решений и соответствующих программ была предпринята в работах [10,21].

На современном этапе развития науки и технологий не только дедуктивные выводы, но и индуктивное обобщение реализуются с целью построения интеллектуализированных систем принятия решений в основном на компьютерах. Однако правополушарные функции интуиции, воображения, обобщения, по-видимому, далеко выходят за пределы класса вычислимых функций. В связи с этим возникает множество вопросов, касающихся реализуемости моделей правополушарных функций на компьютерах. Частично эти сложные вопросы затронуты в настоящей книге.

Говоря об эмпирической индукции, нельзя не упомянуть важное, постоянно развивающееся научное математическое направление – индуктивный синтез математических оптимизационных моделей выбора решений. Это направление является расширением задачи индуктивной классификации и включает индуктивные модели регрессии [8,9,11,12,13,21]. Еще более широким направлением является информационное индуктивное моделирование в целом [8,17,18].

Литература к главе 1

1. Айзерман М.А. Теоретические основы метода потенциальных функций в задаче об обучении автоматов разделению ситуаций на классы / М.А. Айзерман, Э.М. Браверман, Л.И. Розоноэр // Автоматика и телемеханика. – 1964. – Т.25. – С. 821 – 837.
2. Блох А. Ш. Об одном алгоритме обучения для задач по распознаванию образов / А.Ш. Блох // Вычислительная техника в машиностроении. – Минск: 1966. - №10. – С. 37 – 43.
3. Бонгард М. М. Моделирование процесса узнавания на цифровой счетной машине / Бонгард М. М. // Биофизика. – 1961. – Вып. 4. – № 2. – с. 17.

4. Де Бройль Л. Роль любопытства, игр, воображения и интуиции в научном исследовании. Тропами науки / Луи де Бройль. – М.: Издательство иностранной литературы, 1962. – С.292 – 295.
5. Бэкон Ф. Сочинения. В 2-х томах. Т. I / Фрэнсис Бэкон. – М.: Мысль (Философское наследие), 1971. – 590с.
6. Бэкон Ф. Новый органон // Сочинения в двух томах. Т. 2 / Фрэнсис Бэкон. – М.: Мысль (Философское наследие), 1978. – 575 с. – С.7-214. Режим доступа:
http://filosof.historic.ru/books/item/f00/s00/z0000451/st000.shtml
7. Вапник В. Н., Червоненкис А. Я. О равномерной сходимости частот появления событий к их вероятностям // ДАН СССР. – 1968. – Т. 181, № 4. – С. 781–784.
8. Гупал А.М. Индуктивный подход в математике / А. М. Гупал, А. А. Вагис // Пробл. упр. и информатики . – 2002. – № 2. – С. 83 – 90.
9. Донской В. И. Дискретные модели принятия решений при неполной информации / В. И. Донской, А.И. Башта. – Симферополь: Таврия. – 1992. – 166 с.
10. Донской В. И. Дуальные экспертные системы / В. И. Донской // Известия РАН. Техническая кибернетика. – 1993. – №5. – С. 111 – 119.
11. Донской В. И. Оценка точности псевдобулевых канонических моделей принятия решений при неполной информации / В. И. Донской // Системн. дослідж. та інформ. технології . – К. – 2004. – № 4. – С. 77–83.
12. Донской В. И. Синтез согласованных линейных оптимизационных моделей по прецедентной информации: подход на основе колмогоровской сложности / В. И. Донской // Таврический вестник информатики и математики. – 2012. – №1. – С. 13 – 23.
13. Донской В.И. Слабоопределенные задачи линейного булева программирования с частично заданным множеством допустимых решений / В. И. Донской // Журн. выч. матем. и матем. физики. – 1988. – Т. 28. – № 9. – С.1379 – 1385.
14. Ерёмин И.И. Вопросы оптимизации и распознавания образов / И. И. Ерёмин, В.Д. Мазуров. – М: Наука, 1979. – 288 с.
15. Ерёмин И.И. Нестационарные процессы математического программирования / И. И. Ерёмин, В.Д. Мазуров. – Свердловск: Средне-Уральское книжное изд-во, 1979. – 64 с.
16. Журавлев Ю. И. О математических принципах классификации предметов и явлений / Ю. И. Журавлев, А. Н. Дмитриев, Ф. П. Кренделев // Дискретный анализ. – 1966. – Вып. 7. – С. 3 – 15.
17. Рудаков К. В., Воронцов К. В. Применение алгебраического подхода в имитационном моделировании клиентских сред / К. В. Рудаков, К.В. Воронцов // Математические методы распознавания образов: Доклады 10-й Всеросс. конф. – М.: 2001. – С. 292–295.
18. Сергієнко І. В., Гупал А.М. Індуктивна математика. – Вісник НАН України. – 2002. – № 5. – С. 19–25.
19. Субботин А.Л. Фрэнсис Бэкон / А.Л. Субботин. – М.: Мысль, 1974. –175 с.

20. Blakeslee T. R. The Right Brain / Thomas R. Blakeslee. – N. Y.: PBJ Books Inc., 1983. – 276 p.
21. Donskoy V. I. Case-, Knowledge-, and Optimization-Based Hybrid Approach in AI / V. I. Donskoy // Lecture Notes in Computer Science. – 1998. – Vol. 1415. – P. 520 – 527.
22. Donskoy V. Pseudo-Boolean scalar optimization models with incomplete information / V. Donskoy// GMOOR Newsletter. – 1996. – № 1/2. – P. 20 –26.
23. Fu K.S. A sequential decision model for optimal recognition / King-Sun Fu / Biological prototypes and scientific systems. Vol.1. – N.Y.: Plenum Press, 1962. – P. 270 – 277.
24. Fu K.S. Learning system heuristics / King-Sun Fu // IEEE Trans. Automat. Contr. – 1966. – Vol. AC-11. – P. 611 – 612.
25. Glovazky A. Determination of redundancies in a set of patterns // IRE Trans. Inform. Theory. – 1956. – Vol. IT2. – P. 151 – 153.
26. Hunt E. B. Concept learning: An information processing problem / Earl B. Hunt. – N. Y.: John Wiley and Co., 1962. – 286 c.
27. Kanal L. Basic principles of some pattern recognition system / L. Kanal, F. Slaymaker, D. Smith, A. Walker // Proc. Nat. Electron. Conf. – 1962. – Vo. 18. – P.279 – 295.
28. Nilsson N. Learning Machines – Foundation of Trainable Pattern-Classifying Systems / N. J. Nilsson. – N.Y.: McGrow-Hill, 1965. – 137 p.
29. Novikoff A. B. On convergence proofs on perceptrons. Symposium on the Mathematical Theory of Automata, 12 / A. B. Novikoff. – Polytechnic Institute of Brooklyn:1962. – P. 615– 622.
30. Rosenblatt F. Principles of Neurodynamics: Perceptrons and the Theory of Brain Mechanisms / Frank Rosenblatt. – Washington D.C.: Spartan Books, 1962. – 616 p.

2. Машинное обучение и обучаемость

2.1 Основные понятия машинного обучения распознаванию (классификации)

Неформально *машинное обучение* можно представить как процесс нахождения неизвестного решающего правила (или неизвестной целевой функции) по некоторой начальной информации, которая не является полной. Эту *неполную начальную информацию* называют *обучающей*.

Говорят, что значения аргументов искомой функции в данной точке в совокупности являются *описанием объекта (точки) в некоторой проблемной области*. Если такое описание является правильным по смыслу решаемой задачи, то такую точку вместе с её описанием называют *допустимым объектом*. В задачах машинного обучения аргументы целевых функций (решающих правил) и, соответственно, их допустимые описания могут быть разнообразными. В отличие от классических математических задач, могут использоваться допустимые описания изображений, текстов, структур данных и многие другие. Это приводит к тому, что при решении задач машинного обучения используются различные разделы математики, подходящие в конкретных случаях.

Обучающая информация, как правило, представляет собой конечную совокупность *примеров* – допустимых точек (описаний) вместе со значениями целевой функции в этих точках. В этом случае обучающую информацию называют *обучающей выборкой*.

Если машинное обучение предполагает нахождение конечного набора неизвестных характеристических функций множеств, то такое обучение обычно называют обучением распознаванию.

Если целевая функция принимает только два значения, то её называют *классифицирующей или классификатором*.

Если целевая функция принимает произвольные значения, то её называют *регрессией*.

В математике существуют два основных типа проблем. Первая состоит в том, что *для заданных математических объектов* (например, множеств, семейств функций, уравнений) требуется определить их математические свойства (мощность, полноту, существование решений и др.). Эта первая проблема и есть математический анализ. Вторая проблема является обратной: даны свойства, которыми должен обладать математический объект, и такой объект нужно найти. Эта проблема называется *математическим синтезом.* Нахождение неизвестного решающего (классифицирующего) правила в задаче машинного обучения представляет собой проблему математического синтеза функции по её заданным свойствам, в частности

– по данному набору значений искомой функции в точках, описание которых дано.

Если нахождение математического объекта по его заданным свойствам происходит конструктивно, и результатом синтеза является как искомый объект, так и алгоритм его построения, то следует говорить об *алгоритмическом синтезе по обучающей информации или машинном (алгоритмическом) обучении классификации*. В силу обобщенного тезиса Чёрча-Тьюринга, являющегося по своей сути определением алгоритма и вычислимой функции, алгоритмическое обучение классификации предполагает нахождение неизвестной частично рекурсивной функции по примерам – её значениям в конечном заданном числе точек.

Собственно *распознавание* состоит в применении найденных в процессе обучения решающих правил для определения принадлежности рассматриваемым множествам (*классам*) объектов, не содержавшихся в начальной информации.

В этой книге мы будем рассматривать главным образом класс задач алгоритмического обучения классификации – самый общий случай машинного обучения. Это предполагает построение и использование алгоритмов обучения и алгоритмов классификации, полученных в результате обучения.

Будем говорить, что *согласованной с обучающей информацией или корректной на ней* называется любая функция, которая в точках, входящих в эту обучающую информацию, принимает точно такие же значения, какие содержатся в примерах из этой обучающей информации. Иначе говоря, если обучающая информация – это набор из l точек с зафиксированными значениями неизвестной функции в этих точках, то корректной будет любая функция, принимающая эти заданные значения в l заданных точках.

Алгоритм обучения называется корректным на обучающей выборке, если он выдаёт согласованную с заданной обучающей информацией вычислимую функцию.

Возникает вопрос: *в чем же отличие обучения от нахождения произвольной корректной на данной обучающей информации функции*?

Будем рассматривать произвольную частично заданную в l точках $\{\tilde{\alpha}_j = (\alpha_{1j},...,\alpha_{nj})\}$, $j = \overline{1,l}$, функцию g такую, что $g(\tilde{\alpha}_j) = \gamma_j$, где γ_j – заданные значения. В остальных допустимых точках функция g может принимать любые допустимые значения. В случае задачи нахождения функции регрессии для определения понятия машинного обучения полезна следующая

Теорема 2.1. Любая частично заданная в l точках $\{\tilde{\alpha}_j = (\alpha_{1j},...,\alpha_{nj})\}$, $j = \overline{1,l}$, рекурсивная функция $g = g(x_1,...,x_n)$, зависящая от n переменных, может быть представлена в виде

$$g(x_1,...,x_n) = \sum_{j=1}^{l} \gamma_j \chi_j^1(x_1,...,x_n) + f(x_1,...,x_n)\Pi_{j=1}^{l}\chi_j^0(x_1,...,x_n), \quad (2.1)$$

где $\gamma_j \in \mathrm{N}_0$ – значения функции g в точках $\tilde{\alpha}_j$, $j = \overline{1,l},$; $f = f(x_1,...,x_n)$ – произвольная рекурсивная функция, а рекурсивные функции χ_j^{δ}, $\delta = 0,1$, имеют вид

$$\chi_j^{\delta}(x_1,...,x_n) = \begin{cases} \delta, & \tilde{x} = \tilde{\alpha}_j; \\ 1-\delta, & \tilde{x} \neq \tilde{\alpha}_j; \end{cases}$$

обозначение $\Pi_{k=1}^{l} a_k = a_1 \times \cdots \times a_l$ определяет произведение.

Доказательство. Известно, что константы, сложение, усеченная разность $a \dot{-} b$, умножение, модуль разности $|a-b|$, а также функция

$$\overline{Sg}(z) = \begin{cases} 1, & z = 0; \\ 0, & z > 0; \end{cases}$$

являются рекурсивными. Легко видеть, что рекурсивными будут функции $\chi_j^1(x_1,...,x_n) = \overline{Sg}(|\alpha_{1j} - x_1|) \times \overline{Sg}(|\alpha_{2j} - x_2|) \times \cdots \times \overline{Sg}(|\alpha_{nj} - x_n|)$ и

$$\chi_j^0(x_1,...,x_n) = 1 \dot{-} \chi_j^1(x_1,...,x_n),$$

которые являются характеристическими функциями точек $\tilde{\alpha}_j$, принимающими, соответственно, значения только 1 и 0:

$$\chi_j^1(\tilde{x}) = \begin{cases} 1, & \tilde{x} = \tilde{\alpha}_j; \\ 0, & \tilde{x} \neq \tilde{\alpha}_j; \end{cases} \quad \chi_j^0(\tilde{x}) = \begin{cases} 1, & \tilde{x} \neq \tilde{\alpha}_j; \\ 0, & \tilde{x} = \tilde{\alpha}_j. \end{cases}$$

Из последних соотношений следует справедливость разложения (2.1). □

Следствие 2.1. Из любой рекурсивной функции путём замены её значений не более чем в l заданных точках можно получить функцию, согласованную с обучающей информацией.

Следствие 2.2. Число корректных на обучающей выборке рекурсивных функций сколь угодно велико.

Доказательство следует из того факта, что кардинальное число множества рекурсивных функций есть $\aleph_0$ [9]. □

Следствие 2.3. Для любой обучающей выборки, состоящей из двоичных чисел–слов заданной ограниченной длины, существует сколь угодно много корректных на этой выборке алгоритмов обучения.

Доказательство. Для каждого алгоритма (машины Тьюринга) существует эквивалентная частично рекурсивная функция. В частности, каждая рекурсивная функция реализуема некоторой машиной Тьюринга. Взяв любую рекурсивную функцию f и подставив её в правую часть равенства (2.1), которое содержит полностью всю обучающую информацию, получим требуемый алгоритм. □

В задаче машинного обучения предполагается существование истинной искомой целевой функции g^*, которая должна совпадать с частично заданной при помощи обучающей информации функцией g в l заданных точках; значения функции g^* в остальных точках неизвестно. Как показано выше, функций, удовлетворяющих такому условию, и соответствующих алгоритмов обучения сколь угодно много. А истинная – одна. Поэтому почти все корректные алгоритмы обучения, не использующие дополнительные условия-ограничения, т.е. дополнительную информацию, будут вычислять функции, отличающиеся от истинной неизвестной функции.

Таким образом, извлечение дополнительной информации должно иметь решающее значение.

В теории машинного обучения применяется ряд подходов к преодолению указанных трудностей.

1. *Сузить на основе дополнительной инф*ормации (или, когда это удаётся, путем анализа обучающих примеров) *класс функций*, в котором содержится истинная искомая функция *настолько*, что обучающая информация вместе с информацией о таком классе будет полной для точного и единственного решения задачи машинного обучения.

В общем случае это можно осуществить только теоретически. Но в частных случаях, при достоверной обучающей информации и, например, дополнительной информации о линейности искомого решающего правила, найти его можно просто путём решения системы линейных уравнений.

2. Если указанное в п.1 радикальное сужение найти не удаётся, то следует использовать как можно более узкий подкласс решающих функций для поиска в нём. Но здесь возникает проблема: а будет ли искомое решение содержаться в используемом классе?

В теории В. Н. Вапника [1,2] в этом направлении получены выдающиеся результаты. Укажем только два главных аспекта его теории: а) *принципиальное место уделяется специально введенной мере сложности классов*; б) *получаемые условия близости найденных решающих правил к истинным правилам не зависят от того, содержится ли истинное неизвестное правило в классе правил, используемом для поиска.*

3. Использовать *дополнительную информацию* о результатах решения задачи машинного обучения классификации *в виде набора алгоритмов, отличающихся друг от друга, но решающих одну и ту же задачу*. Соответствующая теория алгебраической коррекции семейств эвристических алгоритмов была создана Ю. И. Журавлёвым [7] и получила развитие в работах ученых его научной школы – В. Л. Матросова, К. В. Рудакова, В. В. Рязанова, А. Г. Дьяконова и др.[10,6]. Модели обучения бустинг (*boosting*) и бэггинг (*bagging*) [17] принципиально примыкают к алгебраической теории распознавания Ю. И. Журавлёва: первая – просто как частный случай, вторая – как снабженная дополнительной эвристикой, направленной на увеличение различия алгоритмов, входящих в корректируемый набор. Нужно заметить, что идеи, заложенные в бэггинг и бустинг, были предложены еще в 1980 г. Л. А. Растригиным [8].

Проблема удачного выбора класса функций для поиска в нем подходящего классификатора является центральной в теории машинного обучения и требует глубоких знаний в этой области.

4. Найти убедительные подтверждения того*, что процесс поиска действительно направлен на построение именно требуемой, истинной целевой функции. В таком случае можно рассчитывать, что будет найден не какой-нибудь корректный на выборке алгоритм классификации, а тот, который нужен. Именно такой процесс следует понимать как обучение*.

Указанному требованию будут удовлетворять алгоритмы, которые строят решающие правила последовательно, пошагово, рассматривая пример за примером обучающую выборку. И только в случае ошибки классификации очередного примера, текущее выстраиваемое правило корректируется; причем, в основном, в локальной области примера, на котором совершается ошибка. Если в результате обучения будет получено решающее правило, корректное на выборке, и при этом из l предъявленных примеров для коррекций (синтеза) использовалось только $r < l$ примеров, то можно считать, что $k = l - r$ примеров *подтверждают* правильность выбора, что удаётся оценить с позиций статистического подхода и подхода на основе колмогоровской сложности и сжатия информации [4, 37].

Уточняя процесс обучения нужно определить следующее:

- информацию о множестве (допустимых) объектов;
- о каком неизвестном решающем правиле или функции идёт речь;
- что предоставляется в качестве начальной информации;
- в каком классе решающих правил будет отыскиваться решение;
- какие дополнительные свойства множества допустимых объектов и функций должны быть учтены;

- как будет осуществляться обучение Естественно предполагать, что используется конечный компьютер или шире – вычислимые функции (заметим, что в сложившейся в настоящее время теории машинного обучения математические построения зачастую выходят за рамки указанных классов). Иначе говоря, определить процесс обучения как алгоритмическое отображение начальной информации в некоторое множество решающих правил.
- как оценивать качество обучения;
- как определять, существует ли возможность достижения требуемого качества обучения при перечисленных условиях (имеет ли место обучаемость);
- как оценивать число обучающих примеров, требуемых для достижения нужного качества обучения.

Уточнения неформальной постановки приводит к большому числу специфических задач машинного обучения и распознавания. Попытка представить классификацию таких задач была предпринята в работе [4]. Важно отметить, что получить уточнения задачи машинного обучения по всем перечисленным выше пунктам удаётся не всегда.

2.2 Машинное обучение классификации по прецедентам. Основные определения

Далее будет *рассматриваться задача машинного обучения классификации по прецедентам* (примерам) в соответствии с принципом эмпирической индукции (обобщения) в следующей постановке.

Множество допустимых объектов X, называемое *признаковым пространством*, состоит из векторов (или *точек признакового пространства*) $\tilde{x} = (x_1,...,x_n)$, значения координат которых в совокупности представляют описания объектов. Предполагается, что *на множестве* X *существует вероятностное распределение* P. Вид этого распределения будет полагаться неизвестным. Неизвестная, но *существующая* (*целевая*) *функция* $\varphi : X \to \{0,1\}$ принадлежит некоторому *семейству* Φ, которое также является неизвестным. Требуется, используя начальную информацию – обучающую выборку длины l, извлечь из выбранного заранее класса решающих правил Ψ такую функцию $\psi : X \to \{0,1\}$, которая как можно более точно приближает неизвестную целевую функции φ. Качество найденной в процессе обучения функции ψ в рассматриваемом случае можно представить как вероятностную меру несовпадения целевой функции φ с найденной в результате обучения функцией ψ. Проще говоря – как *вероятность ошибки функции* ψ, которая может быть выражена при

помощи интеграла Лебега при условии измеримости соответствующих функций:

$$Err_{\psi} = \int_{\tilde{x}\in X} |\psi(\tilde{x}) - \varphi(\tilde{x})| dP(\tilde{x}).$$

Чем меньше вероятность ошибки Err_{ψ} выбранного при обучении решающего правила ψ, тем лучше результат обучения. Но величину Err_{ψ} определить невозможно, поскольку неизвестна целевая функция φ и в подавляющем большинстве случаев неизвестна вероятностная мера P. Поэтому в статистической теории обучения используются подходящие оценки вероятности Err_{ψ} снизу и сверху.

Обучающая выборка $X_l = \{(\tilde{x}_j, \alpha_j)\}_{j=1}^{l}$ состоит из *примеров* – пар «точка – значение неизвестной функции в этой точке»: $\alpha_j = \varphi(\tilde{x}_j)$. Точки, входящие в выборку, извлекаются из множества X случайно и независимо в соответствии с распределением P. В широком классе постановок задач машинного обучения обучающие выборки могут содержать ошибки. Но мы будем рассматривать, если не оговаривается противное, только тот случай, когда *обучающие выборки абсолютно точные, не содержат ошибок*.

Естественно потребовать, чтобы с ростом длины обучающей выборки (с увеличением числа обучающих примеров) величина Err_{ψ} стремилась к нулю. В общих чертах это характеризует *обучаемость*, как возможность достижения нужной точности извлекаемой в процессе обучения решающей функции ψ.

Понятие обучаемости возможно строго определить не единственным способом, и это приводит к существенным различиям в постановке задачи и построении моделей обучения.

Если $\Pr(Err_{\psi} > \varepsilon) < \delta$, где $\delta = \delta(l, \varepsilon)$, то величину ε называют точностью, а $(1-\delta)$ – надежностью оценки выбранного решающего правила ψ.

Процесс машинного обучения может быть упрощенно представлен схемой на рис.2.1, в соответствии с которой следует обратить внимание на следующие обстоятельства.

Выборка может быть извлечена различными способами, и это должно уточняться – должна быть определена *схема извлечения выборки*.

Результат обучения – решающая функция ψ – может быть извлечена из семейства Ψ различными методами. Понятие *метода* или *алгоритма обучения* является центральным, поскольку именно его выбор определяет: будет ли иметь место обучаемость. Алгоритм обучения управляет процес-

сом выбора решения ψ, используя обучающую выборку. С точки зрения постановки задачи, предполагая компьютерную реализацию, целесообразно говорить именно об *алгоритме обучения*. А с точки зрения центральной роли этого алгоритма в схеме машинного обучения, следуя К. В. Воронцову [3], представляется возможным применение термина «*метод обучения*». Далее всё же будет использоваться термин «алгоритм обучения».

Любой алгоритм обучения A представляет собой отображение множества всех допустимых обучающих выборок во множество $\operatorname{Im} A \subseteq \Psi$ – образ отображения A.

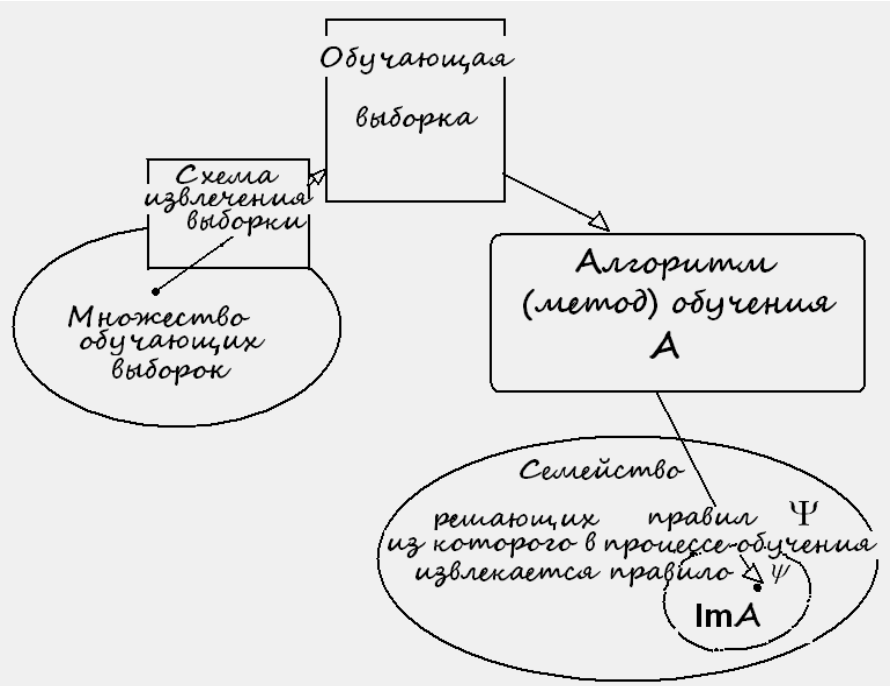

Рис.2.1. Схематическое представление процесса обучения

Будем называть приведенное выше уточнение задачи машинного обучения *функциональным*. В большинстве современных научных работ, посвященных машинному обучению, даётся другое – теоретико-множественное уточнение.

Концептами называют собственные подмножества X. Классом концептов называют семейство $H \subseteq 2^{\mathrm{X}}$ концептов. В дальнейшем полагается, что семейства концептов состоят из борелевских множеств. Задание классифицирующей функции $\varphi : \mathrm{X} \to \{0,1\}$ взаимно-однозначно определяет концепт h_φ как множество $h_\varphi = Dom_1(\varphi) = \{\tilde{x} \in \mathrm{X} : \varphi(\tilde{x}) = 1\}$. Множество, на котором функция φ принимает значение 0, является дополнением концепта h_φ во множестве X. *Примером* (*обучающим примером*) концепта $h \in H$ называют пару $(\tilde{x}, \alpha)$, где $\alpha = 1$, если $\tilde{x} \in h$, и

$\alpha = 0$, если $\tilde{x} \notin h$. *Выборка* – это множество примеров некоторого концепта. Длина выборки – это число содержащихся в ней примеров.

Если класс концептов H является перечислимым ($h_1, h_2, ...$), то его можно представлять перечислением конечных бинарных строк $s(h_1), s(h_2), ...$, определенным образом описывающих входящие в класс концепты. Такой подход позволяет рассматривать сложность концепта как длину кратчайшей описывающей его строки. Это приводит к понятию колмогоровской сложности концепта, которая, в общем случае, не является вычислимой функцией. Но можно использовать любые другие найденные короткие строковые описания концептов с целью оценивания его сложности сверху [4].

Некоторый выделенный концепт $g_\varphi \in G$ называют целевым, а соответствующую ему функцию φ – целевой. Целевая функция полагается неизвестной и принадлежащей некоторому классу $\Phi = \Phi(G)$.

Обучающий алгоритм A использует выборку X_l длины $l = l_A(\varepsilon, \delta)$ в соответствии с вероятностным распределением P на X и вычисляет концепт-гипотезу $h_A = h_A(X_l) \in H$ по этой обучающей выборке. В общем случае используемый для поиска решения концепт H может не совпадать с целевым концептом G.

Таким образом, имеет место следующее соответствие (табл. 2.1):

Табл. 2.1. Классифицирующие функции и концепты

Неизвестная заранее целевая классифицирующая функция φ	Неизвестный целевой концепт – множество $g_\varphi = Dom_1(\varphi)$
Неизвестный класс функций Φ, которому принадлежит функция φ	Неизвестный класс концептов G, содержащий целевой концепт g_f; $G = \bigcup_{\{\varphi:\varphi\in\Phi\}} Dom_1(\varphi)$
Решающая функция ψ – результат обучения	Результирующий концепт – множество $h_\psi = Dom_1(\psi)$
Известный, заранее выбранный класс функций Ψ, из которого в процессе обучения извлекается функция ψ	Известный, заранее выбранный класс концептов H, содержащий извлекаемый при обучении концепт h_ψ; $H = \bigcup_{\{\psi:\psi\in\Psi\}} Dom_1(\psi)$

Из приведенной таблицы видно, что использование концептов приводит к постановке задачи обучения на теоретико-множественной основе,

которая эквивалентна постановке этой же задачи при использовании функционального подхода. Оба подхода имеют свои преимущества, и в силу эквивалентности представленных в таблице теоретико-множественных и функциональных описаний их можно и нужно использовать по мере проявления нужных преимуществ.

2.3. Обучаемость

Говоря неформально, понятие обучаемости необходимо для того, чтобы иметь возможность находить ответ на вопрос: удастся ли при некоторых заданных алгоритмах обучения и семействах функций, из которых извлекается решающее правило, достигнуть приближения этого правила к неизвестной целевой функции с нужной точностью? Т. е. можно ли в результате обучения получить достаточно точную аппроксимацию неизвестной целевой функции?

Фундаментальную роль в исследовании обучаемости моделей построения алгоритмов классификации по прецедентной информации играет теория равномерной сходимости В.Н. Вапника – А.Я. Червоненкиса [2] и особенно – введенное ими понятие *ёмкости класса решающих правил*, в котором отыскивается классифицирующий алгоритм. Эта характеристика сложности функциональных семейств получила название $VC-$размерности ($VC-dimension$) или VCD. Аббревиатура содержит первые буквы фамилий авторов теории равномерной сходимости.

Основное содержание излагаемой теории, основные элементы которой вкратце приведены ниже, связано со следующим положением [1,2]. *Решающую функцию* h *следует выбирать из такого класса* H *, который удовлетворяет определенному соотношению между величиной, характеризующей качество приближения функции к заданной совокупности эмпирических данных, и величиной, характеризующей «сложность» любой выбранной приближающей функции*.

Эмпирическая частота ошибок выбранного в результате обучения по данной выборке $(\tilde{x}_j,\alpha_j)_{j=1}^{l}$ решающего правила $h\in H$ или, иначе говоря, эмпирический функционал качества есть

$$Err_l(h)=\frac{1}{l}\,|\,\{(\tilde{x},\alpha)\in X_l : h(\tilde{x})\neq\alpha\}\,|=\nu_l(h)=\frac{1}{l}\sum_{j=1}^{l}|\,h(x_j)-\alpha_j\,|.$$

Недостаток оценивания качества приближения выбранного правила h к неизвестному, представленному лишь обучающими примерами, правилу $\varphi=\varphi(\tilde{x})\in\{0,1\}$ заключается в следующем. *Оценивается только одно фиксированное выбранное правило* $h\in H$. Но одно выбранное правило, настроенное на эмпирическую выборку и безошибочное на ней, может

оказаться таким, что оно сколь угодно часто будет давать неправильные ответы $h(\tilde{x})$ для произвольных объектов $\tilde{x}$, лежащих вне обучающей выборки. Например, следующее правило, которое можно назвать правилом «точного совпадения с эталоном»,

$$h(x) = \begin{cases} \alpha_j, \ \textit{если} \ \tilde{x} = \tilde{x}_j \ \textit{для какого – нибудь} \ j; \\ \overline{\alpha}_j, \ \textit{если} \ \tilde{x} \neq \tilde{x}_j \ \textit{для всех} \ j; \end{cases}$$

соответствует безошибочной настройке на обучающую выборку, но вне этой выборки не определяет никакой разумный ответ.

Рассмотрим функцию $\pi(l) = \sup\limits_{h \in H} | P^l(h) - \nu_l(h) |$, определяющую наибольшее по классу H уклонение частоты от вероятности. Отметим, что $\pi(l)$ является функцией точек в X^l, она измерима и является случайной величиной. Если величина $\pi(l)$ стремится (по вероятности) к нулю при неограниченном увеличении длины выборки l, то говорят, что *частота ошибок функций системы* H *стремится* (*по вероятности*) *к вероятностям* этих ошибок *равномерно по классу* H.

Далее выясняются условия, при которых для любого $\varepsilon > 0$ выполняется соотношение $\lim\limits_{l \to \infty} P^l(\pi(l) > \varepsilon) = 0$. *В отличие от закона больших чисел, равномерная сходимость частот к вероятностям может иметь или не иметь места в зависимости от того, как выбрана система* H *и как задана вероятностная мера* P^l.

Если равномерная сходимость по классу H имеет место, то гарантируется сходимость частот к вероятностям для любого правила из H, в том числе – и для конкретного правила, построенного по данной обучающей выборке.

Каждый элемент $h \in H$, $h = h(\tilde{x}) \in \{0,1\}$ для произвольной последовательности точек $\tilde{x}_1,...,\tilde{x}_l \in \mathbf{X}$ определяет подпоследовательность X_h, состоящую из тех $\tilde{x}$, для которых имеет место событие $h(\tilde{x}) = 1$. Говорят, что h *индуцирует подпоследовательность* X_h и тем самым разбивает последовательность $\tilde{x}_1,...,\tilde{x}_l$ на элементы X_h и их дополнение в ней.

Обозначим $\Delta^H(\tilde{x}_1,...,\tilde{x}_l)$ *число различных подпоследовательностей* X_h, *индуцируемых всеми элементами* $h \in H$ (число различных разбиений выборки $\tilde{x}_1,...,\tilde{x}_l$ всеми различными элементами $h \in H$). Очевидно,

$\Delta^H(\tilde{x}_1,...,\tilde{x}_l) \le 2^l$, т. е. не превышает числа всевозможных двоичных наборов длины l.

Число $\Delta^H(\tilde{x}_1,...,\tilde{x}_l)$ называется *индексом системы* H относительно выборки $\tilde{x}_1,...,\tilde{x}_l$. Функция

$$m^H(l) = \max_{\tilde{x}_1,...\tilde{x}_l} \Delta^H(\tilde{x}_1,...,\tilde{x}_l),$$

где максимум берется по всем последовательностям $\tilde{x}_1,...,\tilde{x}_l$ длины l, называется *функцией роста системы* H.

Теорема 2.2 [2]. Функция роста $m^H(l)$ либо тождественно равна 2^l, либо, если это не так, мажорируется функцией $\sum_{i=0}^{\mu-1} C_l^i$, где μ - минимальное значение l, при котором $m^H(l) \ne 2^l$. Иначе говоря,

$$m^H(l) \begin{cases} \equiv 2^l, \\ < \sum_{i=0}^{\mu-1} C_l^i, \ \textit{если она не равна тождественно} \ 2^l. \end{cases}$$

Имеет место оценка: $\sum_{i=0}^{\mu-1} C_l^i \le 1{,}5\dfrac{l^{\mu-1}}{(\mu-1)!}$. □

Фигурирующее в теореме число μ имеет следующий смысл: никакие $l = \mu$ точек, извлеченные из $\mathbf{X}$, не могут быть разбиты на два класса всеми возможными способами. В то же время как найдутся $\mu - 1$ точек, которые могут быть разбиты на два класса всеми способами, если $l < \mu$.

Определение 2.1. Говорят, что класс H имеет емкость d, если справедливо неравенство

$$m^H(l) < 1{,}5\frac{l^d}{d!}, \quad l > d.$$

В случае $m^H(l) \equiv 2^l$ говорят, что емкость класса бесконечна: $d = \infty$. □

Величину d называют также VC – размерностью класса функций H и обозначают $VCD(H)$. Она характеризует разнообразие класса функций H и определяет наибольшую длины выборки, которую ещё можно классифицировать на два класса всеми возможными способами (такая выборка из d точек найдется), и тогда функция роста может быть оценена сверху полиномиально. Если конечное число d не существует для класса H, то его функция роста тождественно равна 2^l.

Если число функций в системе H конечно, $|H| = N$, то из условия $2^d \le N$ следует оценка $d = VCD(H) \le \log_2 N$.

Теорема 2.3 [2]**.** Вероятность того, что хотя бы для одной функции h из класса H частота ошибки на обучающей выборке длины l отклонится от её вероятности более чем на ε, удовлетворяет неравенствам

$$P\left(\sup_{h\in H} |P^l(h) - \nu_l(h)| > \varepsilon\right) < 6m^H(2l)e^{-\frac{\varepsilon^2 l}{4}};$$

$$P\left(\sup_{h\in H} |P^l(h) - \nu_l(h)| > \varepsilon\right) < 9\frac{(2l)^{VCD(H}}{(VCD(H)!}e^{-\frac{\varepsilon^2 l}{4}}.$$

Следствие 2.4 [2]. Для того, чтобы частота ошибки любого решающего правила $h \in H$ сходилась (по вероятности) к соответствующей вероятности, достаточно, чтобы емкость $d = VCD(H)$ класса H была конечной.

Действительно, если емкость d является конечной, то $m^H(l) < 1{,}5\frac{l^d}{d!}$, и тогда $P\left(\sup_{h\in H} |P^l(h) - \nu_l(h)| > \varepsilon\right) \to 0$ при $l \to \infty$. □

Для понятия обучаемости существует ряд различных определений.

Определение 2.2 (***PAC***-learning, ***P****robably* ***A****pproximately* ***C****orrect-learning*).

1) Будем говорить, что класс концептов $G \subset 2^X$ является PAC-*обучаемым* (или (ε,δ)-*обучаемым*) с использованием класса концептов $H \subset 2^X$, если найдется (обучающий) алгоритм A, который *при любом вероятностном распределении* P на X, *при любом целевом концепте* $g \in G$, для любых $\varepsilon,\delta : 0 < \varepsilon,\delta < \frac{1}{2}$, вычисляет по обучающей выборке X_l, извлеченной в соответствии с распределением P на X, концепт-гипотезу h_A, и при этом существует функция $l = l(\varepsilon,\delta)$, которая определяет длину обучающей выборки, обеспечивающую выполнение неравенства

$$\Pr\{P(h_A \Delta g) \le \varepsilon\} \ge 1 - \delta,$$

где $h_A \Delta g = (h_A \setminus g) \cup (g \setminus h_A)$, а $\Pr\{Z\}$ – вероятность того, что событие Z – истинно.

Классы концептов H и G, в частности, могут совпадать. В этом случае будем называть алгоритм обучения A *собственным или согласо-*

ванным с целевым концептом: $A(X_l) \in G$. Вариант модели *PAC* обучаемости, когда целевой концепт g заведомо содержится в используемом для обучения классе концептов H, называют *реализуемой PAC* моделью (*The Realizable PAC Model*) или *правильной* PAC-обучаемостью [15].

2) *Полиномиальная* PAC *обучаемость* (***RBPAC*** *– Resource Bounded PAC*) при всех перечисленных в первой части определения условиях дополнительно требует, чтобы алгоритм A обеспечивал (ε, δ)-обучение (выполнялся) за число *шагов, ограниченное полиномом* от $1/\varepsilon$, $1/\delta$, числа n переменных-признаков, длины описания $s(H)$ класса концептов H, и также использовал длину обучающей выборки, ограниченную полиномом от всех указанных величин.

Наименьшее число примеров, обеспечивающее полиномиальную PAC обучаемость называют *сложностью выборки* относительно алгоритма обучения A. □

Важно обратить внимание на то, что в определении *PAC*-обучаемости не оговариваются никакие (кроме сложностных в $RBPAC$) свойства алгоритма обучения. Может применяться любой удовлетворяющий определению алгоритм A. Но при этом область его значений как алгоритмического отображения точно не оговаривается: возможно, что она совпадает с классом концептов H, но не исключается, что она существенно у́же класса H. При этом распределение вероятностей P на X может быть любым. В силу такой широкой трактовки понятия *PAC* обучаемости, необходимым и достаточным условием для её достижения является конечность VC размерности класса, из которого извлекается концепт:

Теорема 2.4 [6]. Класс концептов H является *PAC* обучаемым тогда и только тогда, когда $VCD(H) < \infty$. □

Сложностные свойства алгоритма обучения, фигурирующие в $RBPAC$ модели, предназначены для гарантии эффективной (полиномиальной) реализуемости обучения. Многие авторы научных работ в области машинного обучения не уделяют внимания сложности обучающих алгоритмов, ограничиваясь только требованием их сходимости. Для $RBPAC$ обучаемости предыдущая теорема верна при условии полиномиальной сложности алгоритма обучения.

Алгоритм обучения (и решающее правило) называют *согласованными* (с обучающей выборкой), если решающее правило правильно классифицирует все примеры обучающей выборки. Если же κ – число примеров, неправильно классифицируемых выбранным при обучении решающим правилом, а l – длина обучающей выборки, то величину $\nu_{emp} = \dfrac{\kappa}{l}$ называют эмпирической частотой ошибок. Согласованные алгоритмы обеспе-

чивают выбор решающих правил, имеющих $\nu_{emp} = 0$. Будем говорить, что алгоритм обучения *частично согласован* с обучающей выборкой, если $\kappa > 0$. Тогда он согласован с некоторой подвыборкой длины $l - \kappa$.

Определение 2.3 (A*gnostic PAC-learning*). Пусть P – вероятностное распределение (неизвестное) на $\mathrm{X} \times \{0,1\}$ и $g : \mathrm{X} \to \{0,1\}$ – заранее неизвестная (целевая) функция. Пусть $H = \{h : \mathrm{X} \to \{0,1\}\}$ – класс гипотез. Пусть $A(X_l) = h \in H$ – гипотеза, извлекаемая по выборке $X_l = (\tilde{x}_j, \alpha_j)_{j=1}^{l}$ обучающим алгоритмом A. Ошибка гипотезы h согласно мере P есть $Err(h) = P\{(\tilde{x}, \alpha) : h(\tilde{x}) \neq \alpha\}$. Эмпирическая ошибка гипотезы h есть $Err_l(h) = \frac{1}{l}\left|\{(\tilde{x}, \alpha) \in X_l : h(\tilde{x}) \neq \alpha\}\right|$. Говорят, что имеет место *agnostic PAC обучаемость*, если для любых положительных $\varepsilon, \delta < 1$, *для любого распределения* P на $\mathrm{X} \times \{0,1\}$ можно указать такое значение $l = l(A, \varepsilon, \delta, H)$, что для любой случайно извлеченной в соответствии с P^l обучающей выборкой X_l длины l имеет место неравенство

$$\Pr\{Err(A(X_l)) - \inf_{h \in H} Err_l(h) \leq \varepsilon\} \geq 1 - \delta. \ \square$$

В определении A*gnostic PAC learning* не фигурирует класс, в котором содержится целевой концепт. Распределение вероятностей полагается произвольным и предполагается использование принципа минимизации эмпирического риска ($\inf_{h \in H} Err_l(h)$). По сравнению с *PAC* обучением, модель A*gnostic PAC learning* шире, но и для неё остаётся справедливым необходимое и достаточное условие обучаемости – конечность ёмкости класса, в котором заведомо содержится образ алгоритма обучения ($\mathrm{Im}\, A$).

GSL обучаемость, определяемая далее, практически является A*gnostic PAC* обучаемостью – «едва заметным» её расширением в случае, когда верхняя грань семейства всевозможных вероятностных распределений не является достижимой.

Определение 2.4 (*Обобщенная статистическая обучаемость, GSL* [35]). При условиях, сформулированных в определении, *статистическая обучаемость* имеет место, если для любого $\varepsilon > 0$ можно указать такое значение длины обучающей выборки $l = l(A, \varepsilon, \delta, H)$, что

$$\sup_{P \in \mathcal{P}} \Pr\{Err(A(X_l)) - \inf_{h \in H} Err_l(h) \leq \varepsilon\} \geq 1 - \delta,$$

где $\mathcal{P}$ – всевозможные вероятностные распределения на $\mathrm{X} \times \{0,1\}$.

Рассмотрим ещё ряд определений обучаемости, встречающихся в научной литературе.

Определение 2.5 [25]. Будем говорить, что при обучении имеет место *равномерная сходимость независимо от распределений* (***DFUC***), если

$$\sup_{P\in\mathcal{P}} \int_{X_l\in \mathrm{X}^l} \{\sup_{h\in H} |\, Err(h) - Err_l(h)\,|\} dP^l \to 0 \quad \text{при} \quad l\to\infty.$$

Определение 2.6 [31]. H называется ε-равномерным классом Гливенко-Кантелли, если

$$\lim_{m\to\infty} \sup_{P\in\mathcal{P}} \Pr\{\sup_{l\geq m} \sup_{h\in H} |\, Err(h) - Err_l(h)\,| > \varepsilon\} = 0.$$

Теорема 2.5 [31]. Пусть H – класс функций из X в $\{0,1\}$. Тогда H является равномерным классом Гливенко-Кантелли (***uGC***), если и только если $VCD(H) < \infty$.

Определение 2.7 [1,2]. (Двусторонняя) р*авномерная сходимость по Вапнику* (***VUC***) имеет место при обучении в классе решающих правил H, если для любого положительного $\varepsilon < 1$

$$\lim_{l\to\infty} P\{\sup_{h\in H} |\, Err(h) - Err_l(h)\,| > \varepsilon\} = 0.$$

В этом определении независимость от вероятностного распределения явно не указана. Речь идет о *некотором имеющемся распределении* на $\mathrm{X}\times\{0,1\}$, в соответствии с которым происходит случайное и независимое извлечение примеров в обучающую выборку. Однако полученное В. Н. Вапником достаточное условие равномерной сходимости – конечность $VCD(H)$ – не зависит от свойств распределения. Также независимым от свойств распределения является необходимое и достаточное условие равномерной сходимости [37, с. 57] для любой вероятностной меры:

$$\lim_{l\to\infty} \frac{G^H(l)}{l} = 0, \tag{2.2}$$

где $G^H(l) = \ln \sup_{(\tilde{x}_1,\ldots,\tilde{x}_l)\in X^l} \Delta^H(\tilde{x}_1,\ldots,\tilde{x}_l)$ – логарифм функции роста семейства H, а $\Delta^H(\tilde{x}_1,\ldots,\tilde{x}_l)$ – число способов разбиения выборки на два класса гипотезами семейства H. *Если условие* (2.2) *не выполняется, то найдётся вероятностная мера на* $\mathrm{X}\times\{0,1\}$, *для которой равномерная сходимость по Вапнику не будет иметь места* [37, с. 72].

При выполнении достаточного условия равномерной сходимости по классу гипотез H – ограниченности $VCD(H)$ – выбор любой гипотезы $h\in H$, минимизирующей эмпирический риск, с ростом длины обучающей выборки будет *гарантировать* со сколь угодно большой вероятностью $1-\delta$ сколь угодно малое отклонение вероятности ошибки выбранной гипотезы h от её эмпирической ошибки на обучающей выборке. Причем

ограниченность $VCD(H)$ гарантирует равномерную сходимость при любом вероятностном распределении P на $X \times \{0,1\}$. Конечность $VCD(H)$ перестаёт быть необходимым условием, если не требовать выполнения равномерной сходимости *для любых распределений*. Так, в работе [33] рассматривается обучаемость в случае неатомических (диффузных) вероятностных мер, и такое сужение условий приводит к некоторому *новому определению модулярной* VC *размерности* $VC(H \bmod \omega_1)$, которая, вообще говоря, может быть конечной при $VCD(H) = \infty$.

Одним из подходов к получению оценок ошибок алгоритмов обучения (эмпирического обобщения) является оценивание их устойчивости. *Под устойчивыми обучающими алгоритмами понимаются такие, которые извлекают гипотезы, незначительно изменяющиеся при малом изменении обучающей выборки*. Получаемые при таком подходе оценки оказываются независимыми от VC размерности используемого пространства гипотез [15,16], а зависят от того, *как алгоритм обучения осуществляет поиск* в этом пространстве, и поэтому можно рассчитывать на обучаемость в случае, когда пространство гипотез имеет бесконечную VC размерность. Но при этом следует оговаривать, о каком определении обучаемости идет речь.

Введение в определение обучаемости дополнительных свойств алгоритма обучения влечёт сужение этого определения, выделяет частный случай из множества ситуаций, когда алгоритм обучения является произвольным, и может ослабить необходимые и достаточные условия обучаемости.

2.4. Устойчивость обучающих алгоритмов

Подход на основе устойчивости обучающих алгоритмов требует введения некоторых окрестностей для выборки (в пространстве обучающих выборок) и для выбираемой гипотезы (в пространстве гипотез). В этом плане он близок к подходу, основанному на оценке подмножества используемых гипотез, которое в силу свойств выбранного алгоритма обучения может оказаться гораздо более узким по сравнению со всем пространством гипотез.

Естественно считать малым изменением заданной выборки удаление из неё ровно одного примера (или замену в ней ровно одного примера на другой произвольный пример). Всевозможные такие удаления образуют своеобразную окрестность выборки. Её называют *Loo окрестностью* (*Leave-one-out*). Обучение в окрестности данной выборки приводит к отбору алгоритмом обучения, вообще говоря, различных гипотез, близость ко-

торых можно оценивать, сравнивая частоты ошибок этих гипотез на выборке.

Пусть α – истинное значение целевой функции в точке $\tilde{x}$, а $h_l = A(X_l)$ – выбранная обучающим алгоритмом A по выборке $(\tilde{x}_j, \alpha_j)_{j=1}^{l}$ длины l решающая функция. На практике оценивание решений часто производится при заданной «цене» ошибки. Чтобы учесть эту «цену», вводят функцию потерь, которая определяет, какой «ценой» обходится та или иная ошибка.

Для рассматриваемого нами класса задач, когда ошибка первого и второго рода не различаются, определяя один общий случай ошибки, такая функция имеет вид

$$L(h, \tilde{x}) = \begin{cases} 0, & h(\tilde{x}) = \alpha; \\ m(\tilde{x}), & h(\tilde{x}) \neq \alpha, \end{cases}$$

где $m(\tilde{x})$ – цена ошибка, которая, вообще говоря, зависит от $\tilde{x}$, но чаще всего задаётся константой. В частности, бинарная функция потерь

$$\lambda(h, \tilde{x}) = \begin{cases} 0, & h(\tilde{x}) = \alpha; \\ 1, & h(\tilde{x}) \neq \alpha. \end{cases}$$

является характеристической функцией ошибки.

Обозначим X_l^j обучающую выборку, из которой удалён пример $(\tilde{x}_j, \alpha_j)$, и $A(X_l^i)$ – найденное обучающим алгоритмом A по этой укороченной на единицу выборке X_l^j решающее правило h. Тогда функция потерь $L(A(X_l^j), \tilde{x}_j)$ примет нулевое значение, если в результате обучения с использованием выборки, из которой удалён пример $(\tilde{x}_j, \alpha_j)$, этот пример будет распознаваться безошибочно.

Определение 2.8. Loo-ошибкой называется усреднённая по всем примерам обучающей выборки $(\tilde{x}_j, \alpha_j)_{j=1}^{l}$ величина функции потерь

$$\frac{1}{l}\sum_{j=1}^{l} L(A(X_l^j), \tilde{x}_j) \ .$$

Определение 2.9 [29]. Алгоритм обучения A называется CV_{Loo} устойчивым (*Cross-Validation Leave-one-out*) независимо от распределения, если *для любой вероятностной меры*, для любой длины выборки $l \geq l_0$ найдутся такие положительные $\varepsilon(l), \delta(l) < 1$, что

$$\forall j \in \{1,...,l\} \quad P^l\{\left|L(A(X_l^j),\tilde{x}_j) - L(A(X_l),\tilde{x}_j)\right| \le \varepsilon(l)\} \ge 1-\delta(l),$$

где $\varepsilon(l) \to 0$ и $\delta(l) \to 0$ при $l \to \infty$.□

Согласно определению, CV_{Loo} устойчивость предполагает сколь угодно близкие значения функции потерь для построенного алгоритмом обучения решающего правила в Loo окрестности обучающей выборки с ростом её длины l для каждого из l вариантов удаления одного примера. А для бинарной функции потерь – предполагает в тех же условиях безошибочную классификацию с надёжностью $1-\delta(l)$. Это объясняется тем, что в этом случае модуль разности $\left|\lambda(A(X_l^j),\tilde{x}_j) - \lambda(A(X_l),\tilde{x}_j)\right|$ может принимать только два значения: 0 или 1.

В общем случае, неформально, CV_{Loo} устойчивость можно объяснить так: «удаление одного примера из обучающей выборки почти не влияет на результат ошибки на этом же самом примере».

Определение 2.10 [29]. Обучающий алгоритм называется *согласованным с семейством гипотез* H, если

$$\forall \varepsilon > 0 \ \lim_{l\to\infty} \sup_{P^l} P^l\{Err(A(X_l,\tilde{x})) > \inf_{h\in H} Err(h) + \varepsilon\} = 0.□$$

В определении согласованности супремум берётся по всем возможным вероятностным мерам на множестве обучающих выборок длины l.

Теорема 2.6 [29, с. 178]. CV_{Loo} устойчивость алгоритма обучения A является необходимым и достаточным условием его согласованности с используемым семейством гипотез H при обучении методом минимизации эмпирического риска.

Определение 2.11 [29]. Алгоритм обучения A называется $ELoo_{err}$ устойчивым независимо от распределения, если для любой вероятностной меры при любом $l > l_0$ найдутся такие положительные $\varepsilon(l), \delta(l) < 1$, что

$$P^l\{|\ \mathbf{E}(L(A(X_l),\tilde{x})) - \frac{1}{l}\sum_{j=1}^{l} L(A(X_l^j),\tilde{x}_j)\ | \le \varepsilon(l)\} \ge 1-\delta(l),$$

где $\varepsilon(l) \to 0$ и $\delta(l) \to 0$ при $l \to \infty$; $\mathbf{E}(L(A(X_l),\tilde{x}))$ – математическое ожидание потерь по вероятностной мере P на $X \times \{0,1\}$.□

В случае бинарной функции потерь $\lambda(A(X_l)$ может принимать только два значения: 0 или 1, поэтому неравенство, фигурирующее в определении, будет иметь вид

$$P^l\{|\ Err(A(X_l)) - \frac{1}{l}\sum_{j=1}^{l} \lambda(A(X_l^j),\tilde{x}_j)\ | \le \varepsilon(l)\} \ge 1-\delta(l),$$

где $Err(A(X_l))$ – вероятность ошибки по мере P на $X \times \{0,1\}$. Используя введенные выше обозначения, это неравенство можно записать так:

$$P^l\{|\, Err(h_l) - \frac{1}{l}\sum_{j=1}^{l}\lambda(A(X_l^j), \tilde{x}_j)\,| \le \varepsilon(l)\} \ge 1 - \delta(l).$$

В отличии от предыдущего определения, $ELoo_{err}$ устойчивость предполагает сходимость по вероятности средней ошибки по *Loo* окрестности с ростом длины выборки l к вероятности ошибки решающего правила классификации.

Определение 2.12. Алгоритм обучения A называется LOO *устойчивым*, если он является одновременно и CV_{Loo} устойчивым, и $ELoo_{err}$ устойчивым. □

Таким образом, LOO устойчивость объединяет требования устойчивости как по каждому малому «отклонению» (по одному примеру), так и в среднем (по малой окрестности).

Доказательство этой теоремы приведено в [29].

Различные определения обучаемости, приведенные выше, некоторым образом связывались с семействами гипотез. Но говорить об обучаемости можно и в более общей постановке как о возможности эмпирического обобщения.

Определение 2.13. Алгоритм обучения A называется *симметричным*, если результат его применения $A(X_l)$ к любой допустимой выборке X_l не изменяется при любой перестановке входящих в эту выборку примеров.

Определение 2.14. *Универсальное эмпирическое обобщение* (*universal generalization*) имеет место, если *для любой выбранной алгоритмом обучения гипотезы* частота ошибки этой гипотезы на обучающей выборке сходится по вероятности к её математическому ожиданию при неограниченном росте длины обучающей выборки *независимо от вероятностного распределения*, то есть

$$\forall \varepsilon > 0 \quad P^l\{\left|Err(A(X_l)) - Err_l(A(X_l))\right| \ge \varepsilon\} \to 0 \text{ при } l \to \infty$$

для любой гипотезы $A(X_l)$ *и для любой меры* P^l. □

Установлено, что при обучении методом минимизации эмпирического риска универсальное эмпирическое обобщение эквивалентно согласованности с используемым семейством гипотез H [30]. Но в общем случае универсальное эмпирическое обобщение является самым «сильным» определением обучаемости.

Теорема 2.7 [29]. LOO устойчивость *симметричного алгоритма* обучения классификации *с ограниченной функцией потерь* является достаточным условием для обеспечения универсального эмпирического обобщения.

Доказательство. Оценим математическое ожидание квадрата отклонения математического ожидания ошибки решающего правила (гипотезы) $h = A(X_l)$, выбранной LOO устойчивым алгоритмом обучения A, от эмпирической ошибки этой гипотезы. И распределение P^l, и семейство H, которому принадлежит гипотеза h, полагаются произвольными.

$$\mathbf{E}_l(Err(A(X_l)) - Err_l(A(X_l)))^2 = \mathbf{E}_l(Err(h) - Err_l(h))^2 =$$

$$= \mathbf{E}_l\left(Err(h) - \frac{1}{l}\sum_{j=1}^{l}\lambda(A(X_l^j), \tilde{x}_j) + \frac{1}{l}\sum_{j=1}^{l}\lambda(A(X_l^j), \tilde{x}_j) - Err_l(h)\right)^2$$

$$\le 2\mathbf{E}_l\left(Err(h) - \frac{1}{l}\sum_{j=1}^{l}\lambda(A(X_l^j), \tilde{x}_j)\right)^2$$

$$+ 2\mathbf{E}_l\left(\frac{1}{l}\sum_{j=1}^{l}\lambda(A(X_l^j), \tilde{x}_j) - Err_l(h)\right)^2 .$$

Верхняя оценка, состоящая из двух слагаемых, получена на основе неравенства $(a+b)^2 \le 2a^2 + 2b^2$). Оценим второе слагаемое

$$2\mathbf{E}_l\left(\frac{1}{l}\sum_{j=1}^{l}\lambda(A(X_l^j), \tilde{x}_j) - Err_l(A(X_l))\right)^2$$

$$= 2\mathbf{E}_l\left(\frac{1}{l}\sum_{i=1}^{l}\lambda(A(X_l), \tilde{x}_j) - \frac{1}{l}\sum_{j=1}^{l}\lambda(A(X_l^j), \tilde{x}_j)\right)^2$$

$$= 2\mathbf{E}_l\frac{1}{l^2}\left|\sum_{j=1}^{l}[\lambda(A(X_l), \tilde{x}_j) - \lambda(A(X_l^j), \tilde{x}_j)]\right|^2$$

$$\le 2M\mathbf{E}_l\frac{1}{l}\left|\sum_{i=1}^{l}[\lambda(A(X_l), \tilde{x}_j) - \lambda(A(X_l^j), \tilde{x}_j)]\right|$$

(Здесь использовано условие ограниченности функции потерь, в силу которого $\left|\sum_{j=1}^{l}[\lambda(A(X_l), \tilde{x}_j) - \lambda(A(X_l^j), \tilde{x}_j)]\right| \le M \cdot l$, где M – константа; *в случае бинарной функции потерь* λ *имеем* $M = 1$)

$$\le 2\mathbf{E}_l\frac{1}{l}\sum_{j=1}^{l}\left|\lambda(A(X_l), \tilde{x}_j) - \lambda(A(X_l^j), \tilde{x}_j)\right|$$

$$= 2\frac{1}{l}\sum_{j=1}^{l}\mathbf{E}_l\left|\lambda(A(X_l), \tilde{x}_j) - \lambda(A(X_l^j), \tilde{x}_j)\right|$$

(Учитывая, что A – симметричный алгоритм, $\mathbf{E}_l\,|\cdot|$ – математическое ожидание по вероятностному распределению P^l на множестве обучающих выборок $(\mathrm{X}\times\{0,1\})^l$, получаем далее)

$$= 2\mathbf{E}_l\left|\lambda(A(X_l),\tilde{x}_j) - \lambda(A(X_l^j),\tilde{x}_j)\right|$$

для любого примера $\tilde{x}_j$ из произвольной выборки X_l. Окончательно получаем неравенство

$$\mathbf{E}_l(Err(A(X_l)) - Err_l(A(X_l)))^2$$

$$\le 2\mathbf{E}_l\left(Err(A(X_l)) - \frac{1}{l}\sum_{j=1}^{l}\lambda(A(X_l^j),\tilde{x}_j)\right)^2$$

$$+ 2\mathbf{E}_l\left|\lambda(A(X_l),\tilde{x}_j) - \lambda(A(X_l^j),\tilde{x}_j)\right|,$$

в правой части которого содержатся два слагаемых. Первое слагаемое соответствует определению $ELoo_{err}$ устойчивости, а второе – CV_{Loo} устойчивости. Если оба эти слагаемые при $l\to\infty$ одновременно стремятся к нулю, то, согласно определению, имеет место LOO устойчивость, что влечёт эмпирическое обобщение, поскольку сумма указанных слагаемых является верхней оценкой вероятности математического ожидания ошибки выбранной гипотезы от её эмпирической ошибки. □

Существуют и другие походы к определению устойчивости алгоритмов обучения.

Определение 2.15. Пусть в обучающей выборке $X_l = \{(\tilde{x}_1,\alpha_1),\ldots,(\tilde{x}_j,\alpha_j),\ldots,(\tilde{x}_l,\alpha_l)\}$ произведена замена ровно одного примера $(\tilde{x}_i,\alpha_i)$ на некоторый другой пример $(\tilde{x},\alpha)$. Будем обозначать полученную после такой замены выборку $\overleftarrow{X}_l^i$ и говорить, что $\overleftarrow{X}_l^i$ получена из X_l по правилу ***RO*** (*Replace One*).

Определение 2.16.

1. Обучающий алгоритм A называется *равномерно **RO** устойчивым* на уровне $\varepsilon_{stable}(l)$, если для всех возможных $\overleftarrow{X}_l^i$ и любого замещающего примера $(\tilde{x},\alpha)$

$$\frac{1}{l}\sum_{i=1}^{l}\left|\kappa(A(\overleftarrow{X}_l^i);(\tilde{x},\alpha)) - \kappa(A(X_l);(\tilde{x},\alpha))\right| \le \varepsilon_{stable}(l),$$

где $\kappa(\cdot)$ – число ошибок гипотезы, извлеченной обучающим алгоритмом A при некоторой заданной обучающей выборке.

2. Обучающий алгоритм A называется ***RO*** *устойчивым в среднем* на уровне $\varepsilon_{stable}(l)$, если

$$\left|\frac{1}{l}\sum_{i=1}^{l}\int_{X_l\in X^l}(\kappa(A(\bar{X}_l^i);(\tilde{x},\alpha))-\kappa(A(X_l);(\tilde{x},\alpha)))dP^l(X_l)\right|\le\varepsilon_{stable}(l).$$

3. *Универсальной* ***RO*** устойчивостью в среднем называется ***RO*** устойчивость в среднем для любого вероятностного распределения P.

Определение 2.17. Алгоритм обучения A называется ***AERM*** *правилом* (*Asymptotic Risk Minimizer*), если

$$\int_{X_l\in X^l}(Err(A(X_l))-\inf_{h\in H}Err_l(h))dP^l\le\varepsilon_{erm}(l),$$

и называется *универсальным* ***AERM*** *правилом*, если ***AERM*** имеет место для любого вероятностного распределения P. В этом случае говорят, что имеет место *универсальная* ***AERM*** *устойчивость*.

Определения устойчивости алгоритмов обучения, основанные на замене одного из примеров обучающей выборки некоторым другим примером (***RO***), достаточно схожи с определениями ***LOO***. Их различие проявляется в некоторых результатах обучения при помощи соответствующих алгоритмов [35].

Теорема 2.8 [35, с. 33]. *При использовании* ***AERM*** *правила универсальная* ***RO*** *устойчивость в среднем является необходимым и достаточным условием для обеспечения универсального эмпирического обобщения*.

Примеры устойчивых алгоритмов представлены в ряде научных работ. А. Елисеевым показана устойчивость алгоритма построения линейной регрессии [16,20] с использованием правила ***RO*** согласно следующему определению.

Определение 2.18 [16]. Обучающий алгоритм A называется β-устойчивым относительно неотрицательной вещественной функции потерь L, если

$$\forall X_l\forall X_l^{i,\tilde{u}}\in X^l,\forall\tilde{x}\in X\quad|L(A(X_l),\tilde{x})-L(A(X_l^{i,\tilde{u}}),\tilde{x})|\le\beta,$$

где $X_l^{i,\tilde{u}}$ – выборка, полученная из выборки X_l путём замены в ней $i-$го примера на некоторый другой пример $\tilde{u}$ (правило ***RO***).

Теорема 2.9 [16]. Пусть A есть β-устойчивый обучающий алгоритм, функция потерь удовлетворяет условию $0\le L(A(X_l),\tilde{x})\le M$ для любой обучающей выборки X_l и любого $\tilde{x}\in X$. Тогда для любых $\varepsilon>0$ и $l\ge1$ имеет место неравенство

$$P^l\{|\,Err_l(A(X_l)) - Err(A(X_l))\,| > \varepsilon + 2\beta M\} \le \exp\left(-\frac{2l\varepsilon^2}{(4l\beta + M)^2}\right),$$

и с вероятностью $1-\delta$, где $\delta \to 0$ при $l \to \infty$, справедлива оценка

$$Err(A(X_l)) \le Err_l(A(X_l)) + 2\beta + (4l\beta + M)\sqrt{\frac{\ln \delta^{-1}}{2l}}\,.\square$$

Из последней теоремы видно, что *обучаемость может иметь место независимо от ёмкости класса гипотез* H, *которому принадлежит полученное* β*-устойчивым алгоритмом обучения решающее правило* $A(X_l)$.

Доказательство этой теоремы основано на следующей теореме Мак-Дьярмида:

Теорема 2.10 [28]. Пусть X_l – произвольная выборка, а $X_l^{i,\tilde{u}}$ – выборка, полученная из X_l по правилу ***RO***. Пусть $F : \mathrm{X}^l \to \mathrm{R}$ – любая измеримая функция и найдутся константы c_i, $i = 1,...,m$, такие что

$$\sup_{X_l \in \mathrm{X}^l,\, \tilde{u} \in \mathrm{X}} |\,F(X_l) - F(X_l^{i,\tilde{u}})\,| \le c_i.$$

Тогда

$$P^l\{|\,F(X_l) - \mathbf{E}_l[F(X_l)]\,| > \varepsilon\} \le \exp\left(-\frac{2\varepsilon^2}{\sum_{i=1}^{l} c_i^2}\right).\square$$

Буске́ и Елисеев показали $\beta-$устойчивость тихоновской регуляризации при построении регрессии. Им же принадлежит результат об устойчивости *SVM – Support Vector Machine* [16].

Для методов потенциальных функций и $k-NN$ устойчивость установлена в работе [18]. В работе Р. Рифкина [34] показана устойчивость бэггинга. Это результат не представляется неожиданным, поскольку можно было предположить, что использование совокупности решающих правил с усреднением должно повлечь устойчивость решений. Не рассматривая подробно устойчивость бэггинга, отметим только, что в упомянутой работе Рифкина используется несколько отличающееся от $\beta-$устойчивости определение $\alpha-$устойчивости, применяемое для случая, когда решающие правила не являются бинарными, а принимают вещественные значения.

Определение 2.19 [34]. Обучающий алгоритм A называется α – устойчивым, если

$$\forall X_l \forall X_l^{i,\tilde{u}} \in \mathrm{X}^l, \forall \tilde{x} \in \mathrm{X} \quad |A(X_l)(\tilde{x}) - A(X_l^{i,\tilde{u}})(\tilde{x})| \le \alpha,$$

где $X_l^{i,\tilde{u}}$ – выборка, полученная из выборки X_l путём замены в ней $i-$го примера на некоторый другой пример $\tilde{u}$.

Определение α-устойчивости, в котором оцениваются построенные алгоритмом обучения решающие правила (функция риска не фигурирует), оказалось более удобным для выполнения операций усреднения при использовании машинного обучения для построения регрессии.

2.5. Сравнение моделей и условий обучаемости

Различные определения обучаемости и устойчивости сведены ниже в таблицу 2 для их сравнительного анализа. Из таблицы видно, что в зависимости от определения обучаемости может быть явно указано или нет, в каком семействе (G) содержится целевой концепт, и из какого семейства (H) извлекается гипотеза. Например, в определении *PAC* обучения эти два семейства содержатся. А в определении *Realizable PAC* обучения даже предполагается, что $G = H$.

В теории В. Н. Вапника в определении равномерной сходимости фигурирует только семейство H. Универсальное эмпирическое обобщение не оговаривает явно ни семейство G, ни семейство H. Тем не менее, при любом подходе к машинному обучению его результатом является некоторая выбранная алгоритмом A гипотеза $h = h(A, X_l)$. Для разных обучающих выборок $X_l \in \mathrm{X}^l$ эта выбранная гипотеза, вообще говоря, может оказаться различной. Поэтому $h \in S(A, \mathrm{X}^l) \subseteq H$, где $S(A, \mathrm{X}^l) = \mathrm{Im}\, A$ – множество всевозможных порождаемых алгоритмом A гипотез, а H – любой содержащий это множество класс, имеющий некоторое точное математическое определение. На практике семейство H непосредственно определено выбором для решения задачи машинного обучения некоторой модели: нейронных сетей, решающих деревьев, SVM или др. Но именно алгоритм обучения A определяет сужение $S(A, \mathrm{X}^l)$, оценка ёмкости которого $VCD(S(A, \mathrm{X}^l))$ не превышает $VCD(H)$, и чем она меньше $VCD(H)$, тем точнее окажется оценка обучаемости, использующая VC размерность.

Считается, что *фундаментальным результатом статистической теории обучения является следующий строго доказанный факт* [5,23]. Если H – класс концептов (решающих правил) над проблемной областью X с произвольной вероятностной мерой и выполняются все необходимее условия измеримости, то *следующие три утверждения эквивалентны*:

i. Для класса H имеет место PAC обучаемость для любой вероятностной меры на X.
ii. H является равномерным классом Гливенко-Кантелли.
iii. $VCD(H)$ является конечной.

Табл. 2. Определения и модели обучаемости

Определение обучаемости	*В каком семействе содержится целевой концепт*	*Из какого семейства извлекается гипотеза*	*Для некоторой фиксированной или для любой вероятностной меры*	*Дополнительные требования к алгоритму обучения*	*Условия обучаемости*
PAC	*G*	*H*	для любой	нет	необходимое и достаточное условие – $VCD(H)<\infty$
Realizable PAC	*H*	*H*	для любой	нет	необходимое и достаточное условие – $VCD(H)<\infty$
Poly PAC	*G*	*H*	для любой	$A \in PTIME$	то же, но $\downarrow$
Realizable Poly PAC	*H*	*H*	для любой	$A \in PTIME$	$A \in PTIME$
Agnostic PAC	*не оговаривается*	*H*	некоторая фиксированная	нет	достаточное условие – $VCD(H)<\infty$
Равномерная сходимость по Вапнику (*VUC*)	*не оговаривается*	*H*	некоторая фиксированная	нет	достаточное условие – $VCD(H)<\infty$
Равномерный класс Гливенко-Кантелли	*не оговаривается*	*H*	Для любой равномерно	нет	необходимое и достаточное условие – $VCD(H)<\infty$
LOO устойчивость	*не оговаривается*	*не оговаривается*	для любой	устойчивость в малой окрестности выборки	*LOO* устойчивость – достаточное условие
Универсальная *PO* устойчивость	*не оговаривается*	*не оговаривается*	для любой	устойчивость в малой окрестности выборки	*PO* устойчивость – необходимое и достаточное
Универсальное эмпирическое обобщение	*не оговаривается*	*H*	для любой	нет	необходимое и достаточное условие – $VCD(H)<\infty$
Универсальное эмпирическое обобщение	*не оговаривается*	*не оговаривается*	для любой	устойчивость	*LOO* устойчивость – достаточное условие; универсальная *PO* устойчивость – необходимое и достаточное условие

Рассмотренные выше подходы к определению обучаемости и устойчивости и полученные на их основе результаты позволяют расширить представление о статистической теории обучения.

Теория равномерной сходимости, PAC обучаемость и универсальная способность к обобщению представляют собой достаточно широко определённые модели. В них не оговариваются ни свойства распределения вероятностей, ни особенности алгоритма обучения, которые могут быть произвольными. Фиксация свойств алгоритма обучения (в частности, его заведомая устойчивость) позволяют сузить модель обучения и вследствие этого получить обучаемость даже в случае бесконечной VC размерности семейства гипотез, в которое вложен образ $\mathrm{Im}\, A$ алгоритма обучения A.

Конечность VC размерности также перестаёт быть необходимым условием в некоторых случаях при конкретизации вероятностной меры (например, в случае диффузных или атомарных мер).

Дополнительно выявленные фундаментальные положения дают объяснение практически наблюдаемой обучаемости при использовании некоторых алгоритмов и моделей обучения, несмотря на кажущееся противоречие с VC теорией: в действительности этого противоречия нет.

2.6. *LOO* – устойчивость и обучаемость модели *ABO*

Будем полагать, что обучающая выборка состоит из двух частей – представителей двух непересекающихся классов K_0 и K_1, соответствующих выборочным значениям классифицирующей функции – 0 и 1:

$$X_l = (\tilde{x}_j, \alpha_j)_{j=1}^{l} = T_0 \cup T_1;\ T_0 \cap T_1 = \varnothing;$$
$$T_0 = \{(\tilde{x}, \alpha) : \alpha = 0\};\ T_1 = \{(\tilde{x}, \alpha) : \alpha = 1\};$$
$$| X_l | = l;\ | T_0 | = k_0;\ | T_1 = k_1 |.$$

Потребуем, чтобы в обучающей выборке не содержалось одинаковых точек (что легко выполняется исключением повторов и противоречий). Обозначим X_{l-1} обучающую выборку, из которой удалён ровно один произвольный пример.

Алгоритм (*метод*) *вычисления оценок* (ABO), предназначенный для построения классификатора по заданной обучающей выборке, определяется следующим образом.

Каждой точке $\tilde{x}$ каждого примера $(\tilde{x}, \alpha)$ обучающей выборки ставится в соответствие неотрицательное число – «*вес*» примера (*эталона*) $\omega(\tilde{x})$.

Задана система (*опорных*) подмножеств множества переменных: $\Omega_A \subset \mathfrak{B}(\{1,2,...,n\})$, где $\mathfrak{B}$ – обозначение булеана.

Каждому опорному $\Omega \in \Omega_A$ поставлено в соответствие неотрицательное число $W(\Omega)$ – «вес» опорного множества.

Полагая координаты (признаки) точек числовыми, определяется расстояние между координатами x_i и y_i точек $\tilde{x}$ и $\tilde{y}$ как $\rho(x_i, y_i) = | x_i - y_i |$, $i = \overline{1,n}$.

Определяется функция близости по опорному множеству:
$B_\Omega(\tilde{x}, \tilde{y}) = 1$, если $|\{i \in \Omega : \rho(x_i, y_i) \le \varepsilon\}| \ge q_0$, иначе $B_\Omega(\tilde{x}, \tilde{y}) = 0$;
здесь ε и q_0 – положительные числовые параметры; $q_0 > \frac{1}{2} | \Omega_A |$.

Определяются числовые оценки за класс $\alpha = 0,1$:

$$\Gamma_\alpha(\tilde{x}) = \frac{1}{k_\alpha} \sum_{\Omega \in \Omega_A} \sum_{\tilde{y} \in T_\alpha} \omega(\tilde{y}) \cdot W(\Omega) \cdot B_\Omega(\tilde{x}, \tilde{y}),$$

k_α – параметр.

Решающее правило – алгоритм классификации, вычисляемый согласно определению ABO, состоит в следующем:

$A(X_l; \tilde{\theta}; \tilde{x}) = \alpha$, если $\Gamma_\alpha(\tilde{x}) > \Gamma_{1-\alpha}(\tilde{x}) + \beta$ при заданном параметре β; иначе $A(X_l; \tilde{\theta}; \tilde{x})$ не определено. Здесь $\tilde{\theta}$ обозначает всю совокупность параметров, входящих в модель ABO: $\tilde{\theta} = (\varepsilon, q_0, \tilde{\omega}, \Omega_A, \tilde{W}, \tilde{k}_\alpha, \beta)$.

Для упрощения записи будем обозначать $A(X_l; \tilde{\theta})$ как $A(X_l) = h$ – алгоритм, получаемый методом ABO.

Функция потерь:

$$\lambda(h, \tilde{x}) = \begin{cases} 0, & h(\tilde{x}) = \alpha; \\ 1, & h(\tilde{x}) \ne \alpha \text{ или } h(\tilde{x}) \text{ не определено.} \end{cases}$$

Модифицированный а*лгоритм вычисления оценок* (ABO^*) отличается от описанного выше ABO только областью суммирования для внутренней суммы в формуле вычисления оценок $\Gamma_\alpha(\tilde{x}) = \Gamma_\alpha(X_l, \tilde{x})$:

$$\Gamma^*_\alpha(X_l, \tilde{x}) = \frac{1}{k_\alpha} \sum_{\Omega \in \Omega_A} \sum_{\{\tilde{y} \in T_\alpha :\ \tilde{y} \ne \tilde{x}\}} \omega(\tilde{y}) \cdot W(\Omega) \cdot B_\Omega(\tilde{x}, \tilde{y}).$$

Область суммирования $\{\tilde{y} \in T_\alpha : \tilde{y} \neq \tilde{x}\}$ исключает вклад в оценку самой оцениваемой точки $\tilde{x}$. Тогда при вычислении оценки $\Gamma_\alpha^*(X_l, \tilde{x}_j)$ объект $\tilde{x}_j$ *сам за себя не голосует*.

Теорема 2.11. Алгоритм ABO^* с фиксированными опорными множествами и фиксированными параметрами обеспечивает универсальное эмпирическое обобщение.

Доказательство. Если $X_l^j = X_l \setminus \tilde{x}_j$, то есть если точка $\tilde{x}_j$ заведомо исключена из выборки, то $\Gamma_\alpha^*(X_l^j, \tilde{x}_j) = \Gamma_\alpha^*(X_l, \tilde{x}_j)$, $\alpha \in \{0,1\}$. Поэтому соответствующий алгоритм принятия решений Γ_α^*, основанный на вычислении оценок, будет удовлетворять условию

$$\left|\lambda(A(X_l^j), \tilde{x}_j) - \lambda(A(X_l), \tilde{x}_j)\right| = 0$$

для любой обучающей выборки длины l и любого входящего в неё примера $(\tilde{x}_j, \alpha_j)$. Следовательно, алгоритм ABO^* обладает CV_{Loo} устойчивостью, т.к. требуемое для этого условие, содержащееся в определении 2.9, выполняется с вероятностью единица.

Пусть $h_l = A(X_l)$ – решающее правило, определяемое алгоритмом ABO^* по обучающей выборке X_l, а $Err(h_l) = p$ – вероятность ошибки этого правила. Пусть $\sum_{j=1}^{l} \lambda(A(X_l^j), \tilde{x}_j) = k$ – число ошибок из l вычислений $\lambda(A(X_l^j), \tilde{x}_j)$, где X_l – произвольная выборка. Оценим вероятность неравенства:

$$P^l\{|\, Err(h_l) - \frac{1}{l}\sum_{j=1}^{l} \lambda(A(X_l^j), \tilde{x}_j) \,| > \varepsilon\} \qquad (2.4)$$

$$= P^l\{|\, p - \frac{k}{l} \,| > \varepsilon\} < \frac{p(1-p)}{l\varepsilon^2} \to 0 \text{ при } l \to \infty,$$

что означает наличие $ELoo_{err}$ устойчивости, требующей сходимости по вероятности для любого положительного $\varepsilon < 1$ и $\delta(l) = \frac{p(1-p)}{l\varepsilon^2}$.

Напомним, что статистика $\frac{1}{l}\sum_{j=1}^{l} \lambda(A(X_l^j), \tilde{x}_j)$ является оценкой математического ожидания по методу скользящего контроля. Известно, что эта оценка являются несмещенной [2, с. 130]. Сходимость по вероятности, в соответствии с неравенством (2.4), является частным случаем выражения этого факта.

Таким образом, алгоритм ABO^* является симметричным и LOO устойчивым, и его применение обеспечивает универсальное эмпирическое обобщение.□

Обучение в модели ABO по методу скользящего контроля можно организовать следующим образом.

При условии зафиксированного семейства опорных множеств Ω_A, для каждого из X_l^j, $j=\overline{1,l}$, вариантов исключения одного примера из обучающей выборки решающее правило достраивается (перестраивается) по весам $\tilde{\omega}$ и $\tilde{W}$ так, чтобы пример $(\tilde{x}_j,\alpha_j)$ распознавался безошибочно. При этом, вообще говоря, нет гарантии корректности итогового решающего правила на всей выборке.

Другой способ состоит в усреднении весов, полученных для каждого из l вариантов «настройки» на один пример.

Покажем, что выбор в качестве семейства опорных множеств тупиковых тестов [7] может сохранить устойчивость ABO при обучении по методу скользящего контроля.

Переменным $x_1,...,x_n$, описывающим произвольную точку $\tilde{x}\in\mathbf{X}$, соответствуют столбцы $\{1,...,n\}$ таблицы обучения $X_l=(\tilde{x}_j,\alpha_j)_{j=1_1}^l$, которая состоит из двух подтаблиц – представителей непересекающихся классов:

$$X_l=T_0\cup T;\ T_0\cap T_1=\varnothing;\ T_0=\{(\tilde{x},\alpha):\alpha=0\};\ T_1=\{(\tilde{x},\alpha):\alpha=1\}.$$

Набор столбцов $\tau=\{i_1,...,i_r\}\subset\{1,...,n\}$ позволяет выделить в строке $\tilde{x}$ подстроку $\tau\tilde{x}=(x_{i_1},...,x_{i_r})$. Для широкого класса таблиц обучения и точек $\tilde{x},\tilde{y}\in\mathbf{X}$ *будем полагать заданным предикат* «*различия*» $S(\tau\tilde{x},\tau\tilde{y})\in\{0,1\}$ и говорить, что две точки $\tilde{x}$ и $\tilde{y}$ *различаются по* (*опорному*) *множеству* τ, если $S(\tau\tilde{x},\tau\tilde{y})=0$, и *не различаются*, если $S(\tau\tilde{x},\tau\tilde{y})=1.$

Непустой набор τ называется *тестом* для таблицы X_l, если для любых $\tilde{x},\tilde{y}$ таких, что $\tilde{x}\in T_0,\tilde{y}\in T_1$ выполняется условие $S(\tau\tilde{x},\tau\tilde{y})=0$. *Тупиковым* называется такой тест, что любое его собственное подмножество не является тестом.

Легко убедиться в том, что если τ является тестом таблицы X_l, то τ будет тестом и для любой таблицы X_l^j, $j=\overline{1,l}$. Если же τ – тупиковый тест таблицы X_l, то для таблицы X_l он либо останется тупиковым тес-

том, либо тупиковым тестом будет некоторое его собственное подмножество τ'. В последнем случае будем говорить, что тупиковые тесты τ и τ' $loo-$*эквивалентны*. Назначение таким $loo-$эквивалентным тестам одинаковых весов позволяет получить LOO устойчивую модель ABO с нефиксированными опорными множествами – тупиковыми тестами.

Если все параметры $\tilde{\theta} = (\varepsilon, q_0, \tilde{\omega}, \Omega_A, \tilde{W}, \tilde{k}_\alpha, \beta)$ модели ABO рациональные, то эта модель не только обеспечивает получение рекурсивных решающих правил, но и извлекает его из заведомо рекурсивного семейства.

Литература к главе 2

1. Вапник В. Н. Восстановление зависимостей по эмпирическим данным / В.Н. Вапник. – М. Наука, 1979. – 447 с.
2. Вапник В. Н., Червоненкис А. Я. Теория распознавания образов / В.Н. Вапник, А. Я. Червоненкис. – М.: Наука, 1974. – 416 с.
3. Воронцов К. В. Обзор современных исследований по проблеме качества обучения алгоритмов / К. В. Воронцов // Таврический вестник информатики и математики, 2004. – № 1. – С. 5–24.
4. Донской В.И. Сложность семейств алгоритмов обучения и оценивание неслучайности извлечения эмпирических закономерностей / В.И.Донской // Кибернетика и системный анализ. – 2012. – № 2. – С.86–96.
5. Донской В. И. Эмпирическое обобщение и распознавание: классы задач, классы математических моделей и применимость теорий. Часть I; Часть II / В.И. Донской // Таврический вестник информатики и математики, 2011. – №1. – С. 15 – 26; №2. – С. 31 – 42.
6. Дьяконов А. Г. Алгебраические замыкания обобщённой модели алгоритмов распознавания, основанных на вычислении оценок: дис. … д-ра физ.-мат. наук: 01.01.09.– М.: МГУ, 2009. – 292 с.
7. Журавлев Ю. И. Об алгебраическом подходе к решению задач распознавания или классификации / Юрий Иванович Журавлев // Проблемы кибернетики. – 1978. – Вып. 33. – С. 5 – 68.
8. Растригин Л.. Коллективные правила распознавания / Л.А. Растригин, Р.Х. Эренштейн. — М.: Энергия, 1981. — Р. 244.
9. Роджерс Х. Д. Теория рекурсивных функций и эффективная вычислимость / Хартли Джером Роджерс. – М: Мир, 1972. – 624 с.
10. Рудаков К. В. Об алгебраической теории универсальных и локальных ограничений для задач классификации / К. В. Рудаков // Распознавание, классификация, прогноз. Математические методы и их применение. Вып. 1.– М.: Наука, 1989. – С. 176 – 200.
11. Andonova S., Elisseeff A. A simple algorithm to learn stable machines / Savina Andonova, Andre Elisseeff, Theodoros Evgeniou, Massimillano Pontil // Proceedings of the 15th European Conference on Artificial Intelligence (ECAI). – 2002. – P. 513–520.

12. Blumer A. Learnability and the Vapnik-Chervonenkis Dimension / A.Blumer, A.Ehrenfeucht, D. Haussler, M. Warmuth // J. Assoc. Comp. Mach., 1989. – 35. – P. 929 – 965.
13. Blumer A. Occam's Razor / A. Blumer, A. Ehrenfeucht, D. Haussler, M. Warmuth // Information Processing Letters, 1987. – Vol. 24(6). – P.377 – 380.
14. Blumer A., Littlestont N. Learning faster than promises by the Vapnik-Chervonenkis dimension / Anselm Blumer, Nick Littlestone // Discrete Applied Mathematics, 1989. – Vol. 24. – Iss. 1-3, – P. 47 – 63.
15. Bousquet O., Elisseeff A. Algorithmic Stability and Generalization Performance / Olivier Bousquet , André Elisseeff // Advances in Neural Information Processing Systems. – 2001. – 13. – P. 196 – 202.
16. Bousquet O., Elisseeff A. Stability and Generalization / Olivier Bousquet , André Elisseeff // Journal of Machine Learning Research. – 2002. – 2. – P. 499-526.
17. Breiman L. Arcing Classifiers // The Annals of Statistics / Leo Breiman. – 1998. – Vol. 26. – No.3. – P.801–849.
18. Devroye L., Wagner T. Distribution-free performance bounds for potential function rules / Luc Devroye, T. Wagner // IEEE Transactions on Information Theory. – 1979. – 25. – P. 601 – 604. Режим доступа: https://www.researchgate.net/publication/3083261_Distribution-free_performance_bounds_for_potential_function_rules
19. Ehrenfeucht A. A general lower bound on the number of examples needed for learning / A. Ehrenfeucht, D. Haussler, M. Kearns, L. Valiant // Inform. Computations, 1989. – 82. – P. 247 – 261.
20. Elisseeff A. A Study About Algorithmic Stability and Their Relation to Generalization Performances // Andre Elisseeff. – Technical report. – Laboratoire ERIC, Univ. Lyon 2,2000. – 19 P.
21. Elisseeff A., Pontil M. Leave-one-out error and stability of learning algorithms with applications / Andre Elisseeff, Massimiliano Pontil // Advances in Learning Theory: Methods, Models and Applications. – 2003. – Vol. 190. – NATO Science Series III: Computer and Systems Sciences, chapter 6. – 15 P.
22. Floyd S., Warmuth M. Sample Compression, learnability, and the Vapnik-Chervonenkis dimension / Sally Floyd, Manfred Warmuth // J. Machine Learning, 1995. – Vol. 21. – Iss. 3. – P. 269 – 304.
23. Freund Y. Self bounded learning algorithms / Y. Freund // In Proc. Of the 11th Ann. Conf. on Computational Learning Theory (COLT-98). – N.Y.: ACM Press. – 1998. – P. 247 – 258.
24. Haussler D. Overview of the Probably Approximately Correct (PAC) Learning Framework / David Haussler // AAAI'90 Proceedings of the eighth National conference on Artificial intelligence, 1990. – Volume 2. – P. 1101–1108. Режим доступа: http://www.cbse.ucsc.edu/sites/default/files/smo_0.pdf
25. Hutter M. Algoritmic complexity // Scholarpedia [Электронный ресурс]. – 2008. – 3(1):2573. Режим доступа:

http://www.scholarpedia.org/article/Algorithmic_complexity#Prefix_Turing_machine

26. Kearns M. J., Vazirani U. V. An Introduction to Computational Learning Theory / M. Kearns, U. Vazirani. – MIT Press 1994. – 221 p.
27. Littlestone L., Warmuth M. Relaring Data Compression and Learnability / Nick Littlestone, Manfred K. Warmuth. – Technical Report. – Santa-Cruz: University of California, 1986. – 13 p. [Электронный ресурс]. – Режим доступа: http://users.soe.ucsc.edu/~manfred/pubs/T1.pdf
28. McDiarmid C. On the method of bounded differences / Colin McDiarmid // In Surveys in Combinatorics. – Cambridge University Press, Cambridge, 1989. – London Math. Soc. Lectures Notes. – 141. – P. 148–188.
29. Mukherjee S. Learning theory: stability is sufficient for generalization and necessary and sufficient for consistency of empirical risk minimization / Sayan Mukherjee, Partha Niyogi, Tomaso Poggio, and Ryan Rifkin // Advances in Computational Mathematics. – 2006. – 25. – P. 161–193.
30. Mukherjee S. Statistical Learning : stability is sufficient for generalization and necessary and sufficient for consistency of Empirical Risk Minimization / Sayan Mukherjee, Partha Niyogi, Tomaso Poggio, and Ryan Rifkin. – Massachusetts Institute of Technology, Cambridge, MA, 2004. – 54 p. [Электронный ресурс]. – Режим доступа: http://cbcl.mit.edu/cbcl/publications/ps/mukherjee-AImemoOctNov.pdf
31. Noga A., Shai B. D. Scale-sentitive Dimensions, Uniform Convergence, and Learnability / Alon Noga, Ben David Shai // Journal of the ACM. – 1997. – 44(4). – p. 615 – 631.
32. Ogielski A. T. Information, Probability, and Learning from Examples. Survey / Andrew Ogielski. – Bell Communication Research, 1990. – 87 p. [Электронный ресурс]. – Режим доступа: http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.29.9797&rep=rep1&type=pdf
33. Pestov V. PAC learnability under non-atomic measures: a problem by Vidyasagar / Vladimir Pestov // 21st Int. Conf. "Algorithmic Learning Theory"(ALT 2010). – Canberra, Australia, 2010. – P. 134 – 147.
34. Rifkin M. R. Everything Old Is New Again: A Fresh Look at Historical Approaches in Machine Learning / Ryan Michael Rifkin. Ph.D. in Operation Research. Thesis, MIT, 2002. – 221 P.
35. Sridharan K. Learning from an Optimization Viepoint / Karthik Sridharan. – Thesis for degree of Philosophy in Computer Science. – Chicago:TTIC, 2012. – 217 p. [Электронный ресурс]. – Режим доступа: http://ttic.uchicago.edu/~karthik/thesis.pdf
36. Valiant L. G. A Theory of the Learnable / Leslie G. Valiant // Communications of the ACM, 1984. – Vol. 27. – N11. – P. 1134 – 1142.
37. Vapnik V. N. The Nature of Statistical Learning Theory / Vladimir N. Vapnik. – 2nd ed. – New York: Springer-Verlag, 2000. – 314 p.

3. Параметрические нейронные сети

3.1 Нейронные сети как суперпозиции функций

Нейронные сети – это функциональные суперпозиции, реализующие отображения входного пространства (признаков) $\mathbf{X}$ в выходное пространство или выходное множество решений $\mathbf{Y}$. В зависимости от того, каким является выходное пространство, нейронные сети называют классифицирующими, реализующими регрессионную зависимость или решающими другие задачи. Пространство признаков состоит из точек $\tilde{x} = (x_1,..., x_n)$, которые называют описаниями объектов; для описания объектов используется n переменных. Суперпозиция может иметь, например, такой вид:

$$F(\tilde{x}) = f_0(\varphi_1(f_1(\tilde{x})),..., \varphi_r(f_r(\tilde{x}))); \quad F : \mathbf{X} \to \mathbf{Y}. \qquad (3.1)$$

Функциональные суперпозиции удобно представлять в виде схем, дающих наглядное представление о строении суперпозиции и порядке подстановок одних функций в другие. Так, суперпозиция (1) может быть представлена схемой, которая показана на рис. 3.1.

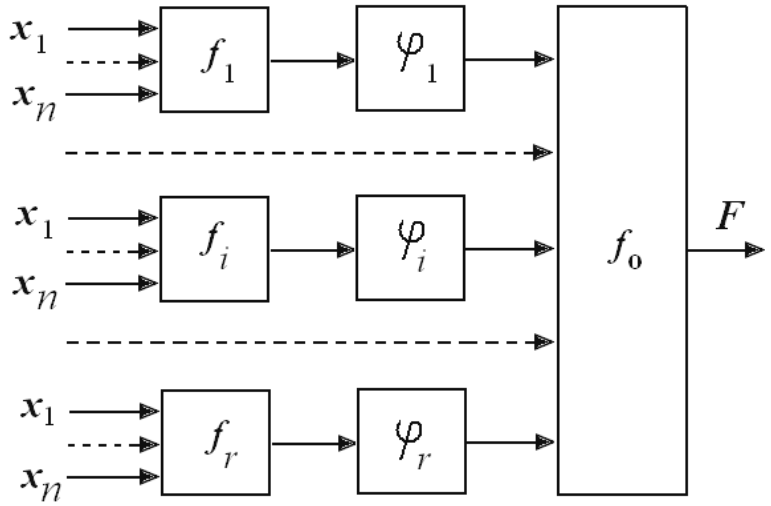

Рис. 3.1. Пример схемного представления суперпозиции

Входящие в суперпозицию функции в общем случае могут зависеть не только от различного числа переменных, но и вообще от разных переменных. Причем количество этих переменных (соответственно – входов элементов схемы) может быть изначально различным для разных элементов или регулироваться в процессе обучения, как это будет показано ниже. Использование случайно выбираемых подмножеств переменных для входов функциональных элементов первого слоя поначалу было удачно угаданной эвристикой. В дальнейшем на основе понятий устойчивости модели обучения оказалось возможным строго обосновать полезность исполь-

зования различных подмножеств переменных при построении суперпозиции (схемы сети).

Возможности каждого класса нейронных сетей определяются набором функций, из которых может быть построена суперпозиция, и ее сложности, которая может оцениваться, например, числом элементарных подстановок – замен одного аргумента значением некоторой функции. О сложности суперпозиции речь пойдет ниже. Но сразу же нужно подчеркнуть, что высокая сложность класса решающих правил, определяемого параметрической нейронной схемой, во многих случаях является её достоинством, дающим возможность обучиться распознаванию сложных объектов. Как было показано выше при рассмотрении обучаемости как свойства модели и алгоритма обучения, для некоторых видов априорных распределений объектов в признаковых пространствах (или наличии устойчивости) даже бесконечная сложность класса решающих правил может не быть препятствием для обучаемости. Напротив, может потребоваться возможность реализации решающих правил достаточно сложного класса. Поэтому рассмотрение возможностей нейронных сетей в плане их сложностных характеристик представляет существенный интерес. В связи с этим ниже приведен ряд теорем о функциональных свойствах суперпозиций функций. Эти теоремы дают представление о возможностях различных по структуре и функциональному составу суперпозиций (и, соответственно, – сетей).

Нейронная сеть может быть формально определена как конечное множество функций с операциями подстановки, сложения и умножения на скаляр. Замыкание множества функций относительно этих операций определяет семейство отображений, реализуемых данной нейронной сетью. С математической точки зрения очень важно оговорить, какие значения принимают переменные $x_1,..., x_n$, и определить, какими являются входящие в суперпозиции функции.

Как правило, полагают, что переменные и функции являются вещественными. Но, строго говоря, при программировании нейронных сетей на компьютерах осуществляется реализация вычислимых (частично рекурсивных) функций и используется дискретное представление (рациональных) чисел в допустимых ограниченных промежутках.

Заметим, что суммирование можно считать функциональным преобразованием: $f_\Sigma(x_1,..., x_k) = \sum_{i=1}^{k} x_i$. Умножение на скаляр также является функцией одной переменной: $f_a(z) = az$.

Будем рассматривать задачи, предполагающие выбор некоторого отображения из заведомо известного класса отображений. Если класс отображений $\mathcal{G}$, в котором содержится отыскиваемое отображение, вложен в семейство отображений $\mathfrak{N}$, реализуемых используемыми нейронными се-

тями, то будем говорить, что семейство $\mathfrak{N}$ является полным относительно рассматриваемого класса задач.

Определение 3.1. Произвольное функциональное семейство $\mathfrak{N}$ называется строго полным относительно другого функционального семейства $\mathcal{G}$, если $\mathcal{G} \subseteq \mathfrak{N}$.

Определение 3.2. Семейство вещественных функций $\mathcal{F}$ вида $f : \mathbf{X} \to \mathrm{R}$ называется ε-полным относительно другого функционального семейства $\mathcal{H} : \mathbf{X} \to \mathrm{R}$,если $\forall h \in \mathcal{H}\ \ \exists f \in \mathcal{F} : \| f - h \| < \varepsilon$, где $\| \cdot \|$ – норма функций, определенных на $\mathbf{X}$.

Если все рассматриваемые функции измеримы относительно меры P, заданной на $\mathbf{X}$, то норму можно задать интегралом Лебега следующим образом:

$$\| f \| = \int_{\mathrm{X}} | f(\tilde{x}) |^2 P(d\tilde{x}).$$

В задачах машинного обучения по прецедентной информации и строгая полнота, и ε-полнота семейств отображений, используемых для нахождения неизвестной функции, имеют важное значение. Строгая полнота гарантирует, что теоретически возможно отыскать точное решение, а ε-полнота гарантирует теоретическую возможность отыскания решения с любой точностью.

В 1957 году А. Н. Колмогоров опубликовал следующий важнейший результат [6].

Теорема 3.1. При любом $n \geq 2$ существуют такие определенные на единичном отрезке $E^1 = [0;1]$ непрерывные действительные функции ψ^{pq}, что каждая определенная на n-мерном единичном кубе E^n непрерывная действительная функция $f = f(x_1,..., x_n)$ представима в виде суперпозиции

$$f(x_1,..., x_n) = \sum_{q=1}^{2n+1} \chi_q [\sum_{p=1}^{n} \psi^{pq}(x_p)],$$

где χ_q – действительные непрерывные функции одной переменной. □

Теорема 3.1 обосновывает существование *конечной двухслойной* нейронной сети, реализующей любую вещественную функцию при условии нормирования значений переменных (гарантирует строгую полноту). Особенно важной является принципиальная возможность использования для построения модели сети конечного и точно определенного числа функциональных элементов. В это число, как видно из теоремы, входит $n(2n+1)$ функциональных элементов ψ^{pq}, образующих нижний слой сети (рис.3.2), $2n+1$ элементов χ_q и $2n+2$ сумматоров. В общей сложно-

сти получается $(n+2)(2n+1)+1=2n^2+5n+3$ элементов. Однако в теореме ничего не говорится об аналитическом виде входящих в суперпозицию функций. Параметров в этой суперпозиции нет. Обучение, которое трудно представить реализуемым, должно заключаться в поиске подходящего набора функций в широчайшем непараметрическом семействе непрерывных функций одной переменной! [5]

Всевозможные функции, удовлетворяющие условию теоремы, при подстановке в суперпозицию Колмогорова определяют полное семейство относительно класса непрерывных функций n переменных.

Кроме полной суперпозиции Колмогорова, существуют также и ε-полные суперпозиции, которые рассматриваются ниже.

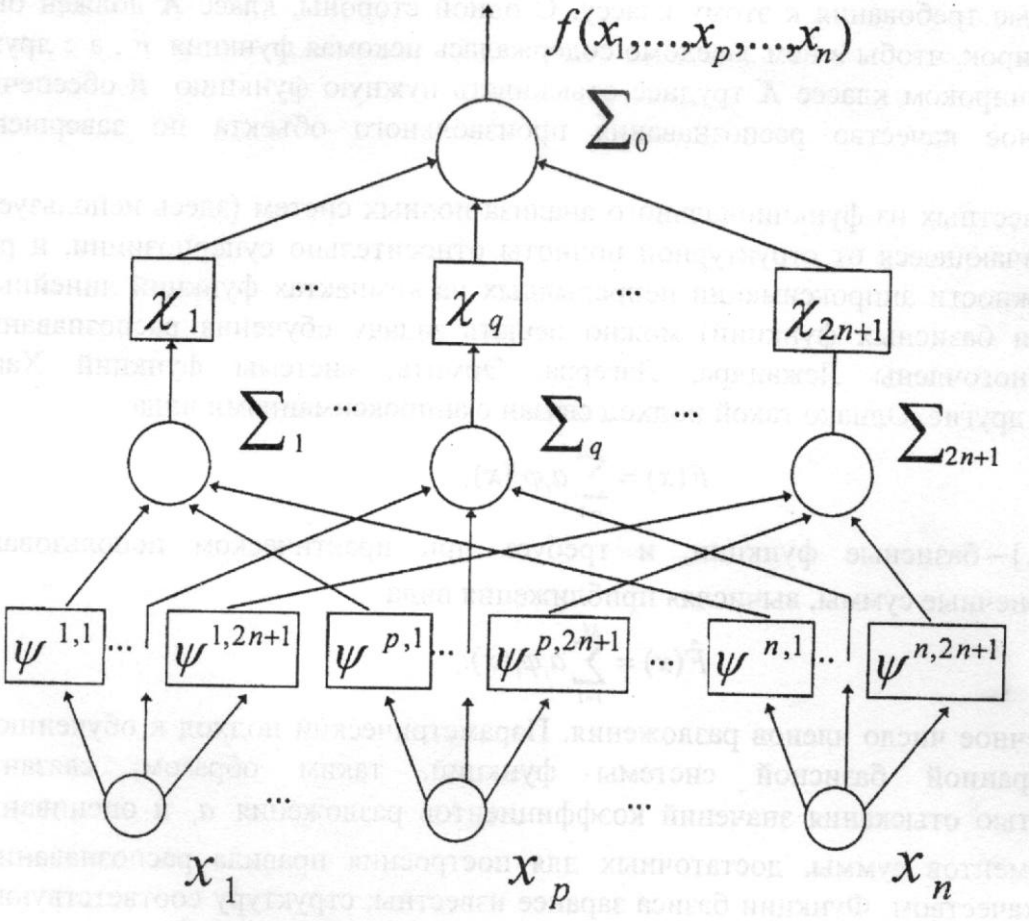

Рис. 3.2. Схема суперпозиции Колмогорова

Обозначим $S^{(n)}$ множество всех многочленов вида $p(\tilde{x})$ от n вещественных переменных $x_1,\ldots, x_n$ с вещественными коэффициентами, где

$$p(\tilde{x})=\sum_{k_1\ldots k_n=0}^{N(p)} a_{k_1\ldots k_n} x_1^{k_1}\cdot\ldots\cdot x_n^{k_n};$$

$N(p)$ – наибольшая степень вхождения переменной в полином $p(\tilde{x})$, которая может принимать любые неотрицательные целые значения; $a_{k_1 \ldots k_n}$ – вещественные коэффициенты. Известен следующий результат, обобщающий теорему Вейерштрасса.

Теорема 3.2 [7]. Пусть $\mathbf{X}$ – произвольное ограниченное множество в n-мерном евклидовом пространстве $\mathbf{R}^n$. Тогда для любой непрерывной на $\mathbf{X}$ функции f и любого $\varepsilon > 0$ найдется многочлен $p(\tilde{x}) \in S^{(n)}$ такой, что

$$\forall \tilde{x} \in \mathrm{X} \quad | f(\tilde{x}) - p(\tilde{x}) | < \varepsilon \quad . \square$$

Согласно этой теореме, конечная, но, вообще говоря, состоящая из *сколь угодно большого числа элементов*, которые реализуют умножение, *однослойная* нейронная сеть, допускающая возможность функционального приближения с любой заданной точностью, может быть реализована в виде полинома – суперпозиции функций, реализующих сложение, умножение на скаляр и перемножение переменных. Вершины такой сети будут соответствовать функциям

$$J_{k_1,\ldots,k_n}(x_1,\ldots,x_n) = x_1^{k_1} \cdot \ldots \cdot x_n^{k_n},$$

где $k_1 + \ldots + k_n = q$ – степень функции (произведения) $J_{k_1,\ldots,k_n}$. Из n переменных можно получить C_{n+q-1}^{q} разных произведений степени q. Поэтому число вершин рассматриваемой сети с одним внутренним слоем (произведений) чрезвычайно велико. Если степени вершин ограничены величиной Q, то число вершин внутреннего слоя будет равно

$$1 + \sum_{q=1}^{Q} C_{n+q-1}^{q} = C_{n+Q}^{Q},$$

и тогда с ростом n (при зафиксированном Q) число вершин внутреннего слоя будет приблизительно равно

$$C_{n+Q}^{Q} \approx \frac{(n+Q)^Q}{Q!}.$$

Так, при 100 входных переменных и наибольшей степени полинома, равной 4, число вершин будет приблизительно равно пяти миллионам. Поэтому такие полиномиальные структуры однослойных нейронных сетей, которые уместно назвать *вертикальными*, применять на практике следует только в тех случаях, когда искомые функции заведомо являются полиномами весьма невысокой степени. Но обеспечение требуемой точности ε может потребовать использования огромного числа вершин внутреннего слоя, и, соответственно, огромного числа параметров $a_{k_1 \ldots k_n}$ – коэффициентов произведений.

Теорема 3.3 [7, с. 34]. Пусть $\mathbf{X}$ – компакт в $\mathbf{R}^n$; $C(\mathbf{X})$ – банахово пространство всех непрерывных функций вида $f:\mathbf{X}\to\mathbf{R}$; $\tilde{x}_1,...,\tilde{x}_m$ – произвольные m точек из $\mathbf{X}$; $\rho(\tilde{x},\tilde{y})$– евклидова метрика. Тогда для любой функции $f\in C(\mathbf{X})$, равной нулю в этих m точках, и для любого $\varepsilon>0$ и некоторого многочлена $Q(\tilde{x})$ найдется многочлен

$$M(\tilde{x})=(\prod_{k=1}^{m}\rho(\tilde{x},\tilde{x}_k))^2 Q(\tilde{x})$$

такой, что

$$\forall\tilde{x}\in \mathrm{X}\quad |f(\tilde{x})-M(\tilde{x})|<\varepsilon. \square$$

Согласно теореме, существует ε-точное приближение любой непрерывной функции, которая определяет соответствующую разделяющую поверхность, задаваемую уравнением $f(\tilde{x})=0$, при заданных m точках на этой поверхности. Здесь, как и в предыдущем случае, можно говорить о вертикальной однослойной нейронной сети, вообще говоря, со сколь угодно большим количеством элементов. При этом гарантируется ε-полная реализация в классе полиномов *согласованных* с любой заданной выборкой (корректных на ней), т. е. дающих нулевую невязку в выборочных точках.

Введем следующие обозначения.

$L_{p,\rho}(\mathbf{R}^n)$ – множество измеримых по Лебегу функций n переменных с конечной нормой

$$\|f(\cdot)\|=\begin{cases}(\int\limits_{\mathbf{R}^n}|f(\tilde{x})\rho(\tilde{x})|^p dx_1...dx_n)^{1/p}, & 1\le p<\infty;\\ ess\sup\limits_{\tilde{x}\in\mathbf{R}^n}|f(\tilde{x})\rho(\tilde{x})|, & p=\infty,\end{cases}$$

где ρ – некоторая весовая функция, $ess\sup$ – существенный супремум.

$C_\rho^0(\mathbf{R}^n)$ – множество всех непрерывных функций n переменных, для которых выполняется условие $\lim\limits_{\tilde{x}\to\infty}|f(\tilde{x})\rho(\tilde{x})|=0$.

Теорема 3.4 (Обобщение теоремы Хехт-Нильсена [1, с.18]). Пусть $\psi=\psi(z)$ – ограниченная, непрерывная, монотонно возрастающая функция. Пусть $1\le p\le\infty$, $f\in L_{p,\rho}(\mathbf{R}^n)$, если $p<\infty$, и $f\in C_\rho^0(\mathbf{R}^n)$, если $p=\infty$. Тогда функцию $f:\mathbf{R}^n\to\mathbf{R}$ можно аппроксимировать в метрике пространства $L_{p,\rho}(\mathbf{R}^n)$ нейронной сетью с двумя слоями, представленной на рис. 3.3 и реализующей функцию $f_\varepsilon:\mathbf{R}^n\to\mathbf{R}$, такую, что $\|f-f_\varepsilon\|_{p,\rho}<\varepsilon$ для любого $\varepsilon>0$. $\square$

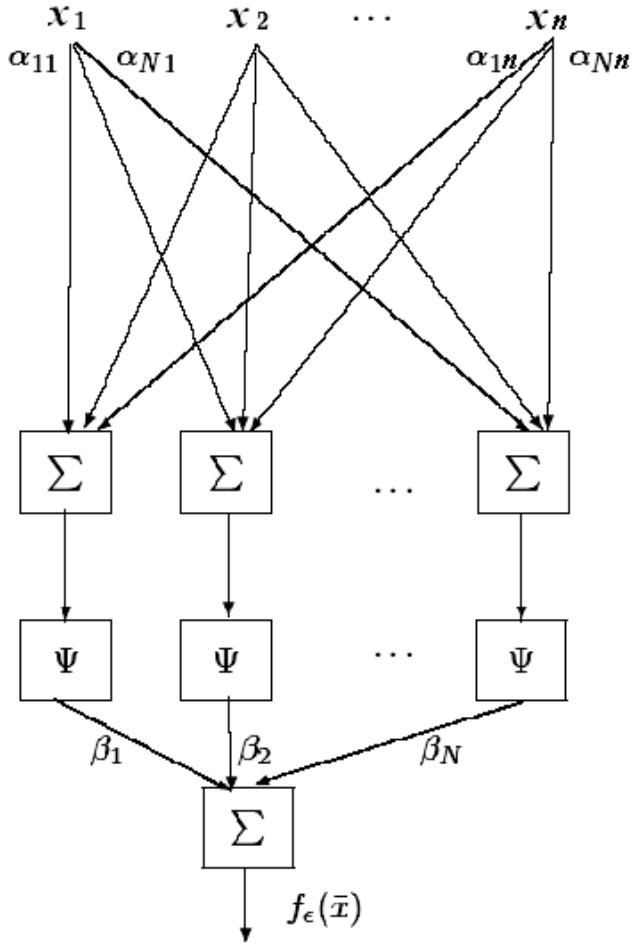

Рис. 3.3. Структура $\mathcal{E}$ -полной сети [1, с.18].

Схема, представленная на рис. 3.3 , эквивалентна суперпозиции функций

$$f_{\varepsilon}(x) = \sum_{i=1}^{N} \beta_i \psi(\sum_{j=1}^{n} a_{ij} x_j),$$

где β_i, a_{ji} – вещественные числовые коэффициенты. Входящую в схему функцию ψ называют функцией активации. Вид этой функции теоремой не определяется. Соответствующая нейронная сеть является *псевдодвухслойной*, содержит $N(n+1)$ числовых параметров и обеспечивает ε-полноту семейства, из которого при обучении извлекается решающая функция. Однако если функция ψ будет выбрана неудачно, то, в общем случае, никакой гарантии возможности обучиться нет.

Функция активации $\psi : \mathbf{R} \to [0,1]$ называется *сигмоидной*, если $\psi(z)$ является монотонно неубывающей на $\mathbf{R}$ и удовлетворяет следующим предельным соотношениям: $\lim\limits_{z \to -\infty} \psi(z) = 0$; $\lim\limits_{z \to +\infty} \psi(z) = 1$. Например, сигмо-

идной является однопараметрическая функция $\psi(z) = \dfrac{1}{1+\exp(-\gamma z)}$, где коэффициент $\gamma > 0$.

Непосредственная реализация нейронных сетей, структура которых определена приведенными выше теоремами, в каждом случае требует выбора для их построения некоторого набора функций. Но даже семейство непрерывных функций одной переменной, определенных на заданном отрезке, имеет мощность континуума. Поэтому задача выбора функций, включаемых в схему сети-суперпозиции, очень сложна.

Сложность семейств полных и ε-полных отображений (суперпозиций, сетей), порождаемых варьированием входящих в них функций и параметров, может оцениваться ёмкостью Вапника-Червоненкиса (VCD) этих семейств. Теоретически VCD может быть неограниченной. Поэтому в общем случае, когда распределение числовых описаний объектов (точек), относящихся к разным классам, произвольно, и на алгоритмы обучения не накладываются специальные ограничения, обучаемость для полных и ε-полных сетей может не иметь места.

Теорема 3.5. Применяемые на практике нейронные сети, реализуемые в виде программ для компьютеров, всегда имеют ограниченную VCD.

Действительно, если все используемые в некоторой суперпозиции $\Im$ функции зафиксированы, а число всех параметров сети равно μ, причем каждый параметр может занимать не более b бит памяти (может записываться только в b-битовые ячейки), то соответствующая компьютерная модель такой сети может иметь не более $2^{\mu b}$ различимых состояний. Следовательно, такая сеть не может реализовать более чем $2^{\mu b}$ решающих функций; поэтому VCD соответствующего класса (обозначим его $\Re(b,\mu)$) ограничена: $VCD(\Re(b,\mu)) \le \mu b$.

Следствие 3.1. Для нейронных сетей, реализуемых в виде компьютерных программ, имеет место равномерная сходимость эмпирических частот ошибок к соответствующим вероятностям с ростом числа примеров.

Чтобы убедиться в справедливости следствия, достаточно вспомнить, что конечность $VCD(\Re(b,\mu))$ является достаточным условием равномерной сходимости частот ошибок к их вероятностям. □

В работе [14] для нейронной сети $NN_{k,1}$ с единственным скрытым слоем, содержащим k элементов, и зафиксированной непараметрической активационной функцией представлена оценка

$$VCD(NN_{k,1}) = (2kn + 4k + 2) \times \log(e(kn + 2k + 1)),$$

где n – размерность признакового пространства. Если такая сеть будет *реализована на компьютере* с использованием b-битовых ячеек для записи параметров, то оценка будет другой. Действительно, число входных параметров для скрытого слоя будет равно $(n+1)k$; число выходных параметров скрытого слоя будет равно $k+1$. Всего получится $nk+2k+1$ параметров, каждый из которых будет использовать не более b бит памяти. Поэтому

$$VCD(NN_{k,1}(b)) \le b(nk+2k+1) \quad [4],$$

и при условии $b < 2\log(e(kn+2k+1))$ последняя оценка будет лучше.

В работе [11] обосновывается положение о том, что для обеспечения способности к обобщению размер весов значит больше чем размер (структурная сложность) нейронной сети. Под размером весов подразумевается положительное вещественное число, ограничивающее сверху весовые коэффициенты сети.

3.2 Нейронные сети и вычислимость

В предисловии к книге [3] А. И. Галушкин (который первым, одновременно и независимо с П. Дж. Вербосом) дал описание метода обучения нейронных сетей, известного как «обратное распространение ошибки», пишет: «Основной идеей создания нейронной ЭВМ – специализированной или универсальной является идея построения ЭВМ как аналогово-цифровой, где "быстрая" – аналоговая часть – выполняет многомерные операции… Алгоритмы настройки коэффициентов нейронных сетей реализуются либо "быстро" в аналоговом виде, либо "медленнее" в виде специализированных цифровых схем, эмулирующих нейронные алгоритмы, либо "медленно" в цифровом виде, например, на универсальной персональной ЭВМ».

Несмотря на то, что физическая реализация некоторых нейросетевых суперпозиций непрерывных функций принципиально возможна, например, на оптоэлектронных элементах, обычно нейронные сети реализуют в виде программ для компьютеров, в которых числовые данные являются дискретными и ограниченными. Поэтому необходимо учитывать следующие строго доказанные утверждения.

1. На цифровых компьютерах реализуются только те функции, которые являются вычислимыми (частично рекурсивными), причем не все, поскольку реальный компьютер, в отличие от машины Тьюринга, имеет конечную память.

2. Класс $\boldsymbol{P}_{comp}$ вычислимых функций значительно у́же класса непрерывных вещественных функций. Более того, класс $\boldsymbol{P}_{comp}$ у́же класса арифметических функций одной целочисленной неотрицательной пере-

менной, принимающих только два значения – ноль и один: $\{f : \mathrm{N} \to \{0,1\}\}$. Последнее легко доказывается диагональным методом с учетом того, что число машин Тьюринга не более чем счетно. Следовательно, *полнота семейств нейросетевых отображений, реализующих класс непрерывных на компакте функций, исключает вычислимость этих отображений в целом*.

3. Реализация нейронных сетей на компьютере приводит к значительному сужению семейства получаемых функций, вложенному в класс $\boldsymbol{P}_{comp}$.

4. Если в одной из рассмотренных сетевых суперпозиций зафиксировать все входящие в нее параметры, то тем самым будет зафиксирована единственная реализуемая сетью функция. Расширение класса реализуемых суперпозицией функций осуществляется за счет варьирования множества параметров сети. При реализации нейронной сети на компьютере каждый параметр может принимать некоторое конечное число значений. Поэтому параметризованные нейронные сети, которые реализуются на компьютерах, всегда представляют некоторые конечные подклассы вычислимых функций.

5. Если структура нейронной сети зафиксирована, все входящие в сеть функции также зафиксированы, то такая сеть реализует конечный класс, состоящий из не более чем $2^{\mu \cdot b}$ функций, где μ – число параметров сети, b – число бит памяти, выделяемых на каждый параметр.

3.3 Обучение нейронной сети прямого распространения (*feed-forward*)

Как показано выше, любая нейронная сеть определяет семейство суперпозиций функций от n входных аргументов вида

$$\tilde{\Im}(x_1,..., x_{n;}\tilde{\omega}),$$

где $\tilde{\omega}$ – конечный набор параметров, задание которых фиксирует одну выбираемую суперпозицию из семейства. При обучении как раз и происходит выбор требуемой суперпозиции. Если суперпозиция на примере $\tilde{x}_j$ из обучающей выборки $(\tilde{x}_j, \alpha_j)_{j=1}^{l}$ определяет выходное значение

$$y_j = \tilde{\Im}(x_{j1},..., x_{jn;}\tilde{\omega}),$$

а правильным выходным значением, согласно обучающей выборке, должно быть α_j, то ошибку можно оценить, например, как $e_j = (\alpha_j - y_j)^2$.
Естественно пытаться минимизировать ошибку, что приводит к задаче

$$\min_{\tilde{w} \in \Omega} \Phi(\tilde{\omega}),$$

где $\Phi(\tilde{\omega}) = \sum_j (\alpha_j - \Im(x_{j1},...,x_{jn;}\tilde{\omega}))^2$; Ω – множество допустимых значений параметров $\tilde{\omega}$.

При условии дифференцируемости функции Φ часто используется градиентный метод, основанный на вычислении градиента

$$\nabla\Phi = (\frac{\partial\Phi}{\partial\omega_1},...,\frac{\partial\Phi}{\partial\omega_\mu}),$$

где μ – число параметров в суперпозиции. Известно, что направление убывания функции Φ в произвольной точке $\tilde{\omega}$ характеризуется её антиградиентом: $-\nabla\Phi$. Произвольно выбирая начальную точку $\tilde{\omega}_0$, согласно градиентному методу вычисляют последовательные приближения (коррекции параметров)

$$\tilde{\omega}_{t+1} := \tilde{\omega}_t - \rho_t \nabla\Phi(\tilde{\omega}_t),$$

где ρ_t – шаговый множитель или шаг спуска, t – порядковый номер шага вычислений. Процесс таких коррекций останавливают, если ошибка по всем примерам обучающей выборки $E(t) = \sum_j (\alpha_j - y_j(t))^2$ становится меньше заданной величины ε или когда $E(t)$ «стабилизируется» – величина ошибки перестаёт уменьшаться.

Хорошо известно, что итерационный градиентный метод и его модификации не гарантируют нахождения глобального минимума многоэкстремальных функций, каковыми чаще всего являются функции ошибок нейросетевых суперпозиций. Но, тем не менее, он лежит в основе всех методов обучения нейронных сетей.

Будем далее рассматривать процесс обучения на примере многослойной нейронной сети, состоящей из входных, внутренних и выходных узлов (вершин) и связывающих их рёбер следующего вида (рис.3.4). В общем случае сеть имеет более одного выхода (узла) и, соответственно, реализует более одной суперпозиции.

Все формальные нейроны рассматриваемой сети будут иметь одинаковые функции активации φ (эти функции в принципе могут быть различными, но это не изменяет процесс обучения). Каждый формальный нейрон, обозначенный на схеме кружком, реализует суперпозицию

$$y_{l,j} = \varphi(\sum_{i=0}^{m(l-1)} \omega_{i,j} y_{l-1,i}),$$

где $y_{l,j}$ - «выход» j-го нейрона слоя l; $y_{l-1,i}, i = \overline{0, m(l-1)}$, - его «входы»; $\omega_{i,j}$ - соответствующие входам «веса»; $m(l)$ - число нейронов в

слое l, $\overline{l=1,L-1}$; φ - функция активации нейрона. Функция φ будет полагаться сигмоидой вида $\varphi(z)=\dfrac{1}{1+\exp(-z)}$.

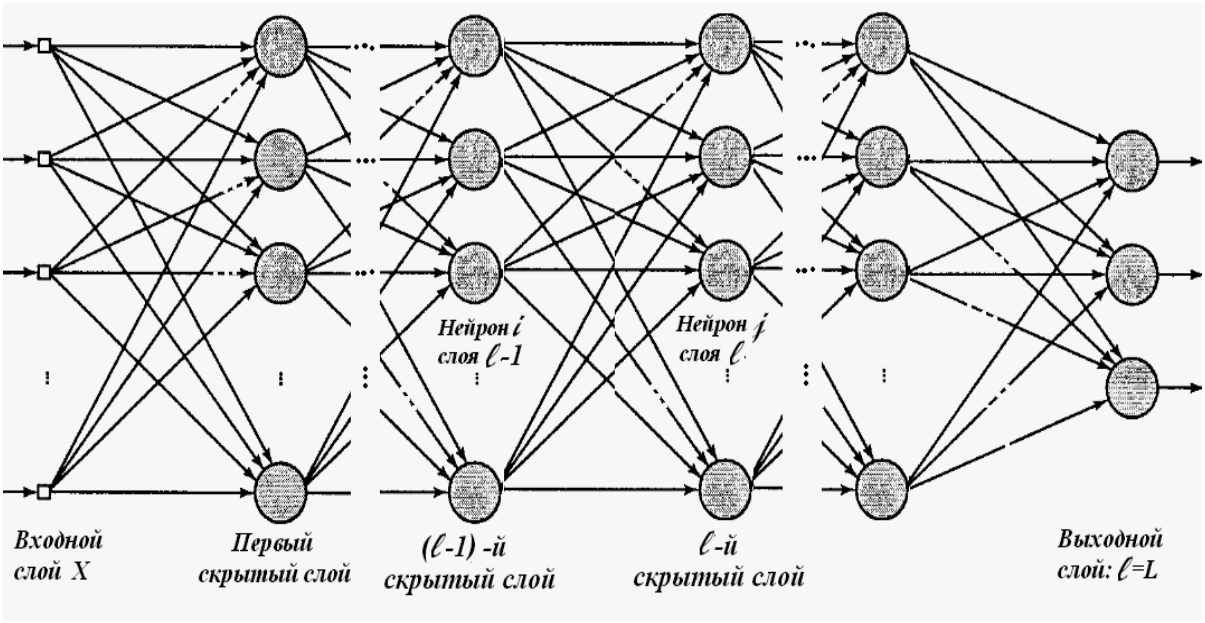

Рис. 3.4. Схема многослойной нейронной сети.

Эта гладкая монотонная нелинейная функция определена для всех вещественных чисел и при этом $0<\varphi(z)<1$, и ее близкий к линейному участок соответствует значениям аргумента z в промежутке $[-1;+1]$.

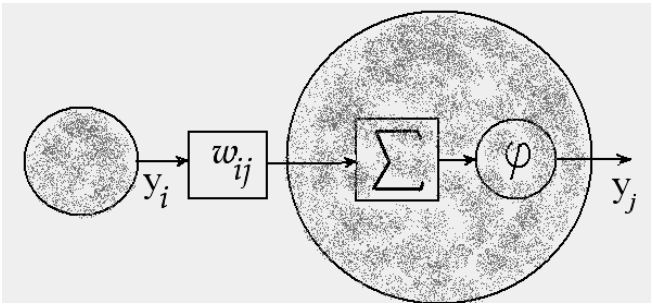

Рис. 3.5. Пояснение к обозначению вершин и ребер на схеме сети.

На схеме многослойной нейронной сети (рис. 3.5) кружками обозначены сумматор и сигмоидная функция, аргументом которой является взвешенная сумма значений выходов нейронов предшествующего слоя. На рис. 3.5 также показано, что «содержится» в кружках схемы сети. Для простоты изображения схемы стрелки на схеме многослойной нейронной сети, соединяющие нейроны i и j, не помечены весами $\omega_{i,j}$. Наличие этих весов полагается «по умолчанию». Такие «взвешенные» стрелки будем называть связями сети.

Входному слою присваивается номер $l=0$. Число точек во входном слое $m(0)$ на единицу больше размерности n исходного пространства векторов (один дополнительный вход соответствует подстановке константы 1). В отличие от внутренних слоёв сети, входной слой предоставляет в качестве своих выходов компоненты исходного входного вектора, подлежащего обработке сетью. Выходной слой сети с номером L в качестве своих выходов содержит результирующие решения (значения сетевой суперпозиции от аргументов – компонент исходного входного вектора). Так, выход нейрона j выходного слоя L, обозначенный $y_{L,j}$, представляется суперпозицией

$$y_{L,j}=\Im_j(\tilde{x})=\varphi(\sum_{i_{L-1}=0}^{m_{L-1}}(\omega_{i_{L-1},j}\varphi(\sum_{i_{L-2}=0}^{m_{L-2}}(\omega_{i_{L-2},i_{L-1}}\cdots$$

$$\cdots(\varphi(\sum_{i_1=0}^{m_1}\omega_{i_1,i_2}(\varphi(\sum_{i_0=0}^{m_0}\omega_{i_0,i_1}x_{i_0})\cdots)\cdots))),$$

где индекс i_0 пробегает «по нейронам» слоя 0 – т.е. по входам сети; индекс i_1 – по слою 1; и так далее, i_{L-1} – по слою $L-1$.

Обучение нейронной сети с вычислительной точки зрения представляет собой нахождение всех неизвестных параметров сети (в рассматриваемом случае это веса $\omega_{i,j}$, взятые по всем определенным в сети индексам). Как и в случае обучения персептрона алгоритмом линейной коррекции Розенблатта-Новикова, обучение нейронной многослойной сети представляет собой процедуру последовательной коррекции весов. Для этого циклически предъявляются примеры из обучающего множества.

3.4 Алгоритм обратного распространения ошибки (*Back Propagation*)

Этот алгоритм обучения нейронных сетей имеет следующее математическое обоснование.

Для упрощения индексации далее индекс i будет использоваться для обозначения нейронов предыдущего слоя, а индекс j – последующего. Будем обозначать буквой k номер нейрона выходного слоя; $k=\overline{1,L}$.

При предъявлении очередного обучающего примера на очередной итерации t нейрон выходного слоя с номером k даёт выходное значение $y_k(t)$, которое может не совпадать с необходимым правильным ответом $\alpha_k(t)$. Ошибка этого нейрона $e_k(t)$ определяется соотношением

$$e_k(t)=\alpha_k(t)-y_k(t).$$

В качестве суммарной ошибки примем сумму

$$E(t) = \frac{1}{2}\sum_k e_k^2(t) = \frac{1}{2}\sum_k (\alpha_k(t) - y_k(t))^2, \qquad (3.2)$$

где индекс k пробегает номера всех нейронов выходного слоя. Очевидно,

$$\frac{\partial E(t)}{\partial e_k(t)} = e_k(t),$$

и такое «удобное» значение производной получается благодаря коэффициенту $\frac{1}{2}$ в формуле (3.2).

Обозначим $v_k(t) = \sum_{i=0}^{m} \omega_{jk}(t) y_j(t))$ суммарное воздействие на нейрон выходного слоя с номером k по связям от всех нейронов предыдущего слоя, включая вес $\omega_{0j}(t)$, умноженный на константу $+1$, соответствующую фиксированному дополнительному входу (свободному члену суммы); m - число нейронов в слое с номером $L-1$. Тогда выход рассматриваемого нейрона определяется по формуле

$$y_k(t) = \varphi(v_k(t)),$$

$$E(t) = \frac{1}{2}\sum_k (\alpha_k(t) - \varphi(v_k(t)))^2 = \frac{1}{2}\sum_k (\alpha_k(t) - \varphi(\sum_{j=0}^{m} \omega_{jk}(t) y_j(t)))^2 .$$

Обучение состоит в изменении (коррекции) всех весов $\omega_{ij}(t)$ сети на величины $\Delta\omega_{ij}(t)$, пропорциональные частным производным $\partial E(t)/\partial\omega_{ij}(t)$. Для весов, соответствующих входным соединениям нейронов выходного слоя

$$\frac{\partial E(t)}{\partial \omega_{jk}(t)} = \frac{\partial E(t)}{\partial e_k(t)} \cdot \frac{\partial e_k(t)}{\partial y_k(t)} \cdot \frac{\partial y_k(t)}{\partial v_k(t)} \cdot \frac{\partial v_k(t)}{\partial \omega_{jk}(t)} = e_k(t) \cdot (-1) \cdot \varphi'(v_k(t)) \cdot y_j(t).$$

Корректирующей добавкой к весам входов выходного слоя сети будет величина

$$\Delta\omega_{jk}(t) = -\gamma \frac{\partial E(t)}{\partial \omega_{jk}(t)} = \gamma \cdot e_k(t) \cdot \varphi'(v_k(t)) \cdot y_j(t)$$

где γ - параметр, позволяющий регулировать скорость обучения (скорость градиентного спуска). Обозначим

$$\delta_k(t) = -\frac{\partial E(t)}{\partial v_k(t)} = e_k(t) \cdot \varphi'(v_k(t))$$

и будем называть $\delta_k(t)$ *локальным градиентом.* Для произвольного нейрона j локальный градиент будем определять по такой же формуле:

$$\delta_j(t) = -\frac{\partial E(t)}{\partial v_j(t)}.$$

Для нейронов выходного слоя локальные градиенты равны произведению ошибки на соответствующем выходе на производную $\varphi'(v_k(t))$. Подчеркнем, что *локальный градиент нейрона пропорционален его ошибке*. Используя локальный градиент, можно записать

$$\Delta\omega_{jk}(t) = \gamma\delta_k(t)y_j(t).$$

Эта формула определяет коррекцию входных весов нейронов выходного слоя, для которых ошибка, используемая при нахождении локального градиента $\delta_k(t)$, вычисляется непосредственно путем сравнения требуемого правильного выходного значения $\alpha_k(t)$, которое известно, с полученным выходным значением $y_k(t)$. Но *для внутренних нейронных слоев требуемые правильные выходные значения неизвестны*.

Пусть теперь j - номер любого из m_j нейронов скрытого слоя, предшествующего выходному слою, а $y_j(t)$ - его выход,

$$y_j(t) = \varphi(v_j(t)) = \varphi(\sum_{i=0}^{m_j}\omega_{ij}(t)y_i(t)).$$

Определим частную производную функционала ошибки $E(t) = \frac{1}{2}\sum_k e_k^2(t)$ по выходу нейрона j, учитывая, что

$$e_k(t) = \alpha_k(t) - y_k(t) = \alpha_k(t) - \varphi(v_k(t)) = \alpha_k(t) - \varphi(\sum_{i=0}^{m}\omega_{jk}(t)y_j(t)):$$

$$\frac{\partial E(t)}{\partial y_j(t)} = \sum_k e_k(t)\frac{\partial e_k(t)}{\partial y_j(t)} = \sum_k e_k(t)\cdot\frac{\partial e_k(t)}{\partial v_k(t)}\cdot\frac{\partial v_k(t)}{\partial y_j(t)} =$$

$$= -\sum_k e_k(t)\cdot\varphi'(v_k(t))\cdot\omega_{jk}(t) = -\sum_k \delta_k(t)\cdot\omega_{jk}(t).$$

Используя полученную частную производную по $y_j(t)$, можно найти

$$\frac{\partial E(t)}{\partial\omega_{ij}(t)} = \frac{\partial E(t)}{\partial y_j(t)}\cdot\frac{\partial y_j(t)}{\partial\omega_{ij}(t)} = -\varphi'(v_j(t))y_i\sum_k\delta_k(t)\cdot\omega_{jk}(t);$$

$$\Delta\omega_{ij}(t) = -\gamma\cdot\frac{\partial E(t)}{\partial\omega_{ij}(t)} = \gamma\varphi'(v_j(t))y_i\sum_k\delta_k(t)\cdot\omega_{jk}(t) = \gamma\delta_j(t)y_i(t), \text{ где}$$

$$\delta_j(t) = -\frac{\partial E(t)}{\partial v_j(t)} = -\frac{\partial E(t)}{\partial y_j(t)} \cdot \frac{\partial y_j(t)}{\partial v_j(t)} = -\frac{\partial E(t)}{\partial y_j(t)} \cdot \varphi'(v_j(t));$$

$$\delta_j(t) = \varphi'(v_j(t)) \sum_k \delta_k(t) \cdot \omega_{jk}(t);$$

$$\Delta\omega_{ij}(t) = \gamma \delta_j(t) y_i(t).$$

Заметим, что для выходного слоя была получена аналогичная формула

$$\Delta\omega_{jk}(t) = \gamma \delta_k(t) y_j(t),$$

поэтому $\delta_j(t)$ – локальный градиент нейрона j скрытого слоя.

Используя получение формулы, получаем следующий окончательный результат.

Для выходного слоя

$$\omega_{jk}(t+1) := \omega_{jk}(t) + \Delta\omega_{jk}(t) = \omega_{jk}(t) + \gamma \delta_k(t) y_j(t).$$

Для внутренних слоев с номерами $2,3,\ldots,L-1$

$$\omega_{ij}(t+1) := \omega_{ij}(t) + \Delta\omega_{ij}(t) = \omega_{ij}(t) + \gamma \delta_j(t) y_i(t).$$

Для слоя с номером 1

$$\omega_{i1}(t+1) := \omega_{i1}(t) + \Delta\omega_{i1}(t) = \omega_{i1}(t) + \gamma \delta_1(t) x_i(t).$$

Необходимые для вычисления приращений значений параметров локальные градиенты вычисляются рекуррентно:

$$\delta_k(t) = e_k(t) \cdot \varphi'(v_k(t));$$

$$\delta_i(t) = \varphi'(v_i(t)) \sum_j \delta_j(t) \cdot \omega_{ij}(t),\ i = \overline{1, L-1}.$$

Сумма в последней формуле берётся по всем нейронам слоя, следующего за слоем, в котором содержится нейрон i.

Полагая, что используется сигмоидная функция вида

$$\varphi(v_j(t)) = \frac{1}{1+e^{-v_j(t)}},$$

можно выразить нужную для вычислений производную $\varphi'(v_i(t))$ следующим образом:

$$\varphi'(v_j(t)) = \frac{e^{-v_j(t)}}{(1+e^{-v_j(t)})^2} = \frac{1}{1+e^{-v_j(t)}} - \frac{1}{(1+e^{-v_j(t)})^2} = \varphi(v_j(t)) - \varphi^2(v_j(t)).$$

Учитывая, что $\varphi(v_j(t)) = y_j(t)$, получаем

$$\varphi'(v_j(t)) = y_j(t)(1 - y_j(t)).$$

Для нейрона k выходного слоя:

$$\delta_k(t) = y_k(t)(1 - y_k(t) e_k(t);$$

$$\omega_{jk}(t+1) := \omega_{jk}(t) + \gamma \delta_k(t) y_j(t),$$

и расчет проводится для всех номеров j нейронов слоя, предшествующего выходному. Для произвольного нейрона i скрытого слоя;

$$\delta_i(t) = y_i(t)(1 - y_i(t)\sum_j \delta_j(t) \cdot \omega_{ij}(t),$$

где сумма берется по всем номерам нейронов слоя, непосредственно следующего за слоем, в котором содержится нейрон i;

$$\omega_{ij}(t+1) := \omega_{ij}(t) + \gamma\delta_j(t)y_i(t).$$

Алгоритм обратного распространения ошибки состоит из следующих этапов.

1º Инициализация – задание начальных значений весам связей сети. Строго обоснования выбора этих начальных значений нет, поскольку невозможно дать начальное приближение, обеспечивающее в результате итераций гарантированное «попадание» в точку глобального экстремума. Представляется удобным задать начальные веса как случайные числа с равномерным распределением, нулевым математическим ожиданием из промежутка $[-1;1]$

2º Итерации, состоящие из двух «проходов». На каждой итерации происходит предъявление очередного вектора обучающей выборки, расчет выходов всех нейронов (прямой проход) и коррекция всех параметров сети, начиная от выходного слоя (обратный проход).

Прямой проход обеспечивает нахождение сумм взвешенных входов и значений выходов всех нейронов сети. При этом вычисления происходят, начиная с первого слоя далее к выходному слою сети: иначе функциональную сетевую суперпозицию вычислить нельзя. Поэтому первый проход называют *прямым*.

Обратный проход реализуется в обратную сторону – от последнего слоя сети к первому, следующему за входным слоем. Сначала используются полученные на прямом проходе ошибки выходного слоя сети и локальные градиенты этого слоя, пропорциональные ошибкам. Зная эти градиенты, можно вычислить локальные градиенты нейронов следующего слоя по направлению ко входу сети. Последовательное вычисление локальных градиентов «в обратную» сторону обеспечивает рекуррентное оценивание ошибок нейронов сети. Именно поэтому рассматриваемый алгоритм называют «*обратным распространение ошибки*».

В рамках настоящей работы вполне достаточно данного выше описания нейронных сетей и наиболее распространенного алгоритма обучения, поскольку мы рассматриваем, главным образом, обучаемость и процессы обучения с алгоритмической точки зрения. Модификации метода обратного распространения ошибки и другие алгоритмы обучения нейронных сетей можно найти в литературе [2, 3, 9, 10].

3.5 Обучение с адаптацией структуры сети по связям

Установлено, что рост сложности нейронной сети в общем случае ухудшает вероятностные оценки точности решений при дальнейшем её применении. В то же время недостаточная сложность сети может заведомо не позволить обучиться для вычисления сложных функций. Поэтому имеет смысл говорить о некоторой оптимальной сложности сети.

Будем говорить, что *нейронная сеть является эмпирически оптимальной по сложности в заданном классе сетей относительно данной обучающей выборки*, если

i) как дальнейшее увеличение её сложности (переобучение),
i) так и намеренное уменьшение её сложности (сужение используемого класса, приводящее к невозможности обучиться)

влечёт рост оценки эмпирической ошибки.

Управление структурой сети возможно за счет введения специальных «параметров соединения» следующим образом. Пусть z – значение, передаваемое по некоторой связи сети. Суперпозиция $u = \beta z$, где β – управляющий параметр связи, принимающий только два значения: 1 – «соединение есть» и 0 – «соединения нет», позволяет подключать или отключать части схемы сети (рис).

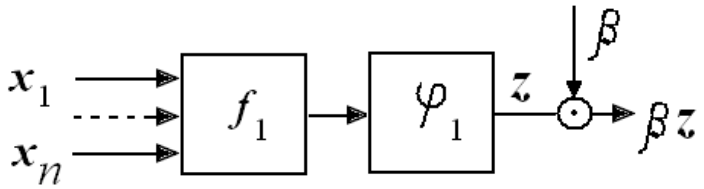

Рис. 3.6. Пример ведение бинарного управляющего параметра

Если связь оценивается настраиваемым параметром β, который принимает значения в некотором промежутке, включающем ноль, то «сброс» этого параметра в ноль приводит к исключению соединения. На основе такого представления веса связей можно рассматривать как управляющие параметры и управлять в процессе обучения структурой сети следующим образом. *По мере обучения параметры связи, принимающие значения, достаточно близкие к нулю*, «*сбрасываются*» *в ноль*. В таком случае структура и, соответственно, сложность нейронной сети может уменьшиться в процессе обучения.

Исходная сеть с заданными функциями активации (возможно, параметрическими) изначально допускается полной по связям: для любой пары узлов смежных слоёв может быть задана ненулевая (по весу) связь. Обозначим $\omega_{i_{k-1}j_k}$ коэффициент связи между узлом номер i_{k-1} слоя $k-1$ и узлом j_k слоя k, $k = 1,...,r$, слой с номером 0 полагается входным. Обозна-

чим $n_0, n_1, ..., n_k, ..., n_r$ – число узлов соответственно в слоях $0,1,...,k,...,r$. Тогда слой k может иметь $n_{k-1} \times n_k$ входов.

При инициализации *каждая вершина* произвольного *внутреннего слоя* k «*нагружается*» *не всеми ненулевыми* n_{k-1} *входами*, а только частью их. Для этого осуществляется случайный выбор $\alpha \cdot n_{k-1}$ связей, где α – эвристический параметр, определяющий долю «нагружаемых» связей.

В процессе обучения коэффициенты связей, значения которых близки к нулю (меньше заданной величины θ – *порога сброса*), принудительно сбрасываются в ноль.

Если в результате выполнения заданного числа итераций не достигается требуемая эмпирическая точность, то производится усложнение структуры – добавление ненулевых связей. Для этого в сеть добавляются Δ случайно выбранных параметров, значения которых были нулевыми с начальным значением q_0. Далее процесс обучения продолжается, если удаётся повышать эмпирическую точность.

Для уточнения и усовершенствования процедур адаптивного структурного обучения сети можно использовать идеи, используемые в генетических алгоритмах.

3.6 Метод опорных векторов (*Support Vector Machine – SVM*)

Линейное разделение точек обучающей выборки $S = (\tilde{x}_j, \alpha_j)_{j=1}^{l}$, $\tilde{x} = (x_1, ..., x_n) \in \mathrm{R}^n$, гиперплоскостью $(\tilde{\omega}, \tilde{x}) - c = 0$ (здесь $(\tilde{\omega}, \tilde{x})$ – скалярное произведение) возможно только тогда, когда выпуклые оболочки двух подмножеств точек, представляющих классы в обучающей выборке,

$$M_0 = \{\tilde{x}_j : \alpha_j = 0 \wedge (\tilde{x}_j, \alpha_j) \in S\} \text{ и}$$

$$M_1 = \{\tilde{x}_j : \alpha_j = 1 \wedge (\tilde{x}_j, \alpha_j) \in S\}$$

не пересекаются. Линейная разделимость эквивалентна существованию вектора $\tilde{\omega}$ и числа c таких, что $(\tilde{\omega}, \tilde{x}_j) - c < 0$, если $\alpha_j = 0$ и $(\tilde{\omega}, \tilde{x}_j) - c > 0$, если $\alpha_j = 1$. Пусть множества M_0 и M_1 линейно разделимы. Обозначим

$$c_1(\tilde{\omega}) = \min_{\tilde{x}_j \in M_1} (\tilde{\omega}, \tilde{x}_j) > c > c_2(\tilde{\omega}) = \max_{\tilde{x}_j \in M_0} (\tilde{\omega}, \tilde{x}_j),$$

$$\rho(\tilde{\omega}) = \frac{1}{2}(c_1(\tilde{\omega}) - c_2(\tilde{\omega})),$$

где $\rho(\tilde{\omega})$ – полусумма расстояний ближайших точек классов M_0 и M_1 до разделяющей гиперплоскости, которую будем называть *зазором*.

Отыскивать вектор $\tilde{\omega}$, определяющий линейное правило разделения классов, имеет смысл так, чтобы зазор между точками – представителями разных классов $\rho(\tilde{\omega})$ был максимальным. Такой максимум существует и является единственным.

Не теряя общности, паре значений номеров классов $\{0,1\}$ для удобства поставить во взаимно однозначное соответствие пару значений $\{-1,+1\}$. Это приведёт к замене значений α_j классифицирующей функции на значения y_j: $y_j = -1$, если $\alpha_j = 0$ и $y_j = +1$, если $\alpha_j = 1$.

Тогда неравенство $f(\tilde{x}) = y_j((\tilde{\omega}, \tilde{x}_j) - c) > 0$ будет выполняться для всех точек $\tilde{x}_j$ обучающей выборки S; $j = 1,...,l$. А если принять $a = \min_j |((\tilde{\omega}, \tilde{x}_j) - c)|$, то для всех j должно выполняться неравенство $y_j \cdot \frac{1}{a}((\tilde{\omega}, \tilde{x}_j) - c) \geq 1$. Из этого следует, что отыскивать неизвестный вектор разделяющей гиперплоскости можно используя систему неравенств вида:

$$y_j \cdot ((\tilde{\omega}, \tilde{x}_j) + b) \geq 1, \quad j = 1,...l,$$

где b – некоторое подходящее число. Эта система определяет разделяющую гиперплоскость $(\tilde{\omega}, \tilde{x}) + b = 0$, которая будет заключения между граничными гиперплоскостями $(\tilde{\omega}, \tilde{x}) + b - 1 = 0$ и $(\tilde{\omega}, \tilde{x}) + b + 1 = 0$. Последние два уравнения можно записать в эквивалентном виде:

$$\frac{(\tilde{\omega}, \tilde{x}_j)}{\| \tilde{\omega} \|} + \frac{b}{\| \tilde{\omega} \|} = \frac{1}{\| \tilde{\omega} \|}; \quad \frac{(\tilde{\omega}, \tilde{x}_j)}{\| \tilde{\omega} \|} + \frac{b}{\| \tilde{\omega} \|} = \frac{-1}{\| \tilde{\omega} \|}.$$

Расстояние между этими двумя гиперплоскостями будет равным $\frac{2}{\| \tilde{\omega} \|}$.

Поэтому *задачу максимизации зазора можно свести к задаче условной минимизации нормы весового вектора* $\| \tilde{\omega} \|$ (*квадратичного функционала*) в форме

$$\begin{cases} \min \sum_{i=1}^{n} \omega_i^2; \\ y_j((\tilde{\omega}, \tilde{x}_j) + b) \geq 1, \quad j = 1,...,l. \end{cases}$$

где b – некоторое число.

Составим функцию Лагранжа

$$L(\tilde{\omega}, b, \tilde{a}) = \frac{1}{2}(\tilde{\omega}, \tilde{\omega}) - \sum_{j=1}^{l} a_j (y_j((\tilde{\omega}, \tilde{x}_j) + b) - 1), \qquad (3.3)$$

где $a_j \geq 0$ – множители Лагранжа; $\tilde{a} = (a_1,...,a_l)$. Для нахождения седловой точки функции (3.3), нужно минимизировать её по $\tilde{\omega}$ и b, а затем максимизировать по неотрицательным множителям a_j.

$$\frac{\partial}{\partial \tilde{\omega}} L(\tilde{\omega}, b, \tilde{a}) = \tilde{\omega} - \sum_{j=1}^{l} a_j y_j \tilde{x}_j = 0$$

$$\tilde{\omega} = \sum_{j=1}^{l} a_j y_j \tilde{x}_j \qquad (3.4)$$

$$\frac{\partial}{\partial b} L(\tilde{\omega}, b, \tilde{a}) = \sum_{j=1}^{l} a_j y_j = 0$$

Подстановка (3.4) в (3.3) с учетом равенства $\sum_{j=1}^{l} a_j y_j = 0$ позволяет получить функцию

$$F(\tilde{a}) = \sum_{j=1}^{l} a_j - \frac{1}{2} \sum_{j=1}^{l} \sum_{k=1}^{l} a_j a_k y_j y_k (\tilde{x}_j, \tilde{x}_k).$$

Функцию $F(\tilde{a})$ нужно максимизировать при условии $\sum_{j=1}^{l} a_j y_j = 0$.

Пусть максимум достигается в точках $a_j = a_j^0$, $j = 1,..,l$, определяя в соответствии с (3.4) параметры оптимальной гиперплоскости

$$\tilde{\omega}^0 = \sum_{j=1}^{l} a_j^0 y_j \tilde{x}_j, \qquad (3.5)$$

$$b_0 = \frac{\min\limits_{y_j=1}(\tilde{\omega}^0, \tilde{x}_j) + \max\limits_{y_j=-1}(\tilde{\omega}^0, \tilde{x}_j)}{2}.$$

Условия Куна-Таккера требуют выполнения следующих соотношений для $\tilde{\omega}^0$ и b_0:

$$a_j^0 (y_j((\tilde{\omega}^0, \tilde{x}_j) + b_0) - 1) = 0, \quad j = 1,..,l.$$

Видно, что множитель a_j^0 может быть большим нуля только при условии $y_j((\tilde{\omega}^0, \tilde{x}_j) + b_0) - 1 = 0$, что выполняется только в случаях, когда

$$(\tilde{\omega}^0, \tilde{x}_j) + b_0 = 1 \quad \text{или} \qquad (3.6)$$

$$(\tilde{\omega}^0, \tilde{x}_j) + b_0 = -1 \qquad (3.7)$$

Точки $\tilde{x}_j$, которые удовлетворяют уравнению (3.6) или (3.7) лежат на граничных плоскостях, называемых *опорными векторами*. Число k опорных векторов может лежать в отрезке $2 \leq k \leq l$. Обозначим опорные векторы

$$\tilde{x}_{s_1},...,\tilde{x}_{s_m},...,\tilde{x}_{s_k},$$

где $s_1,\ldots,s_m,\ldots,s_k$ подмножество номеров из множества $\{1,2,\ldots,l\}$. Искомый вектор весов оптимальной разделяющей гиперплоскости определяется из (3.5) с учетом удаления из суммы нулевых коэффициентов

$$\tilde{\omega}^0 = \sum_{m=1}^{k} a_{s_m}^0 y_{s_m} \tilde{x}_{s_m}$$

и является линейной комбинацией опорных векторов. Оптимальная разделяющая гиперплоскость определяется уравнением

$$\sum_{m=1}^{k} a_{s_m}^0 y_{s_m} (\tilde{x}_{s_m}, \tilde{x}) \quad + \quad b_0 = 0,$$

а решающее правило классификации имеет вид

$$h(\tilde{x}) = sign\{ \sum_{\substack{\text{по номерам } j, \\ \text{принадлежащим} \\ \text{множеству номеров} \\ \text{опорнных векторов}}} a_j^0 y_j (\tilde{x}_j, \tilde{x}) \quad + \quad b_0 \}.$$

На рис. 3.7.А приведены две граничные прямые, разделяющие точки двух классов, на которых лежат 5 опорных точек.

Если линейная разделимость невозможна (как показано на рис.3.7.Б), то применяется переход $\mathrm{R}^n \xrightarrow{\varphi} \mathcal{H}$ в так называемое *спрямляющее пространство* более высокой размерности со скалярным произведением так, чтобы образы $\tilde{\varphi}(\tilde{x})$ точек $\tilde{x}$ разных классов из обучающей выборки оказались линейно разделимыми в $\mathcal{H}$. Тогда оптимальную разделяющую гиперплоскость в новом пространстве $\mathcal{H}$ можно найти при помощи метода опорных векторов. Линейная разделяющая поверхность в $\mathcal{H}$ будет иметь в качестве прообраза некоторую нелинейную разделяющую поверхность $F(\tilde{x}) = 0$ в пространстве R^n.

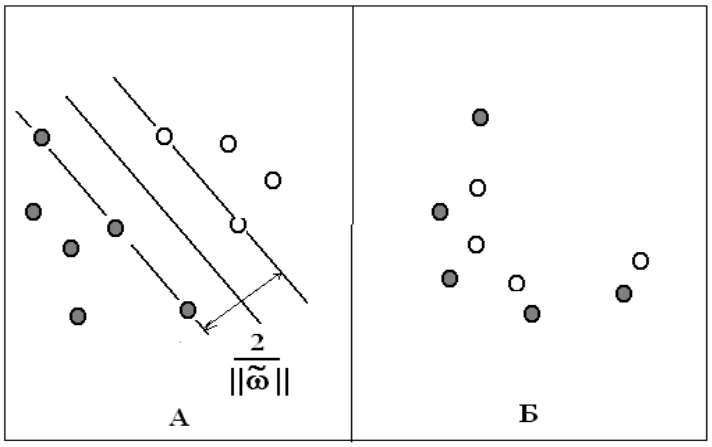

Рис. 3.7. Линейно разделимые (А) и линейно неразделимые (Б) подмножества

Скалярное произведение в пространстве $\mathcal{H}$, имеющее вид $(\tilde{\varphi}(\tilde{x}),\tilde{\varphi}(\tilde{y})) = \mathcal{K}(\tilde{x},\tilde{y})$ называется ядром. В результате перехода в спрямляющее пространство, решающее (нелинейное) правило будет иметь вид

$$h_{\mathcal{H}}(\tilde{x}) = sign\{\sum_{\substack{\text{по номерам } j,\\ \text{принадлежащим}\\ \text{множеству номеров}\\ \text{опорных векторов}}} a_j^0 y_j \mathcal{K}(\tilde{x}_j,\tilde{x}) + b_0\}. \qquad (3.8)$$

Нелинейная суперпозиция (3.8) может быть представлена схемой (рис. 3.8), аналогичной нейронной сети с одним скрытым слоем. Использование подхода на основе метода опорных векторов позволяет определить число k нейронов в скрытом слое и коэффициентов $a_j^0 y_j$, b_0. Но для нелинейной SVM суперпозиции общий поход к выбору ядер не разработан. Более подробные сведения о разработке и применению алгоритмов SVM можно найти в обширной литературе [2,8,13,15,16].

С точки зрения теории машинного обучения, обоснования обучаемости, SVM подход интересен тем, что обеспечивает сжатие информации об обучающей выборке до числа k опорных векторов. Остальные $l-k$ векторов, согласно формуле (3.8), не определяют решающее правило. Таким образом, в случае SVM можно говорить об обучении сжатием. Не удивительно, что известны оценки точности и надежности SVM, в которых число k является определяющим параметром.

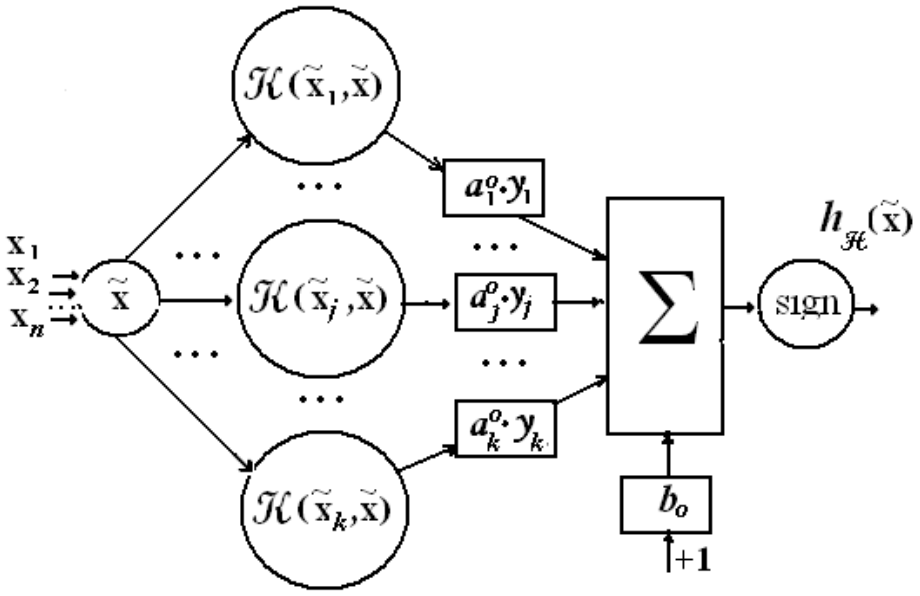

Рис. 3.8. Схематическое представление SVM

Теорема 3.6[16, с. 139]. Пусть обучающая выборка содержит l примеров, разделённых гиперплоскостями с максимальным зазором. Тогда математическое ожидание (по множеству обучающих выборок) вероятности ошибки ограничено математическим ожиданием минимума трёх величин: отношения $\frac{k}{l}$, где k – число опорных векторов; отношения $\frac{r_2 \mid \omega \mid^2}{l}$, где r – радиус сферы, содержащей выборочные данные, $\mid \omega \mid$ – величина зазора; и отношения $\frac{n}{l}$, где n – размерность входного пространства:

$$EP_{error} \le E \min\{ \frac{k}{l}, \frac{r_2 \mid \omega \mid^2}{l}, \frac{n}{l} \}.$$

Литература к главе 3

1. Алексеев Д. В. Приближение функций нескольких переменных нейронными сетями / Д. В. Алексеев // Фундаментальная и прикладная математика. – 2009. – 15(3). – С. 9-21.
2. Воронцов К. В. Лекции по искусственным нейронным сетям // Константин Вячеславович Воронцов. – М.: ВЦ РАН, 2009. – 20 с. www.machinelearning.ru/wiki/images/c/cc/Voron-ML-NeuralNets.pdf
   Лекции по методу опорных векторов. – М.: ВЦ РАН, 2007. – 18 с. http://www.ccas.ru/voron/download/SVM.pdf
3. Галушкин А. И. Теория нейронных сетей // А. И. Галушкин. – М.:ИПРЖР, 2000. – 416 с.
4. Донской В.И. Оценки ёмкости основных классов алгоритмов эмпирического обобщения, полученные pVCD методом / В. И. Донской // Ученые записки Таврического национального университета им. В. И. Вернадского. Серия «Физико-математические науки». – 2010. – Т. 23(62). – №2. – С. 56-65.
5. Донской В.И., Махина Г.А. Обучение нейроподобной структуры, основанной на суперпозиции Колмогорова / В. И. Донской, Г. А. Махина // Искусственный интеллект. – 1999. – № 2. – С.166-170.
6. Колмогоров А. Н. О представлении непрерывных функций нескольких переменных в виде суперпозиций непрерывных функций одного переменного и сложения / Андрей Николаевич Колмогоров // ДАН СССР. – 1957. – Т. 114. – С. 953-956.
7. Коробейник Ю. Ф. Теорема Стоуна-Вейерштрасса / Ю. Ф. Коробейник. – Ростов-на-Дону: Изд-во Ростовского ун-та, 1992. – 144 с.
8. Норкин В.И. Об эффективности методов классификации, основанных на минимизации эмпирического риска / В.И. Норкин, М.А. Кайзер // Кибернетика и системный анализ. – 2009. – Т. 45. – №.5. – С. 93 –105.

9. Першин Д. Обзор некоторых видов нейронных сетей. Препринт // Денис Першин. – Новосибирск: Институт систем информатики РАН, 2000. – 26 с.
10. Хайкин С. Нейронные сети. Полный курс // Саймон Хайкин. – М. Издательский дом «Вильямс», 2006. – 1104 с.
11. Bartlett P. L. For valid generalization, the size of the weights is more important than the size of network / Peter L. Bartlett // In *Advances in Neural Information Processing Systems.* – MIT Press, Cambridge, 1997. – P. 134–140.
12. Bartlett P. L., Maass W. Vapnik-Chervonenkis Dimension of Neural Nets / Peter L. Bartlett, Wolfgang Maass // In The Handbook of Brain Theory and Neural Networks, editor M. A. Arbib. – MIT Press, Cambridge, 2003. – P. 1188–1192.
13. Burges C. J. A Tutorial on Support Vector Machines for Pattern Recognition / C. J. Burges // Data Mining and Knowledge Discovery. – 1998. – 2. – P. 121–167.
14. Sontag E.D. VC dimension of Neural Networks / E. D. Sontag // In Neural Networks and Machine Learning. – Berlin: Springer, 1998. – P. 69–95.
15. Theodoros E., Massimiliano P. / Evgeniou Theodoros, Pontil Massimiliano. Support Vector Machines: Theory and Applications. // Lecture Notes in Computer Science. – 2001. – Vol. 2049. – P. 249 – 25
16. Vapnik V.N. The Nature of Statistical Learning Theory / Vladimir N. Vapnik. – N.Y.: Springer-Verlag, 2000. – 314 p.

4. Колмогоровская сложность в машинном обучении

4.1. Основные понятия колмогоровской сложности

Определение 4.1. [13, с. 221] Колмогоровская сложность слова x при заданном способе описания – вычислимой функции (декомпрессоре) D есть

$$KS_D(x) = \min\{l(p) \mid D(p) = x\},$$

если существует хотя бы одно слово p такое что $D(p) = x$. Иначе полагается, что значение сложности не ограничено $(+\infty)$. Будем говорить, что в этом случае колмогоровская сложность не определена.

Определение 4.2. Условная колмогоровская сложность слова x при заданном слове y есть

$$KS_D(x \mid y) = \min\{l(p) \mid D(p, y) = x\}.$$

Если y – пустое слово, то $KS_D(x \mid y) = KS_D(x)$.

Определение 4.3. Говорят, что декомпрессор D_1 слова x не хуже декомпрессора D_2, если $KS_{D_1}(x) \le KS_{D_2}(x) + O(1)$. Декомпрессор называют оптимальным, если он не хуже любого другого декомпрессора.

Теорема 4.1 (*Соломонова–Колмогорова*) [13]**.** Существуют оптимальные декомпрессоры.

Доказательство. Покажем, что найдется такая частично рекурсивная функция-декомпрессор A, что для любой другой частично рекурсивной функции-декомпрессора $D = D(p, y)$ будет выполнено неравенство

$$KS_A(x \mid y) \le KS_D(x \mid y) + c_D.$$

Здесь c_D – константа, не зависящая от x и y.

Используя универсальную частично рекурсивную функцию U с подходящим номером n, для любого декомпрессора D можно записать равенство

$$D(p, y) = U(n, (p, y)) = x.$$

Колмогоровская сложность относительно декомпрессора D есть

$$KS_D(x \mid y) = l(p).$$

Далее, осуществляя группировку аргументов, можно определить функцию A следующим образом:

$$A((n, p), y) = U(n, (p, y)) = x.$$

Здесь пара слов (n, p) рассматривается как их конкатенация, длина которой есть $l(np) = l(p) + l(n)$. Тогда $A((n, p), y) = D(p, y) = x$ для любого допустимого декомпрессора D. Поэтому для любого номера функции

n, определяющего декомпрессор D, найдется константа $c_D \geq l(n)$, зависящая только от выбора этого декомпрессора, такая, что

$$KS_A(x \mid y) = l(np) \leq l(n) + l(p) + \delta = KS_D(x \mid y) + c_D,$$

где константа δ определяет дополнительное число бит, которое может потребоваться для того, чтобы входящий в конкатенацию np номер используемой универсальной функции n мог быть отделен от аргумента p. Это можно сделать разными способами независимо от слова p, например, при помощи специального так называемого самоограничивающего кодирования. Подробнее это будет показано ниже при разборе определения колмогоровской сложности, данного Витаньи и Ли. □

Далее запись $KS(x)$ будет обозначать колмогоровскую сложность строки x по некоторому оптимальному декомпрессору.

Замечание. Конкатенация xy двух строк x и y не может рассматриваться как пара (x, y), поскольку в конкатенации, вообще говоря, не содержится информации о нужном разделении строки xy на две подстроки. Поэтому конкатенация дополняется информацией, обеспечивающей её правильное разделение.

Определение 4.4. Функция $f(x)$ называется *перечислимой сверху*, если существуют вычислимая функция $F(x,k)$, определенная для всех слов x и всех натуральных чисел k, для которой $F(x,0) \geq F(x,1) \geq F(x,2) \geq \cdots$ и $f(x) = \lim\limits_{k \to \infty} F(x,k)$ для каждого значения x. При любом k значение $F(x,k)$ является верхней оценкой для $f(x)$. Функция $f(x)$ называется *перечислимой снизу*, если существует аналогичная нижняя оценка $L(x,k)$.

Теорема 4.2. Функция KS перечислима сверху, причем $|\{x : KS(x) < n\}| < 2^n$ для всех $n > 0$.

Доказательство. Покажем, что множество пар $\{\langle n, x\rangle : KS(x) < n\}$, где n – натуральное число, а x – двоичное слово, перечислимо. Если $KS(x) < n$, то существует фигурирующая в определении KS вычислимая функция – декомпрессор D. Используя установленный стандартный порядок двоичных слов, можно организовать вычисления, начиная с $k = 0$, в соответствии с этим порядком. Т. е. перебирать слова p по мере роста их длины, соблюдая условие $KS(x) < n$. Будут перебираться все слова, длина которых не превышает n. Как только окажется, что $D(p) = x$, перечисляющий алгоритм будет выдавать пару $\langle l(p) + k, x\rangle$ и увеличивать

k на единицу. Если первая выдача будет парой $\langle l(p)+0, x\rangle$, то выдаваемая перечисляющая последовательность будет иметь вид $\langle l(p)+0, x\rangle, \langle l(p)+1, x\rangle, \langle l(p)+2, x\rangle, \langle l(p)+3, x\rangle \cdots$. Поскольку перебираются все слова длины не больше n, то сумма этих длин $\sum_{i=0}^{n-1} 2^{-i} = 2^n - 1 < 2^n$. Поэтому $|\{x : KS(x) < n\}| < 2^n$.

Определим функцию $F(x,k) = l(p) + n - k$ как последовательность оценок сверху сложности $KS(x)$, *полагая* $F(k,x) = \infty$ *при* $k > n$. Тогда $F(x,0) \geq F(x,1) \geq \cdots$ и $KS(x) = \lim\limits_{k\to\infty} F(x,k)$, поскольку это предельное соотношение соответствует неравенству $k > n$ для любого заданного n. □

Лемма 4.1. Для любой вычислимой функции $f(x)$ имеет место неравенство $KS(f(x)) \leq KS(x) + O(1)$ для всех тех значений x, когда функция $f(x)$ определена.

Доказательство. Пусть D – оптимальный декомпрессор в определении $KS(x) = KS_D(x) = \min l(p) : D(p) = x$. Возьмем в качестве другого декомпрессора композицию вычислимых функций $f \circ D$ и рассмотрим

$$\begin{aligned} KS_{f\circ D}(f(x)) &= (\min l(p) : f(D(p)) = f(x)) \\ &= (\min l(p) : D(p) = x) \\ &= KS(x). \end{aligned}$$

$$KS(f(x)) \leq KS_{f\circ D}(f(x)) + O(1) = KS(x) + O(1).$$

Теорема 4.3. Любая частично рекурсивная (вычислимая) функция $L(x)$ такая, что $L(x) \leq KS(x)$ в тех точках, в которых $L(x)$ определена, ограничена некоторой константой C, то есть $L(x) \leq C$ для всех x.

Доказательство. Предположим, что существует вычислимая функция $L(x)$, являющаяся оценкой снизу колмогоровской сложности: $L(x) \leq KS(x)$. Определим функцию $A(n)$, которая ставит в соответствие натуральному числу n минимальное в порядке перечисления значение x такое, что $L(x) \geq n$. Функция $A(n)$ будет вычислимой в силу предположения, что $L(x)$ вычислима. Тогда $L(A(n)) \leq KS(A(n)$ по сделанному предположению, что $L(x) \leq KS(x)$. Согласно определению функции $A(n)$, имеет место неравенство $L(A(n)) \geq n$. Согласно лемме, $KS(A(n)) \leq KS(n) + c_1$. Получается цепочка неравенств:

$$n \leq L(A(n)) \leq KS(A(n)) \leq KS(n) + c_1 \leq \log n + c_2,$$

где c_1 и c_2 – некоторые константы. Но вытекающее из этой цепочки неравенство $n \le \log n + c_2$ не выполняется для всех n, больших некоторого значения n_0. Полученное противоречие доказывает теорему.□

Теорема 4.4. Колмогоровская сложность KS не является вычислимой функцией.

Доказательство. Предположив, что KS вычислима, получим, что вычислима функция $f(x) = KS(x) - 1$, и тогда $f(x) \le KS(x)$ для всех непустых строк x. Но такой нижней оценки для колмогоровской сложности не существует, согласно теореме.□

Замечание. Невычислимость колмогоровской сложности влечёт невычислимость любой неограниченной функции, являющейся её нижней оценкой; однако подходящие константы такими оценками служить могут.

Теорема 4.5. Колмогоровская сложность конечной строки x $KS_D(x) = \min\{l(p) \mid D(p) = x\}$ определена тогда и только тогда, когда существует машина Тьюринга T_C (компрессор) такая, что $T_C(x) = p$.

Доказательство. Действительно, если существует машина Тьюринга D такая, что $D(p) = x$, то существует система подстановок Маркова M_D, алгоритмически эквивалентная МТ D (реализующая тот же самый алфавитный оператор). Применение M_D к слову p даст $x = M_D(p)$. Зафиксируем выполненную при этом последовательность марковских подстановок:

$$\tilde{S}(M_D, p, x) = \{s_1, ..., s_j, ..., s_\mu : \ s_j = \lambda_j \to \rho_j\},$$

где λ_j – левая часть подстановки (замещаемое подслово), а ρ_j – правая часть подстановки (замещающее подслово), вместе с последовательностью $k_1, ..., k_j, ..., k_\mu$ номеров символов текущего обрабатываемого слова, начиная с которых реализуются подстановки. Тогда компрессор T_C может быть композицией машин Тьюринга двух типов: подвода головки к символу с номером k_j (обозначим эти машины T_j^1) и заменой подслова ρ_j на подслово λ_j (обозначим их T_j^2). Применение к слову x последовательно машин $T_\mu^1, T_\mu^2, ..., T_j^1, T_j^2, ..., T_1^1, T_1^2$ даёт композицию T_C такую, что $T_C(x) = p$ (машина T_1^2 должна быть снабжена заключительным состоянием).

Аналогично доказывается, что если для строки x существует машина Тьюринга T_C (компрессор) такая, что $T_C(x) = p$, где p - некоторая

строка, то можно указать соответствующую ей машину-декомпрессор D_{T_C} такую, что $D_{T_C}(p) = x$, и тогда колмогоровская сложность $KS_D(x) = \min\{l(y) \mid D(p) = x\}$ будет определена.

Определение 4.5. Назовем *точной колмогоровской сложностью* строки x

$$KC(x) = \min_{\{D|D(p)=x\}} \min\{l(p) \mid D(p) = x\}.$$

Точная колмогоровская сложность определяется наилучшим декомпрессором.

Теорема 4.6. Точная колмогоровская сложность KC не является вычислимой функцией.

Доказательство. Если бы KC была вычислима, то она была бы нижней оценкой колмогоровской сложности KS: $KC(x) \le KS(x)$. Но таковой оценки не существует по теореме. □

Определение 4.6. Пусть x – конечная строка, и множество её компрессоров $\mathrm{T}_C(x) = \{T_C \mid T_C(x) = p\}$ не является пустым. Назовем $K_T(x) = \min_{T_C \in \mathrm{T}_C}\{l(p) \mid T_C(x) = p\}$ сжатием строки x *наилучшим компрессором*.

Очевидно, для конечной строки x сжатие удовлетворяет двойному неравенству $0 \le K_T(x) \le l(x)$. Значение 0 соответствует пустой строке.

Теорема 4.7. Если $l(x) < \infty$, то $KC(x) = K_T(x)$ (точная колмогоровская сложность равна сжатию наилучшим компрессором).

Доказательство. Предположим, что $KC(x) < K_T(X)$. Зафиксируем наилучший декомпрессор D^*, соответствующий значению $KC(x) = l(p^*)$ на слове p^*. Зафиксируем это слово – кратчайшее описание p^* строки x. Используя марковское представление декомпрессора D^*, построим, как это было сделано при доказательстве теоремы, алгоритм-компрессор T_{D^*} такой, что $T_{D^*}(x) = p^*$. Но тогда $K_T(x) \le KC(x)$. Точно также, предположив, что $K_T(X) < KC(x)$, используем наилучший компрессор для построения соответствующего декомпрессора, и получим $K_T(x) \ge KC(x)$. □

Следствие 4.1. Функция $K_T(X)$ сжатия наилучшим компрессором не является вычислимой.

Доказательство следует из равенства $KC(x) = K_T(x)$ и невычислимости $KC(x)$. □

Напомним, что произвольное множество строк $\mathfrak{I}$ называется префиксным кодом, если для любых двух строк $S_1 \in \mathfrak{I}$ и $S_2 \in \mathfrak{I}$ таких, что $S_1 \neq S_2$, ни одна из этих строк не является префиксом другой, т.е. не существует непустой строки $W \in \mathfrak{I}$ такой, то $S_2 = S_1 W$.

В работе [47,48] исходная колмогоровская сложность определяется, на первый взгляд, иначе. Используется понятие с*амоограничивающего кода* $\bar{x}$ заданной бинарной строки $x_1 x_2 \ldots x_n$, который определяется соотношением $\bar{x} = x_1 x_1 x_2 x_2 \ldots x_{n-1} x_{n-1} x_n \neg x_n$, определяющим *префиксный код*. В этом коде в каждой, начиная слева, смежной паре символов, кроме последней, символы повторяются. Но в последней паре последний символ $\neg x_n$ строки $\bar{x}$ является инверсией предпоследнего символа x_n.

Способность этого кода определять собственную длину очевидна, что соответствует названию «самоограничивающий». Покажем, что этот код – префиксный. Действительно, пусть x и y – две бинарных строки такие, что x является префиксом строки y, то есть $y = x\tau$ при непустом окончании τ. Обозначим длины этих строк $l(x) = n$ и $l(y) = m > n$. Убедимся, что код $\bar{x}$ не будет префиксом кода $\bar{y}$:

$$x = x_1 x_2 \ldots x_n;$$

$$y = x_1 x_2 \ldots x_n y_{n+1} \ldots y_m;$$

$$\bar{x} = x_1 x_1 x_2 x_2 \ldots x_{n-1} x_{n-1} x_n \neg x_n;$$

$$\bar{y} = x_1 x_1 x_2 x_2 \ldots x_{n-1} x_{n-1} x_n \; x_n y_{n+1} y_{n+1} \ldots y_m \neg y_m.$$

Из приведенных соотношений видно, что код $\bar{x}$ не является префиксом кода $\bar{y}$.

Используя построенный выше префиксный код, определяют *стандартный самоограничивающий код* x' для любой строки x согласно соотношению $x' = \overline{l(x)}x$. Это соотношение определяет, что *к исходной строке приписывается префикс, являющийся самоограничивающим кодом ее длины*, и $l(x') = n + 2]\log n[$, где $n = l(x)$.

Теорема 4.8 (о сложности конкатенации строк). Пусть xy – конкатенация строк x и y. Тогда выполняется неравенство

$$KS(xy) \leq KS(x) + 2\log KS(x) + KS(y) + c, \qquad (4.1)$$

где c – некоторая константа.

Доказательство. Пусть p и q – такие слова, что $KS(x) = l(p)$ и $KS(y) = l(q)$. Пусть D' – произвольный декомпрессор. Предположим, что имеет место равенство $D'(pq) = xy = D(p)D(q)$. Но $D'(pq)$ не может быть определено однозначно, поскольку разные разбиения слова pq на части $p_1q_1 = p_2q_2 = pq$ могут давать различные результаты декомпрессии. Чтобы разделение конкатенации pq было корректным, можно применить самоограничивающий код $\overline{l(p)}pq$, чем обеспечивается выполнение $D'(\overline{l(p)}pq) = D(p)D(q) = xy$. Тогда

$$KS_{D'}(xy) = 2\log l(p) + l(p) + l(q);$$

$$KS_{D'}(xy) = KS(x) + 2\log KS(x) + KS(y).$$

Переходя от декомпрессора D' к оптимальной машине, согласно теореме Соломонова-Колмогорова получаем неравенство (4.1) с константой c, не зависящей от x и y.

Приведем без доказательства еще одну теорему, полезную при использовании математического аппарата колмогоровской сложности.

Теорема 4.9 (Колмогорова-Левина о декомпозиции сложности пары строк) [12].

$$KS(x, y) = K(x) + K(y \mid x) + O(\log K(x, y)).$$

Определение 4.7 [48]**.** Пусть $T_1, T_2, ..., T_i, ...$- стандартное перечисление машин Тьюринга, а $\phi_1, \phi_2, ..., \phi_i, ...$ – перечисление соответствующих этим машинам частично рекурсивных функций. Колмогоровская сложность строки x по заданной строке y определяется выражением

$$C(x \mid y) = \min_{p,i}\{l(i' p) : \phi_i(p, y) = x, \;\; p \in \{0,1\}^*, i \in \mathrm{N}\};$$

$$C(x) = C(x \mid \lambda). \;\square$$

В этом определении Витаньи и Ли сложность слова x определяется длиной конкатенации номера i' машины-декомпрессора D_i, представленного в самоограничивающем коде, и кода p слова x.

Пусть $(p^*, i^*) = \arg\min_{p,i}\{l(i' p) : \phi_i(p, y) = x, \quad p \in \{0,1\}^*, i \in \mathrm{N}\}.$

По слову i', представленному в самоограниченном коде, можно определить описание декомпрессора (машины) i и отделить его от слова p. Затем можно выполнить программу i (промоделировать её) на любом другом допустимом декомпрессоре – машине D. Тогда

$$C\ (x \mid y) \le C_D(x \mid y) + l(i'),$$

Откуда следует, что $C(x \mid y) = KS(x \mid y)$ – колмогоровская сложность относительно некоторого оптимального способа описания D_{i^*}.

4.2. Префиксная сложность

Префиксная сложность является модификацией простой колмогоровской сложности, приспособленной для построения универсальной вероятностной меры на множестве последовательностей. Напомним, что если $\mathfrak{I}$ – некоторое множество строк, в котором любая пара строк удовлетворяет условию: одна из них не является префиксом другой, то множество $\mathfrak{I}$ называют безпрефиксным. Вычислимая функция $U(p, y)$ двух переменных называется *префиксно-корректной* по первому аргументу, если для любого y множество строк p, на которых эта функция определена, является безпрефиксным. Иногда такую функцию называют *самоограниченным декомпрессором*. Определение распространяется на случай пустой сроки y: $U(p, \lambda) = U(p)$. Если $U(p) = x$ для некоторой строки x, то множество $\{p : U(p) = x\}$ является безпрефиксным. И тогда компрессор T_C (см. ниже теорему 4.11) порождает для всех допустимых конечных строк x безпрефиксное множество.

Определение 4.8. Пусть U – произвольная вычислимая префиксно-корректная функция. Условная префиксная колмогоровская сложность строки x при условии y есть

$$KP_U(x \mid y) = \begin{cases} \min\{l(p) \mid U(p, y) = x\}, & \exists p\ U(p, y) = x \\ \infty, & \forall p\ U(p, y) \ne x. \end{cases}$$

Теорема 4.10. Существует такая (универсальная) префиксно-корректная функция $A = A(p, y)$, что для любой вычислимой префиксно-корректной функции $U = U(p, y)$ и для всех x и y имеет место неравенство

$$KP_A(x \mid y) \le KP_U(p, y) + O(1).$$

Доказательство аналогично доказательству теоремы Соломонова – Колмогорова для сложности KS.

Определение 4.9. Условной префиксной сложностью $KP(x \mid y)$ называют условную префиксную сложность $KP_A(x \mid y)$ по любой зафиксированной универсальной префиксно-корректной функции A.

Определение 4.10. Назовем точной условной префиксной сложностью

$$KPC(x \mid y) = \min_{\{U \mid U(p,y)=x\}} \min\{ l(p) \mid U(p,y) = x\},$$

если множество префиксно-корректных функций $\{U : U(p,y) = x\}$ не пусто, иначе будем говорить, что точная префиксная сложность не определена, и полагать, что $KPC(x \mid y) = \infty$. □

Если точная префиксная сложность определена, то для любой универсальной вычислимой префиксно-корректной функции U и для любой универсальной префиксно-корректной функции A

$$KPC(x \mid y) \le KP_A(x \mid y) \le KP_U(p,y) + O(1),$$
$$KPC(x \mid y) \le KP(x \mid y).$$

Поэтому точную префиксную сложность $KPC(x \mid y)$ можно считать условной префиксной сложностью $KP(x \mid y)$ (по некоторой наилучшей универсальной вычислимой префиксно-корректной функции U^*). Это позволяет освободиться от латентной константы.

В определении префиксной сложности можно использовать в качестве функции U так называемую префиксную машину Тьюринга. Это приводит к эквивалентному понятию и оказывается полезным для дальнейшего изложения.

Префиксной называют машину Тьюринга T, описываемую, например, следующим образом [15]. Предполагается, что у такой машины помимо рабочей ленты есть входная лента, на которой имеется *односторонняя читающая головка*. Крайняя левая клетка ленты содержит специальный маркер, справа от которого может быть записана любая последовательность нулей и единиц. Изначально читающая головка находится у левого края входной ленты под специальным маркером. Шаги вычислений машины Тьюринга определяются как символом, который «видит» читающая головка, так и символом, который «видит» головка на рабочей ленте. В зависимости от этих символов и текущего состояния машина предпринимает то или иное действие. Это действие состоит в изменении внутреннего состояния, записи нового символа на рабочей ленте, а также может включать в себя сдвиг и влево, и вправо на рабочей ленте и *сдвиг только вправо читающей головки входной ленты*. Результат работы машины обычным образом записывается на рабочей ленте, которая изначально является пустой. Когда машина останавливается, читающая головка входной ленты находится в точности над первым пробелом, следующим за заданным на входной ленте словом.

Теорема 4.11. Областью определения префиксной машины является безпрефиксное множество.

Доказательство. Пусть S – множество строк, для которых результат работы префиксной машины T определен. Если $x \in S$, то машина T останавливается при условии, что выполнены все необходимые вычисления, на рабочую ленту выдано результирующее слово $z = T(x)$ и на входной ленте прочитаны в точности все символы строки x, но не более. Последнее условие соответствует нахождению входной головки на символе, следующем за последним символом строки x.

Рассмотрим две строки: $x \in S$ и $y \in S$. Предположим, что x является префиксом строки y, то есть $y = x\tau$ при непустом окончании τ. Но тогда, начав работу над словом y, машина T сначала произведёт в точности такие же действия, как при работе над словом x, и затем она остановится, не продолжая просмотр окончания τ слова y. Но тогда результат работы машины на слове y не может быть определен. Это противоречие доказывает, что область определения префиксной машины T – безпрефиксное множество. □

В литературе встречаются другие, эквивалентные определения префиксной машины. В работе [30] префиксная машина Тьюринга T определяется так. Эта машина снабжена тремя лентами: однонаправленной входной лентой (только для чтения), однонаправленной выходной лентой (только для записи) и двунаправленной рабочей лентой. Вдоль однонаправленных лент головка перемещается только слева направо. Все ленты – двоичные, пустой символ не используется. Рабочая лента инициализируется нулями. Машина T останавливается на входе p, выдавая $z = T(p)$, если p находится слева от входной головки, и z находится слева от выходной головки. Множество таких слов p образуют префиксный код. Такие коды называют самоограничивающими программами.

Префиксная машина всегда предполагает существование способа, позволяющего указать, где именно на ленте ограничивается входное слово.

Теорема 4.12. Для любой префиксной МТ можно указать эквивалентную ей обычную МТ.

Доказательство. Пусть T – произвольная префиксная машина, заданная своей таблицей команд, а x – произвольная входная строка. Рассмотрим подпрограмму-функцию $Input(x, k)$, возвращающую k-й символ входной строки x. Подпрограмма реализуется подтаблицей с конечным множеством дополнительных состояний. Чтобы получить обычную машину Тьюринга T_1, эквивалентную префиксной машине T, достаточно реализовать указанную подпрограмму внутри последовательности вычис-

лений однолентoчной машины. Машина T_1 начинает работу, положив $k = 0$, и пропускает (пройдя до конца вправо) входное слово. Эти действия имитируют подготовку входной ленты префиксной машины. Далее она выполняет шаги, логически эквивалентные последовательности вычислений машины T, вне зоны записи любого входного слова. Аналогом обращения к выделенной входной ленте префиксной машины T будет обращение к подпрограмме $Input(x,k)$. При таком обращении будет происходить следующее:

вычисление $k := k+1$;

запоминание при помощи специального маркера ячейки ленты, на которой прерываются вычисления;

переход в начальное состояние подтаблицы-подпрограммы; считывание символа $x[k]$,

подвод к ячейке ленты, соответствующей точке возврата;

возврат в следующее по логике обработки машины T состояние.

Замечание. МТ, суммирующая любую входную конечную двоичную последовательность x, применима к любому её префиксу. Но такой сумматор не реализуем на префиксной МТ. Поэтому

Следствие 4.2. Префиксные МТ образуют специфический собственный подкласс машин Тьюринга.

Следствие 4.3. Любая префиксно-корректная вычислимая функция вычислима на МТ без маркера конца входа.

В справедливости последнего следствия можно убедиться иным способом [5].

Для префиксной сложности KP справедлива такая же теорема о несуществовании нетривиальной вычислимой оценки снизу, как и для колмогоровской сложности KS. Из этой теоремы следует, что префиксная сложность не является вычислимой. Её доказательство [14], такое же, как и доказательство аналогичной теоремы для колмогоровской сложности KS.

Лемма 4.2. $KPC(x,y) \le KPC(x) + KPC(y)$.

Доказательство. Пусть слово x восстанавливается по кратчайшему слову p наилучшей машиной T_1, соответствующей точной префиксной сложности $KPC(x)$, а слово y восстанавливается по кратчайшему слову q наилучшей машиной T_2, соответствующей точной префиксной сложности $KPC(y)$. По следствию ? обе эти машины могут не использовать маркер конца входа. Тогда $T_1 \circ T_2\,(pq) = xy$, где $T_1 \circ T_2$ – композиция машин Тьюринга. Сначала машина T_1 применяется к слову p и выдаёт x.

После её работы головка машины T_2 будет обозревать первый символ слова q. Следовательно,

$$KP_{T_1 \circ T_2}(x, y) = KPC(x) + KPC(y) = |p| + |q|.$$

Тогда для любой наилучшей машины $KPC(x, y) \le KP_{T_1 \circ T_2}(x, y)$.

Теорема 4.13. Любая частично рекурсивная (вычислимая) функция $L(x)$ такая, что $L(x) \le KP(x)$ в тех точках, в которых $L(x)$ определена, ограничена некоторой константой C, то есть $L(x) \le C$ для всех x.

Теорема 4.14. Префиксная сложность не является вычислимой.

Теорема 4.15. Обычная и префиксная сложности связаны неравенством $\forall x \; KS(x) \le KP(x) + O(1)$, причем разность $(KP(x) - KS(x))$ стремится к бесконечности с ростом длины строки x [14].

Теорема 4.16 [14]. Существует всюду определённая вычислимая функция f, оценивающая сверху KS и на бесконечном множестве равная KS.

Теорема 4.17 [14]. Существует всюду определённая вычислимая функция f, оценивающая сверху KP и на бесконечном множестве равная KP.

4.3. Универсальное распределение

« – Я теперь считаю так: меры нет.
Вместо меры наши мысли,
заключенные в предмет»
Даниил Хармс

Определение 4.11. Вещественнозначная функция $f : \mathrm{N} \to \mathrm{R}$ называется *перечислимой*, если существует МТ, вычисляющая рекурсивную функцию φ такую, что $\varphi(< x,t >) =< p,q >$, где $\frac{p}{q}$ есть t-е рациональное приближение значение $f(x)$. В этом смысле функцию f, допускающую указанную аппроксимацию, называют рекурсивной.

Определение 4.12. Будем называть функцию $P : \mathrm{N} \to [0,1]$ *вероятностным распределением*, если $\sum_{x \in \mathrm{N}} P(x) \le 1$. Неравенство (вместо равенства) вводится для удобства, и полагается, что недостающая вероятность $\varepsilon = 1 - \sum_{x \in \mathrm{N}} P(x)$ сосредоточена на неопределенном элементе $u \notin \mathrm{N}$. В этом случае P называют *полумерой.*

Определение 4.13. Рассмотрим семейство полумер (вероятностных распределений) P_ε на N (эквивалентно – на $\{0,1\}^*$). Назовем перечислимую снизу полумеру $m \in \mathrm{P}$ *максимальной*, если для любой другой перечислимой снизу полумеры μ для некоторой константы c и для всех x выполнено неравенство $\mu(x) \le cm(x)$. □

Можно сказать, что максимальная полумера m «выделяет» так много вероятности каждому объекту, как любое другое распределение семейства P_ε с точностью до мультипликативного множителя. В этом смысле она является *универсальной относительно априорной неопределенности.* В некоторых случаях использование меры m в пространстве $\{0,1\}^\infty$ приводит к тем же результатам, которые даёт использование истинного неизвестного априорного распределения.

Теорема 4.18. P_ε содержит элемент m, который мультипликативно доминирует все элементы из P_ε. То есть, для любой полумеры $P \in \mathrm{P}_\varepsilon$ существует константа c такая, что $cm(x) > P(x)$ для всех $x \in \mathrm{N}$.

Доказательство этой теоремы можно найти в работах [5, 15]. □

Назовем в указанном смысле максимальную перечислимую снизу полумеру m *универсальным распределением*.

Теорема 4.19.

$$-\log m(x) = KP(x) + O(1).$$

Доказательство. Сначала докажем неравенство $-\log m(x) \le KP(x) + O(1)$. Перепишем неравенство в эквивалентной форме $2^{-KP(x)} \le cm(x)$, где $c \ne 0$ – некоторая константа. В силу максимальности полумеры $m(x)$ достаточно показать, что функция $2^{-KP(x)}$ является a) перечислимой снизу b) полумерой.

Убедимся в справедливости b). Неравенство $\sum_x 2^{-KP(x)} \le 1$ для полумеры действительно выполняется, так как префиксная сложность $KP(x) = l(x)$ – минимальная длина слова – определена для совокупности слов x, образующий префиксный код. А для префиксного кода справедливо неравенство Крафта $\sum_x 2^{-l(x)} \le 1$.

Убедимся в справедливости a). Известно, что функция префиксной сложности $KP(x)$ перечислима сверху: существует вычислимая функция F такая, что $KP(x) < F(x,k)$ для любого натурального k. Тогда $2^{-KP(x)} > L(x,k) = 2^{-F(x,k)}$, следовательно, $2^{-KP(x)}$ перечислима снизу.

Теперь докажем обратное неравенство: $-\log m(x) \geq KP(x) + O(1)$. Как уже было показано, функция $2^{-KP(x)} = \mu(x) > 0$ является полумерой; $m(x) > 0$, поскольку $cm(x) \geq \mu(x) > 0$. Обозначим

$$\eta = \sup_x |\, m(x) - \mu(x) \,| < 1, \quad \delta = \inf_x\, m(x) > 0.$$

Тогда $\mu(x) \geq m(x) - \eta \geq cm(x)$, для любой константы c такой, что $c \leq 1 - \eta / m(x)$. В качестве c можно взять $1 - \eta / \delta$, получая $\mu(x) \geq (1 - \eta / \sigma) m(x)$ или $2^{-KP(x)} \geq (1 - \eta / \delta) m(x)$, и тогда $-KP(x) \geq \log m(x) + O(1)$ или $-\log m(x) \geq KP(x) + O(1)$. □

Следствие 4.4.

$$-\log m(x) = KPC(x) + O(1).$$

4.4. Принцип «Бритвы Оккама» (*Occam's Razor*) и обучаемость

Определение 4.14[18]**.** *Алгоритмом Оккама* с параметрами $\alpha \geq 1$ и $\beta : 0 \leq \beta < 1$ над классом (целевых) гипотез G, в котором сложность любой гипотезы не превышает n, называется алгоритм обучения, который:

(i) выполняется за полиномиальное время от длины выборки l и
(ii) в результате обучения выдаёт гипотезу, имеющую сложность, не превышающую $n^{\alpha} l^{\beta}$.

В этом определении не оговаривается, является ли полученная гипотеза согласованной с обучающей выборкой; кроме этого, выбранная гипотеза может даже не принадлежать классу G. □

Теорема 4.20 [18]**.** Для алгоритма Оккама над классом (целевых) гипотез G, в котором сложность любой гипотезы не превышает n, независимо от распределения вероятностей на признаковом пространстве (ε, δ), обучаемость имеет место при длине выборки l, оцениваемой как

$$l = O(\frac{1}{\varepsilon} \ln \frac{1}{\delta} + (n^{\alpha} / \varepsilon)^{1/(1-\beta)}),$$

где $\alpha \geq 1$ и $\beta : 0 \leq \beta < 1$.

В случае согласованности алгоритма Оккама с обучающей выборкой $\beta = 0$, и тогда

$$l = O(\frac{1}{\varepsilon}(n^{\alpha} + \ln \frac{1}{\delta})).$$

Теорема 4.21 (*Occam's Razor Theorem* **).** Пусть G и H – классы концептов. Пусть $g \in G$ – целевой концепт**;** $n(g)$ – длина его бинарного

представления $s(g)$. Пусть A – алгоритм обучения, и даны константы $\alpha \geq 1$ и $\beta : 0 \leq \beta < 1$. Предположим, что алгоритм A, используя выборку X_l длины l, извлеченную из признакового пространства в соответствии с вероятностным распределением на нём, выдаёт гипотезу $h \in H$. Пусть эта гипотеза согласована как минимум с $(1-\varepsilon/2)l$ примерами из X_l, и её строчное бинарное описание $s(h)$ имеет длину не большую, чем $n(g)^{\alpha} l^{\beta}$. Тогда, если

$$l = O(\max(\frac{1}{\varepsilon}\log\frac{1}{\delta}, (\frac{n(g)^{\alpha}}{\varepsilon})^{\frac{1}{1-\beta}}))$$

или $\quad l = O(\frac{n(g)^{\alpha}}{\varepsilon})$ при $\beta = 0$,

то *полиномиальная обучаемость имеет место*. □

Оценка длины выборки, которая требуется для *PAC* обучаемости в сложностной версии *Occam's Razor* теоремы, основанной на длине описания $s(h) \leq n(g)^{\alpha} l^{\beta}$ выбираемого при обучении концепта h, может быть уточнена[32]:

$$l = \max(\frac{2}{\varepsilon}\ln\frac{1}{\delta}, (\frac{(2\ln 2)n(g)^{\alpha}}{\varepsilon})^{\frac{1}{1-\beta}}).$$

Константы α и β, фигурирующие в *Occam's Razor* теореме, можно интерпретировать следующим образом. Бинарное описание выбранной гипотезы должно иметь длину, не превышающую $n(g)^{\alpha} l^{\beta}$, где α *– степень сжатия описания целевого концепта*, а β *– степень сжатия описания выборки*.

Попытки уточнения *Occam's Razor* теоремы привели к следующей формуле для длины выборки, необходимой для (ε, δ)– обучаемости, и определяемой сжатием описания выбираемого при обучении концепта h [32]:

$$l = \max(\frac{2}{\varepsilon}\ln\frac{2}{\delta}, (\frac{(2\ln 2)p(n,s,\delta/2)}{\varepsilon})^{\frac{1}{1-\beta}}),$$

где $p(n,s,\delta/2)$– характеризующая сжатие описания концепта h оценочная функция такая, что $KP(h) < p(n,s,\delta/2)l^{\beta}$; n– размерность признакового пространства, s – верхняя граница возможных длин описаний по допустимым классам концептов. Если можно указать оценку сверху M_h

такую, что $p(n,s,\delta/2)l^{\beta} \le M_h$ для всех допустимых значений параметров функции $p(\cdot)$, то требуемая длина выборки будет определяться как

$$l = \max(\frac{2}{\varepsilon}\ln\frac{2}{\delta},(\frac{(2\ln 2)M_h}{\varepsilon})^{\frac{1}{1-\beta}}),$$

и при полном сжатии выборки при $\beta = 0$ как

$$l = \max(\frac{2}{\varepsilon}\ln\frac{2}{\delta},(\frac{(2\ln 2)M_h}{\varepsilon})).$$

Оценка M_h может быть получена $pVCD$ методом [7,8].

Версия *Occam's Razor* теоремы, основанной на вапниковской ёмкости $VCD(H)$ семейства концептов H, из которого извлекается концепт h, определяет следующую оценку выборки, требуемую для PAC– обучаемости [18, 26,32]:

$$\max(\frac{VCD(H)-1}{32\varepsilon},\frac{1}{\varepsilon}\ln\frac{1}{\delta}) < l(H,\delta,\varepsilon) \le \frac{4}{\varepsilon}(VCD(H)\log\frac{12}{\varepsilon}+\log\frac{2}{\delta}).$$

Из приведенных оценок видно, что колмогоровская сложность $KP(h)$ выбранной гипотезы $h \in H$ и $VCD(H)$ при их использовании для оценивания результатов машинного обучения дают близкие результаты. Действительно, выбор семейства гипотез наименьшей ёмкости влечёт минимизацию колмогоровской сложности этого семейства, что следует из установленного в [7, 25] неравенства для конечных семейств гипотез – соответствующих классов рекурсивных функций:

$$VCD(H) < K_l(H) \le VCD(H)\log l.$$

В случае конечного семейства гипотез H оценка длины выборки, обеспечивающей обучаемость для любого согласованного с выборкой концепта $h \in H$, имеет вид:

$$l(H,\delta,\varepsilon) \ge \frac{1}{\varepsilon}\ln\frac{|H|}{\delta}.$$

Это неравенство, как и многие другие фундаментальные результаты, связанные с обучаемостью, были получены В. Н. Вапником еще в начале 1970-х годов.

4.5. Обучение и сжатие

Связь между сжатием обучающей выборки, обучаемостью и VCD была изучена в работе Флойда и Вармута [27] не основе следующих понятий. Для любого $Y \subset \mathrm{X}$ (X – признаковое пространство) и произвольного

класса концептов C вводится обозначение $C|Y = \{c \cap Y : c \in C\}$ – ограничение концепта по области (множеству) Y.

Схема сжатия выборки размера не более k для класса концептов C описывается *функцией сжатия, функцией реконструкции* и их применением следующим образом. Используя конечную обучающую выборку, согласованную с классом концептов C, функция сжатия K отбирает из неё так называемое *множество сжатия* A, *состоящее из не более* k *помеченных обучающих примеров*. Функция реконструкции φ использует это множество сжатия для построения концепта-гипотезы $c_A = \varphi(A)$ – результата обучения. При этом гипотеза c_A, вообще говоря, может не содержаться в классе C, но должна быть согласованной со всеми примерами исходной обучающей выборки.

Пример. Рассмотрим класс C_{L^o} однородных линейных концептов в R^n и согласованную выборку D длины $l > n$, состоящую из точек $\tilde{x} = x_1,...,x_n$, удовлетворяющих уравнению $a_1x_1 + ... + a_nx_n = 0$. Неизвестные коэффициенты $\tilde{a} = a_1,...,a_n$ определяют один из концептов $c_{\tilde{a}} \in C_L$. Пусть множество сжатия A_L состоит из любых $k = n$ попарно различных примеров обучающей выборки. Тогда, используя эти k примеров, функция реконструкции φ, определяемая алгоритмом решения системы однородных линейных уравнений, однозначно восстанавливает $\varphi(A_L) = c_{\tilde{a}}$. Заметим, что $VCD(C_{L^o}) = n$. Если $l < n$, то функции реконструкции, обеспечивающая безошибочное нахождение неизвестного целевого концепта, для этого примера не существует, так как по $l < n$ точкам невозможно однозначное восстановление линейного концепта.

Для класса неоднородных линейных концептов C_L, соответствующих уравнениям $a_1x_1 + ... + a_nx_n = a_0$, параметр сжатия k должен быть не меньше $d = VCD(C_L) = n+1$. □

Класс концептов называется *максимальным*, если добавление любого концепта к этому классу увеличивает его VCD. Класс концептов C, имеющий $VCD(C) = d$, называется *классом-максимумом*, если для каждого конечного подмножества $Y \subseteq C$, при условии $|Y| = m > d$ семейство $C|Y$ содержит $\Phi_d(|Y|) = \sum_{i=0}^{d} C_{|Y|}^i$ концептов.

Теорема 4.22 [27]**.** Пусть класс концептов $C \subseteq 2^X$ является классом-максимумом, $VCD(C) = d$, $|X_l| = l \geq d$. Тогда для любого концепта

$c \in C$ найдётся множество сжатия $A \subseteq X_l$, состоящее ровно из d примеров, и функция реконструкции такие, что $c_A = c$.

Теорема 4.23. Пусть класс концептов $C \subseteq 2^X$ является классом-максимумом, $VCD(C) = d$, и выборочное пространство может быть бесконечным. Тогда для класса концептов C при длине обучающей выборки l существует схема сжатия размера k, удовлетворяющего неравенству $d < k \le d \log l$.

Теорема 4.24. Пусть $C \subseteq 2^X$ класс концептов со схемой сжатия размером не более $d = VCD(C)$. Тогда для любых ε, δ таких, что $0 < \varepsilon, \delta < 1$, использование обучающего алгоритма, соответствующего этой схеме компрессии, обеспечит (ε, δ) обучаемость при длине выборки, удовлетворяющей неравенству

$$l \ge \frac{1}{1-\beta}\left(\frac{1}{\varepsilon}\ln\frac{1}{\delta} + VCD(C) + \frac{VCD(C)}{\varepsilon}\ln\frac{1}{\beta\varepsilon}\right)$$

для любого $\beta : 0 < \beta < 1$. □

Нужно подчеркнуть, что сжатие в последних теоремах характеризуется длиной выборки, а не длиной бинарной строки. Но, тем не менее, в указанных условиях возможно сжатие информации о семействе концептов ёмкости d до бинарной строки, длина которой не будет превышать $O(d \log l)$.

Теоретически, колмогоровская сложность произвольного класса вычислимых функций может быть равной его ёмкости d, в силу чего, с учетом перечислимости колмогоровской сложности сверху, возможно сжатие информации о таком классе до строки длины d.

В работе [36] схема компрессии размера k уточняется следующим образом. Функция сжатия K ставит в соответствие каждой обучающей выборке X_l длины l единственную её подвыборку $V = V(X_l)$ длины k, называемую *ядром сжатия*. Функция K в схеме k – сжатия полагается зафиксированной. Функция реконструкции $\varphi = \varphi(V, \tilde{x})$ тоже зафиксирована и ставит в соответствие паре ядро-точка значение 1 или 0. Таким образом определяется решающее правило и некоторый концепт $c_V = c_\varphi(K, \varphi, \tilde{x})$. Этот концепт c_V, вообще говоря, может не принадлежать классу концептов C. Но для любого целевого концепта семейства C и для любой заданной выборки длины l функция реконструкции согласована со всеми точками этой выборки.

Ядерным размером называется минимальная мощность ядра сжатия по всем возможным схемам сжатия (варьируются функции сжатия, реконструкции и выборки длины l).

Если зафиксировать любую схему компрессии с ядерным размером k и использовать определяемую ею функцию реконструкции φ^*, то в соответствии с данными выше определениями, применение этой функции к произвольным точкам признакового пространства, вообще говоря, может давать ошибки. Нужно убедиться, что использование функции φ^* обеспечивает обучаемость.

Характеризация сжатия ядерным размером позволяет считать произвольным признаковое пространство, поскольку речь идёт о числе примеров в ядре, а не о битовой строке, кодирующей сложность.

Будем полагать, что концепты класса C и функция реконструкции измеримы по Борелю. Из этого следует измеримость множеств, определённых ниже при доказательстве теоремы, и правомочность использования теоремы Фубини.

Теорема 4.25 [36]. Для любой схемы компрессии с ядерным размером k при длине выборки $l > k$, ошибка Err функции реконструкции как решающего правила, определяющего принадлежность произвольной точки x целевому концепту G, может быть оценена неравенством

$$P(Err > \varepsilon) < C_l^k (1-\varepsilon)^{l-k}.$$

Доказательство. Пусть X^l – множество любых выборок длины l; $(\tilde{x}_1,...,\tilde{x}_l) = X_l \in \mathrm{X}^l$ – произвольная выборка длины l; P^l – вероятностная мера на множестве выборок длины l (по этой мере оценивается вероятность события $Err > \varepsilon$). Будем обозначать A^* – ядро сжатия произвольной схемы компрессии с ядерным размером k; $\varphi^*(A^*, \tilde{x})$ – результат применения функции реконструкции, определяющий, возможно с ошибкой, принадлежность точки $\tilde{x}$ концепту c; $c(\tilde{x})$ – истинное значение этой принадлежности. Обозначим

$$E = \{X_l \in \mathrm{X}^l : \Pr(\tilde{x} \in X_l \wedge \varphi^*(A^*, \tilde{x}) \neq c(\tilde{x})) > \varepsilon\}$$

множество всевозможных выборок длины l, точки которых классифицируются функцией φ^* с вероятностью ошибки, превышающей ε. Эквивалентное определение –

$$E = \{X_l \in \mathrm{X}^l : \Pr(\tilde{x} \in X_l \wedge \varphi^*(A^*, \tilde{x}) = c(\tilde{x})) < 1-\varepsilon\}.$$

Пусть T – множество всех C_l^k подпоследовательностей номеров любых k точек выборки; $\tilde{t} = (t_1,...,t_k) \in T$. Набор $\tilde{t}$ определяет подпоследовательность выборки $\tilde{x}_{t_1},...,\tilde{x}_{t_k}$. Введём следующие обозначения.

$A_{\tilde{t}}$ – множество всех выборок длины l, для которых по каждой выборке $\tilde{x}_1,..,\tilde{x}_l$ функция сжатия K выделяет ядро, состоящее из $\tilde{x}_{t_1},...,\tilde{x}_{t_k}$ этой выборки. Очевидно, $\bigcup\limits_{\tilde{t}\in T} A_{\tilde{t}} = X_l$.

$E_{\tilde{t}} \subseteq A_{\tilde{t}}$ – такое подмножество выборок, на котором применение функции реконструкции φ^* даёт правильное решение с вероятностью, меньшей $1-\varepsilon$. То есть, $E_{\tilde{t}}$ – это все выборки, для которых функция сжатия K выделяет ядро, состоящее из точек этих выборок с номерами $t_1,...,t_k$, а функция реконструкции даёт правильное решение с вероятностью, меньшей $1-\varepsilon$.

По определению соответствующих подмножеств, $E_{\tilde{t}} = E \cap A_{\tilde{t}}$, откуда с учётом равенства $\bigcup\limits_{\tilde{t}\in T} A_{\tilde{t}} = X_l$ следует $E = \bigcup\limits_{\tilde{t}\in T} E_{\tilde{t}}$.

Обозначим далее:

$U_{\tilde{t}}$ – множество всех выборок длины l, для которых вероятность правильной классификации при помощи функции реконструкции φ^* с выделяемой функцией компрессии K ядром $\{x_{t_1},...,x_{t_k}\}$ ограничена величиной $1-\varepsilon$. Тогда $E_{\tilde{t}} = U_{\tilde{t}} \cap A_{\tilde{t}}$.

$B_{\tilde{t}}$ – множество всех выборок длины l таких, что входящие в них точки с номерами вне множества $\{t_1,...,t_k\}$ правильно классифицируются функцией реконструкции.

Если выборка принадлежит множеству $A_{\tilde{t}}$, то функция сжатия K выделяет из выборок этого множества ядро, состоящее из $x_{t_1},...,x_{t_k}$ этой выборки. По определению схемы сжатия, все остальные точки этой выборки с номерами вне множества $\{t_1,...,t_k\}$ должны правильно классифицироваться. Поэтому $A_{\tilde{t}} \subset B_{\tilde{t}}$. Вместе с равенством $E_{\tilde{t}} = U_{\tilde{t}} \cap A_{\tilde{t}}$ это даёт

$$P^l(E_{\tilde{t}}) = P^l(A_{\tilde{t}} \cap U_{\tilde{t}}) \le P^l(B_{\tilde{t}} \cap U_{\tilde{t}}).$$

Пусть $\pi_{\tilde{t}}$ такая перестановка координат точек выборки $\tilde{x}_1,...,\tilde{x}_i,...,\tilde{x}_l$, что $t_i \mapsto i, i=1,...,k$; $\pi_{\tilde{t}} : X_l \to X_l$. Тогда $\pi_{\tilde{t}}(U_{\tilde{t}})$ – множество всех выборок длины l, для которых вероятность правильной классификации входящих в них точек при помощи функции реконструкции φ^* с ядром $\{x_1,...,x_k\}$ ограничена величиной $1-\varepsilon$. Перестановка вводится для удобства дальнейших рассуждений: без потери общности применяется замена (переименование) $\{x_{t_1},...,x_{t_k}\} \mapsto \{x_1,...,x_k\}$.

$$P^l(E_{\tilde{t}}) \le P^l(B_{\tilde{t}} \cap U_{\tilde{t}}) = P^l(\pi_{\tilde{t}}(B_{\tilde{t}}) \cap \pi_{\tilde{t}}(U_{\tilde{t}})).$$

$$P^l(\pi_{\tilde{t}}(B_{\tilde{t}}) \cap \pi_{\tilde{t}}(U_{\tilde{t}})) = \int_{\pi_{\tilde{t}}(U_{\tilde{t}})} I(\pi_{\tilde{t}}(B_{\tilde{t}}))dP^l,$$

где $I(\pi_{\tilde{t}}(B_{\tilde{t}}))$ – характеристическая функция множества $\pi_{\tilde{t}}(B_{\tilde{t}})$, которая выделяет из всех выборок длины l такие выборки, что входящие в них точки с номерами вне множества $\{t_1,...,t_k\}$ правильно классифицируются функцией реконструкции, т. е. правильно классифицируются $l-k$ точек.

Интегрирование производится по множеству $\pi_{\tilde{t}}(U_{\tilde{t}})$ выборок таких, что вероятность правильной классификации входящих в них точек при помощи функции реконструкции φ^* с ядром $\{x_1,...,x_k\}$ ограничена величиной $1-\varepsilon$.

Ядра компрессии извлекаются из выборок, поэтому существует некоторое множество $V_{\tilde{t}}$ ядер размера k такое, что $\pi_{\tilde{t}}(U_{\tilde{t}}) = V_{\tilde{t}} \times X^{l-k}$.

По теореме Фубини

$$\int_{\pi_{\tilde{t}}(U_{\tilde{t}})} I(\pi_{\tilde{t}}(B_{\tilde{t}}))dP^l = \int_{V_{\tilde{t}}} dP^k \int_{X^{l-k}} I(\pi_{\tilde{t}}(B_{\tilde{t}}))dP^{l-k}.$$

Обозначим $W_{x_1,..,x_k}$ – множество точек выборки X_l, правильно классифицируемое функцией реконструкции φ^* с ядром $x_1,...,x_k$. Тогда

$$(x_1,...,x_k) \times X^{l-k} \cap \pi_{\tilde{t}}(B_{\tilde{t}}) = (x_1,...,x_k) \times W^{l-k}_{x_1,..,x_k}.$$

$$P^l(E_{\tilde{t}}) < \int_{X^{l-k}} I(\pi_{\tilde{t}}(B_{\tilde{t}}))dP^{l-k} = \int_{W^{l-k}_{x_1,..,x_k}} dP^{l-k} < (1-\varepsilon)^{l-k}.$$

$$P^l(E_{\tilde{t}}) < (1-\varepsilon)^{l-k}.$$

Число различных подпоследовательностей длины k последовательности $x_1,...,x_l$ равно C_l^k. Поэтому

$$P^l(E) = P^l(\underset{\tilde{t}\in T}{\cup} E_{\tilde{t}}) < \sum_{\tilde{t}\in T} P^l(E_{\tilde{t}}) = |T| \cdot (1-\varepsilon)^{l-k} = C_l^k (1-\varepsilon)^{l-k}.$$

Теорема 4.26 [36]. Для любой схемы компрессии, имеющей ядерный размер k, (ε, δ)-обучаемость имеет место при длине выборки l, определяемой неравенством

$$l \geq \max\{\frac{2}{\varepsilon}\ln\frac{1}{\delta},\ \frac{2k}{\varepsilon}\ln\frac{4k}{\varepsilon} + 2k\}.$$

Доказательство. Преобразуем неравенство

$$C_l^k (1-\varepsilon)^{l-k} < \delta$$

(см. предыдущую теорему) в эквивалентное неравенство

$$l \geq \frac{\ln\frac{1}{\delta} + \ln C_l^k}{-\ln(1-\varepsilon)} + k,$$

которое выполняется при условии

$$l \geq \frac{1}{\varepsilon}(\ln\frac{1}{\delta} + k\ln l) + k = \frac{1}{\varepsilon}\ln\frac{1}{\delta} + k(\frac{1}{\varepsilon}\ln l + 1),$$

поскольку $l^k > C_l^k$, и для малых ε выполняется: $-\ln(1-\varepsilon) > \varepsilon$. В оценку входят два слагаемых. Поэтому неравенство будет иметь место, если одновременно каждое слагаемое будет не больше величины $\frac{l}{2}$, что приводит к системе из двух неравенств:

$$\begin{cases} \frac{l}{2} \geq \frac{1}{\varepsilon}\ln\frac{1}{\delta}, \\ \frac{l}{2} \geq k(\frac{1}{\varepsilon}\ln l + 1). \end{cases}$$

Второе из этих двух неравенств путём подстановки в правую часть оценки для l можно преобразовать следующим образом:

$$l \geq 2k(\frac{1}{\varepsilon}\ln(2k(\frac{1}{\varepsilon}\ln l + 1)) + 1);\ \ l \geq 2k(\frac{1}{\varepsilon}\ln\frac{4k}{\varepsilon} + 1);$$

$$l \geq 2k\frac{1}{\varepsilon}\ln\frac{4k}{\varepsilon} + 2k.$$

Полученная система неравенств

$$\begin{cases} l \geq \dfrac{2}{\varepsilon}\ln\dfrac{1}{\delta}, \\ l \geq \dfrac{2k}{\varepsilon}\ln\dfrac{4k}{\varepsilon} + 2k \end{cases}$$

дают оценку

$$l \geq \max\left(\frac{2}{\varepsilon}\ln\frac{1}{\delta}, \frac{2k}{\varepsilon}\ln\frac{4k}{\varepsilon} + 2k\right). \square$$

Сравнивая эту оценку длины выборки, требуемой для обучаемости с параметром размера сжатия k, с аналогичной оценкой обучаемости Блумера и Литтлстоуна [19], которая была получена на основе размерности Вапника-Червоненкиса $d = VCD(\mathfrak{F})$ класса функций $\mathfrak{F}$, используемого для обучения, –

$$l \geq \max\left(\frac{4}{\varepsilon}\ln\frac{2}{\delta}, \frac{8\cdot d}{\varepsilon}\ln\frac{8d}{\varepsilon}\right),$$

можно заметить, что эти оценки достаточно близки в случае $k \approx d$.

Литтлстоном и Вармутом [36] также получены аналогичные результаты для схемы сжатия размера k с дополнительной информацией, обозначаемой Q – некоторым множеством, добавляемым отображением сжатия к ядру выборки. Это отображение ставит в соответствие любой выборке пару: множество Q и ядро размера k. Так что сжатие оценивается числом элементов в Q и размером ядра k

Теорема 4.27 [36]. Для любой схемы компрессии с ядерным размером k и дополнительной информацией Q при длине выборки $l > k$, ошибка Err функции реконструкции как решающего правила, определяющего принадлежность произвольной точки x целевому концепту C, может быть оценена неравенством

$$P(Err > \varepsilon) < |Q| C_l^k (1-\varepsilon)^{l-k}. \square$$

Если схему компрессии ослабить так, что классификация выборки, по которой найдено ядро, при помощи функции реконструкции, допускает ошибку в $s < l - k$ её точках, то будет иметь место следующий результат:

Теорема 4.28 [36]. Для любой схемы компрессии с ядерным размером k, допускающей не более s ошибок при длине выборки $l > k$, ошибка Err функции реконструкции как решающего правила, определяющего принадлежность произвольной точки x целевому концепту C, может быть оценена неравенством

$$P(Err > \varepsilon) < C_{k+s}^{k} C_{l}^{k} (1-\varepsilon)^{l-k}.$$

4.6. Использование универсального распределения для аппроксимации неизвестного распределения

Определение 4.15 [34, с.6]. Распределение P называется *простым*, если существует перечислимое распределение Q такое, что $\exists c: \ \forall x(cQ(x) \geq P(x))$, где $c \leq 2^{KP(Q)+O(1)}$ – константа. Говорят, что P доминируется перечислимым распределением Q.

Теорема 4.29 [33, с. 361, 34, с.9] Полиномиальная обучаемость над универсальным распределением m имеет место тогда и только тогда, когда имеет место полиномиальная обучаемость над любым простым распределением P, при условии что выборка извлекается в соответствии с распределением m.

Доказательство. Пусть P – любое простое распределение: найдется константа $c_P > 0$ такая, что $c_P m(x) \geq P(x)$.

Предположим, что имеет место обучаемость над распределением m с ошибкой ε / c_p и имеется соответствующий определению полиномиальной обучаемости алгоритм A полиномиальной сложности. Зафиксируем его. Пусть Err – множество объектов, на которых обученный концепт даёт ошибку. Тогда с вероятностью не меньшей $1-\delta$

$$\sum_{x \in Err} m(x) \leq \varepsilon / c_P \text{ и } \sum_{x \in Err} P(x) \leq c_P \sum_{x \in Err} m(x) \leq \varepsilon.$$

Поскольку алгоритм A извлекает обучающую выборку *всегда в соответствии с распределением* m, то его точное выполнение в условиях распределения P должно давать в качестве результата тот же самый концепт, определяющий множество Err. Следовательно, из полиномиальной обучаемости над универсальным распределение m следует полиномиальная обучаемость над любым простым распределением P.

Пусть теперь имеет место полиномиальная обучаемость над любым распределением P, вероятность ошибки не больше ε, и обученный алгоритм даёт ошибку только на множестве Err. Но по условию теоремы, извлечение выборки происходит в соответствии с распределением m, поэтому $\sum_{x \in Err} m(x) \leq \varepsilon$, что доказывает полиномиальную обучаемость над m. □

Замечание, касающееся теоремы. Параметр обучаемости ε / c_P требует знания константы $c_P = KP(P)$ – префиксной сложности неизвестного простого распределения P. При решении задач обучения прихо-

дится иметь дело с некоторыми подмножествами признакового пространства, и для таких подмножеств D использовать условные распределения $m(\cdot \mid D)$. В связи с этим Ли и Витаньи получили более тонкий критерий обучаемости, который будет приведен ниже без доказательства.

Определение 4.16 [34, с. 5]. Вероятностное распределение $P : S \to \mathrm{R}$, где $S = \mathrm{N} \cup u$, $\sum_{x \in \mathrm{N}} P(x) \le 1$, $\sum_{x \in S} P(x) = 1$, u – некоторый неопределенный элемент, называется *перечислимым*, если множество точек $\{(x, y) : x \in \mathrm{N}, y \in Q, P(x) > y\}$ рекурсивно перечислимо.

Теорема 4.30 [33]. Если распределение $P(x \mid y)$ перечислимо, то для всех допустимых x, y имеет место неравенство

$$2^{KP(P)} m(x \mid y) \ge P(x \mid y).$$

Теорема 4.31 [34, с.9]. Пусть H – класс концептов, $D \subseteq \mathrm{N}$ – выборочное пространство, $\mu = \min\{ l(s(h)) : h \in H \}$ – минимальная длина описания концепта по классу H и c – некоторая константа. Класс H полиномиально обучаем над универсальным распределением m тогда и только тогда, когда он полиномиально обучаем над любым простым условным распределением $P(\cdot \mid D)$ таким, что существует перечислимое распределение Q, доминирующее P, которое удовлетворяет условию $KP(Q) \le c \log \mu + O(1)$, и кроме этого, выполняется одно из следующих условий:

(i) выборка извлекается согласно условному распределению $m(\cdot \mid D)$;

(ii) $KP(D) \le c \log \mu + O(1)$ и выборка формируется так, что полиномиальное число примеров извлекаются в соответствии с безусловным распределением $m(\cdot)$, причем степень полинома зависит от константы c.

4.7. Байесовский подход к обучению и *MDL*

Правило Байеса определяет наиболее вероятную гипотезу h при заданном обучающем множестве D согласно соотношению

$$\Pr\{h \mid D\} = \frac{\Pr\{D \mid h\} \Pr(h)}{\Pr\{D\}},$$

которое может быть представлено в эквивалентной форме:

$$-\log \Pr\{h \mid D\} = -\log \Pr\{D \mid h\} - \log \Pr\{h\} + \log \Pr\{D\}.$$

Наиболее вероятная гипотеза h при заданном обучающем множестве D должна максимизировать $\log \Pr\{h \mid D\}$ или, равносильно, минимизировать $-\log \Pr\{h \mid D\}$. Поскольку $\log \Pr\{D\}$ не меняется при выборе гипотез,

байесовское правило выбора гипотезы из семейства H может быть представлено в виде:

$$h^* = \arg\min_{h \in H}(-\log \Pr\{D \mid h\} - \log \Pr\{h\}).$$

Использование универсального распределения приводит к соотношению

$$\hat{h}^* = \arg\min_{h \in H}(-\log m\{D \mid h\} - \log m\{h\})$$

и далее, с учетом соотношения $-\log m(x) = KP(x) + O(1)$, к правилу

$$\hat{\hat{h}}^* = \arg\min_{h \in H}(KP(D \mid h) + KP(h)).$$

Последнее соотношение является выражением ***принципа MDL*** (*Minimum Description Length*), который является одной из формализаций «бритвы Оккама»: *наилучшая гипотеза для данного набора данных та, которая минимизирует сумму длины описания кода гипотезы* (*также называемой моделью*) *и длины описания множества данных относительно этой гипотезы* [47].

Основанная на строгом математическом обосновании, применении колмогоровской сложности и универсальной меры m, уточнённая версия *MDL* называется ***идеальным MDL.*** Применение и обоснование идеального *MDL* иллюстрируется на байесовской схеме выбора гипотезы [47]..

Имеет место *фундаментальное неравенство*:

$$KP(D|h) + KP(h) - \alpha(P,h) \le -\log \Pr\{D \mid h\} - \log \Pr(h) \le KP(D|h) + KP(h),$$

где $\alpha(\Pr, h) = KP(\Pr\{\cdot \mid h\}) + KP(\Pr(h))$. При малом значении $\alpha(P,h)$ левая и правая оценки становятся приблизительно равными друг к другу, и тогда $KP(D|h) + KP(h) \approx -\log \Pr(D \mid h) - \log P(h)$. Это рассуждение лежит в основе доказательства следующего утверждения.

Теорема 4.32. Байесовское правило и идеальный *MDL* при извлечении решения из допустимого класса гипотез H выбирают одну и ту же гипотезу: $h^* = \hat{\hat{h}}^*$ при условии, что величина $\alpha(P,h)$ является достаточно малой. □

Таким образом, минимизация суммы $KP(D \mid h) + KP(h)$ обеспечивает выбор гипотезы $\hat{\hat{h}}^*$ в соответствии с правилом Байеса, которое, как известно, является оптимальным: обеспечивает минимум среднего риска.

Казалось бы, если правило Байеса является оптимальным, то его и нужно применять, не изобретая новых способов выбора решений. Но непосредственное использование байесовского правила требует знания априорного распределения вероятностей, а оно, как правило, неизвестно: в задачах машинного обучения в качестве начальной информации представляется обучающая выборка, по которой приходится аппроксимировать неиз-

вестное распределение. Идеальный MDL позволяет *обойтись без информации об истинном априорном распределении*. Но возникают другие трудности: и колмогоровская сложность $KP(x)$, и универсальное распределение $m(x) = 2^{-KP(x)-O(1)}$ не являются вычислимыми. Поэтому нужно рассчитывать на *использование вычислимых оценок колмогоровской сложности*.

Рассмотрим условную сложность $KP(D \mid h)$, входящую в минимизируемую сумму $KP(D \mid h) + KP(h)$. По определению префиксной колмогоровской сложности, $KP(D \mid h) = \min\{ l(p) : U(p,h) = D\}$ для некоторого оптимального декомпрессора U. Здесь декомпрессор U – префиксная машина Тьюринга, которая принимает вход в виде пары строк (p – сжатого описания и h – применяемой гипотезы) и в результате выдает обучающую информацию в виде строки D. Если $KP(D \mid h) = 0$, то $U(\lambda, h) = D$, где λ – пустое слово. В таком случае будем говорить, что *гипотеза* h *полностью описывает данные* D. Действительно, декомпрессор U точно восстанавливает данные D, используя при этом в качестве входа только описание гипотезы h. В противном случае будем использовать запись $KP(D \mid h) = K(D \setminus \hat{D})$, где $\hat{D} = \hat{D}(h)$ часть обучающих данных, которые правильно описываются гипотезой h. Обозначим $D \setminus \hat{D} = \overline{D}(h)$ – выделенную подпоследовательность последовательности-строки D и будем говорить, что $\overline{D}(h)$ – остаток данных, не описанных гипотезой h. Тогда принцип MDL принимает эквивалентный вид

$$\hat{\hat{h}}^* = \arg\min_{h \in H}(KP\{\overline{D}(h)\} + KP\{h\})$$

и формулируется так: *наилучшая гипотеза для данного набора данных та, которая минимизирует сумму длины описания кода гипотезы* (*также называемой моделью*) *и длины описания множества данных, не описываемых* (*не объясняемых*) *этой гипотезой*.

Для согласованных с данными D гипотез это правило будет выглядеть так:

$$\hat{\hat{h}}^* = \arg\min_{h \in H_c(D)}(KP\{h\}),$$

где $H_c(D)$– класс гипотез, согласованных с данными D.

Лемма 4.3. Пусть в процессе обучения выбрана гипотеза h, не согласованная ровно с d примерами обучающей выборки, $0 < d < l/2$, но согласованная со всеми остальными примерами. Тогда

$$KP(D \mid h) \geq d \log l .$$

Доказательство. $KP(D \mid h) = KP(\overline{D}(h))$ – сложность «необъяснённой» или, что равносильно, неверно классифицированной правилом h части таблицы. Поэтому для получения информации о значении класса одной точки, которая не классифицируется правилом h, следует реализовать одно обращение к имеющейся таблице данных D, имеющей l «входов». Сложность такого обращения не меньше $\log l$. □

4.8. Вапниковская интерпретация принципа *MDL*

Обучающее множество как совокупность пар $(\tilde{x}_1, \alpha_l),...,(\tilde{x}_l, \alpha_l)$, содержит две строки: строку $\tilde{x}_1,...\tilde{x}_i,...\tilde{x}_l$, описывающую l точек признакового пространства X, и бинарную строку $\tilde{\alpha} = \alpha_1,...\alpha_i,...\alpha_l$ классификации этих точек неизвестной функцией $g : \mathrm{X} \to \{0,1\}$. Значение $\alpha_i = g(x_i)$ зависит только от точки x_i, и не зависит от точек x_j, $j \neq i$, поскольку предполагается, что все пары извлекаются в обучающее множество случайно и независимо.

Рассмотрим следующую модель [46]. Пусть имеется набор способов кодирования C_b, содержащий $N << 2^l$ различных таблиц кодирования $T_s, s = 1,..., N$. Каждая таблица реализует некоторое отображение, согласно которому любой строке $x_1,...x_i,...x_l$ ставится в соответствие некоторая бинарная строка $\tilde{\beta} = \beta_1,...\beta_i,...\beta_l$. По таблице $T \in C_b$ можно вычислить $\beta = T(x)$ только для одной точки признакового пространства $\mathbf{X}$. В этом случае будем говорить о $T : \mathrm{X} \to \{0,1\}$ как о *решающем правиле*.

Будем отыскивать в C_b таблицу T, которая ставит в соответствие строке $\tilde{x}_1,...\tilde{x}_i,...\tilde{x}_l$ такую бинарную строку $\tilde{\beta}^* = \beta_1^*,...\beta_i^*,...\beta_l^*$, что $\rho(\tilde{\alpha}, \tilde{\beta}^*) = \min_{T \in C_b} \rho(\tilde{\alpha}, \tilde{\beta})$, где $\rho(.,.)$ – расстояние Хэмминга между булевыми векторами.

Таблица T_0, если таковая существует, будет обозначать такую таблицу, что $\rho(\tilde{\alpha}, \tilde{\beta}^*) = 0$. Будем говорить, что эта *совершенная таблица* декодирует сроку $\tilde{\alpha}$. Таблица T_0 может быть однозначно определена своим номером во множестве C_b, для описания которого потребуется $]\log N[<< l$ бит. Тогда используя набор способов кодирования C_b, со-

держащий совершенную таблицу, можно сжать длину l исходного описания строки $\tilde{\alpha}$ с коэффициентом $K(T_0) = \dfrac{]\log N[}{l}$. Согласно данной интерпретации, $]\log N[$ бит являются мерой сложности совершенной таблицы.

Будем называть $K(T)$ коэффициентом сжатия строки $\tilde{\alpha}$. В общем случае набор способов кодирования C_b может не содержать совершенной таблицы, и тогда $\min\limits_{T \in C_b} \rho(\tilde{\alpha}, \tilde{\beta}) = d > 0$. Не теряя общности, можно считать, что $d \le l/2$

При фиксированном значении $d = d(T)$ существует C_l^d различных исправлений кода $\tilde{\beta} = \tilde{\beta}(T)$, отличающегося по некоторым d разрядам от кода $\tilde{\alpha}$. Иначе говоря, существует C_l^d доопределений кода $\tilde{\beta}$ до нужного кода $\tilde{\alpha}$. Чтобы выделить один из таких способов, указав тем самым нужное доопределение, требуется $]\log C_l^d[$ бит. Таким образом, для описания строки $\tilde{\alpha}$ потребуется: $]\log N[$ бит для определения номера таблицы, $]\log C_l^d[$ бит для описания доопределения, а также $]\log d[+\Delta_d$ бит для числа коррекций d, где $\Delta_d < 2\log\log d$ при $d > 2$. Из этого подсчета следует, что

$$K(T) = \frac{]\log N[+]\log C_l^d[+]\log d[+\Delta_d}{l}.$$

Слагаемое $]\log N[$ оценивает сложность таблицы (гипотезы) T, а слагаемые $]\log C_l^d[+]\log d[+\Delta_d$ оценивают сложность обучающей выборки (данных) при условии использования этой таблицы (гипотезы).

Чем меньше коэффициент сжатия $K(T)$, тем лучше декодирующая таблица T аппроксимирует неизвестное функциональное отношение между $\tilde{x}$ и $\tilde{\alpha}$, представленное обучающей выборкой.

Теорема 4.33. При заданном семействе C_b и любой выбранной таблице кодирования $T \in C_b$, обеспечивающей сжатие с коэффициентом $K(T)$, с вероятностью не меньшей $1-\eta$, $0 < \eta < 1$, можно утверждать, что при использовании T как решающего правила будет выполняться неравенство

$$R(t) < 2(K(T) - \frac{\ln \eta}{l}),$$

где $R(T)$ – вероятность ошибки решающего правила T (риск ошибки, оценивающий несовпадение решающего правила T с неизвестной, заданной обучающей выборкой функцией).

Доказательство. Для случая обучения, когда решающее правило извлекается из конечного семейства, содержащего N функций (в нашем случае это конечный класс таблиц C_b, $T_i \in C_b$), с вероятностью не меньшей $1-\eta$, одновременно для всех функций семейства выполняется неравенство

$$R(T_i) \le R_{emp}(T_i) + \frac{\ln N - \ln \eta}{l}\left(1 + \sqrt{1 + \frac{2R_{emp}(T_i)l}{\ln N - \ln \eta}}\right),$$

где эмпирический риск равен коэффициенту компрессии $R_{emp}(T_i) = \frac{d}{l}$.

При условии $d < l/2$ и $l > 6$ из этого неравенства получается

$$\begin{aligned}
&\frac{d}{l} + \frac{\ln N - \ln \eta}{l}\left(1 + \sqrt{1 + \frac{2d}{\ln N - \ln \eta}}\right) \\
&< \frac{d}{l} + \frac{\ln N - \ln \eta}{l} + \frac{\ln N - \ln \eta}{l}\sqrt{1 + \frac{2d}{\ln N - \ln \eta}} \\
&< \frac{d}{l} + \frac{\ln N - \ln \eta}{l} + \frac{\ln N - \ln \eta}{l}\left(1 + \frac{d}{\ln N - \ln \eta}\right) \\
&< \frac{2}{l}(d + \ln N - \ln \eta) \\
&< \frac{2}{l}(]\log N[+]\log C_l^d[+]\log d[+\Delta_d - \ln \eta) = 2(K(T) - \frac{\ln \eta}{l}) .
\end{aligned}$$

Заметим, что $-\ln \eta = \ln \frac{1}{\eta} >> 0$ при малых η; для нетривиальности оценки требуется выполнение условия $\frac{l}{2} > \ln \frac{1}{\eta}$ и условия $d < l/2$. □

4.9. Индуктивное обучение как синтез наилучшего компрессора

В процессе обучения происходит как можно большее сжатие описания начальной информации путём выбора соответствующего компрессора.

Начальное описание данных D длины l_0 соответствует широкому классу решений $\mathfrak{I}_0$. Этот класс состоит из таких компрессоров T – префиксных машин Тьюринга, которые обеспечивают сжатие $p = T(D),\ l(p) \le l_0$. Назовём пару компрессоров T_1 и T_2 из $\mathfrak{I}_0$ эквивалентными, если $T_1(D) = T_2(D)$. Обозначим классы эквивалентности $\mathrm{T}_{p,s} = \{T : T(D) = p \ \wedge \ l(p) = s\}$. Пусть $KC(D)$ – точная колмогоровская сложность выборки D. Поскольку s может принимать любые значения из множества $\{ KC(D), KC(D)+1, ..., l_0\}$ и $\left|\{p : l(p) = s\}\right| = 2^s$, число таких классов эквивалентности равно

$$\sum_{s=KC(D)}^{l_0} 2^s = 2^{l_0+1} - 2^{KC(D)}.$$

Процесс обучения может быть реализован посредством сжатия исходных выборочных данных. Тогда выбор кратчайшего (в идеальном случае) или близкого к кратчайшему компрессора с длиной описания μ фиксирует не только одно описание-структуру, но и определяет класс $\mathfrak{I}_\mu$ компрессоров, выстраивающих такое же по структуре описание длины μ. Мощность этого класса не превосходит 2^μ. Таким образом, обучение «сжатием» приводит к сужению используемого семейства $\mathfrak{I}_\mu$, из которого выбирается решение.

В последние десятилетия интенсивно развиваются подходы к обоснованию и оцениванию методов эмпирического обобщения на основе понятия алгоритмической сложности. Прежде всего, имеется в виду колмогоровский поход и предложенный на его основе метод MDL. Предположение, что более „простые” решающие правила чаще дают правильные решения, чем „сложные”, оправдалась на практике и многие годы воспринималась как „гипотеза простой структурной закономерности”.

Цель исследований в указанном направлении, связанном со сжатием и поиском как можно более коротких описаний решающих правил – понять природу сложности и получить на основе её изучения методы нахождения оценок качества алгоритмов обучения (эмпирического обобщения). Несмотря на некоторое продвижение в теории, такие оценки до сих пор не получены для многих классов алгоритмов. Это связано, прежде всего, с математическими трудностями вывода логико-комбинаторных оценок и отсутствием общего приёма их получения.

Ниже представлен именно общий приём к оцениванию – так называемый $pVCD$ метод, – который удалось разработать, ограничив все рассматриваемые семейства моделей эмпирического обобщения до классов,

реализуемых на компьютерах, и шире, – рассматривая их частично-рекурсивные представления. В рамках алгоритмического подхода введено понятие колмогоровской сложности классов алгоритмов распознавания свойств или извлечения закономерностей. На основе этого понятия предложен метод оценивания неслучайности извлечения эмпирических закономерностей.

4.10. Оценивание сложности семейств алгоритмов эмпирического обобщения на основе колмогоровского подхода

Далее будем полагать, что координаты точек обучающих выборок – аргументов рассматриваемых частично рекурсивных функций (алгоритмов) – принимают значения либо из расширенного натурального ряда, $x_i \in \{0,1,2,...\}$, либо из его ограниченного отрезка $x_i \in \{0,1,...,2^M - 1\}$, когда это будет специально оговариваться. Тогда натуральное число M можно считать заданной разрядностью применяемого для решения рассматриваемых задач компьютера. Как и ранее, $X_l = (\tilde{x}_j, \alpha_j)_{j=1}^{l}$ обозначает обучающую выборку длины l. Отдельно обозначим $\hat{X}_l = \{\tilde{x}_j\}_{j=1}^{l}$ набор из l точек $\tilde{x} = (x_1,...,x_i,...,x_n)$, входящих в обучающую выборку, без соответствующих значений $\{\alpha_j\}_{j=1}^{l}$ неизвестной классифицирующей функции.

Определение 4.17. Пусть U – такая частично рекурсивная функция, что для каждого алгоритма $\mathfrak{a}$ из заданного семейства алгоритмов $\mathfrak{A}$ и для любой обучающей выборки X_l найдётся двоичное слово p, которое обеспечивает выполнение равенства $U(p, \hat{X}_l) = \tilde{y}$, где $\tilde{y} = \mathfrak{a}(\tilde{x}_1),...,\mathfrak{a}(\tilde{x}_l)$ – двоичное слово (строка) длины l, содержащая результаты применения алгоритма $\mathfrak{a}$ к *точкам набора* $\hat{X}_l$. Каждый алгоритм $\mathfrak{a} \in \mathfrak{A}$ полагается определенным на каждой выборке $X_l \in \mathrm{X}^l$. Функция U с указанными свойствами существует в силу существования универсальной функции двух аргументов для любого семейства частично рекурсивных функций одного аргумента.

1º Сложность алгоритма $\mathfrak{a}$ относительно выборки X_l по частично рекурсивной функции U есть $K_U(\mathfrak{a} \mid X_l) = \min\ len(p):\ U(p, \hat{X}_l) = \tilde{y}$.

2º Сложность алгоритма $\mathfrak{a}$ на множестве X^l по частично рекурсивной функции U есть $K_{U,\mathrm{X}^l}(\mathfrak{a}) = \max\limits_{X_l \in \mathrm{X}^l} K_U(\mathfrak{a} \mid \hat{X}_l)$.

3º Сложность семейства алгоритмов $\mathfrak{A}$ на множестве X^l по частично рекурсивной функции U есть $K_{U,\mathrm{X}^l}(\mathfrak{A}) = \max\limits_{\alpha\in\mathfrak{A}} K_{U,\mathrm{X}^l}(\alpha)$.

4º Сложность семейства алгоритмов $\mathfrak{A}$ на множестве X^l есть

$$K_l(\mathfrak{A}) = \min_{U\in P_{p.r.}} K_{U,\mathrm{X}^l}(\mathfrak{A}).\square$$

Приведенное определение легко поясняется следующим образом. Сложность $K_l(\mathfrak{A})$ семейства алгоритмов $\mathfrak{A}$ на множестве всех возможных выборок X^l длины l – это наименьшая длина двоичного слова (программы) p, обеспечивающего вычисление по ней самого сложного (и поэтому – любого) алгоритма $\alpha\in\mathfrak{A}$. Важно, что это слово p обрабатывается одной и той же функцией (программой) U^*, причём, согласно пункту 4º данного выше определения, – наилучшей в следующем смысле. *Программа U^* обеспечивает наибольшее сжатие информации о семействе $\mathfrak{A}$ в слово p длины $K_l(\mathfrak{A})$*. Никакие дополнительные требования на программу U^* не накладываются. Поэтому можно получить мажоранту сложности для $K_l(\mathfrak{A})$, если точно указать структуру обеспечивающего восстановление алгоритмов семейства $\mathfrak{A}$ слова p', подлежащего расшифровке, и его длину в битах, а также предоставить алгоритм обработки этого слова U', который будет использоваться вместо программы U^* для оценивания сложности сверху.

Если снять ограничение $x_i \in \{0,1,\ldots,2^M-1\}$ и полагать, что значения переменных x_i могут быть любыми из расширенного натурального ряда N, то рассматриваемые семейства $\mathfrak{A}$ можно полагать бесконечными. Бесконечные семейства функций, тем не менее, могут иметь конечную ёмкость $VCD(\mathfrak{A}) = h_{\mathfrak{A}}$ (что и требуется для гарантированной обучаемости согласно теории Вапника-Червоненкиса). Но при этом функция роста семейства $\mathfrak{A}$ будет расти с ростом l, оставаясь полиномиальной. Колмогоровская сложность $K_l(\mathfrak{A})$ бесконечного семейства $\mathfrak{A}$, вообще говоря, тоже может расти с ростом длины l обучающей последовательности.

Теорема 4.34. Пусть не обязательно конечная система общерекурсивных функций $\mathfrak{A}$ вида $\alpha : \mathbf{X}^n \to \{0,1\}$ имеет ограниченную емкость $VCD(\mathfrak{A}) = h_{\mathfrak{A}}$ и колмогоровскую сложность $K_l(\mathfrak{A})$. Тогда при конечных значениях $h_{\mathfrak{A}} \geq 2$ и $l > h_{\mathfrak{A}}$ имеет место двойное неравенство:

$$h_{\mathfrak{A}} \leq K_l(\mathfrak{A}) < h_{\mathfrak{A}} \log l .$$

Доказательство. Для семейства функций $\mathfrak{A}$ сложность $K_l(\mathfrak{A})$ определена выше с использованием соотношения $U(p, X_l) = \tilde{y}$, в котором булев вектор $\tilde{y}$ длины l принимает значения, соответствующие различным вариантам разбиения всевозможных наборов $\hat{X}_l$ из X^l на два подмножества. Обозначим $\tilde{y} = \mathfrak{A}(\hat{X}_l)$ – результат применения алгоритма $\mathfrak{A}$ к набору точек $\hat{X}_l$ ровно l раз. *Все возможные варианты разбиений набора* $\hat{X}_l$ определяются функциями семейства $\mathfrak{A}$: $\tilde{y} = \alpha(X_l)$, $\alpha \in \mathfrak{A}$. При этом одинаковые разбиения порождают подклассы эквивалентных в этом смысле на выборке $\hat{X}_l$ элементов α из семейства $\mathfrak{A}$. Выберем из каждого такого класса эквивалентности по одной функции (алгоритму). Согласно определению функции роста, будет выделено $m^{\mathfrak{A}}(l)$ функций, где $m^{\mathfrak{A}}(l)$ – функция роста системы $\mathfrak{A}$, определяющая наибольшее число разбиений (наибольшее возможное число различных векторов $\tilde{y}$) по всем выборкам из X^l. Обозначим выбранные функции $\alpha_0,...,\alpha_i,...,\alpha_{m^{\mathfrak{A}}(l)-1}$. Для того, чтобы равенство $U(p, \hat{X}_l) = \alpha(\hat{X}_l)$ при зафиксированной частично рекурсивной функции U выполнялось для всех $\alpha \in \mathfrak{A}$ и на каждом наборе $\hat{X}_l$, аргумент p, определяющий номера функций $\alpha_0,...,\alpha_i,...,\alpha_{m^{\mathfrak{A}}(l)-1}$, должен принимать при зафиксированном l не менее $m^{\mathfrak{A}}(l)$ значений. Поэтому, с учетом того, что U является функцией, должно выполняться неравенство $l(p) \geq]\log m^{\mathfrak{A}}(l)[$, т.е. $K_l(\mathfrak{A}) \geq]\log m^{\mathfrak{A}}(l)[$.

Покажем теперь, что $\min\limits_{U \in P_{p.r.}} K_{U,\mathrm{X}^l}(\mathfrak{A}) =]\log m^{\mathfrak{A}}(l)[$. Для этого, с учетом уже доказанного неравенства $K_l(\mathfrak{A}) \geq]\log m^{\mathfrak{A}}(l)[$, достаточно указать такую функцию $U^* \in P_{p.r.}$, что $K_{U^*,\mathrm{X}^l}(\mathfrak{A}) =]\log m^{\mathfrak{A}}(l)[$. Построение такой функции U^* можно пояснить таблицей 4.1, имеющей в общем случае неограниченное вправо число столбцов. Каждая строка таблицы с номером i, $0 \leq i \leq m^{\mathfrak{A}}(l) - 1$, соответствует алгоритму α_i из выбранного выше множества $\{\alpha_0,...,\alpha_i,...,\alpha_{m^{\mathfrak{A}}(l)-1}\}$ и числовому значению i кода программы p для этого алгоритма. Значения $\tilde{y}_{i,j}$, $j = 0,1,2,...$, содержащиеся в

таблице, являются результатами применения алгоритмов $\mathfrak{a}_i$ к наборам $\hat{X}_l^{(j)}$, *являются двоичными кодами длины* l и отождествляются с соответствующими числами расширенного натурального ряда. Также числами интерпретируются выборки $\hat{X}_l$ и коды p, $p = 0,1,\ldots, m^{\mathfrak{A}}(l) - 1$. Для набора $\{\mathfrak{a}_0,\ldots,\mathfrak{a}_i,\ldots,\mathfrak{a}_{m^{\mathfrak{A}}(l)-1}\}$ из $m^{\mathfrak{A}}(l)$ общерекурсивных функций найдется универсальная функция U^* двух аргументов, обеспечивающая выполнение равенства $U^*(p, \hat{X}_l) = \mathfrak{a}_{i=i(p)}(\hat{X}_l)$ для каждого из $m^{\mathfrak{A}}(l)$ различных значений слова p длины $]\, m^{\mathfrak{A}}(l)[$. Поэтому $\min\limits_{U \in P_{p.r.}} K_{U,X^l}(\mathfrak{A}) =]\log m^{\mathfrak{A}}(l)[$ достигается для этой функции U^*.

Таблица 4.1. Пояснение к определению функции U^*

Код (*номер программы*) p	*Код* (*номер*) *набора* $\hat{X}_l$			
	$\hat{X}_l^{(0)}$	…	$\hat{X}_l^{(j)}$	…
0	…	…	…	…
…	…	…	…	…
i	$\tilde{y}_{i,0}$		$\tilde{y}_{i,j}$	
…	…	…	…	…
$m^{\mathfrak{A}}(l) - 1$	…	…	…	…

Для класса событий ограниченной емкости $h_{\mathfrak{A}} \geq 2$ при $l > h_{\mathfrak{A}}$ справедливы соотношения:

$$2^{h_{\mathfrak{A}}} \leq m^{\mathfrak{A}}(l) < 1{,}5\frac{l^{h_{\mathfrak{A}}}}{h_{\mathfrak{A}}!} < l^{h_{\mathfrak{A}}} = 2^{h_{\mathfrak{A}} \log l},$$

$$h_{\mathfrak{A}} \leq \log m^{\mathfrak{A}}(l) < h_{\mathfrak{A}} \log l.$$

С учетом равенства $\min\limits_{U \in P_{p.r.}} K_{U,X^l}(\mathfrak{A}) =]\log m^{\mathfrak{A}}(l)[$, получаем

$$h_{\mathfrak{A}} \leq K_l(\mathfrak{A}) < h_{\mathfrak{A}} \log l.$$

Следствие 4.5. Колмогоровская сложность семейства алгоритмов $\mathfrak{A}$ равна наименьшему целому, большему или равному логарифму функции роста этого семейства: $K_l(\mathfrak{A}) =]\ \log m^{\mathfrak{A}}(l)[$.

Следствие 4.6. $0 \leq K_l(\mathfrak{A}) \leq l$.

Доказательство. 1) Укажем семейство $\mathfrak{A}$, для которого $K_l(\mathfrak{A}) = 0$. Последнее соотношение имеет место, если для получения равенства $U(p, X_l) = \tilde{y}$ наличие слова p вообще не требуется: оно может быть пустым. Например, рассмотрим семейство $\mathfrak{A}$, в котором каждый алгоритм $\mathfrak{a} = \mathfrak{a}(\tilde{x})$ выдает значение суммы по модулю два всех символов входной бинарной строки $\tilde{x}$. Тогда каждая выборка будет классифицироваться единственным способом, поэтому $m^{\mathfrak{A}}(l) = 1,\ \log m^{\mathfrak{A}}(l) = 0$ и $K_l(\mathfrak{A}) = 0$. Заметим, что алгоритмы такого простого семейства $\mathfrak{A}$, будучи эквивалентными, могут быть различными по их построению. Например, прямое суммирование по модулю; вычисление числа единиц в строке и последующая проверка его четности по младшему двоичному разряду; последовательное инвертирование при прохождении единиц слова $\tilde{x}$.

2) Поскольку $m^{\mathfrak{A}}(l) \le 2^l$, то $K_l(\mathfrak{A}) =]\ \log m^{\mathfrak{A}}(l)[\le l$. □

Следствие 4.7. Если $\lim\limits_{l\to\infty} \dfrac{K_l(\mathfrak{A})}{l} = 0$, то имеет место равномерная сходимость частот ошибок к их вероятностям по всему классу $\mathfrak{A}$.

Доказательство. Напомним [2], что $\Delta^{\mathfrak{A}}(\tilde{x}_1,...,\tilde{x}_l)$ – индекс системы $\mathfrak{A}$ – есть число различных разбиений набора точек $\tilde{x}_1,...,\tilde{x}_l$ всеми элементами $\mathfrak{a} \in \mathfrak{A}$. Очевидно, $\Delta^{\mathfrak{A}}(\tilde{x}_1,...,\tilde{x}_l) \le 2^l$, т. е. не превышает числа всевозможных двоичных наборов длины l; $H^{\mathfrak{A}}(l) = \mathbf{E}\log\Delta^S(\tilde{x}_1,...,\tilde{x}_l)$ – математическое ожидание логарифма индекса семейства $\mathfrak{A}$ относительно набора $(\tilde{x}_1,...,\tilde{x}_l)$; $m^{\mathfrak{A}}(l) = \max\limits_{\tilde{x}_1,...,\tilde{x}_l \in X^l} \Delta^{\mathfrak{A}}(\tilde{x}_1,...,\tilde{x}_l)$ – функция роста семейства $\mathfrak{A}$. Легко видеть, что $\log m^{\mathfrak{A}}(l) \ge H^{\mathfrak{A}}(l)$, поэтому

$$\lim_{l\to\infty} \frac{\log m^{\mathfrak{A}}(l)}{l} = \lim_{l\to\infty} \frac{K_l(\mathfrak{A})}{l} \ge \lim_{l\to\infty} \frac{H^{\mathfrak{A}}(l)}{l} = 0.$$ □

4.11. Метод программирования колмогоровской и вапниковской оценки сложности классов решающих правил

Сложность $K_l(\mathfrak{A})$ класса алгоритмов $\mathfrak{A}$ определяется наименьшей длиной слова (программы) p, по которому при помощи соответствующей частично рекурсивной функции (наилучшему внешнему алгоритму) U^*

можно определить слово $\tilde{y} = \mathfrak{a}(\tilde{x}_1),...,\mathfrak{a}(\tilde{x}_l)$ в наиболее «трудном» (на множестве всех наборов $\hat{X}^l$, взятых из X^l, и алгоритмов семейства $\mathfrak{A}$) случае. Очевидно, $K_l(\mathfrak{A}) \le K_{U,X^l}(\mathfrak{A})$ для произвольной функции $U \in P_{p.r.}$, поэтому для оценивания $K_l(\mathfrak{A})$ сверху в качестве алгоритма U может быть взята, например, машина Тьюринга MT, вычисляющая $\tilde{y} = MT(p, \hat{X}_l)$, или подходящая программа π на каком-нибудь языке программирования такая, что $\pi(p, \hat{X}_l) = \tilde{y}$ для входа $(p, \hat{X}_l)$, и тогда, согласно доказанной теореме, $h_{\mathfrak{A}} = VCD(\mathfrak{A}) \le len(p)$.

Подход к оцениванию VCD на основе соотношения $VCD(\mathfrak{A}) \le len(p)$: $U(p, \hat{X}_l) = \tilde{y} = (\mathfrak{a}(\tilde{x}_1),...,\mathfrak{a}(\tilde{x}_l))$ называется *методом программирования оценки* VCD, *сокращенно* – ***pVCD***. Используя соотношение $K_{U^*,X^l}(\mathfrak{A}) = \underline{\log m^{\mathfrak{A}}(l)}$, получаем:

$$K_{U,X^l}(\mathfrak{A}) \ge \underline{]\log m^{\mathfrak{A}}(l)[} = K_l(\mathfrak{A}) \ge h_{\mathfrak{A}} = VCD(\mathfrak{A}), \qquad U \in P_{p.r.}.$$

Подход к оцениванию функции роста $m^{\mathfrak{A}}(l)$ на основе соотношения $m^{\mathfrak{A}}(l) \le 2^{len(p)}$, аналогичный методу программирования оценки VCD, называется методом программирования оценки $m^{\mathfrak{A}}(l)$, сокращенно – $pm^{\mathfrak{A}}(l)$. Вводятся обозначения $len(p) = pVCD(\mathfrak{A})$ и $2^{len(p)} = pm^{\mathfrak{A}}(l)$.

Этапы реализации метода $pVCD$ ($pm^{\mathfrak{A}}(l)$).

1° Изучение класса $\mathfrak{A}$ и определение как можно меньшей совокупности свойств (параметров, структурных особенностей) этого класса, указания значений которых достаточно, чтобы сформировать из них слово p, описывающее любой алгоритм $\mathfrak{a} \in \mathfrak{A}$. Предъявить алгоритм U (машину Тьюринга, частично рекурсивную функцию, программу для конечного компьютера) такую, что $\forall \mathfrak{a} \in \mathfrak{A}\ \exists p_A : U(p_A, \hat{X}_l) = (\mathfrak{a}(\tilde{x}_1),...,\mathfrak{a}(\tilde{x}_l))$.

2° Определение максимальной длины $len(p_A)$ слова p_A, $\mathfrak{a} \in \mathfrak{A}$, как оценки $VCD(\mathfrak{A})$ сверху ($2^{len(p_A)}$ как оценки $m^{\mathfrak{A}}(l)$ сверху).

Метод $pVCD$ предполагает *конструирование сжатого описания* p всего класса $\mathfrak{A}$ и указания алгоритма U, обрабатывающего вход $(p, \hat{X}_l)$. Во многих случаях достаточно очевидности существования такого алгоритма, но может оказаться, что применение $pVCD$ потребует искусства программирования и организации данных p, чтобы получить нетривиальную $pVCD$ оценку.

Сужая круг решающих правил до реализуемых на компьютерах разрядности M, как будет показано ниже, можно получить оценку $pVCD(\mathfrak{U})$ с указанием констант.

Теорема 4.35 (об аддитивности $pVCD$ оценки композиции алгоритмов). Пусть $S_0^r = \{f = f_1 \circ \cdots \circ f_r : \ f_1 \in S_1, ..., f_r \in S_r\}$ – такой класс композиций алгоритмов, принадлежащих семействам $S_1, ..., S_r$, что каждый алгоритм используется в композиции ровно один раз. Пусть известны оценки $pVCD(S_1) = L_1, \ldots, \ pVCD(S_r) = L_r$. Тогда справедлива оценка

$$pVCD(S_0^r) = \sum_{j=1,...,r} L_j + c \cdot r, \qquad (4.1)$$

где c – константа.

Доказательство. Любая композиция из S_0^r определяется совокупностью слов $p_1, ..., p_j, ..., p_r$, имеющих длины $L_1, ..., L_j, ..., L_r$. Для обработки этих слов, согласно методу программирования оценок и соотношению $U_j(p_j, \hat{X}_l) = \tilde{y}$, указаны алгоритмы U_j, $j = \overline{1,r}$, каждый из которых по слову p_j восстанавливает алгоритм f_j. Поэтому легко указать алгоритм (программу) $U_{S_0^r}$, обрабатывающую конкатенацию $p_0 = p_1 p_2 \cdots p_r$ и соответствующую композиции $f_1 \circ \cdots \circ f_r$. Такая программа будет содержать подпрограммы U_j, $j = \overline{1,r}$, которые восстанавливают все алгоритмы $f_1, ..., f_r$, и переходы между ними, предопределенные структурой композиции и известными длинами $L_1, ..., L_j, ..., L_r$ подслов, входящих в конкатенацию $p_0 = p_1 p_2 \cdots p_r$. Но для правильной расшифровки слова p_0 входящие в нее слова $p_1, p_2, \cdots, p_r$ должны быть снабжены разделителями для их вычисления. Для этой цели достаточно заменить каждое слово p_j его самоограничивающим кодом $p_j' = \overline{l(p_j)} p_j$, получив код p_0' длины $l(p_0') = l(p) + 2\sum_{j=1}^{r}]\log L_j[$. Тогда в качестве константы c в формуле (4.1) можно взять $2 \max\limits_{l}]\log L_j[$.

Следствие 4.8. $pVCD$ оценка суперпозиции алгоритмов $S_0^r = \{f = f_1 \circ \cdots \circ f_r : \ f_1 \in S_1, ..., f_r \in S_r\}$ имеет, в частности, вид $pVCD(S_0^r) = \log l \sum_{j=1,...,r} h_{S_j} + c \cdot r$, где $h_{s_1}, ..., h_{s_r}$ – емкости классов $S_1, ..., S_r$.

Доказательство усматривается из неравенства $K_l(S) < h_S \log l$.

Замечание. Согласно следствию 1, колмогоровская сложность $K_l(\mathfrak{A})$ должна зависеть от длины выборки l. Однако при использовании $pVCD(\mathfrak{A})$ может быть получена мажоранта сложности, определяемая длиной слова p и не зависящая от l. Это объясняется тем, что класс $\mathfrak{A}$ может оказаться конечным или тем, что функция $m^{\mathfrak{A}}(l)$ растёт не быстрее чем $O(l)$.

4.12. Примеры программирования *pVCD* оценок сложности

Оценка для ДНФ. Дизъюнктивной нормальной формой (*ДНФ*) представления булевых функций называется выражение вида $\bigvee_{j=1}^{\mu}(x_{j1}^{\sigma_{j1}} \,\&\, x_{j2}^{\sigma_{j2}} \,\&\ldots\&\, x_{jk_j}^{\sigma_{jk_j}})$, где $x^{\delta} = x$ при $\delta = 1$ (положительный литерал); $x^{\delta} = \bar{x}$ при $\delta = 0$ (отрицательный литерал); μ – число конъюнкций в *ДНФ*; $L = \sum_{j=1}^{\mu} k_j$ – длина, количество литералов в *ДНФ*.

Пусть класс $DNF_{L,\mu,n}$ – это семейство булевых функций вида $f : \{0,1\}^n \to \{0,1\}$, представимых в виде *ДНФ* длины не более L, содержащих не более чем μ конъюнкций. Используя $pVCD$ метод, можно получить оценку $VCD(DNF_{L,\mu,n}) < L + (\mu - 1 + L)]\log(n+1)[$ следующим образом. Слово p_f, позволяющее закодировать информацию о любой *ДНФ* длины $L = \sum_{j=1}^{\mu} k_j$, состоящей не более чем из μ конъюнкций над n переменными, можно представить конкатенацией μ двоичных слов сформированных из таких блоков, как показано в таблице 4.2.

Таблица 4.2. Фрагмент слова, кодирующего литерал

Номер переменной x_j, входящей в конъюнкцию, $j \in \{1,\ldots,n\}$, или ноль – разделитель блоков	Двоичная цифра 1, если x_j входит в конъюнкцию с инверсией, или 0 – в противном случае

Чтобы представить в двоичном коде один любой номер переменной или ноль, достаточно зарезервировать $]\log(n+1)[$ двоичных разрядов. Поскольку номера переменных начинаются с единицы, *ноль можно использовать как признак разделения конъюнкций в строке*. Для того чтобы указать знак литерала – с инверсией или без неё – достаточно одного двоичного разряда. При таком кодировании на каждый литерал в слове p_f бу-

дет расходоваться $]\log(n+1)[+1$ бит. На j-ю конъюнкцию будет расходоваться $k_j(]\log(n+1)[+1)$ бит для представления литералов. $(\mu-1)]\log(n+1)[$ бит понадобится для разделителей. Поэтому длина слова p_f не превысит

$$(\mu-1)]\log(n+1)+\sum_{j=1}^{\mu}k_j(]\log(n+1)[+1)$$
$$=(\mu-1)]\log(n+1)[+L]\log(n+1)[+L$$
$$=L+(\mu-1+L)]\log(n+1)[.$$

Если ДНФ содержит $m<\mu$ конъюнкций, то последние $\mu-m$ блоков слова p_f заполняются нулями.

Таблица 4.3. Расшифровка ДНФ по слову p_f

Цифры слова p_f	Пояснение
3	Взять переменную x_3;
1	x_3 берётся без инверсии;
5	Взять в текущую конъюнкцию следующую переменную x_5;
0	x_5 берётся с инверсией;
0	Вместо номера переменной – ноль; получена конъюнкция $x_3\bar{x}_5$, и далее начинается описание следующей конъюнкции, если за считанным нулём не последует второй ноль; счетчик выделенных конъюнкций увеличивается на единицу.
2	Цифра не равна нулю; включить в текущую конъюнкцию переменную x_2;
0	x_2 берётся с инверсией;
4	Цифра не равна нулю; взять в текущую конъюнкцию переменную x_4;
1	x_4 берётся без инверсии;
0	Поскольку вместо номера переменной – ноль, то получена конъюнкция $\bar{x}_2x_4$; счетчик выделенных конъюнкций увеличивается на единицу и становится равным двум. Значение $\mu=2$ свидетельствует об окончании слова p_f и представлении результата расшифровки – $x_3\bar{x}_5\vee\bar{x}_2x_4$.

Пусть, например, дана *ДНФ* $x_3\bar{x}_5\vee\bar{x}_2x_4$ из класса $DNF_{10,2,5}$ – длины не более 10 и не более чем с двумя конъюнкциями. Пусть число булевых переменных $n=5$. Десятичная (для облегчения восприятия) интерпретация слова p_f будет иметь вид |3|1|5|0|0|2|0|4|1|0|. Расшифровка этого слова (алгоритм U) поясняется таблицей 4.3. Поскольку $n=5$ и $]\log(n+1)[=3$, двоичное представление слова p_f будет следующим:

|011|1|101|0|000|010|0|100|1|000|. Здесь знак «|» сохранен для удобства восприятия структуры слова, но этот знак в слове p_f не содержится.

Оценивание VCD нейронной сети с единственным скрытым слоем, содержащим k элементов (класс $NN_{k,1}$). В работе [43] для нейронной сети с единственным скрытым слоем, содержащим элементов, и зафиксированной непараметрической активационной функцией σ представлена оценка

$$VCD(NN_{k,1}) = (2kn + 4k + 2) \times \log(e(kn + 2k + 1)).$$

Используя $pVCD$ метод, легко получить оценку [8]

$$VCD(NN_{k,1}) = M(kn + 2k + 1),$$

где M – число бит памяти, выделяемых для записи одного параметра; n – размерность входа.

Действительно, нейронные сети рассматриваемого класса полностью определяются $nk + 2k + 1$ параметрами: nk параметров соответствуют коэффициентам связи каждой из k внутренних вершин с каждым из n входов; k параметров определяют пороги суммирования для внутренних вершин и один параметр соответствует порогу выходной вершины сети. Если для каждого параметра используется M бит памяти, то каждую сеть рассматриваемого класса можно задать словом p длины $M(nk + 2k + 1)$. Алгоритм расшифровки этого слова состоит в последовательном считывании параметров (по M бит) согласно единому зафиксированному их порядку по всему классу. Считанные параметры подставляются в зафиксированные участки памяти алгоритма расшифровки.

Оценка, полученная $pVCD$ методом, будет лучше известной [43] при условии $M < 2\log(e(kn + 2k + 1))$, и ее выигрыш растет с ростом размерности задачи n.

Оценивание VCD класса $N_{k,m}$ нейронных сетей с k элементами в каждом из m скрытых слоев. Для этого класса аналогичным образом получена оценка [8]

$$pVCD(N_{k,m}) = M(nk + 2mk^2).$$

Оценивание VCD суперпозиции $f(F_1,...,F_k)$ с фиксированным логическим корректором $f \in P_2(k)$. **Пусть** $F_1,...,F_k$ – некоторые семейства алгоритмов вида $a : X^n \to \{0,1\}$, имеющие емкости соответственно $VCD(F_1),...,VCD(F_k)$, и f – зафиксированная булева функция. Обозна-

чим $f(F_1,...,F_k)=\{f(f_1,...,f_k): f_i \in F_i, i=\overline{1,k}\}$. В работе [43] получена оценка

$$VCD(f(F_1,...,F_k)) \le 2k\log(e\cdot k)\max_i\{VCD(F_i)\}.$$

Используя $pVCD$ метод (см. теорему), можно получить оценку

$$VCD(f(F_1,...,F_k)) = \sum_{i=1}^{k}(pVCD(F_i)+2k]\log VCD(F_i)[)$$
$$< k\max_i\{VCD(F_i)\}+c,$$

где c – дополнительная часть оценки – константа самоограничивающего кодирования. Основная часть оценки, полученной $pVCD$ методом, лучше в $2\log(ek)$ раз.

Оценивание VCD ***класса*** $BFT_{n,m}$ ***бинарных решающих деревьев с*** μ ***листьями*** $pVCD$ метод позволяет получить оценку

$$pVCD(BFT_{n,m}) = (\mu-1)(]\log n[+]\log(\mu+3)[),$$

где n – число булевых переменных. Логико-комбинаторным методом ранее удалось получить оценку

$$VCD(BFT_{n,m}) < (\mu-1)\log n+\mu-1+\sum_{j=2}^{\mu-1}\ln j \text{ [6].}$$

Сравнение последних двух оценок показывает, что $pVCD$ оценка точнее.

Для класса $BSP_{n,m}$ [23] композиций бинарных решающих деревьев *не более чем* с μ листьями и линейными предикатами во внутренних вершинах, зависящих от n числовых переменных, занимающих по M бит каждая, $pVCD$ оценка имеет вид:

$$pVCD(BSP_{n,m}) =$$
$$= (\mu-1)(]\log n[+]\log(\mu+3)[+(n+1)M+2]\log((n+1)M)[).$$

VCD структурной композиции линейного алгебраического корректора k эвристических моделей $F_1,...,F_k$ (класс $L(F_1,...,F_k)$). Для указанной совокупности эвристических алгоритмов с произвольным линейным корректором легко получить оценку

$$pVCD(L(F_1,...,F_k)) = Mk+\sum_{i=1}^{k}(pVCD(F_i)+2k]\log VCD(F_i)[).$$

Оценивание VCD ***интервальных множественных автоматов*** (IMA)**.** Класс решающих функций $\Im_{IMA}$, порождаемый IMA, описывается двумя следующими определениями.

Определение 4.18 [16]**.** Множественным автоматом (MA) называется пятёрка $< Q, \Sigma, \delta, q_0, F >$, где Q – конечное множество состояний, Σ – конечный алфавит, $\delta : Q \times Z \to 2^Q$ – множественная функция переходов, $q_0 \in Q$ – начальное состояние, $F \subset Q$ – множество финальных состояний. Последовательность $p_0, p_1, ..., p_n$ называется *принимаемым путём* для входа $\omega_1, ..., \omega_n$, если $p_0 = q_0$; $p_i = \delta(p_{i-1}, \omega_i)$ для любого $i = 1, ..., n$ и $p_n \in F$. Автомат MA вычисляет функцию $f_{MA} : \Sigma^* \to \{0,1\}$, где $f_{MA}(\omega) = 1$, если число принимаемых путей для $\omega = (\omega_1, ..., \omega_n)$ является нечётным, и $f_{MA}(\omega) = 0$, если это число – чётное.

Определение 4.19 [16]**.** Интервальным множественным автоматом (IMA) называется пара $< A, C >$, где A – множественный автомат с алфавитом $\Sigma = \{0,1,..., \mu - 1\}$, C – множество, состоящее из $\mu - 1$ вещественных чисел: $C = \{c_0, c_1, ..., c_{\mu-1}\}$, $c_0 = -\infty$, $c_0 < c_1 < ... < c_{\mu-1}$. Индексом числа a, обозначаемым $ind_C(a)$, называется $\max\{i : c_i \le a\}$. Функция $f_{<A,C>}$, вычисляемая IMA $< A, C >$, ставит в соответствие вещественной числовой последовательности $(x_1, ..., x_n) \in \mathrm{R}^n$ значение $f_{<A,C>}(ind_C(x_1), ..., ind_C(x_n))$.

В работе [16] получена оценка

$$VCD(\Im_{IMA}) = O(\mu(\log \mu + r^2)),$$

где $\mu = |\Sigma|$, $r = |Q|$. Сначала авторы работы [16] оценили сверху число способов обработки автоматом IMA входной последовательности как

$$(VCD(\Im_{IMA}) \cdot n + 2)^{\mu} \cdot 2^{O(\mu r^2)},$$

а затем получили окончательный результат.

Применение $pVCD$ метода даёт существенно лучшую оценку

$$pVCD(\Im_{IMA}) = \mu(M + r^2) + r.$$

4.13. Колмогоровская сложность классов решающих функций и оценивание эмпирических закономерностей

Определение 4.20. Пусть $X_l = (\tilde{x}_j, \alpha_j)_{j=1}^l$ – зафиксированная обучающая выборка, S – семейство алгоритмов, используемое для обучения. Выбор решения f^* функциональной системы (1), если оно существует,

$$\begin{cases} f(\tilde{x}_1)=\alpha_1 \\ f(\tilde{x}_2)=\alpha_2 \\ \cdots\cdots\cdots\cdots \\ f(\tilde{x}_l)=\alpha_l \\ f\in S, \end{cases} \quad (1) \qquad \begin{cases} f(\tilde{x}_{j_1})=\alpha_{j_1} \\ f(\tilde{x}_{j_2})=\alpha_{j_2} \\ \cdots\cdots\cdots\cdots \\ f(\tilde{x}_{j_k})=\alpha_{j_k} \\ f\in S, \end{cases} \quad (2)$$

называется безошибочной настройкой на выборку X_l. Выбор решения функциональной системы (2), если оно существует, называется настройкой на k $(1<k<l)$ фиксированных элементов $\tilde{x}_{j_1},\tilde{x}_{j_2},\cdots,\tilde{x}_{j_k}$ выборки X_l и является настройкой на подвыборку X_k выборки X_l.

Будем полагать, что обучающая выборка извлекается случайно и независимо из множества обучающих выборок X^l. В случайно извлеченной обучающей выборке $(\tilde{x}_j,\alpha_j)_{j=1}^l$ булев вектор $\tilde{\alpha}_l=(\alpha_1,...,\alpha_j,...,\tilde{\alpha}_l)$ может появиться с некоторой вероятностью.

Теорема 4.36. Пусть вероятностная модель извлечения выборки из генеральной совокупности X^l такова, что *появление любого булевого вектора* $\tilde{\alpha}_l$ в произвольно извлеченной выборке $(\tilde{x}_j,\alpha_j)_{j=1}^l$ *равновероятно*. Тогда вероятность $P(S,l,\delta l)$ случайной настройки на какие-нибудь $l-\delta l$ элементов выборки $(\tilde{x}_j,\alpha_j)_{j=1}^l$ при извлечении решающего правила из семейства S удовлетворяет неравенству

$$P(S,l,\delta l)<C_l^{\delta l}2^{-(l-K_l(S)-\delta l)},$$

где $K_l(S)$ - колмогоровская сложность семейства S, а δl - число ошибок, допущенных на обучающей выборке $(\tilde{x}_j,\alpha_j)_{j=1}^l$ выбранным из семейства S алгоритмом.

Доказательство. Семейство S однозначно порождает конечное множество $M_S(X_l)$ разных способов классификации любой выборки X_l. Мощность этого множества $|M_S(X_l)|$ *не превышает* $m^S(l)$. Точная настройка на все l элементов выборки может произойти только тогда, когда способ $\tilde{\alpha}_l$ классификации последовательности X_l на два класса содержится во множестве $M_S(\tilde{X}_l)$. Можно сказать, что точная настройка произойдет тогда, когда входящий в обучающую выборку $(\tilde{x}_j,\alpha_j)_{j=1}^l$ вектор $\tilde{\alpha}_l$ случайно "попадёт" в такую же точку $\tilde{\alpha}_l$ множества $M_S(\tilde{X}_l)$. Веро-

ятность такого события равна вероятностной мере множества $M_S(\tilde{X}_l)$: $\Pr\{M_S(\tilde{X}_l)\} = |M_S(\tilde{X}_l)| / 2^l \le m^S(l)/2^l$, поскольку любой вектор $\tilde{\alpha}_l$ может появиться в выборке равновероятно по условию теоремы. Поэтому вероятность точной настройки *на фиксированную часть* выборки длины $l - \delta l$ *не превысит* $m^S(l) \cdot 2^{\delta l} / 2^l$. Выбрать $l - \delta l$ элементов из l можно $C_l^{\delta l}$ способами. В результате получается оценка $P(S,l,\delta l) < C_l^{\delta l} m^S(l)/2^{(l-\delta l)}$. Поскольку $K_l(S) =]\log m^S(l)[$, то $2^{K_l(S)} \ge m^S(l)$. Поэтому $P(S,l,\delta l) < C_l^{\delta l} 2^{-(l-K_l(S)-\delta l)}$.

Следствие 4.9. Пусть вероятностная модель извлечения выборки из генеральной совокупности X^l такова, что появление любого булевого вектора $\tilde{\alpha}_l$ в произвольной обучающей выборке $(\tilde{x}_j, \alpha_j)_{j=1}^l$ равновероятно. Тогда вероятность $P(S,l,0)$ точной случайной настройки на выборку $(\tilde{x}_j, \alpha_j)_{j=1}^l$ удовлетворяет неравенству $P(S,l,0) < 2^{-(l-K_l(S))}$.

Следствие 4.10. Пусть вероятностная модель извлечения выборки из генеральной совокупности X^l такова, что появление любого булевого вектора $\tilde{\alpha}_l$ в произвольной обучающей выборке $(\tilde{x}_j, \alpha_j)_{j=1}^l$ равновероятно, колмогоровская сложность оценена (например, при помощи $pVCD$ метода) и получено неравенство $K_l(S) \le len(p)$. Тогда $P(S,l,0) < 2^{-(l-len(p))}$. □

Следуя А. Н. Колмогорову, мы придерживаемся мнения о *закономерности как неслучайности*. С такой точки зрения *вероятность неслучайной настройки* или, иначе говоря, *обнаружения закономерности*, соответственно оценивается величиной $1 - 2^{-(l-len(p))}$.

Если $l - K_l(S) \ge 5$, то $P(S,l,0) < 2^{-5} = 0{,}03125$, и тогда вероятность неслучайного обнаружения закономерности не меньше $0{,}96$. Это вполне приемлемо на практике и позволяет сформулировать следующее

Правило "плюс пять": *Для обеспечения надёжного извлечения закономерности (решающего правила или алгоритма) из используемого семейства алгоритмов длина обучающей последовательности должна быть хотя бы на 5 единиц больше, чем колмогоровская сложность этого семейства*.

Применим для примера правило "*плюс пять*" для класса решающих правил, имеющих вид *ДНФ* над $n = 100$ переменными длины не более

$L = 20$ не более чем с $\mu = 7$ конъюнкциями. В соответствии с полученной оценкой

$$pVCD(DNF_{L,\mu,n}) < L + (\mu - 1 + L)]\log(n+1)[= 20 + (6+20)\cdot 7 = 202$$

определяем, что найденная $DNF_{20,7,100}$ – закономерность, безошибочно классифицирующая всю обучающую выборку длины $l \geq 207$, может считаться неслучайной с вероятностью не менее $0{,}96$.

Для понимания и применения правила "плюс пять" нужно учитывать, что задачи синтеза закономерностей (классификаторов) по прецедентной информации являются частным случаем проблемы принятия решений в условиях неопределённости. Это означает, что решения отыскиваются в широкой области, порождённой частичной информацией. Для любой задачи из рассматриваемого класса Z с начальной информацией I эта область неопределённости $\mathfrak{D}(Z, I)$ содержит огромное количество решений, включая нужное решение g. Кроме этого, о вероятностном распределении решений в области $\mathfrak{D}(Z, I)$ ничего не известно. Поэтому представляется естественным:

а) предположить такое распределение равномерным, что соответствует случаю наибольшей неопределённости;

б) попытаться как можно больше сузить (сжать) область $\mathfrak{D}(Z, I)$ до области $\mathfrak{D}'(Z, I)$, не потеряв при этом теоретическую возможность нахождения правильного решения: $g \in \mathfrak{D}'(Z, I) \subset \mathfrak{D}(Z, I)$.

В этом смысле выше шла речь об *обучении сжатием и* $pVCD$ *методе как аппарате такого обучения и оценивания классификаторов и закономерностей, синтезированных по начальной прецедентной информации*. $pVCD$ метод является одним из возможных вариантов обоснования эмпирических индукторов, и в этом направлении проводятся широкие научные исследования [3,4].

Литература к главе 4

1. Вапник В. Н. Восстановление зависимостей по эмпирическим данным / В.Н.Вапник. – М. Наука, 1979. – 447 с.
2. Вапник В. Н., Червоненкис А. Я. Теория распознавания образов / В. Н. Вапник, А. Я. Червоненкис. – М.: Наука, 1974. – 416 с.
3. Воронцов К. В. Обзор современных исследований по проблеме качества обучения алгоритмов / К. В. Воронцов // Таврический вестник информатики и математики, 2004. – № 1. – С. 5–24.
4. Воронцов К. В. Слабая вероятностная аксиоматика и надёжность эмпирических предсказаний / К. В. Воронцов // Математические методы распознавания образов-13. – М.: МАКС Пресс, 2007. – С. 21–25.
5. Вьюгин В. В. Колмогоровская сложность и алгоритмическая случайность / В.В.Вьюгин. – М.: МФТИ, 2012. – 131 с.
6. Донской В.И. Асимптотика числа бинарных решающих деревьев / В.И. Донской // Ученые записки Таврического национального университета им. В.И. Вернадского, Серия "Математика". – 2001. – Т. 14(53), №1. – С.36-38.
7. Донской В. И. Колмогоровская сложность классов общерекурсивных функций с ограниченной ёмкостью / В. И. Донской // Таврический вестник математики и информатики, 2005. – №1. – С. 25 – 34.
8. Донской В. И. Оценки ёмкости основных классов алгоритмов эмпирического обобщения, полученные *pVCD* методом / В. И. Донской // Ученые записки ТНУ им. В. И. Вернадского. Серия «Физико-математические науки», 2010. – Т. 23(62). – №2. – С. 56 – 65.
9. Донской В. И. Сложность семейств алгоритмов обучения и оценивание неслучайности извлечения эмпирических закономерностей / В.И. Донской // Кибернетика и системный анализ, 2012. – №2. – С. 86 – 96.
10. Донской В. И. Эмпирическое обобщение и распознавание: классы задач, классы математических моделей и применимость теорий. Часть I; Часть II / В. И. Донской // Таврический вестник информатики и математики, 2011. – №1. – С. 15 – 26; №2. – С. 31 – 42.
11. Донской В. И. Эмпирическое обобщение и распознавание: классы задач, классы математических моделей и применимость теорий. Часть II / В. И. Донской // Таврический вестник информатики и математики, 2011. – №2. – С. 86 – 96.
12. Звонкин А. К., Левин Л. А. Сложность конечных объектов и обоснование понятий информации и случайности с помощью теории алгоритмов / А. К. Звонкин, Л. А. Левин // Успехи математических наук, 1970. – Т. 25:6(156). – С. 85 – 127.
13. Колмогоров А. Н. Теория информации и теория алгоритмов // А.Н.Колмогоров. – М.: Наука, 1987. – 304 с.
14. Мучник А. А., Семенов А. Л. Гиперпростые множества, возникающие при вычислимой аппроксимации сверху префиксной сложности [Электронный

ресурс] / А. А. Мучник, А. Л. Семенов. – ВЦ РАН, Отделение кибернетики, 2002. – 9 с. – Режим доступа: http://alexander.shen.free.fr/muchnik/publications/hh-simple.pdf

15. Успенский В. А., Верещагин Н. К., Шень А. Колмогоровская сложность и алгоритмическая случайность. – М.:МЦНМО, 2010. – 556 с.
16. Beimel A., Kushilevitz E. Learning Unions of High Dimentional Boxes over the Reals / A. Beimel, E. Kushilevitz // Inf. Proc. Letters. – 2000. – Vol.73. – Issue 5–6. – P. 213–220.
17. Blumer A. Learnability and the Vapnik-Chervonenkis Dimension / A. Blumer, A.Ehrenfeucht, D. Haussler, M. Warmuth // J. Assoc. Comp. Mach., 1989. – 35. – P. 929 – 965.
18. Blumer A. Occam's Razor / A. Blumer, A. Ehrenfeucht, D. Haussler, M. War muth // Information Processing Letters, 1987. – Vol. 24(6). – P.377 – 380.
19. Blumer A., Littlestont N. Learning faster than promise by the Vapnik-Chervonenkis dimension / Anselm Blumer, Nick Littlestone // Discrete Applied Mathematics, 1989. – Vol. 24. – Iss. 1-3, – P. 47 – 63.
20. Bousquet O., Elisseeff A. Algorithmic Stability and Generalization Perfor mance / Olivier Bousquet , André Elisseeff // Advances in Neural Information Processing Systems. – 2001. – 13. – P. 196 – 202.
21. Bousquet O., Elisseeff A. Stability and Generalization / Olivier Bousquet, André Elisseeff // Journal of Machine Learning Research. – 2002. – 2. – P. 499-526.
22. Elisseeff F. A Study About Algorithmic Stability and Their Relation to Generalization Performances // Andre Elisseeff. – Technical report. – Laboratoire ERIC, Univ. Lyon 2, 2000. – 19 P.
23. Devroye L. A. Probabilistic Theory of Pattern Recognition / L. A. Devroye, L. Gyorfi, G. Lugosi. – NY: Springer-Verlag, 1996. – 636 p.
24. Devroye L., Wagner T. Distribution-free performance bounds for potential function rules [Электронный ресурс] / Luc Devroye, T. Wagner // IEEE Transactions on Information Theory. – 1979. – 25. – P. 601 – 604. – Режим доступа: https://www.researchgate.net/publication/3083261_Distribution-free_performance_bounds_for_potential_function_rules
25. Donskoy V. I. The estimations based on the Kolmogorov Complexity and Machine Learning from Examples/ V. I. Donskoy // Proc. of the 5-th Int. Conf. "Neural Networks and ArtificialIntelligence"(ICNNAI'2008). – Minsk:INNS. – 2008. – P. 292–297.
26. Ehrenfeucht A. A general lower bound on the number of examples needed for learning / A. Ehrenfeucht, D. Haussler, M. Kearns, L. Valiant // Inform. Computations, 1989. – 82. – P. 247 – 261.
27. Floyd S., Warmuth M. Sample Compression, learnability, and the Vapnik-Chervonenkis dimension / Sally Floyd, Manfred Warmuth // J. Machine Learning, 1995. – Vol. 21. – Iss. 3. – P. 269 – 304.

28. Freund Y. Self bounded learning algorithms / Y. Freund // In Proc. Of the 11th Ann. Conf. on Computational Learning Theory (COLT-98). – N.Y.: ACM Press. – 1998. – P. 247 – 258.
29. Haussler D. Overview of the Probably Approximately Correct (PAC) Learning Framework / David Haussler // AAAI'90 Proceedings of the eighth National conference on Artificial intelligence, 1990. – Volume 2. – P. 1101–1108. http://www.cbse.ucsc.edu/sites/default/files/smo_0.pdf
30. Hutter M. Algoritmic complexity // Scholarpedia [Электронный ресурс]. – 2008. – 3(1):2573. – Режим доступа: http://www.scholarpedia.org/article/Algorithmic_complexity#Prefix_Turing_machine
31. Kearns M. J., Vazirani U. V. An Introduction to Computational Learning Theory / M. Kearns, U. Vazirani. – MIT Press 1994. – 221 p.
32. Li M., Tromp J., Vitányi P. Sharpening Occam's Razor / Ming Li, John Tromb, Paul M. B. Vitányi. – Research Rep. CT-94-03. – Amsterdam: ILLC, 1994. – 13 p. http://www.illc.uva.nl/Research/Reports/CT-1994-03.text.pdf
33. Li M., Vitányi P. An introduction to Kolmogorov complexity and its applications / Ming Li, Paul M. B. Vitányi. – New York: Springer-Verlag, 1997. – 637 p.
34. Li M., Vitányi P. Learning Simple Concepts under Simple Distributions / Ming Li, Paul M. B. Vitányi // SIAM J. Comput. – Vol. 20. – Iss. 5. – P. 911–935.
35. Li M., Vitányi P. Theories of Learning / Ming Li, Paul M. B. Vitányi [Электронный ресурс] // In Proc. Int. Conf. Of Young Computer Scientists. – Beijing, China. – 1993. – 8 P. – Режим доступа: http://www.google.com.ua/url?sa=t&rct=j&q=Can+computers+learn%3F++Recent+research+on+learning+theory+suggests&source=web&cd=1&cad=rja&ved=0CCEQFjAA&url=http%3A%2F%2Fhomepages.cwi.nl%2F~paulv%2Fpapers%2Ficycs93.ps&ei=oNtYUKHHKsXptQbAyIDoCw&usg=AFQjCNG9z7RLrMoMxuWPI8VqLtmS91pbHA
36. Littlestone L., Warmuth M. Relaring Data Compression and Learnability [Электронный ресурс] / Nick Littlestone, Manfred K. Warmuth. – Technical Report. – Santa-Cruz: University of California, 1986. – 13 p. Режим доступа: http://users.soe.ucsc.edu/~manfred/pubs/T1.pdf
37. McDiarmid C. On the method of bounded differences / Colin McDiarmid // In Surveys in Combinatorics. – Cambridge University Press, Cambridge, 1989. – London Math. Soc. Lectures Notes. – 141. – P. 148–188.
38. Mukherjee S. Learning theory: stability is sufficient for generalization and necessary and sufficient for consistency of empirical risk minimization / Sayan Mukherjee, Partha Niyogi, Tomaso Poggio, and Ryan Rifkin // Advances in Computational Mathematics. – 2006. – 25. – P. 161–193.
39. Noga A., Shai B. D. Scale-sentitive Dimensions, Uniform Convergence, and Learnability / Alon Noga, Ben David Shai // Journal of the ACM. – 1997. – 44(4). – p. 615 – 631.

40. Ogielski A. T. Information, Probability, and Learning from Examples. Survey / [Электронный ресурс] Andrew Ogielski. – Bell Communication Research, 1990. – 87 p. Режим доступа: http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.29.9797&rep=rep1&type=pdf
41. Pestov V. PAC learnability under non-atomic measures: a problem by Vidyasagar / Vladimir Pestov // 21st Int. Conf. "Algorithmic Learning Theory"(ALT 2010). – Canberra, Australia, 2010. – P. 134 – 147.
42. Rifkin M. R. Everything Old Is New Again: A Fresh Look at Historical Approaches in Machine Learning / Ryan Michael Rifkin. Ph.D. in Operation Research. Thesis, MIT, 2002. – 221 P.
43. Sontag E.D. VC dimension of Neural Networks / E. D. Sontag // In Neural Networks and Machine Learning. – Berlin: Springer, 1998. – P. 6995.
44. Sridharan K. Learning from an Optimization Viepoint / Karthik Sridharan. – Thesis for degree of Philosophy in Computer Science [Электронный ресурс].– Chicago:TTIC, 2012. – 217 p. – Режим доступа: http://ttic.uchicago.edu/~karthik/thesis.pdf
45. Valiant L. G. A Theory of the Learnable / Leslie G. Valiant // Communications of the ACM, 1984. – Vol. 27. – N11. – P. 1134 – 1142.
46. Vapnik V. N. The Nature of Statistical Learning Theory / Vladimir N. Vapnik. – 2nd ed. – New York: Springer-Verlag, 2000. – 314 p.
47. Vitányi P., Li M. Ideal MDL and Its Relation to Bayesianism / Paul M. B. Vitányi, Ming Li // In Proc. ISIS: Information, Statistic and Induction in Science. – Singapore: World Scientific, 1996. – P. 282 – 291.
48. Vitányi P., Li M. Minimum description length induction, Bayesianism, and Kolmogorov complexity / Paul M. B. Vitányi, Ming Li // IEEE Transactions on Information Theory, 2000 – Vol.46. – N2. – P.446–464.

5. Синтез бинарных классифицирующих деревьев как задача машинного обучения

«Любили вы петь и считали,
что музыка – ваша звезда?
– Да.
– Имели вы слух или голос
и знали хотя бы предмет?
– Нет.
– Вы знали ли женщину
с узкою трубочкой рта?
И дом с фонарем
отражался в пруду,
Как бубновый валет?
– Нет»
А. Вознесенский

5.1. Основные понятия, связанные с деревьями классификации

Идеи построения и применения деревьев решений в машинном обучении и распознавании впервые появились в статьях Ханта и Ховленда в 50-х годах XX века. Но центральной работой, привлекшей внимание математиков и программистов к этому направлению во всем мире, явилась книга Ханта, Марина и Стоуна [47], увидевшая свет в 1966г.

В Советском союзе научное направление, связанное с решающими деревьями и граф-схемами алгоритмов, начало развиваться примерно в то же время в научной школе А. Ш. Блоха [3]. Из многочисленных работ этой школы (см. обзор в работе [14]) следует обратить особое внимание на исследование В. А. Орлова [36], который первым, еще в начале 70-х годов прошлого века, – более чем на 10 лет раньше Росса Куинлана – предложил энтропийный критерий ветвления и алгоритм синтеза решающих деревьев, который принципиально не отличался от широко используемого в настоящее время алгоритма ID3 [63].

Синтез бинарных решающих деревьев, вообще говоря, состоит из двух этапов: *выбора признаковых предикатов и собственно построения дерева решений*. Эти этапы могут быть совмещены, что часто реализуется при синтезе деревьев, например, соответствующих разбиениям вещественного признакового пространства гиперпараллелепипедами. Далее предполагается, что применяется именно двухэтапный подход, причем все *рассмотрение сосредоточено в основном на вопросе синтеза БРД при уже найденном наборе признаковых предикатов и их значений, зафиксированных в логических таблицах обучения*. Проблема поиска признаковых предикатов рассматривается отдельно.

Каждой внутренней вершине БРД ставится в соответствие некоторый (признаковый) предикат. В каждую внутреннюю вершину, кроме выделенной – корневой, – входит одно ребро. Из каждой внутренней вершины БРД исходят два ребра, соответствующие нулевому и единичному значению предиката, приписанного этой вершине. Каждая ветвь БРД не содержит одинаковых предикатов в своих внутренних вершинах и заканчивается концевой вершиной-листом, который помечен номером класса.

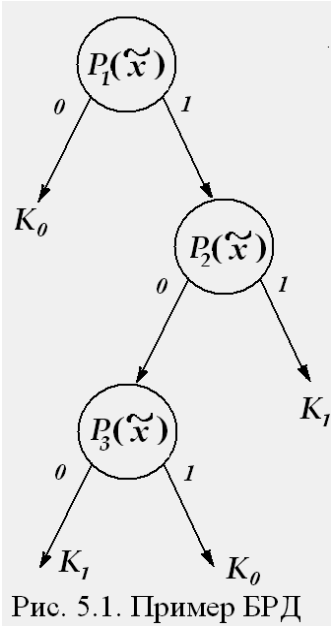

Рис. 5.1. Пример БРД

Алгоритм распознавания, определяемый БРД, относит объекты (точки признакового пространства), для которых все предикаты в ветви дерева, заканчивающейся этим листом, обращаются в единицу (выполняются) к тому классу, метка которого находится в концевой вершине этой ветви.

Пример *БРД* с тремя внутренними вершинами, четырьмя листьями, использующего три признаковых предиката $P_1(\tilde{x}),\ P_2(\tilde{x}),\ P_3(\tilde{x})$ и реализующего классификатор на два класса K_0, K_1, приведен на рис. 5.1. «Логика» этого классификатора или алгоритм классификации следующий.

1º Если $P_1(\tilde{x}) = 0$, то выдать ответ $\tilde{x} \in K_0$ и остановиться; иначе (если $P_1(\tilde{x}) = 1$) то перейти на 2º;

2º Если $P_2(\tilde{x}) = 1$, то выдать ответ $\tilde{x} \in K_1$ и остановиться; иначе (если $P_2(\tilde{x}) = 0$) то перейти на 3º;

3° Если $P_3(\tilde{x}) = 0$, то выдать ответ $\tilde{x} \in K_1$ и остановиться; иначе выдать ответ $\tilde{x} \in K_0$ и остановиться.

Эквивалентными приведенному алгоритму являются две непосредственно выписываемые по *БРД* решающие (классифицирующие) булевы функции:

$$f_{K_0}(P_1(\tilde{x}), P_2(\tilde{x}), P_3(\tilde{x})) = \overline{P}_1(\tilde{x}) \vee P_1(\tilde{x}) \,\&\, \overline{P}_2(\tilde{x}) \,\&\, P_3(\tilde{x});$$

$$f_{K_1}(P_1(\tilde{x}), P_2(\tilde{x}), P_3(\tilde{x})) = P_1(\tilde{x}) \,\&\, P_2(\tilde{x}) \vee P_1(\tilde{x}) \,\&\, \overline{P}_2(\tilde{x}) \,\&\, \overline{P}_3(\tilde{x}).\square$$

Легко проверяется хорошо известное *свойство БРД*: *число его внутренних вершин всегда на единицу меньше числа листьев*. Поэтому минимизация числа листьев и числа тестов в вершинах – эквивалентны.

Длиной ветви называют число содержащейся в ней вершин. Высотой *БРД* называют длину его ветви, содержащей наибольшее число вершин.

Дерево называют равномерным (сбалансированным), если все его ветви имеют равную длину.

В современных интеллектуализированных информационных технологиях *БРД* (в англоязычной литературе – *Binary Decision Trees*) занимают важное место, особенно в связи с развитием таких направлений, как *Machine Learning*, *Case-Based Reasoning*, *Data Mining*.

Перечислим вкратце основные свойства *БРД*, определяющие возможности их реализации и значимость для использования в указанных информационных технологиях.

1º. *БРД* – класс понятных, легко интерпретируемых и воспринимаемых решающих правил, применяемых для распознавания, классификации, формирования понятий, слабоопределенной оптимизации и др.

2º. *БРД* с ограниченным (и небольшим) числом листьев определяют для случая двухэлементных решений (бинарной классификации) чрезвычайно узкий класс булевых функций, асимптотически (при числе аргументов $n \to \infty$) сколь угодно узкий по сравнению даже с классом линейных булевых функций $L(n) \subset P_2(n)$ [12].

3º. Любая булева функция из $P_2(n)$ может быть представлена в виде *БРД*.

4º. Если μ – число листьев, то для класса $D(n, \mu)$ булевых функций, представимых *БРД*, при условии $2 \le \mu \le 2^n$ справедливо включение $D(n, \mu) \subset D(n, \mu + 1)$. Свойства 2º,3º,4º обосновывают возможность оптимизационного синтеза *БРД*-индуктора, корректного на непротиворечивой начальной обучающей информации, путем минимизации параметра μ [11].

5º. Синтез по заданной конечной корректной начальной информации *БРД* с минимальным числом листьев μ является сложной экстремальной задачей из класса NPC [48] (к ней сводится, например, NP-полная задача о точном покрытии).

6º. Построенное *БРД* с μ листьями (μ – константа) далее позволяет со сложностью $O(n)$ получить логическое описание синтезированных классов в виде дизъюнктивных нормальных форм (ДНФ). Конъюнкции, входящие в эти ДНФ, являются эмпирическими закономерностями и могут быть использованы, кроме прочего, для синтеза эмпирических продукций, пополняющих базы знаний.

7º. Свойства 4º,5º определяют актуальность построения эвристических алгоритмов синтеза *БРД*, близких к оптимальным, и усилия разработчиков интеллектуализированного программного обеспечения, настойчиво проявляемые в этом направлении.

8º. Путём выбора подходящего набора признаковых предикатов *БРД* можно использовать для классификации объектов, описанных разнотипными признаками.

Исследование и выбор признаковых предикатов имеет едва ли не решающее значение. В этой связи нужно упомянуть работу И. Б. Сироджи [40], в которой решающее дерево как граф отношений сопоставляется со специальной программной регулярной грамматикой, порождающей регулярный язык структурно-аналитического описания образов. В этой работе даны определения структурно-полной и неизбыточной систем свойств-предикатов по отношению к обучающей выборке.

Г.С. Лбов в работе [33] предложил эвристический алгоритм формирования логических решающих функций с выделением признаковых предикатов, позволяющий строить понятия при разнотипных признаках, описывающих объекты.

5.2. Булевы функции, критерии ветвления и бинарные деревья классификации

Будем полагать, что на основе анализа предметной области уже выбрано n признаковых предикатов для синтеза *БРД*. Отождествим эти признаковые предикаты с булевыми переменными $x_1,\ldots,x_n$.

Класс булевых функций, представимых *БРД*, полон: при помощи бинарного дерева можно построить алгоритм реализации любой булевой функции. Это важное свойство легко доказывается путем последовательного разложения Шеннона по одной переменной (рис. 5.2):

$$f(x_1,\ldots,x_{i-1},x_i,x_{i+1},\ldots,x_n) =$$
$$= x_i f(x_1,\ldots,x_{i-1},1,x_{i+1},\ldots,x_n) \vee x_i f(x_1,\ldots,x_{i-1},0,x_{i+1},\ldots,x_n)$$

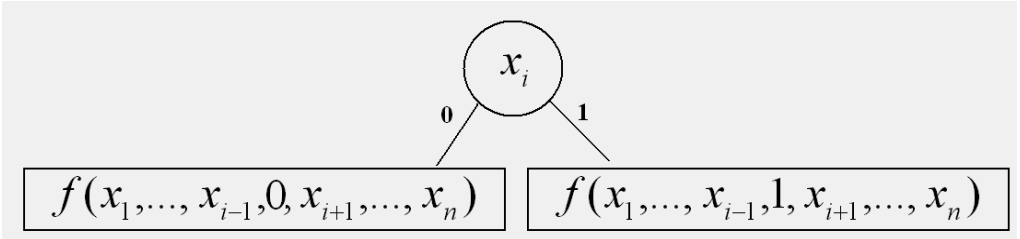

Рис. 5.2. Шаг ветвления соответствует шагу разложения по одной переменной

Разложение по r переменным вдоль любой ветви *БРД* определяет интервал ранга r (рис. 5.3) в разбиении множества вершин единичного n-мерного куба B^n на совокупность непересекающихся интервалов, помеченных номерами классов, к которым *БРД* относит эти интервалы. Кодами интервалов являются наборы значений предикатов, размещенных во внутренних вершинах соответствующих ветвей, а их размерность равна 2^{n-r}.

Ниже рассмотрение процесса ветвления как последовательного разбиения B^n на интервалы основано на теоретико-множественном подходе, развитом в работах Ю. И. Журавлева [31]. Этот подход оказался плодотворным и послужил толчком к разработке ряда критериев ветвления на основе понятия отделимости [30]. Синтез *БРД* с минимальным числом листьев равносилен синтезу кратчайшего ортогонального покрытия, корректного относительно размещения точек из обучающей выборки по интервалам разбиения.

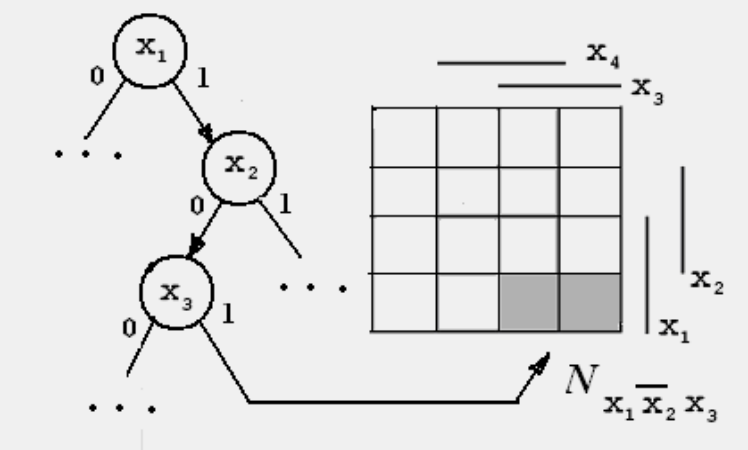

Рис. 5.3. Ветвь *БРД* соответствует интервалу $N_{x_1\bar{x}_2x_3}$

Число листьев μ *БРД* является естественной мерой его сложности, поскольку число внутренних вершин $\mu-1$ определяет количество однотипных шагов, выполняемых при «наращивании» дерева в процессе синтеза.

Обозначим q – заданное число классов объектов, а $\mathfrak{D}(n,q,\mu)$ – семейство *БРД*, имеющих ровно μ листьев. Точная формула для числа $d=|\mathfrak{D}(n,q,\mu)|$ неизвестна. Произвольная булева функция представима *БРД*, вообще говоря, не единственным образом.

В работе [12] получена асимптотическая оценка

$$d(n,q,\mu)\sim(\mu-1)![q(q-1)]^{\mu-1}n(n-1)^{\mu-2} \text{ при } n\to\infty,$$

и доказано, что число $b(n,2,\mu)$ булевых функций (случай $q=2$), представимых *БРД* с ровно μ листьями, удовлетворяет неравенству

$$b(n,2,\mu)<(\mu-1)!2^{\mu-1}n^{\mu-1}.$$

Подробнее об этих оценках см. в п. 5.10.

Для VCD конечного класса $\mathfrak{B}(n,2,\mu)$ решающих функций, представимых в виде *БРД* с числом листьев, *не превышающим* μ, в случае двух классов $pVCD$ методом [20] получена оценка [15]:

$$VCD(\mathcal{B}(n,2,\mu)) < (\mu-1)(\log(n+1)+\log\mu+1).$$

И теоретические исследования, и практическое применение БРД свидетельствуют, что наилучшими как по статистической надёжности, так и согласно колмогоровскому подходу и принципу MDL в подавляющем большинстве случаев являются БРД с минимальным числом листьев. Однако вычислительная сложность задачи минимизации *БРД*-индуктора с минимальным числом листьев не позволяет рассчитывать на использование точных алгоритмов. Кроме этого возникает ряд дополнительных эвристических соображений, которые учитываются разработчиками процедур синтеза.

Главным элементом алгоритмов синтеза БРД по заданным бинарным обучающим таблицам является выбор на каждом шаге переменной для ветвления или, что равносильно, для разбиения некоторого интервала N_t (на первом шаге ветвления – всего куба B^n как интервала ранга 0). Интервал разбивается на два интервала N_t^1 и N_t^2 таких, что $N_t^1 \cup N_t^2 = N_t;\ \ N_t^1 \cap N_t^2 = \emptyset;$ при условии, что в интервале N_t непременно содержатся точки различных классов.

Обозначим k – номер переменной, выбранной для разбиения интервала N_t. Поскольку именно выбранная переменная определяет разбиение, будем обозначать интервалы $N_t^1(k)$ и $N_t^2(k)$. Определим $A(k) = N_t^1(k) \cap T_{l,n}$ – множество точек из обучающей выборки $T_{l,n} = \{\tilde{x}_j : \tilde{x}_j \in (\tilde{x}_j, \alpha_j)\}_{j=1}^{l}$, попавших в интервал $N_t^1(k)$; $B(k) = N_t^2(k) \cap T_{l,n}$ – множество точек из обучающей выборки, попавших в интервал $N_t^2(k)$. Пусть $|A(k)| = m_1(k);\ |B(k)| = m_2(k)$.

Будем говорить, что некоторый предикат $S(k)$ является критерием ветвления, если переменная x_k выбирается для ветвления в том и только в том случае, когда этот предикат принимает истинное (единичное) значение. Критерии ветвления могут быть различными.

Рассмотрим следующие *критерии ветвления* – условия, определяющие выбор для ветвления переменной с номером k.

Критерий $\boldsymbol{S_2}$ (*полной отделимости*). $S_2(k) = 1$, если множество $A(k)$ содержит точки только одного класса, множество $B(k)$ содержит точки только одного класса и классы наборов в $A(k)$ и $B(k)$ различны; иначе – $S_2(k) = 0$.

Критерий S_1 [14] (*частичной отделимости*). $S_1(k)=1$, если множество $A(k)$ содержит точки только одного класса или множество $B(k)$ содержит точки только одного класса; иначе – $S_1(k)=0$. Легко видеть, что событие «$S_2(k)=1$» влечет событие «$S_1(k)=1$».

Критерий D [14] (*равномерного разделения пар*). Пусть $T_{m_t,n}=T_{l,n}\cap N_t$ – подмножество точек обучающей выборки, попавших в интервал N_t, а $K_t(k)$ – число пар наборов разных классов в подмножестве $T_{m_t,n}$, которые различаются по переменной x_k. Если $k^*=\arg\max\limits_k K_t(k)$ и для разбиения используется переменная x_{k^*}, то будем говорить, что для ветвления используется критерий D.

Свойства критерия D.

1° Пусть число точек, подлежащих разбиению, зафиксировано. Пусть возможны любые размещения этих точек и их пометок номерами классов в разбиваемом интервале N_t.

Утверждение 5.1. Для того, чтобы при заданной обучающей выборке и заданном интервале, подлежащем разбиению, величина $D(k^*)=\max\limits_k K_t(k)$ имела максимальное возможное значение, необходимо и достаточно одновременное выполнение двух следующих условий:

(i) Класс любой точки множества $A(k^*)$ отличен от класса любой точки множества $B(k^*)$.

(ii) Разбиение является равномерным: $m_1(k^*)=m_2(k^*)$ при четном значении $m_{1,2}$ или $|m_1(k^*)-m_2(k^*)|=1$ при нечетном $m_{1,2}$, где $m_{1,2}=m_1(k^*)+m_2(k^*)$ – число точек обучающей выборки, попавших в разбиваемый интервал N_t.

Достаточность. Предположим, что величина $D(k)$ может быть увеличена. Следовательно, можно увеличить число пар точек разных классов в интервалах разбиения (при зафиксированной величине $m_{1,2}$). Тогда: либо существуют точки одного и того же класса во множестве $A(k)$ (или в $B(k)$), и тогда такие точки можно переносить в соседний интервал разбиения $N_t^2(k)$ (или $N_t^1(k)$); либо, если условие (i) выполнено, величина $m_1(k)(m_{1,2}-m_1(k))$ не достигает максимума. Но тогда не выполняется условие (ii).

Необходимость. Если в разбиваемом интервале число пар наборов разных классов, которые различаются по переменной x_k, является максимально возможным, то наборов одного и того же класса ни во множестве $A(k)$, ни во множестве $B(k)$ быть не может (i). При этом условие (ii) является необходимым условием экстремума при целочисленных величинах $m_1(k)$ и $m_2(k)$.

2° Критерий D может применяться в случаях любых признаковых пространств и любых разделяющих предикатах.

Критерий DKM (Dietterich, Kearns, Mansour). Этот критерий был предложен в [50], и был рассчитан на случай двух классов. Если в двух интервалах разбиения $N_t^1(k)$ и $N_t^2(k)$ соответственно s_{11} точек первого класса и s_{22} точек второго класса, то $DKM(k) = 2\sqrt{\frac{s_{11}s_{22}}{m_{1,2}}} = 2\sqrt{\hat{p}_{11}\hat{p}_{22}}$. Здесь $\hat{p}_{11}$ и $\hat{p}_{22}$ - оценки вероятностей появления точек первого класса в интервале $N_t^1(k)$ и второго класса – в интервале $N_t^2(k)$. В работе показано, что использование критерия *DKM* в задачах синтеза *БРД* предпочтительнее, чем использование энтропийного критерия E и критерия Джини G (см. ниже).

Свойства критерия DKM.

1° $DKM(k) = 1$, если в каждом из интервалов разбиения содержатся точки только одного класса.

2° Критерий DKM обладает таким же свойством равномерности, как и критерий D.

3° Критерий D обладает преимуществом перед критерием *DKM*: может использоваться при числе классов, большем двух.

Критерий TWO (Twoing).

Пусть для случая двух классов, как обозначалось выше, в интервале разбиения $N_t^1(k)$ содержатся s_{11} точек первого и s_{21} точек второго классов, а в интервале $N_t^2(k)$ - s_{12} точек первого и s_{22} точек второго класса; всего в интервале $N_t^1(k)$ содержится $m_1 = s_{11} + s_{21}$ точек из обучающей выборки, а в интервале $N_t^2(k)$ – $m_2 = s_{12} + s_{22}$ точек. Разбиению подлежат $m_{1,2} = m_1 + m_2$ точек. Тогда критерий *Twoing* определяется выражением

$$TWO = \frac{m_1 m_2}{m_{1,2}^2}\left(\left|\frac{s_{11}}{m_1} - \frac{s_{12}}{m_2}\right| + \left|\frac{s_{21}}{m_1} - \frac{s_{22}}{m_2}\right| \right)^2.$$

$$TWO = \hat{p}\hat{q}(\ |\hat{p}_{11} - \hat{p}_{12}| + |\hat{p}_{21} - \hat{p}_{22}|\ )^2,$$

где $\hat{p} = \frac{m_1}{m_{1,2}}$, $\hat{q} = \frac{m_2}{m_{1,2}}$, $\hat{p} + \hat{q} = 1$. При безошибочном разделении $s_{12} = s_{21} = 0$ и тогда $TWO = 4\hat{p}\hat{q}$. Если при этом имеет место равномерное распределение точек выборки по интервалам разбиения – т.е. $\hat{p} = \hat{q} = \frac{1}{2}$, то $TWO = 1$.

Свойства критерия TWO в основном совпадают со свойствами критерия DKM.

***Критерий* Ω.** [14] Пусть при разбиении по переменной x_k в интервале $N_t^1(k)$ оказались точки $J_1(k)$ разных классов, а в интервале $N_t^2(k)$ – точки $J_2(k)$ разных классов. Обозначим $\Omega(k^*) = \min_k(J_1(k) + J_2(k))$. Будем говорить, что используется критерий Ω, если для разбиения выбирается переменная k^* и при этом классы хотя бы одной пары точек из разных интервалов разбиения $N_t^1(k^*)$ и $N_t^2(k^*)$ различны.

***Свойства критерия* Ω.**

1° Имеет место эквивалентность $(\Omega(k) = 2) \Leftrightarrow (S_2(k) = 1)$.

2° Если значение $\Omega(k)$ равно числу q классов объектов в исходной задаче, то разбиение по переменной x_k приводит к тому, что объекты каждого из классов попадут только в один из двух интервалов разбиения. Назовем это свойство *чувствительностью к иерархическому разделению классов.*

Критерий E (*энтропийный*). Пусть $s_{i,j}$ – количество точек класса i в интервалах разбиения $N_t^j(k)$, $j = 1,2$, полученных при разбиении интервала N_t по переменной x_k. В общем случае $m_{1,2}$ точек обучающей выборки распределятся по двум полученным в результате разбиения интервалам так, как показано на рис. 5.4 (где для наглядности полагается, что число классов в выборке равно двум).

$N_t^1(k)$	$N_t^2(k)$
содержит $m_1(k)$ точек;	содержит $m_2(k)$ точек;
из них $s_{1,1}$ точек – класса 1	из них $s_{1,2}$ точек – класса 1
и $s_{2,1}$ точек – класса 2.	и $s_{2,2}$ точек – класса 2.

Рис.5.4. Распределение точек по интервалам

Вероятность того, что произвольный объект из $N_t^j(k)$ принадлежит классу i, может быть оценена как $\hat{p}_{i,j} = s_{i,j} / m_j(k)$, где $m_j(k)$ – число точек выборки, попавших в интервал $N_t^j(k)$. Заметим, что эта оценка условной вероятности $\hat{p}_{i,j}$ – смещенная.

Оценкой энтропии интервала $N_t^j(k)$ будет $I_j(k) = -\sum_i \hat{p}_{i,j} \log_2 \hat{p}_{i,j}$. А оценкой средней энтропии по двум интервалам $N_t^1(k)$ и $N_t^2(k)$ будет величина

$$E(k) = \frac{m_1(k)}{m_{1,2}} I_1(k) + \frac{m_2(k)}{m_{1,2}} I_2(k),$$

поскольку $\dfrac{m_j(k)}{m_{1,2}}$ является оценкой вероятностной меры интервала $N_t^j(k)$, и тогда $E(k)$– среднестатистическая оценка.

Критерий E выбора переменной для разбиения (ветвления) интервала N_t состоит в выборе переменной с номером

$$k^* = \arg\min_k E(k),$$

что соответствует минимизации неопределенности в результате разбиения текущего интервала.

Свойства критерия E.

1° Энтропийный критерий E не чувствителен к равномерности разбиения – может давать одинаковые значения в случаях, когда количество объектов в интервалах равно и когда различается вплоть до 1 и $m_{1,2} - 1$.

Действительно, если в каком либо интервале j содержатся объекты только одного класса i, то оценка вероятности $\hat{p}_{i,j} = s_{i,j} / m_j(k)$ будет равна единице независимо от величины $m_j(k)$. В частности, рассмотрим две таблицы на рис. 5.5.

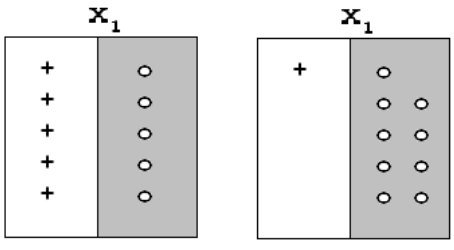

Рис. 5.5. Неравномерное распределение объектов по интервалам

И в одном, и в другом случае критерий E принимает нулевое значение. Заметим, что критерий D в этих случаях примет различные значения: 25 и 9.

2° Критерий E нечувствителен к иерархическому разделению классов. Это свойство иллюстрируется следующим рис. 5.6.

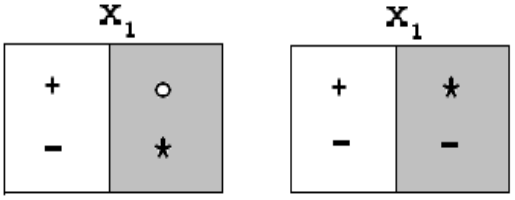

Рис. 5.6. Два случая, когда значения критерия E совпадают и равны 1.

Критерий информационного выигрыша (*Information gain, IGain*) [62,63] рассчитан на выбор переменной для ветвления на основе энтроприйного подхода. Критерий усовершенствован так, чтобы оценивать средний прирост информации (выигрыш) от выполнения шага ветвления.

Начальное среднее количество информации, необходимое для определения класса произвольного объекта определяется как

$$Info(T) = -\sum_{j=1}^{q} \frac{s_j}{l} \log \frac{s_j}{l} = -\sum_{j=1}^{q} \hat{p}_j \log \hat{p}_j,$$

где T – обучающая выборка; l – число примеров в обучающей выборке; q – число различных классов (значений целевой переменной); s_j – число точек из обучающей выборки, помеченных классом j; $\hat{p}_j$ – оценка вероятности появления класса j, вычисленная по данной обучающей выборке.

Критерий выбора переменной x_k – *по максимуму информационного выигрыша* $Gain(k) = Info(T) - Info(k) = Info(T) - E(k)$, где $E(k)$ – величина определенного выше критерия E – есть средняя энтропия по интервалам разбиения при выборе для ветвления переменной x_k.

Критерий MEE (***M****inimum* ***E****rror* ***E****ntropy*)[56].

Сначала рассмотрим случай двух классов – ω_1 и ω_2. Пусть x_k – переменная-кандидат для ветвления, а ω_1 – номер класса – кандидат для пометки интервала разбиения $N_t^1(k)$ (левой ветви) в случае разбиения по переменной x_k. Тогда правая ветвь (и интервал $N_t^2(k)$) предположительно помечается оставшимся классом – ω_2. Если считать такое ветвление правильным, то любая точка из обучающей выборки, попадающая в интервал

$N_t^1(k)$ и принадлежащая классу ω_2, будет классифицироваться неверно. Обозначим соответственно число таких ошибочных точек в интервалах $N_t^1(k)$ и $N_t^2(k)$ как r_{12} и r_{21}. Тогда оценками вероятностей ошибок типа «перепутывания классов» в разбиваемом интервале $N_t = N_t^1 \cup N_t^2$ будут $\hat{P}_{12} = \frac{r_{12}}{m_{1,2}}$ и $\hat{P}_{21} = \frac{r_{21}}{m_{1,2}}$, где $m_{1,2}$ - число точек выборки, попадающих в интервал N_t. Величина $1 - \hat{P}_{12} - \hat{P}_{21}$ будет оценкой вероятности правильного вычисления классов вершиной с распознавателем x_k и метками ω_1 и ω_2. Числовая оценка для рассматриваемого критерия *MEE* задается формулой

$$EE = EE(N_t, k, \hat{P}_{12}, \hat{P}_{21}) = -\hat{P}_{12}\log\hat{P}_{12} - \hat{P}_{21}\log\hat{P}_{21} - (1-\hat{P}_{12}-\hat{P}_{21})\ln(1-\hat{P}_{12}-\hat{P}_{21})$$

и называется энтропией ошибки. Правило ветвления *MEE* состоит в выборе для разбиения допустимого интервала N_t и допустимой переменной с таким номером k, чтобы достигалось минимальное значение энтропии ошибки

$$\min_{N_t, k} EE(N_t, k, \hat{P}_{12}, \hat{P}_{21}).$$

Свойства критерия MEE.

1° Минимальное значение оценки $EE = 0$ имеет место в случае правильной классификации вершиной всех точек выборки, попавших в интервал разбиения. Максимальное – $EE = 1$ имеет место при «полном перепутывании» точек в интервалах разбиения, когда $\hat{P}_{12} = \hat{P}_{21} = \frac{1}{2}$.

2° *С ростом «перепутывания» классов оценка* EE *возрастает. Заметим, что в этом случае и значение критерия* Ω *возрастает.*

3° В случае частичной отделимости, например, при $\hat{P}_{12} = 0$, если при этом $\hat{P}_{21} = \frac{1}{2}$, вычисления также дают $EE = 1$. Поэтому критерий *MEE* иногда может не различать случаи частичной и полной разделимости классов.

Критерий G (*основанный на индексе Джини*). Индекс Джини интервала $N_t^j(k)$ равен $g(N_t^j(k)) = 1 - \sum_i \hat{p}_{i,j}^2 = 1 - \sum_i (s_{i,j}/m_j(k))^2$. Суммируются квадраты оценок условных вероятностей всех классов в данном интервале. Если в интервале содержатся точки только одного класса, то

его индекс достигает минимального значения, равного нулю. Критерий G для ветвления определяется по формуле

$$G(k) = g(N_t^1(k)) + g(N_t^2(k)).$$

Выбор переменной осуществляется по правилу $k^* = \arg\min_k G(k)$.

Свойства критерия G.

1° Если в интервале содержатся точки только одного класса, то его индекс достигает минимального значения, равного нулю. Поэтому критерий G определяет частичную отделимость.

2° $(G(k) = 0) \Leftrightarrow (S_2(k) = 1)$, что означает способность критерия G определять полную отделимость.

В работах [65, с.7, 68] показано, что применение критерия Джини может привести к неразличению иерархической отделимости классов, и приведен пример (рис. 5.7). На рис. 5.7 представлены два случая разбиений. Случай A соответствует полной отделимости двух пар классов. Но по критерию Джини более предпочтительным оказывается разбиение B.

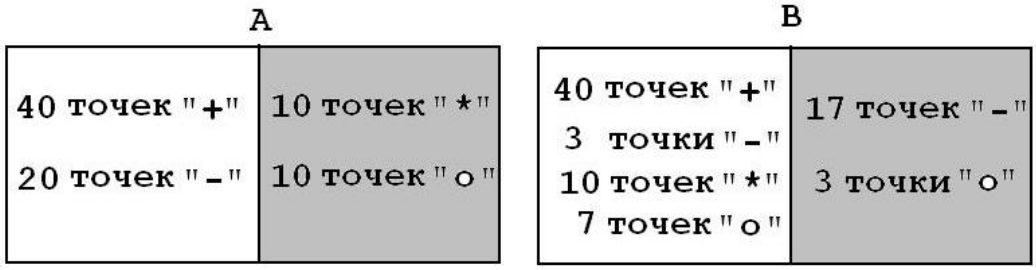

Рис. 5.7. Критерий Джини может не различать иерархическую отделимость

Сравним результаты использования различных критериев.

Пример. Дан интервал размерности 5, в котором содержатся 9 точек трех различных классов, обозначенных метками $+, -, *$ (рис. 5.8).

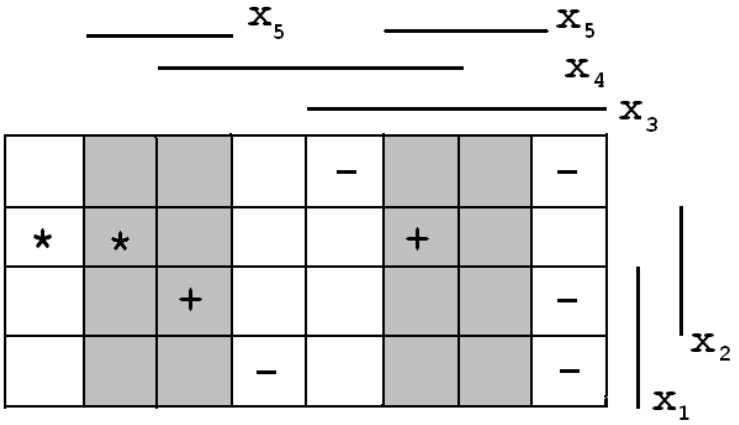

Рис. 5.8.

Значения критериев ветвления при выборе переменных $x_1,\ldots,x_5$ представлены на рис. 5.9.

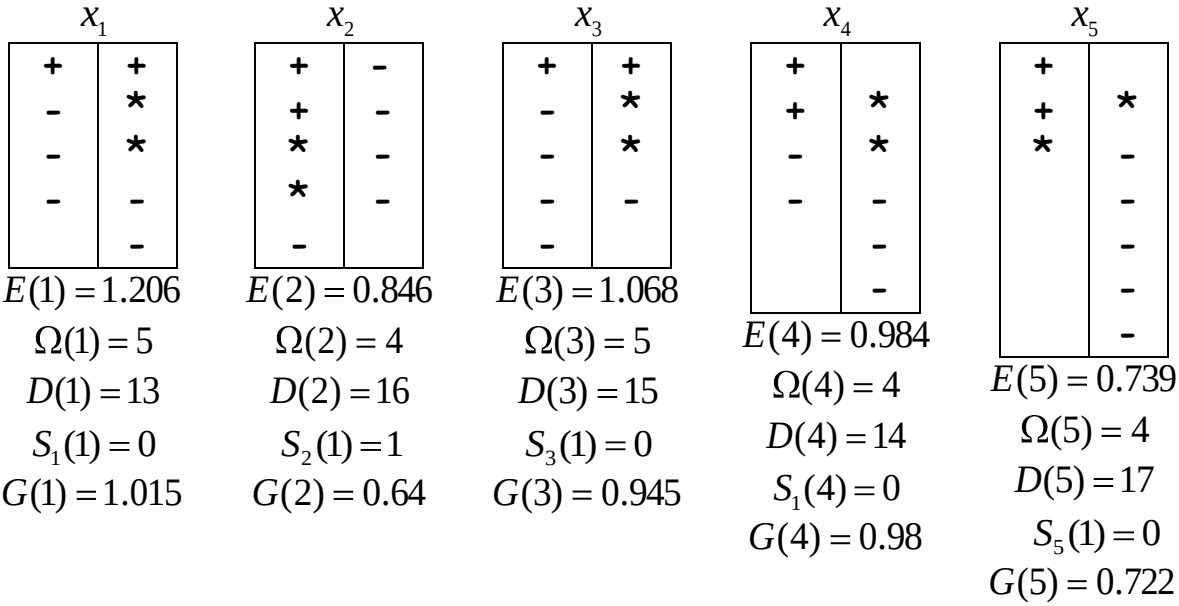

Рис. 5.9. Значения критериев в разных случаях распределения точек в интервале.

Сравнение значений критериев показывает, что они, за исключением критериев S_1 и G, согласованы: определяют один и тот же выбор переменной – x_5. Критерии S_1 и G, в свою очередь, согласованы друг с другом и выделяют случай частичной отделимости. Если упорядочить переменные по убыванию значения критерия E, то значения критерия D, как видно из таблицы 5.1 и рис. 5.10, будут возрастать, но монотонность роста нарушается: для переменной x_3 увеличенное значение $D(3)=15$ объясняется большей «чувствительностью» критерия D к частичной отделимости по сравнению с критерием E.

Таблица 5.1

Критерии	x_1	x_3	x_4	x_2	x_5
E	1.206	1.068	0.984	0.846	0.739
D	13	15	14	16	17

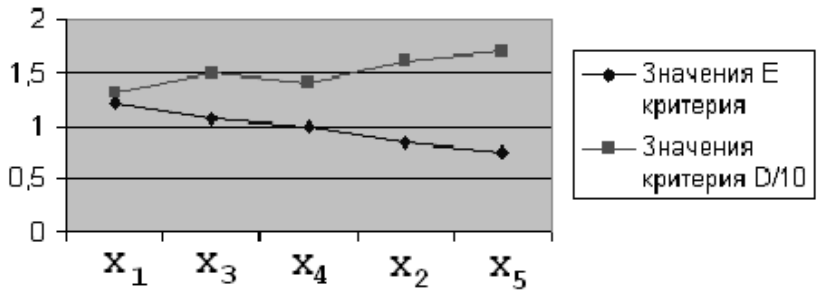

Рис. 5.10. Сравнение критериев D и E

Пример. Дан интервал размерности 4, в котором содержатся 10 точек пяти различных классов (рис. 5.11).

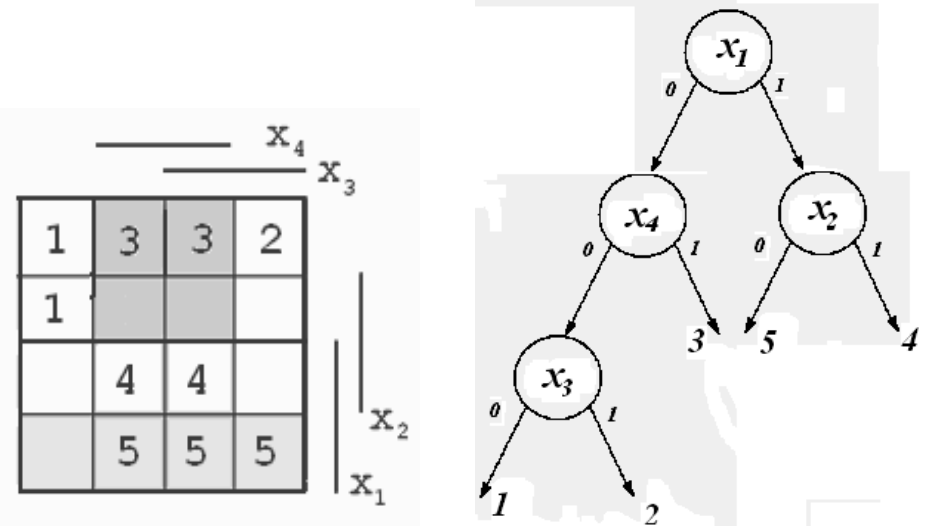

Рис. 5.11. Распределение точек и оптимальное дерево из примера

В этом примере на всех шагах синтеза значения критериям E и D совпадают. Приведем их значения только для первого шага разбиения (табл. 5.2).

Таблица 5.2.

Номер переменной / *Значение критерия*	x_1	x_2	x_3	x_4
D	25	20	21	22
E	1.246	1.565	1.922	1.551

Легко видеть, что в случае, когда в каждом интервале разбиения будут содержаться точки только одного класса, критерий E будет давать нулевое значение. Вследствие утверждения 5.1, в этом случае выбор по критериям E и D всегда будет совпадать.

Согласно многократным экспериментам по применению различных критериев ветвления, в работе [56] представлены сравнительные результаты. В частности, сравнивалось число листьев в полученных в результате синтеза решающих деревьях. Оценивание производилось на 36 реальных задачах. В таблице 5.3 приведены данные: сколько раз использование каждого из пяти сравниваемых критериев приводило к получению деревьев с

Таблица 5.3.

Алгоритмы	*Gini*	*Info Gain*	*Twoing*	*C4.5*	*MEE*
Число выигрышей	11	9	8	1	18
Число проигрышей	4	3	3	24	7

наименьшим по сравнению со всеми другими алгоритмами листьев (лучшие результаты) и наибольшим числом листьев (худшие результаты).

Данные, приведенные в таблице 5.3, подтверждают, прежде всего, что *нельзя указать критерий ветвления, который дает лучшие результаты во всех случаях – при любых допустимых входных данных*. Но, тем не менее, согласно таблице 5.3, алгоритм *MEE* побеждает как минимум вдвое чаще других. Несколько неожиданным представляется то, что по результатам рассматриваемых экспериментов алгоритм *C4.5*, который очень часто используют в приложениях, оказался худшим.

В работе [14] проводились экспериментальные исследования алгоритмов синтеза *БРД*. В статистических экспериментах точки – вершины единичного n-мерного куба ($n = 25$) генерировались равновероятно; также равновероятно каждой сгенерированной точке присваивался номер одного из заданного числа классов (таблица 5.4).

Таблица 5.4. Статистические испытания трех алгоритмов ветвления

Алгоритмы	*Среднее по 15 экспериментам число листьев*		
	25 признаков 5 классов 50 объектов в выборке	*25 признаков 2 класса 50 объектов в выборке*	*25 признаков 5 классов 100 объектов в выборке*
LISTBB	23,1	13,3	44,7
LISTD	24,5	14,1	46,7
LISTB	44,9	34,9	—

Алгоритм *LISTBB*, показавший лучшие в этом эксперименте результаты (см. ниже), является гибридной процедурой ситуативного выбора критерия ветвления, зависящего от начального значения критерия Ω и наличия полной или частичной отделимости. Алгоритм *LISTD* использует только критерий *D*; алгоритм *LISTB* реализует произвольный порядок выбора признаков для ветвления, полученный в результате случайной генерации.

Алгоритм *LISTBB* в первую очередь вычисляет значение критерия Ω, который логически наиболее близок к критерию *MEE*.

5.3. Алгоритмы синтеза бинарных деревьев решений по прецедентной информации

Алгоритм CLS (***C****oncept* ***L****earning* ***S****ystem*). Это – классический алгоритм Ханта [47], который явился основой для подавляющего большинства разработок в области синтеза решающих деревьев в процессе машинного обучения. Алгоритм *CLS* циклически разбивает точки обучающей выборки на подмножества в соответствии со значениями переменных, имеющих

наибольшую разделяющую способность. Разбиение заканчивается, когда в подмножестве оказываются объекты лишь одного класса. В ходе процесса разбиений формируется дерево решений.

Алгоритм ID3 [63] был предложен Россом Куинланом в 1986 г. и основывался на алгоритме Ханта, учеником которого был Куинлан. Алгоритм *ID3* был основан на использовании критерия информационного выигрыша для выбора вершины и переменной для ветвления. Синтез решающего дерева завершался либо в случае достижения его корректности относительно выборки, либо когда ветвление ни в одной некорректной вершине не приводило к увеличению информационного выигрыша.

Алгоритм C4.5 [62]. Этот алгоритм явился развитием идей, реализованных в *ID3*, был разработан Р. Куинланом в 1993 г. и использовал отношение выигрыша (*gain ratio*) в качестве критерия ветвления. Процесс синтеза (добавления вершин) в алгоритме *C4.5* прекращался, когда число точек для разбиения становилось меньше некоторого порога.

Алгоритм CART [42]. Аббревиатура *CART* взята из названия «***C****lassification* ***A****nd* ***R****egression* ***T****rees*». Алгоритм предназначен для синтеза бинарных решающих деревьев. Для ветвления используется критерий *Twoing*. *CART* рассчитан, кроме прочего, на построение деревьев регрессии, в корневых вершинах которых вместо меток классов помещаются вещественные числа. В этих случаях ветвление осуществляется по минимуму среднеквадратической ошибки.

Алгоритм CHAID [49] (***CH****isquare–****A****utomatic–****I****nteraction–****D****etection* – *интерактивное обнаружение на основе критерия* χ^2). Применение методов прикладной статистики для реализации ветвления при синтезе решающих деревьев получило развитие в начале 70-х годов. *CHAID* являлся «развитием» алгоритма *AID* (***A****utomatic* ***I****ntegration* ***D****etection*) [66] и был ориентирован на выбор групп значений переменных для ветвления следующим образом.

Для каждой переменной находились такие пары ее значений, которые незначительно изменялись при изменении целевого признака во входных данных. В зависимости от типов переменных-признаков незначительность такого изменения оценивалась разными статистическими критериями: Пирсона χ^2 – для номинальных переменных, Фишера – для непрерывных переменных, критерием правдоподобия – для ранговых переменных. Статистически значимо неразличимые пары значений переменных объединялись в однородную группу значений, и процесс повторялся, пока находились «неразличимые» пары.

Для ветвления (построения текущей вершины) интерактивно выбиралась такая переменная, которая разделяла группы однородных значений. Синтез дерева прекращался при выполнении любого из следующих условий:

1º. Достижение максимальной заданной глубины дерева;

2º. Число точек выборки для дальнейшего разбиения в терминальных вершинах или в любой получаемой дочерней вершине меньше заданного значения.

При этом пропущенные значения переменных (если таковые имелись в начальной информации) выделялись в отдельные группы значений.

Алгоритм QUEST (*Quick Unbiased Efficient Statistical Tree*) [53]. Для осуществления ветвления связь между каждой входной переменной и целевой переменной оценивалась на основе F-критерия *ANOVA* (*Analysis Of Variances*) или теста Левене [51] однородности дисперсий для порядковых или непрерывных переменных или критерия χ^2 для номинальных переменных. Для многоклассовых целевых переменных применялся кластерный анализ для объединения в два «сверхкласса». Для ветвления использовалась переменная, имеющая наибольшую оценку статистической связи с целевым признаком.

Для подрезания деревьев использовался скользящий контроль, применение которого давало основание авторам говорить о несмещенности статистических оценок.

Здесь отмечена только часть особенностей алгоритма *QUEST*, касающихся выбора переменных для ветвления. *QUEST* можно классифицировать как сложную систему анализа данных, дающую возможность исследовать различные варианты предикторов и применять оптимизационные процедуры для их выбора.

Алгоритм SLIQ (***S****upervised* ***L****earning* ***I****n* ***Q****UEST*) [57]. Этот алгоритм рассчитан на применение в области Data Mining и работу с большими объемами исходных данных. Для ветвления используется индекс Джини и специальные методы быстрой сортировки.

Алгоритм PUBLIC (***Pru****ning and* ***B****uil****d****ing* ***I****ntegrate* ***C****lassifier*) [64]. *Выбор порогового значения переменной для ветвления осуществляется на основе построения гистограмм распределения классов.* Каждая точка на гистограмме, рассматриваемая как кандидат для определения порога ветвления, оценивается энтропийным критерием, который используется для окончательного выбора переменной и порога.

Алгоритмы – CAL5 [58]***, FACT*** (ранняя версия алгоритма *QUEST*)***, LMDT*** [43]***, T1*** [46]***, MARS*** [45] и многие другие – принципиально не отличаются от рассмотренных выше.

Ниже приведена таблица 5.5, в которой приведены данные об использовании алгоритмов синтеза деревьев решений в медицинских задачах.

Таблица 5.5. Частота использования алгоритмов в медицинских приложениях [67]

Алгоритм	*Частота использования* (%)
ID3	68
C4.5	54.55
CART	40.9
SLIQ	27.27
Public	13.6
CLS	9

5. 4. Гибридный алгоритм LISTBB, основанный на использовании совокупности критериев ветвления

Критерии ветвления S_2, S_1, Z_1, D, Ω, разработанные автором еще в 1979-80 годах, легли в основу алгоритма *LISTBB*. Название *LISTBB* алгоритма синтеза *БРД* объясняется тем, что его первая реализация на автокоде компьютера $M222$ была осуществлена на основе спискового представления дерева (*LIST*); *Branching* (B) – обозначало ветвление, а *Boolean* – второе B – обозначало случай булевых переменных. Алгоритм *LISTBB* и его модификации *LISTD*, *LISTBB*(*P*) многократно применялись при решении практических задач и использовались при создании программных комплексов *РАДИУС*-222, *ТРИОЛЬ*, *ИНТМАН* [13,17,18]. Главная особенность алгоритма *LISTBB* состоит в том, что он «заточен» именно на минимизацию отыскиваемого *БРД* индуктора по числу листьев.

Алгоритм LISTBB ***выбора переменной для ветвления (разбиения интервала)***

1^o Вычислить множество номеров переменных, для которых достигается минимум критерия Ω:

$$\tilde{\kappa}_\Omega = \{k_0 : k_0 = \arg\min_k \Omega(k)\},$$

где k пробегает номера свободных переменных разбиваемого интервала.

2^o Если $|\tilde{\kappa}_\Omega| = 1$, т.е. минимум критерия Ω достигается только для одной переменной, то выбрать эту переменную k_0 и завершить алгоритм выбора.

3^o Если $\min_k \Omega(k) = q$, где q – исходное число классов, то выбрать для разбиения любую переменную k^* такую, что

$$k^* = \arg\max_{k \in \tilde{\kappa}_\Omega} D(k),$$

и завершить алгоритм выбора.

4^o Если частичная отделимость не имеет места, т.е. $\forall k \in \tilde{\kappa}_\Omega(S_1(k) = 0)$, то выбрать для разбиения любую переменную k^* такую, что $k^* = \arg\max_{k \in \tilde{\kappa}_\Omega} D(k)$, и завершить алгоритм выбора.

5^o Если частичная отделимость имеет место, то выбрать для разбиения любую переменную k^* по максимуму частичной отделимости, т.е. такую, что $k^* = \arg\max_{k \in \tilde{\kappa}_\Omega} Z_1(k)$, и завершить алгоритм выбора. □

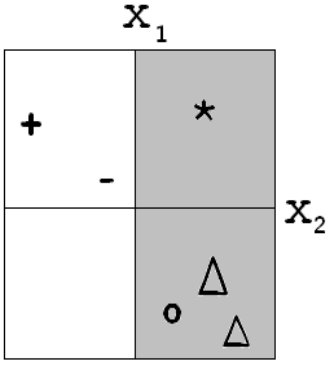

Рис. 5.12

Шаги $1^o - 3^o$ алгоритма *LISTBB* «нацелены» на «улавливание» иерархической разделимости по классам. Для пояснения шага 3^o можно привести следующий пример (рис. 5.12). Шесть точек в разбиваемой области принадлежат пяти различным классам, которые обозначены символами **+, -,*, o, Δ.** Разбиение по переменным, условно обозначенным как x_1 и x_2, дает $\Omega(1) = \Omega(2) = 5;\ D(1) = 8$, но $D(2) = 9$. Из этого примера следует, что при равных значениях критерия Ω для двух разных переменных, значение критерия D для этих переменных в то же время может отличаться.

Шаги $4^o - 5^o$ алгоритма «нацелены» на «улавливание» максимальной частичной отделимости. Критерий D используется в алгоритме *LISTBB* в случаях, когда нет возможности реализовать иерархическое разделение классов или частичную отделимость.

В процессе построения *БРД* выполняются шаги ветвления, и поэтому число листьев синтезируемого дерева растет. При этом существует нижняя оценка числа листьев *БРД*, которое получится в итоге процедуры синтеза. В зависимости от выбора стратегий ветвления и по мере приближения к завершению синтеза эта нижняя оценка может изменяться. Поэтому будем назвать ее *текущей*.

Утверждение 5.2. Текущей оценкой снизу для числа листьев синтезируемого корректного *БРД* является величина $\mu_t + \Omega(k^*) - 1$, где μ_t - текущее число листьев *БРД* до выполнения шага ветвления очередного интервала, $\Omega(k^*)$ – минимальное значение критерия Ω, достигаемое при выборе для ветвления переменной x_k.

Доказательство. Действительно, на шаге t построена некоторая часть дерева, концевые вершины которого (листья) могут содержать объ-

екты различных классов и соответствовать некоторым интервалам. Пусть построенная часть дерева имеет μ_t листьев. Интервал N_t, соответствующий одному такому листу, разбивается на два интервала, поэтому к $\mu_t - 1$ оставшимся листьям будут добавлено не менее $\Omega(k^*)$ листьев, поскольку все точки различных классов в интервалах разбиения $N_t^1(k^*)$ и $N_t^2(k^*)$ для достижения корректности *БРД* должны быть разделены.

Замечание. Поскольку $q \le \Omega(k) \le 2q$, где q – изначальное число классов в обучающей выборке, то при малых величинах q, равных двум или трем, полезность оценки, полученной в утверждении 5.2, небольшая. Но с увеличением значения q эта оценка может действительно стать полезной.

Утверждение 5.3. При выборе в алгоритме *LISTBB* переменной для ветвления согласно шагу 5^o имеет место оценка

$$\min_k \Omega(k) - 1 \le \Delta\mu_t \le m_{1,2} - Z_1(k^*) + 1,$$

где $\Delta\mu_t$ - приращение числа листьев *БРД* после выполнения ветвления по переменной x_{k^*}.

Доказательство. Левая часть неравенства доказана в предыдущем утверждении, а правая часть неравенства становится очевидной, если заметить, что разделению подлежат $m_{1,2}$ точек разбиваемого интервала, и в худшем случае пришлось бы отделять каждую точку отдельным листом дерева. Заметим, что на шаге 5^o для ветвления выбирается переменная с номером $k^* \in \tilde{\kappa}_\Omega = \{k_0 : k_0 = \arg\min_k \Omega(k)\}$. Но при частичной отделимости $Z_1(k^*)$ точек появится один интервал, не подлежащий дальнейшему дроблению, и к синтезируемому дереву добавится один соответствующий лист. А второй интервал разбиения будет содержать $m_{1,2} - Z_1(k^*)$ точек, которые в худшем случае в дальнейшем будут разделены интервалами по одной точке в каждом. □

Согласно утверждениям 5.2 и 5.3, *алгоритм LISTBB*, являясь эвристическим, *направлен на выбор переменной для ветвления так, чтобы минимизировать и нижнюю, и верхнюю оценку приращения число листьев*. Но его «пристрастие» к частичной отделимости может приводить к случаям, когда $Z_1(k^*)$ слишком мало, например, $Z_1(k^*)$=1, и тогда выигрыш от выбора переменной для ветвления по частичной отделимости может оказаться невыгодным.

Параметрический вариант алгоритма $LISTBB(p)$ содержит параметр p, который определяет ветвление в пункте 5^o следующим образом:

5^o Если частичная отделимость имеет место и $Z_1(k^*) > p$, то выбрать для разбиения любую переменную k^* по максимуму частичной отделимости: такую, что $k^* = \arg\max_{k \in \tilde{\tilde{\kappa}}_\Omega} Z_1(k)$, и завершить алгоритм выбора; иначе – выбрать любую переменную k^* такую, что $k^* = \arg\max_{k \in \tilde{\tilde{\kappa}}_\Omega} D(k)$, и завершить алгоритм выбора.

5.5. Правила остановки при обучении и подрезание решающих деревьев

«(Правила простые совсем;
всего – семь).
1. Берутся классики, свёртываются в трубку
и пропускаются через мясорубку.
2. Что получится, то
откидывается на решето…»
В. Маяковский

Решающее дерево называют *корректным* (относительно данной обучающей выборки), если все примеры этой выборки классифицируется деревом правильно. Разбиение пространства признаков, порождаемое корректным решающим деревом таково, что каждое *терминальное* множество, входящее в полученное разбиение, содержит точки, принадлежащие только одному классу. Терминальные множества соответствуют листьям дерева. Каждое из них наследует номер класса, которым помечен соответствующий лист.

***Правило* 1**. Процесс синтеза решающего дерева (ветвление) продолжается до тех пор, пока оно не станет корректным. Это возможно только в том случае, когда предикатные описания всех пар объектов обучающей выборки, принадлежащих различным классом, различны.

***Правило* 2**. Процесс синтеза прекращается, когда число листьев достигает заданной пороговой величины.

***Правило* 3**. Процесс синтеза прекращается, когда информационный выигрыш (*Information gain*) невозможно увеличить за счет замены ни одного листа новой внутренней вершиной.

***Правило* 4**. Процесс синтеза прекращается, когда длины всех ветвей достигли заданной величины.

***Правило* 5**. Процесс синтеза прекращается, когда терминальные множества, подлежащие ветвлению, содержат число точек, меньшее заданного порогового значения.

***Правило* 6**. Момент остановки при синтезе дерева определяется на основе принципа минимальной длины описания (*Minimum Description Length*), согласованного с выбором наиболее вероятных гипотез по правилу Байеса. Этот подход соответствует парадигме *Ideal MDL* [69]. Он является одной из формализаций «бритвы Оккама»: *наилучшей гипотезой является та, которая минимизирует сумму длины описания кода гипотезы* (*называемой моделью*) *и длины описания множества данных относительно этой гипотезы*. В рассматриваемом случае кодом модели является бинарное описание решающего дерева (в виде некоторой строки), а описанием данных – бинарное строковое описание некоторой совокупности обучающих примеров. Это правило для случая *БРД* подробно описано в [20].

***Правило* 7**. Остановка на основе теоретической оценки вероятности ошибки происходит тогда, когда добавление любой дополнительной вершины к строящемуся дереву уже не приводит к уменьшению ошибки. Такой подход описан во многих работах, в частности, в [15,20]. □

Последнее правило остановки представляется теоретически наиболее обоснованными.

Любое из перечисленных правил может быть применено с некоторым одним или совокупностью критериев ветвления и дать «новый» алгоритм машинного обучения, основанный на построении дерева решений. Что и наблюдается в многочисленных публикациях, посвященных синтезу эмпирических индукторов рассматриваемого класса.

***Правила подрезания* (*редуцирования*),** как правило, определяют максимально возможную длину ветвей дерева. Если какая-нибудь ветвь имеет длину, больше заданного ограничения, то она укорачивается, и вместо последней вершины ветвления в редуцированной ветви ставится метка класса. Эта метка определяется тем, точек какого класса содержится больше в интервале, соответствующем редуцированной ветви. Редуцированием можно считать также и ограничение числа листьев дерева.

Редуцирование приходится применять, когда попытка синтезировать корректное решающее дерево приводит к его неоправданной сложности.

Будем считать, что набор из l точек $T_{l,n}$ в обучающей выборке состоит из случайно и независимо выбранных из множества $\{0,1\}^n$ векторов, для каждого из которых достоверно указана принадлежность одному из двух $\{0,1\}$ классов; одинаковых векторов (строк) в таблице $T_{l,n}$, принадлежащих разным классам, нет. Такие обучающие таблицы называются корректными и достоверными.

Конъюнктивной закономерностью класса $\omega \in \{0,1\}$ ранга r ($КЗ_r$) называется любая конъюнкция ранга r, обращающаяся в единицу на векторах $\tilde{x} \in T_{l,n}$, заведомо принадлежащих классу ω, и в ноль – на векторах $\tilde{x} \in T_{l,n}$, заведомо принадлежащие классу $\overline{\omega}$. С точки зрения теоретико-множественного подхода, $КЗ_r$ K_r соответствует интервалу N_{K_r} такому, что множество $N_{K_r} \cap T_{l,n}$ содержит точки только одного класса.

Один из подходов к оцениванию эмпирических закономерностей и решающих правил основывается на представлении о закономерности как неслучайности. А. Д. Закревский показал [27], что вероятность $P_{случ}$ того, что в таблице $T_{l,n}$, состоящей из случайно и независимо выбранных булевых векторов, найдётся $КЗ_r$ ранга r, удовлетворяет неравенству

$$P_{случ}(n,l,r) < C_n^r (n-r) 2^{-(l-2^r)} \qquad (5.1)$$

при выполнении условия $l > 2^r$. Неравенство (5.1) позволяет оценить допустимый ранг конъюнктивной закономерности (допустимую длину ветви *БРД*) следующим образом. Потребовав, чтобы $P_{случ}(n,l,r)$ было меньше заданного $\varepsilon > 0$, из уравнения $C_n^r (n-r) 2^{-(l-2^r)} = \varepsilon$ переборным расчетом находят наибольший допустимый ранг r. Ветви, имеющие длину выше r, подлежат редукции. На практике в типичных случаях это приводит к отсечению ветвей дерева таким образом, чтобы они содержали не более семи условных (внутренних) вершин.

Эмпирическое *БРД* с μ листьями определяет сразу μ $КЗ$ рангов $r_1,...,r_j,...,r_\mu$, соответствующих интервалам $N_{r_1},...,N_{r_j},...,N_{r_\mu}$, таким, что $N_{r_1} \cup ... \cup N_{r_\mu} = \{0,1\}^n$. Иначе говоря, *БРД* с μ листьями может являться совокупной эмпирической закономерностью. Используя свойство ортогональности конъюнкций $\{КЗ_{r_j}\}_1^\mu$ (интервалов $\{N_{r_j}\}_1^\mu$) и формулу полной вероятности

$$P_{случ}^\mu = \sum P(N_{r_j}) P(случ./N_{r_j}),$$

легко получить неравенство

$$P_{случ}^\mu < \sum_{j=1}^{\mu} 2^{-r_j} C_n^{r_j} (n-r_j) 2^{-(l-2^{r_j})}, \qquad (5.2)$$

где $P_{случ}^\mu$- вероятность случайного появления в $T_{l,n}$ совокупной закономерности, состоящей из μ $КЗ$ и соответствующей μ-*БРД*.

Из неравенства (5.2) следует

Утверждение 5.4. Вероятность неслучайного обнаружения по таблице $T_{l,n}$ μ-*БРД* закономерности при $l > 2^{r_j}$, $j = 1,2,...,\mu$, больше, чем

$$1 - \sum_{j=1}^{\mu} (n - r_j) C_n^{r_j} 2^{-(l - r_j - 2^{r_j})}. \square \qquad (5.3)$$

Используя (5.3), можно определить число листьев μ^*, при достижении которого процесс синтеза *БРД* должен завершаться. Для этого при заданном $\varepsilon > 0$ переборным расчетом находится наибольшее допустимое число листьев μ^* из уравнения

$$\sum_{j=1}^{\mu} (n - r_j) C_n^{r_j} 2^{-(l - r_j - 2^{r_j})} = \varepsilon.$$

Перебор выполняется по неизвестной переменной $\mu = 2,3,..., \mu^*$.

5.6. Правило Байеса и оптимальная остановка при обучении

Обучение принципиально отличается от настройки на обучающую выборку или её прямой аппроксимации тем, что предполагает организацию *последовательного процесса усложнения решающего правила (гипотезы) с целью достижения его способности к эмпирическому обобщению*. По отношению к самой выборке, способность к обобщению проявляется в том, что часть её примеров, не использованных на некотором этапе обучения, правильно классифицируется сформированным на этом этапе решающим правилом.

В этом смысле показательна обучающая процедура линейной коррекции Розенблатта-Новикова [59], в которой вектор коэффициентов решающего правила – линейного отделителя – *корректируется только при ошибочной классификации очередного обучающего примера*. Коррекция происходит путём использования этого примера – добавления его с регулирующим скорость сходимости коэффициентом к вектору линейного отделителя.

Можно представить процесс обучения как последовательный подбор решающего правила, при котором его сложность постепенно увеличивается, а обобщающая способность оценивается на каждом шаге. Обозначая решающее правило, полученное на шаге t, как h_t, получаем последовательность $h_0, h_1, ..., h_t, ..., h_s$, где s – номер шага остановки. При этом сложность синтезируемого правила обычно не убывает:

$$KP(h_0) \le KP(h_1) \le ... \le KP(h_t) \le ... \le KP(h_s).$$

По мере обучения все большее число примеров классифицируется правильно, поэтому *условная сложность обучающей выборки* – данных D, обозначаемая $KP(D \mid h_t)$, не возрастает:

$$KP(D \mid h_0) \geq KP(D \mid h_1) \geq ... \geq KP(D \mid h_t) \geq ... \geq KP(D \mid h_s).$$

В соответствии с байесовским подходом, следует рассматривать последовательность суммарных сложностей

$$KP(D \mid h_t) + KP(h_t)$$

и минимизировать эту сумму. Поэтому следует остановиться на том шаге t_{opt}, когда указанная суммарная сложность в процессе обучения перестанет убывать. Учитывая, что

$$KP(D \mid h_t) - KP(D \mid h_{t-1}) \leq 0; \quad KP(h_t) - KP(h_{t-1}) \geq 0,$$

условие остановки можно определить следующим образом:

$$t_{opt} = \min t: \; KP(h_t) - KP(h_{t-1}) - (KP(D \mid h_{t-1}) - KP(D \mid h_t)) \geq 0.$$

Это неравенство определяет шаг t_{opt}, на котором приращение $KP(h_t) - KP(h_{t-1})$ сложности синтезируемого решающего правила становится больше, чем величина $KP(D \mid h_{t-1}) - KP(D \mid h_t)$, характеризующая уменьшение условной сложности данных за счет правильной классификации (объяснения) большего числа примеров выборки «растущим» в процессе обучения правилом h_t.

Коррекцию, определяемую на шаге t_{opt}, производить не нужно, и результатом обучения считается правило $h_{t_{opt}-1}$.

Проиллюстрируем этот подход на примере последовательного обучения *БРД*.

Процесс коррекции на одном шаге обучения приводит к увеличению числа внутренних вершин бинарного дерева на единицу, что влечёт увеличение решающих вершин – листьев μ также на единицу: $\mu_t = \mu_{t-1} + 1$. Используя $pVCD$ метод [15], можно получить оценку сложности *БРД* с μ листьями следующим образом. Программирование слова p для декомпрессии любого *БРД* с μ листьями с целью получения оценки сложности $KP(h_\mu)$ основано на представлении каждой из $\mu - 1$ вершин ветвления словом-атомом, состоящим из двух частей:

Код номера переменной или значение решающей функции (0 или 1)	Номер следующего атома в конкатенации или значение решающей функции (0 или 1)

Префикс атома может иметь $n+1$ значение, поскольку 0 и 1 резервируются на значения классифицирующей функции, а значениями $2,3,\ldots, n+1$ кодируются номера признаков $1,2,\ldots,n$. Окончание атома может иметь μ значений: 0 и 1 резервируются как в префиксе. Остальные $\mu-2$ значений соответствуют направленным рёбрам дерева, являющимися указателями на решающие вершины дерева (атомы списка). Указатель на одну (начальную вершину дерева) не требуется: нужны указатели только на $\mu-2$ внутренних вершин. Всего получается μ значений для окончания атома.

Использование стандартного самоограничивающего кода позволяет получить $pVCD$оценку

$$KP(h_\mu) < 2(]\log\log n[+]\log\log\mu\ [) + (\mu-1)(]\log(n+1)[+]\log\mu\ [),$$

и приближенно принять

$$KP(h_\mu) \approx 2(\log\log n + \log\log\mu) + (\mu-1)(\log(n+1) + \log\mu).$$

Усложнение *БРД* при добавлении ровно одной условной вершины приводит к увеличению сложности $KP(h_\mu)$ на длину одного атома, приблизительно равную

$$\log(n+1) + \log\mu.$$

Если при этом число ошибочно классифицируемых примеров выборки уменьшится на единицу, то сложность $KP(D \mid h_\mu)$ уменьшится на величину $\log l$. Эта величина характеризует *сложность одного необъясненного правилом h_μ примера из данных D – имеющейся в наличии таблицы, содержащей l примеров*. Поэтому она оценивается сложностью одного обращения к одной строке таблицы D. В таком случае оптимальная остановка ветвления (синтеза *БРД*) определяется условием

$$\log(n+1) + \log\mu > \log l.$$

При больших n для оценки μ можно применять неравенство $\log n\mu > \log l$. Тогда условие остановки синтеза определяется соотношением $\mu > l/n$. Так, если в обучающей выборке содержится $l = 300$ примеров, а число признаков $n = 20$, то увеличивать сложность *БРД* ради правильной классификации ещё только одного примера не следует при $\mu \geq 15$.

При уменьшении числа ошибок классификации на величину k на одном шаге усложнения *БРД* оценивающее неравенство примет вид

$$\log(n+1) + \log\mu > k\log l.$$

Приблизительное соотношение в этом случае будет иметь вид $\mu > l^k / n$, и можно сделать вывод, что в большинстве случаев ради исключения хотя бы двух ошибок следует продолжать ветвление. При этом может оказаться, что построенное *БРД* будет излишне сложным.

5.7. Случай *k*-значных переменных. Обобщение *БРД* до *k* -решающих деревьев

Если независимые переменные принимают значения из множества $E_k = \{0,1,...,(k-1)\}$, то для реализации алгоритмических отображений из класса

$$A_k = \{ f : \underbrace{E_k \times \cdots \times E_k}_{n} \to \{0,1,...,(k-1)\}$$

могут быть использованы классифицирующие *решающие деревья, в которых из каждой вершины выходят не более k ребер и число классов* q *не превышает* k. *Назовем такие деревья k- РД* [24].

Теорема 5.1. Любое алгоритмическое отображение из класса A_k может быть построено в виде *k*-РД.

Доказательство. Пусть $f \in A_k$. Прямой проверкой легко убедиться в справедливости разложения f по одной (любой) переменной:

$$f(x_1,...,x_{i-1},x_i,x_{i+1},...,x_n) = \bigvee_{\{\sigma \in E_k\}} I_\sigma(x_i) \,\&\, f(x_1,...,x_{i-1},\sigma,x_{i+1},...,x_n).$$

Здесь $\alpha \vee \beta = \max\{\alpha,\beta\}$, $\alpha \,\&\, \beta = \min\{\alpha,\beta\}$ и

$$I_\sigma(x) = \begin{cases} 0, x \neq \sigma, \\ k-1, x = \sigma. \end{cases}$$

На первом шаге построения алгоритмического отображения f реализуется корневая вершина дерева (рис.5.13).

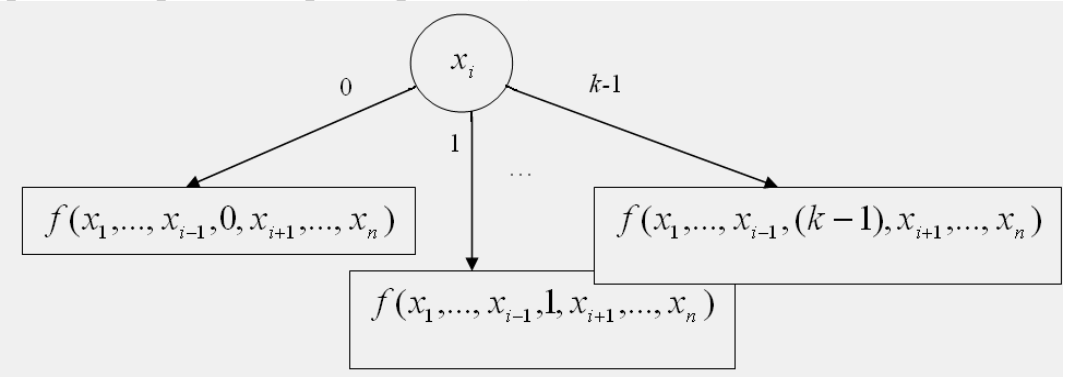

Рис. 5.13. Первый шаг ветвления. Из корневой вершины выходят k ребер

Если хотя бы одна из функций f_σ, $\sigma \in E_k$, полученных после построения корневой вершины и не зависящих от x_i, где

$$f_\sigma(x_1,...,x_{i-1},x_{i+1},...,x_n) = f(x_1,...,x_{i-1},\sigma,x_{i+1},...,x_n),$$

не является константой, то к ней, как к функции $(n-1)$-й переменной, снова применяется разложение по одной, но уже другой переменной, которое определяет следующий шаг построения *k-РД*. Если же для некоторого $\sigma \in E_k$ выполняется $f(x_1,...,x_{i-1},\sigma,x_{i+1},...,x) = \gamma = const$, т.е. оставшиеся незафиксированными переменные не являются существенными, то лист дерева, соответствующий ребру $x_i = \sigma$, становится терминальным и помечается константой γ.□

Основным элементом процедур синтеза *k-РД* является выбор переменной для ветвления (построения внутренней вершины дерева).

Обобщим D-критерий ветвления, описанный выше и используемый для синтеза БРД, на случай синтеза k-РД при k>2. Будем полагать, что обучающая информация состоит из l векторов, случайно и независимо выбранных из E_k^n, для каждого из которых *достоверно известно, какому из* q *классов он принадлежит*, причем среди них *нет одинаковых векторов с указанной принадлежностью разным классам.* Обучающая информация, удовлетворяющая указанным свойствам, называется *допустимой*, обозначается $T_{l,n,q}$ и является целочисленной таблицей из l строк и $n+1$ столбцов. Последний столбец служит для указания классов и не используется при выполнении теоретико-множественных операций над таблицей.

Определение5.1. Назовем *k-значным интервалом ранга* r в E_k^n множество $N_r = \{(x_1,...,x_n) \in E_k^n : x_{i_1} = \sigma_1,...,x_{i_r} = \sigma_r\}$, где $\sigma_1,...,\sigma_r \in E_k$; $0 \le r \le n$. Набор номеров переменных $I_r = \{i_1,...,i_r\}$ называется *направлением интервала*, а набор значений $(\sigma_1,...,\sigma_r)$ – *кодом интервала*. □

Если $r = 0$, то $N_r = E_k^n$; если $r = n$, то N_r состоит из единственной точки, принадлежащей E_k^n.

Пусть на шаге ветвления *t* при синтезе *k*-РД разбиению подлежит интервал $N_r^{(t)}$. Для ветвления, вообще говоря, может быть выбрана любая переменная, номер которой j не принадлежит направлению интервала $N_r^{(t)}$. Обозначим $K_t(j)$ число пар наборов различных классов в непустой подтаблице $T_{l,n,q} \cap N_r^{(t)}$, различающихся по переменной x_j. Если

$K_t(j^*) = \max_j K_t(j)$ и для ветвления выбирается переменная x_{j^*}, будем говорить, что используется *D-критерий ветвления.*

Определение5.2. (Δ, m)-сужением *k*-значного интервала N_r ранга *r* по переменной $x_m, m \notin I_r$, называется множество точек $N_r^{(\Delta,m)} = \{(x_1,...,x_m,...,x_n) \in N_r : x_m \in \Delta, \Delta \subset E_k\}$.□

Множество Δ в связи с таким определением можно условно считать одним выделенным значением, заменяющим набор из $|\Delta|$ значений.

D_Δ-критерий ветвления определим так, что при вычислении чисел $K_t(j)$ используется подтаблица $T_{l,n,q} \setminus N_r^{(\Delta,m)(t)}$, где $N_r^{(\Delta,m)(t)}$ – сужение интервала $N_r^{(t)}$, подлежащего разбиению. Если переменная x_{j^*} выбирается по D_Δ-критерию, то множество ребер, выходящих из внутренней вершины, соответствующей переменной x_{j^*}, состоит из группы ребер, соответствующих значениям $\sigma \in \{E_k \setminus \Delta\}$ и еще одного ребра, соответствующего множеству значений Δ.

D_Δ-критерий особенно полезен при синтезе *k-РД* по начальной информации $T_{l,n,q}$, имеющей пропуски – неизмеренные или неизвестные значения некоторых переменных. В этом случае символу Δ сопоставляется пропуск значения в таблице $T_{l,n,q}$. Совокупность строк из $T_{l,n,q}$, имеющих пропуск (Δ) значения переменной x_{j^*}, образует подтаблицу, которая далее используется для синтеза дерева с условием, что при последующих ветвлениях переменная x_{j^*} использоваться не будет. Шаг ветвления, допускающий пропуски, поясняется рисунком 5.14.

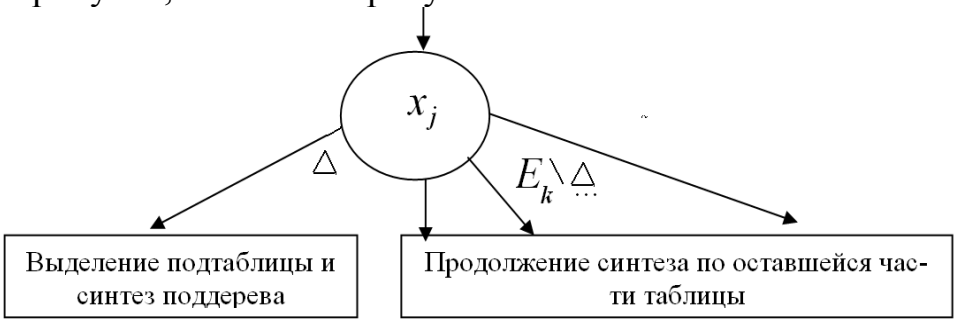

Рис. 5.14. Шаг ветвления с выделением подтаблицы по множеству Δ

Аналогично могут быть обобщены и другие критерии ветвления S_2, S_1, Z_1, Ω. На основе такого обобщения осуществляется эвристический синтез *k-РД*, близких к оптимальным, алгоритмом – аналогом *LISTBB,*

который можно отнести к типу *GREEDY*. Класс *k-РД* при этом расширяется до класса $(k+1)$-*РД*, допускающих принятие решений при наличии пропусков в информации.

Алгоритмы принятия решений, основанные на решающих деревьях и допускающие работу с пропусками в начальной информации, дают возможность полнее использовать обучающую информацию.

5.8. Эмпирический лес

Совокупность деревьев – отдельных компонент, являющихся связными графами и не имеющих циклов, называют лесом. Если имеется набор деревьев – эмпирических индукторов, то такую совокупность называют *эмпирическим лесом.* Будем полагать, что каждое дерево эмпирического леса *решает одну и ту же задачу классификации*, и деревья различаются тем, что реализуют *отличающиеся друг от друга алгоритмы*. Алгоритмы с такими свойствами заложены в основу парадигмы бэггинга.

Алгоритм синтеза r-корректного эмпирического леса «по ссылкам», представленный ниже, *имеет существенные отличия от декларированных приёмов бэггинга и бустинга*.

Алгоритм DFBSA построения r-корректного эмпирического леса (***D**ecision **F**orest **B**uilding **S**equencing **A**lgorithm*) [23]. .

Для построения леса используется непротиворечивая эмпирическая (обучающая) таблица $T_{l,n,q}$, содержащая l булевых наборов значений n переменных-признаков с указанной принадлежностью одному из q классов. Таблица непротиворечива: в ней нет двух одинаковых наборов, принадлежащих разным классам.

1° По заданному $\varepsilon > 0$ и значениям l, n находится такой допустимый ранг r конъюнктивной закономерности, что вероятность случайного обнаружения закономерности ранга r в случайно выбранной таблице не превысит ε (см. п. 5.5).

2° Строится бинарное решающее дерево одним из известных методов с учетом следующего правила отсечения: если при достройке *БРД* ранг ветви оказывается больше r, то в этой ветви остается r внутренних вершин, а листья, исходящие из последней по порядку вершины ветви, помечаются следующим образом. Если какой-нибудь лист из этих двух листьев соответствует интервалу из B^n, в который *попадают наборы только одного класса* из $T_{l,n,q}$, *то этот лист помечается меткой соответствующего класса*. Иначе лист помечается *указателем* (ссылкой) *на корневую вершину следующего дерева, которое предстоит построить*. Такое пра-

вило отсечения приводит к получению *БРД*, листья которого помечены либо метками классов, либо ссылками на следующее дерево.

3º Пусть уже построено $k \geq 1$ деревьев. Использованные при построении деревьев переменные (признаки) заносятся в список, называемый далее *USED.*

Синтез эмпирического леса завершается, если:

а) k -ое *БРД* не содержит ссылок в листьях (а содержит только метки классов),

б) $k = k_{\max}$, где константа $k_{\max}$ задает ограничение на возможное число деревьев эмпирического леса,

в) при переходе к построению нового дерева на предыдущих шагах синтеза леса уже были использованы все переменные (список *USED* полон).

Получается либо корректный, либо некорректный относительно обучающей таблицы лес.

4° Если условие прекращения синтеза не выполняется, то начинается синтез следующего дерева. Выделяются все наборы таблицы $T_{l,n,q}$, которые «попали» в интервалы, соответствующие ветвям, заканчивающимися листьями со ссылками от последнего построенного дерева. Эти наборы составляют некоторую подтаблицу $T_{m,n,q} \subset T_{l,n,q}$, $m < l$. Строится следующее дерево с использованием таблицы $T_{m,l,q}$ с *учетом нового порядка отбора переменных для внутренних вершин*. *Сначала используются переменные, не вошедшие в список USED, упорядоченные по используемому критерию выбора*. И только если их не хватает для синтеза ветвей допустимого ранга, используются переменные списка *USED.* Затем, после завершения построения дерева снова проверяется условие прекращения синтеза.□

Суть алгоритма ***DFBSA*** состоит в том, что последовательно строится набор из некоторого числа d эмпирических деревьев не более чем с $\mu \leq 2^r$ листьями каждое с учетом подрезания ветвей по пороговому рангу r. В итоге в лесе получается не более чем $d \cdot 2^r$ решающих ветвей (конъюнкций). Если каждое отдельное *БРД*, входящее в r-корректный лес, определяет ортогональную *ДНФ*, содержащую не более 2^r конъюнкций, то в целом по всему лесу конъюнкции, соответствующие разным деревьям, могут быть и неортогональными. Это становится очевидным, если предположить, например, что при большом числе переменных n два разных дерева, входящие в лес, используют во внутренних вершинах непересекающиеся подмножества этих переменных-признаков.

На рис. 5.15 схематически изображен эмпирический лес, построенный алгоритмом ***DFBSA***.

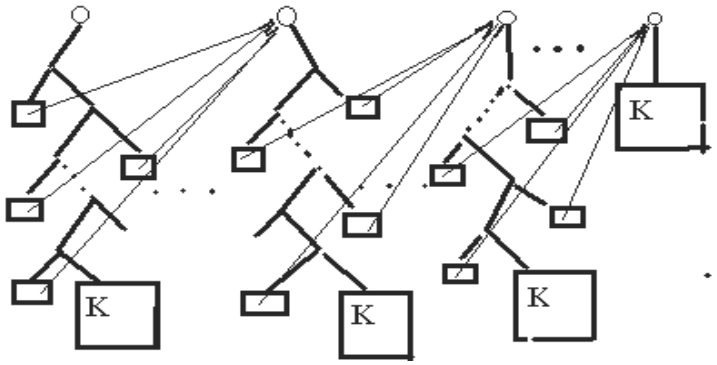

Рис. 5.15. Схема эмпирического леса

Буквами K помечены листья леса, указывающие на метки классов. Именно им соответствуют решающие конъюнкции ранга не выше r. Остальные листья соответствуют ссылкам на корневые вершины (помеченные кружком) некоторых деревьев, входящих в лес. Этим ссылочным листьям соответствуют конъюнкции, определяющие области «некомпетентности» отдельных деревьев, и они не используются совокупным решающим правилам эмпирического леса. В результате выполнения ссылки из области некомпетентности определяется другая решающая конъюнкция. Таким образом, в этой, ссылочной части, алгоритм ***DFBSA*** реализует идею Л.А. Расстригина о коллективе алгоритмов, отобранных по областям компетентности [37].

Совокупность конъюнкций, собранная по отметкам K, составляет набор дизъюнктивных нормальных форм, описывающих классы. Если число классов $q = 2$ (значения меток классов в этом случае можно считать 0 и 1), то весь эмпирический лес можно считать эквивалентным одной *ДНФ*, содержащей $d \cdot 2^r$ конъюнкций.

Приведем оценки VCD эмпирического леса. Доказательства можно найти в [25].

Число любых конъюнкций ранга не более r равно $\sum_{i=1}^{r} 2^i C_n^i$. Несложно проверить, что выполняется двойное неравенство

$$\frac{2^r (n-r)^r}{r!} < \sum_{i=0}^{r} 2^i C_n^i < \frac{(2n)^{r+1} - 1}{2n-1}.$$

Известна оценка мощности и VCD конечного класса $DNF(n, \mu, r)$ решающих правил, образованных дизъюнктивными нормальными формами, содержащими не более μ конъюнкций ранга не более r, состоящими из литералов n переменных [25].

$$\frac{\left(\left(\frac{2n}{r}-1\right)^r-\mu\right)^\mu}{\mu!} < \mid DNF(n,\mu,r) \mid < \frac{1.5^\mu n^{r\mu}}{\mu!};$$

$$\mid DNF(n,\mu,r) \mid = \Theta(n^{r\mu});$$

$$VCD(DNF(n,\mu,r)) < r\mu \log n - \mu \log \frac{\mu}{2} = O(\log n);$$

$$VCD(DNF(n,\mu,r)) = \Theta(\log n).$$

Теорема 5.2 [25]. Пусть $BDF(n,\mu,r,q)$ – класс $r-$ корректных решающих лесов, содержащий не более μ конъюнкций ранга не выше r для случая n булевых признаков и q классов. Тогда выполняется двойное неравенство

$$\max(\mu q, \log n) < VCD(BDF(n,\mu,r,q) < r\mu q \log n - \mu q \log \frac{\mu q}{2}. \square$$

Следствие 5.1. $VCD(BDF(n,\mu,r,q)) = \Theta(\log n). \square$

Приведенные оценки позволяют сделать вывод, что емкость класса эмпирических лесов с ограничением на суммарное число листьев и ранги ветвей деревьев имеет один порядок роста с емкостью класса решающих деревьев с ограниченным числом листьев.

Полезность совместного использования набора эмпирических *БРД* для принятия решений обусловлена повышением надежности результатов и возможностью применения набора *БРД* в случае наличия большого числа пропусков в информации, поступающей для принятия решения по синтезированному набору деревьев. Дополнительно поясняя роль набора эмпирических *БРД*, уместно применить фразу «судить (делать вывод) по разным признакам».

Поскольку *БРД* с μ листьями использует не более $\mu-1$ внутренних вершин, то μ-*БРД* использует не более чем $\mu-1$ переменную из n, и при $\mu << n$ используется малая часть переменных (если $\mu = const$, то при $n \to \infty$ почти все переменные из n не используются в μ-*БРД*).

Напомним, что для случая целочисленных (*k*-значных) признаков известны понятия теста и тупикового теста [41]. *Подмножество столбцов* Ω_A *таблицы обучения* $T_{l,n,q}$ *называется тестом*, если любые две строки подтаблицы, образованной данными столбцами, различны при условии их принадлежности разным классам. *Тупиковым называется тест*, любое собственное подмножество которого не является тестом. Тупиковый тест, состоящий из минимального числа столбцов по сравнению с другими ту-

пиковыми тестами таблицы, называется *минимальным тупиковым тестом*.

Тупиковый тест – это минимальная подсистема признаков, разделяющая эталоны (примеры) разных классов.

***Пример*.** Рассмотрим таблицу обучения $T_{4,4,2}$ – четыре точки, четыре признака, два класса K_1 и K_2.

$$\left.\begin{matrix}\tilde{x}_1 & 0111\\ \tilde{x}_2 & 1010\end{matrix}\right\} \in K_1$$

$$\left.\begin{matrix}\tilde{x}_3 & 0011\\ \tilde{x}_4 & 1000\end{matrix}\right\} \in K_2$$

Эта таблица имеет два тупиковых теста: {2,3,4},{1,2,3}. □

Очевидно, что *для возможности построения корректного БРД, использующего только признаки с номерами* $i_1,...,i_s$ *по информации* (*обучающей таблице*) $T_{l,n,q}$, *необходимо и достаточно, чтобы множество* $\{i_1,...,i_s\}$ *было тестом таблицы* $T_{l,n,q}$.

Известно, что для почти всех таблиц при $n \to \infty,\ l \to \infty$ и условии $\lim\limits_{n\to\infty} l / 2^{n/2} = 0$, для любого положительного ε любые $2(1+\varepsilon)\log_2 l$ столбцов таблицы образуют тест [29, 41]. Для всех таблиц с l строками средняя длина тупикового теста s_{cp} заключена в отрезке [61]

$$]\log_2 l[\le s_{cp} \le 2]\log_2 l[.$$

Следовательно, для широкого класса произвольных булевых таблиц при синтезе *БРД* можно получить набор корректных деревьев, использующих полностью (или частично) разные переменные.

5.9. Поиск признаковых предикатов

« Долго он не мог распознать, какого пола была фигура: баба или мужик. Платье на ней было совершенно неопределенное, похожее очень на женский капот, на голове колпак, какой носят деревенские дворовые бабы, только один голос показался ему несколько сиплым для женщины"
Н.В.Гоголь. Мертвые души. Гл. VI

Признаковые предикаты могут рассматриваться как элементарные классификаторы вида $P_i : \mathbf{X} \to \{0,1\}$, $i = 1,2,...,n$, где $\mathbf{X}$ – множество

исходных описаний допустимых объектов произвольной размерности. Во многих случаях, когда $\mathbf{X}$ состоит из наборов однотипных переменных – вещественных (если не учитывать частично рекурсивную реализацию), рациональных или целых чисел, то в качестве признаков эти числа и берутся (возможно, с нормировкой). Но если переменные-признаки разнотипные, то их можно свести к бинарным, выбрав некоторые признаковые предикаты. Признаковые предикаты можно называть *синдромами* или *вторичными признаками*. Для применения *БРД* признаковые предикаты необходимы принципиально.

Набор признаковых предикатов будем обозначать $\tilde{P} = \{P_1,...,P_n\}$. Это набор реализует отображение (будем обозначать его тем же символом) $\tilde{P} : \mathbf{X} \to B^n$, где $B^n = \{0,1\}^n$. В случае классификации с q классами отображение $\tilde{P}$ должно обеспечивать вычисление номера класса в соответствии с композицией

$$\tilde{P} \circ F, \quad F : B^n \to \{0,1,...,q\}. \tag{5.4}$$

Очевидно, что для существования композиции вида (5.4) при достаточно разнообразном исходном пространстве описаний $\mathbf{X}$ (как минимум при $|\mathbf{X}| \geq q$) необходимо и достаточно выполнения условия $n \geq]\log q[$.

Определение 5.3. Множество признаковых предикатов $\tilde{P}$ называется *допустимым*, если существует композиция (5.4).

Определение 5.4. Допустимое множество признаковых предикатов $\tilde{P}$ называется *корректным относительно обучающей информации*, если для любой точки $\tilde{x}_j$ из обучающей информации $(\tilde{x}_j, \alpha_j)_{j=1}^{l}$ выполняется условие

$$(\tilde{P} \circ F)(\tilde{x}_j) = \alpha_j . \square$$

Отображение $\tilde{P}$ переводит обучающую информацию $(\tilde{x}_j, \alpha_j)_{j=1}^{l}$ в булеву таблицу $T_{l,n,q}$, каждая строка которой является булевым описанием точки из обучающей выборки с указанием её принадлежности одному из классов.

Напомним, что *обучающая информация $(\tilde{x}_j, \alpha_j)_{j=1}^{l}$ называется корректной*, если в ней для любой пары точек выполняется условие $(\alpha_j \neq \alpha_v) \Rightarrow (\tilde{x}_j \neq \tilde{x}_v),\ 1 \leq j < v \leq l$.

Утверждение 5.5. *Свойство корректности исходной обучающей информации сохраняется для таблицы $T_{l,n,q}$ в случае корректного множества предикатов $\tilde{P}$.*

Действительно, пусть какие-нибудь две строки таблицы $T_{l,n,q}$ с номерами $u, w,\ \ 1 \le u < w \le l$, которые помечены метками классов $\alpha_u \ne \alpha_w$ совпадают: $\tilde{y}_u = \tilde{y}_w$. Тогда это одна и та же совпавшая строка, полученная *в результате применения корректного отображения* $\tilde{P}$ к некоторой точке $\tilde{x}_j$ обучающей выборки. Поэтому $\tilde{y}_u = \tilde{P}(\tilde{x}_j),\ \ \tilde{y}_w = \tilde{P}(\tilde{x}_j),$ $\tilde{y}_u = \tilde{y}_w$, и тогда метки классов α_u и α_w обязаны совпадать: $\alpha_u = \alpha_w$.□

Утверждение 5.5 обосновывает следующий алгоритм отбора корректного множества признаковых предикатов.

1° Выбрать первый предикат P_1 и положить $n = 1.$

2º Построить, используя начальную обучающую информацию $(\tilde{x}_j, \alpha_j)_{j=1}^{l}$, таблицу $T_{l,n,q}$ и проверить её корректность.

3° Если таблица корректна, то перейти на 4°; иначе выбрать следующий признаковый предикат, положить $n := n+1$ и прейти на 2°

4° Конец алгоритма.

Каким образом реализуется выбор признаковых предикатов на шагах приведенного алгоритма – не определено. Такой выбор является трудноформализуемой задачей, для решения которой может использоваться широкий арсенал средств математической статистики, интерактивный подход, экспертные оценки. Полезными являются следующие эвристические принципы поиска признаковых предикатов.

1. Более предпочтительным является класс предикатов, имеющий меньшую структурно-алгоритмическую сложность.

2. Более предпочтительными являются предикаты, позволяющие выделять наиболее значимые статистические закономерности. Для поиска используются точечные и интервальные оценки, уравнения регрессии, сравнение распределений, вычисление оптимального по Байесу порога (или набора порогов [70]), оценка связи между группами признаков и др.

Покажем, как должен выбираться оптимальный по Байесу одноместный предикат $P(x) = «x > a»$, если имеются условные плотности распределения $p(x/K_1)$ и $p(x/K_2)$ некоторого числового признака x и двух классов K_1 и K_2 (рис. 5.16).

Плотности условных вероятностей в решаемой задаче $p(x/K_1)$ и $p(x/K_2)$ на самом деле неизвестны; выбор признаковых предикатов осуществляется на основе заданной таблицы обучения, что в случае использования байесовского подхода предполагает восстановление указанных плотностей.

Выбрав из таблицы обучения столбец значений для какой-нибудь одной переменной x и в нем элементы с пометками $\alpha = K_1$ можно построить гистограмму распределения $\hat{p}(x/K_1)$. Точно так же можно построить гистограмму $\hat{p}(x/K_2)$. Затем выяснить: существует ли точка $\hat{a}$ такая, что $\hat{p}(\hat{a}/K_1) = \hat{p}(\hat{a}/K_2)$ и в окрестности этой точки $\hat{a}$ одна гистограмма убывает, а другая – возрастает. Не исключается случай, что такая ситуация в выборочных данных возникла случайно и не соответствует реальной закономерности для условных плотностей распределения.

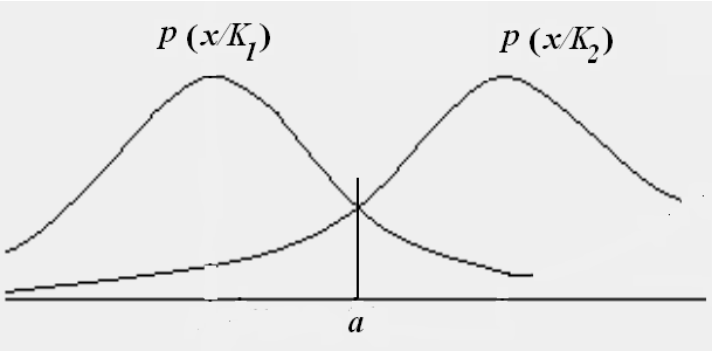

Рис. 5.16. Байесовский выбор одноместного признакового предиката «$x > a$»

Если же построить общую по всем классам гистограмму плотности распределения одной переменной x (не используя информацию о принадлежности точек классам, которая имеется в обучающей выборке), то возможность выделения байесовских порогов будет определяться фактом существования в этой общей гистограмме более одной моды. Желательно, чтобы этот факт был статистически достоверным.

Ниже представлен подход к оценке достоверности существования локального минимума и, следовательно, не менее двух мод плотности распределения числового признака на основании данных, извлеченных из обучающей выборки.

Теорема 5.3. Пусть функция распределения случайной величины $F(x) = \int_{-\infty}^{x} f(u)du$ непрерывна, имеет производную $f(x)$ в каждой точке отрезка $[a,b]$, и заданы точки $a < x_1 < x_2 < x_3 < b$ так, что $x_2 - x_1 = x_1 - a = \Delta_1$ и $b - x_3 = x_3 - x_2 = \Delta_2$. Тогда при одновременном выполнении неравенств

$$2F(x_1) - F(a) - F(x_2) > 0, \tag{5.5}$$

$$F(b) - 2F(x_3) + F(x_2) > 0 \tag{5.6}$$

плотность распределения $f(x)$ в интервале (a,b) имеет локальный минимум.

Доказательство. Перепишем неравенства (5.5) и (5.6) в виде

$$F(x_1) - F(a) > F(x_2) - F(x_1), \tag{5.7}$$

$$F(b) - F(x_3) > F(x_3) - F(x_2). \tag{5.8}$$

По теореме Лагранжа

$$\exists \xi \in (a, x_1) : F(x_1) - F(a) = f(\xi)\Delta_1,$$
$$\exists \eta \in (x_1, x_2) : F(x_2) - F(x_1) = f(\eta)\Delta_1,$$
$$\exists \tau \in (x_2, x_3) : F(x_3) - F(x_2) = f(\tau)\Delta_2,$$
$$\exists \lambda \in (x_3, x_b) : F(b) - F(x_3) = f(\lambda)\Delta_2,$$

откуда с учётом (5.7) и (5.8) следует, что для точек $a < \xi < \eta < \tau < \lambda < b$ выполняются неравенства $f(\xi) > f(\eta)$ и $f(\tau) < f(\lambda)$, доказывающие теорему.

Следствие 5.2. Если плотность распределения $f(x)$ определена для всех $x \in R$ и выполнены условия теоремы, то она имеет более одной моды.

Замечание. Неравенства и точки, введенные в условие теоремы, служат для удобства статистического оценивания и специально подобраны для этой цели. □

При обработке статистических данных возможно использование только эмпирической функции распределения $\hat{F}(x)$.

Обозначим

$$\delta = \sup_{-\infty < x < \infty} |F(x) - \hat{F}(x)|. \tag{5.9}$$

Теорема 5.4. При выполнении неравенств

$$2\hat{F}(x_1) - \hat{F}(a) - \hat{F}(x_2) > 4\delta, \tag{5.10}$$

$$\hat{F}(b) - 2\hat{F}(x_3) + \hat{F}(x_2) > 4\delta \tag{5.11}$$

выполняются условия теоремы 5.3, и плотность $f(x)$ имеет более одной моды.

Доказательство. Неравенство (5.10) можно переписать в виде

$$2(\hat{F}(x_1) - \delta) - (\hat{F}(a) + \delta) - (\hat{F}(x_2) + \delta) > 0. \tag{5.12}$$

Из (5.9) следует, что

$$F(x_1) \geq \hat{F}(x_1) - \delta;$$
$$\hat{F}(a) + \delta \geq F(a);$$
$$\hat{F}(x_2) + \delta \geq F(x_2).$$

Используя эти неравенства, из (5.12) получаем

$$2F(x_1) - F(a) - F(x_2) > 0.$$

Аналогично показывается справедливость неравенства

$$F(b) - 2F(x_3) + F(x_2) > 0 .\square$$

Для осуществления статистической проверки существования локального минимума плотности распределения используются значения числового признака, которые извлечены из обучающей выборки. Предполагается выполнение следующих этапов.

1º Построить гистограмму плотности распределения $\hat{f}(x)$ признака x (рис. 5.17) и на её основе – кумулятивную оценочную функцию распределения $\hat{F}(x)$ (рис. 5.18). Выбрать точки a, x_1, x_2, x_3, b так, чтобы выполнялись условия теоремы 5.4 (точка x_2 должна соответствовать локальному минимуму на гистограмме).

2º Определить величину отклонения

$$d = \min\{2\hat{F}(x_1) - \hat{F}(a) - \hat{F}(x_2);\ \hat{F}(b) - 2\hat{F}(x_3) + \hat{F}(x_2)\}.$$

Из теоремы 5.4 следует, что при выполнении условия

$$\sup_{-\infty<x<\infty} |F(x) - \hat{F}(x)| < d/4, \qquad (5.13)$$

плотность распределения будет иметь более одной моды. Неравенство (5.13) может иметь место с некоторой вероятностью, которую можно оценить при помощи критерия А.Н. Колмогорова [1].

Критерий Колмогорова применяется для проверки непараметрической гипотезы, согласно которой независимые одинаково распределенные случайные величины имеют непрерывную функцию распределения $F(x)$. Согласно теореме Колмогорова

$$P(\sup_{-\infty<x<\infty} |F(x) - \hat{F}(x)| < \lambda/\sqrt{l}) \to K(\lambda) \text{ при } l \to \infty,$$

где

$$K(\lambda) = \sum_{i=-\infty}^{\infty} (-1)^i e^{-2i^2\lambda^2} .$$

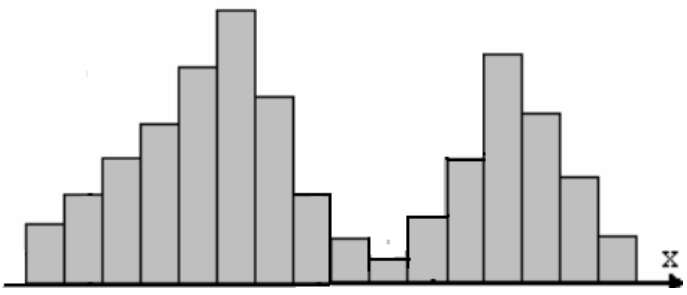

Рис. 5.17. Гистограмма плотности распределения значений признака

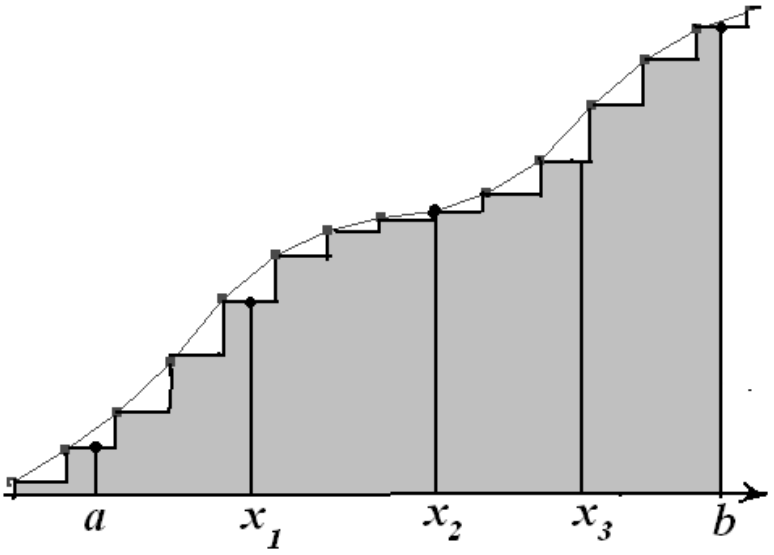

Рис. 5.18. Кумулятивная оценочная функция распределения $\hat{F}(x)$

Смысл использования критерия Колмогорова состоит в том, чтобы оценить какова *вероятность случайного обнаружения локального минимума в гистограмме плотности вероятности*.

Если неравенство (5.13) достоверно при найденном значении d, то локальный минимум достоверно существует. Вероятность

$$P(\sup_{-\infty<x<\infty} |F(x)-\hat{F}(x)|< d/4)$$

оценивает меру случайности эмпирического обнаружения минимума.

3º Найти величину $\lambda = \frac{d\sqrt{l}}{4}$ (она определяется из соотношения $\frac{\lambda}{\sqrt{l}} = \frac{d}{4}$) и значение $K(\lambda)$ по таблице распределения статистики Колмогорова [1].

Чтобы проиллюстрировать, какие при этом получаются результаты, приведена таблица 5.6.

4º Если минимум на гистограмме оказывается значимым, то искомый признаковый предикат имеет вид «$x > x_2$».□

Из таблицы 5.6, в частности, видно, что обнаружение минимума в гистограмме, построенной по выборке из 100 наблюдений, при наличии отклонения $d = 0.1$ позволяет с высокой вероятностью 0,9958 принять гипотезу о существовании в плотности распределения признака локального минимума.

Таблица 5.6. Расчет вероятности существования минимума плотности распределения по критерию Колмогорова

№	Отклонение d	Длина выборки l	$\lambda = \frac{d\sqrt{l}}{4}$	Вероятность существования локального минимума $K(\lambda)$
1	0,04	250	1,7643	0,9960
2	0,05	190	1,6750	0,9925
3	0,10	60	1,0500	0,7400
4	0,10	70	1,2290	0,8980
5	0,10	80	1,4000	0,9600
6	0,10	90	1,5820	0,8960
7	0,10	95	1,6700	0,9924
8	0,10	100	1,7590	0,9958
9	0,20	40	1,3960	0,9603
10	0,20	50	1,7500	0,9956
11	0,30	32	1,6700	0,9924

5.10. Подходы к оцениванию качества деревьев решений как эмпирических индукторов

Оценить точность корректного на обучающей выборке *БРД*, использующего в качестве предикатов во внутренних вершинах бинарные переменные-признаки, можно, если известно число решающих правил в семействе, из которого решающее дерево было выбрано.

Семейство, состоящее из *БРД*, имеющих ровно μ листьев и реализующих отображения вида $f:\{0,1\}\to\{0,1,...,(k-1)\}$, обозначим $D(n,k,\mu)$. Здесь n – число булевых признаков, k – число классов. Обозначим $d(n,k,\mu)=|D(n,k,\mu)|$ – число различных *БРД* в семействе $D(n,k,\mu)$. Точное значение $d(n,k,\mu)$ неизвестно. Ниже будут получены оценки для этого комбинаторного числа.

Теорема 5.5 [12]. При заданных константах k,μ и $n\to\infty$ имеет место асимптотика

$$d(n,k,\mu)\sim(\mu-1)![k(k-1)]^{\mu-1}n(n-1)^{\mu-2}.$$

Доказательство. Очевидно, что $d(n,k,2)=n\cdot 2\cdot C_k^2$. Переход от *БРД* с j листьями к *БРД* с $j+1$ листьями связан с заменой какой-нибудь

одной концевой вершины на новую внутреннюю вершину и добавлением двух новых листьев. Такой процесс «достройки» предполагает:

а) выбрать любой из j листьев;

б) заменить выбранный лист вершиной с предикатом – переменной, *не встречавшейся в ветви, которая заканчивалась замещаемым листом*;

в) выбрать два разных значения из $\{0,1,\ldots,k\}$ для пометки двух новых листьев.

Поскольку наибольшее возможное число неконцевых вершин *в одной ветви БРД* с j листьями равно $j-1$, то выбрать переменную для новой внутренней вершины при $n > j-1$ можно не менее чем $n-j+1$ способами. Тогда

$$d(n,k,j+1) > d(n,k,j)\cdot j\cdot 2\cdot C_k^2\cdot(n-j+1).$$

Из этого неравенства при $n > \mu-2$ получается нижняя оценка

$$d(n,k,\mu) > L(n,k,\mu) = n(\mu-1)![k(k-1)]^{\mu-1}\cdot(n-\mu+2)^{\mu-2}.$$

С другой стороны, выбрать переменную для замены листа внутренней вершиной можно не более чем $n-1$ способами, а для дерева, в котором каждая ветвь содержит более одной внутренней вершины – уже менее чем $n-1$ способами, поэтому

$$d(n,k,j+1) < d(n,k,j)\cdot j\cdot 2\cdot C_k^2\cdot(n-1). \qquad (5.14)$$

Из неравенства (5.14) получается верхняя оценка

$$d(n,k,\mu) < H(n,k,\mu) = n(\mu-1)![k(k-1)]^{\mu-1}\cdot(n-1)^{\mu-2}.$$

Легко убедиться, что

$$\lim_{n\to\infty}\frac{L(n,k,\mu)}{H(n,k,\mu)} = 1.$$

Следовательно $H(n,k,\mu)\sim L(n,k,\mu)$ при $n\to\infty$ и

$$d(n,k,\mu)\sim(\mu-1)![k(k-1)]^{\mu-1}n(n-1)^{\mu-2}.$$

Следствие 5.3. Число булевых функций $b(n,2,\mu)$ от n переменных, представимых БРД с ровно μ листьями, удовлетворяет неравенству

$$b(n,2,\mu) < (\mu-1)!\cdot 2^{\mu-1}\cdot n^{\mu-1}.$$

Следствие 5.4. Класс $P^2_{БРД}(n,\mu)$ булевых функций, представимых *БРД* с ровно μ листьями, при $n\to\infty$ сколь угодно узок по сравнению с классом $L(n)$ линейных булевых функций из P_2.

Доказательство.

$$|L(n)| = 2^{n+1};\ \frac{|P^2_{БРД}(n,\mu)|}{|L(n)|} < \frac{(\mu-1)!\cdot 2^{\mu-1}\cdot n^{\mu-1}}{2^{n+1}};$$

$$| P^2_{БРД}(n,\mu) | = o(2^{n+1}) \text{ при } n \to \infty.$$

Теорема 5.6. С вероятностью $1-\delta$ можно утверждать, что вероятность ошибочной классификации объектов, описываемых n булевыми признаками, при помощи корректного на обучающей выборке длины l БРД будет меньше ε, если *только длина обучающей последовательности будет не меньше*

$$l = \frac{(\mu-1)\log(2n) + \sum_{j=2}^{\mu-1}\log j - \ln\delta}{-\ln(1-\varepsilon)}. \qquad (5.15)$$

Доказательство. Оценка (5.15) является частным случаем оценки, представленной в работе[4, Теорема 5.2, с.106] для случая обучения распознаванию в детерминистской постановке с конечным классом используемых решающих правил, содержащем N гипотез

$$l = \frac{\ln N - \ln\delta}{-\ln(1-\varepsilon)}.\square$$

Точность *БРД* как эмпирического индуктора можно *оценить по контрольной выборке.* В этом случае используется следующая вероятностная схема. Точки контрольной выборки извлекаются из генеральной совокупности случайно и независимо. Контрольная выборка не содержит общих примеров с использованной обучающей выборкой. Контрольные точки снабжены точной информацией о принадлежности классам, и их появление не зависит и от того, какое дерево было построено при обучении. Тогда частота ошибок на контрольной выборке будет несмещенной оценкой вероятности ошибки построенного решающего дерева на генеральной совокупности. Распределение числа ошибок в этом случае будет биномиальным. Независимое от используемого класса гипотез оценивание по контрольной выборке рассматривается в главе 6.

Теорема 5.7. Если классификатор *БРД*, имеющий μ листьев, на контрольной последовательности длины l_C допустил δl_C ошибок, где $0 \le \delta < 1$, то для любого ε такого, что $1 > \varepsilon > \delta$, имеет место неравенство

$$\Pr(P(E) \ge \varepsilon) \le \frac{\mu}{4l_C(\varepsilon-\delta)^2}. \qquad (5.16)$$

Замечание. Оценка (5.16) содержит параметр μ – число листьев *БРД*, который определяет действующую ёмкость использованного класса индукторов. Она нужна для того, чтобы указать на важность синтеза *БРД* именно с минимальным числом листьев.

Доказательство. Обозначим пометки листьев *БРД* $\omega_1,...,\omega_s,...,\omega_\mu$ так, чтобы пометка ω_s определяла класс точек, попавших в интервал разбиения N_s, который соответствует s-той ветви дерева, заканчивающейся листом ω_s. Вероятностную меру интервала N_s обозначим $P(N_s)=\Pr\{\tilde{x}\in N_s\}$. Для упрощения записи будем обозначать N_s и интервал, и событие «$\tilde{x}\in N_s$», а ω_s – и номер класса, и событие, заключающееся в появлении точки именно этого класса. Интервалы $N_1,...,N_s,..,N_\mu$ образуют разбиение множества B^n, поэтому

$$\sum_{s=1}^{\mu}P(N_s)=1;\;\; P(E)=\sum_{s=1}^{\mu}P(E/N_s)P(N_s);$$

$$P(E/N_s)=1-P(\omega_s/N_s);\;\; P(\omega_s,N_s)=P(\omega_s/N_s)P(N_s);$$

$$P(E)=\sum_{s=1}^{\mu}(1-P(\omega_s/N_s)P(N_s)=$$

$$=\sum_{s=1}^{\mu}(P(N_s)-\sum_{s=1}^{\mu}P(\omega_s,N_s)=1-\sum_{s=1}^{\mu}P(\omega_s,N_s).$$

Для каждого интервала разбиения частоты

$$\nu(\omega_s,N_s)=\frac{n(\omega_s,N_s)}{l_C}$$

определяются числами $n(\omega_s,N_s)$ точек из контрольной выборки, попавших в интервал N_s и отнесенных к классу ω_s. Эти точки классифицируется деревом правильно. Обозначим число точек контрольной выборки, попавших в интервал N_s и классифицируемых неправильно, как k_s. Тогда

$$\sum_{s=1}^{\mu}(n(\omega_s,N_s)+k_s)=l_C;\;\; \sum_{s=1}^{\mu}\frac{n(\omega_s,N_s)}{l_C}+\sum_{s=1}^{\mu}\frac{k_s}{l_C}=1;$$

$$\sum_{s=1}^{\mu}\nu(\omega_s,N_s)+\delta=1, \qquad (6.1)$$

где $\delta=\frac{1}{l_C}\sum_{s=1}^{\mu}k_s$ – доля ошибок на контрольной выборке. Подставим левую часть равенства (6.1) вместо единицы в формулу, определяющую ошибку *БРД*:

$$P(E)=1-\sum_{s=1}^{\mu}P(\omega_s,N_s)=\sum_{s=1}^{\mu}\nu(\omega_s,N_s)-\sum_{s=1}^{\mu}P(\omega_s,N_s)+\delta.$$

Событие «$P(E) \geq \varepsilon$» равносильно событию

$$\sum_{s=1}^{\mu} \nu(\omega_s, N_s) - \sum_{s=1}^{\mu} P(\omega_s, N_s) \geq \varepsilon - \delta .$$

Найдем математическое ожидание и дисперсию случайной величины $\xi = \sum_{s=1}^{\mu} \nu(\omega_s, N_s)$ – суммы независимых случайных величин.

$$\mathrm{M}[\xi] = \sum_{s=1}^{\mu} \mathrm{M}[\nu(\omega_s, N_s)] = \sum_{s=1}^{\mu} P(\omega_s, N_s);$$

$$\mathrm{D}[\xi] = \sum_{s=1}^{\mu} \mathrm{D}[\nu(\omega_s, N_s)] = \sum_{s=1}^{\mu} \mathrm{M}[(\frac{n(\omega_s, N_s)}{l_C} - P(\omega_s, N_s))^2] =$$

$$= l_C^{-2} \sum_{s=1}^{\mu} \mathrm{M}[(n(\omega_s, N_s) - l_C P(\omega_s, N_s))^2].$$

Здесь $l_C P(\omega_s, N_s)$ – математическое ожидание случайной величины «ω_s, N_s», а $\mathrm{M}[(n(\omega_s, N_s) - l_C P(\omega_s, N_s))^2]$ – дисперсия этой случайной величины, равная $l_C P(\omega_s, N_s)(1 - P(\omega_s, N_s)) \leq \frac{l_C}{4}$. Отсюда получаем неравенство $\mathrm{D}[\xi] \leq \frac{\mu}{4 l_s}$. Используя неравенство Чебышёва

$$(\forall \varepsilon > 0) \quad \Pr(|\varsigma - \mathrm{M}[\varsigma]| \geq \varepsilon) \leq \mathrm{D}[\varsigma] / \varepsilon^2,$$

Получаем

$$\Pr(P(E) \geq \varepsilon) \leq \frac{\mu}{4 l_C (\varepsilon - \delta)^2}. \qquad (5.17)$$

Следствие 5.5. Чем меньше число листьев решающего дерева, тем выше его статистическая надежность. □

Применение вместо неравенства Чебышёва неравенства Бернштейна дает оценку

$$\Pr(P(E) \geq \varepsilon) < \exp\{-\frac{(\varepsilon - \delta)^2 l_c}{\mu}\}. \qquad (5.18)$$

Замечание. Правые части оценок (5.17, 5.18) не содержат переменную n – размерность признакового пространства. Если во внутренних вершинах предикаты являются одноместными, зависят только от одной переменной, то число всех использованных в *БРД* переменных не превысит $\mu - 1$. Таким образом, размерность n входит в оценки неявно: $n < \mu - 1$.

Оценивание точности BSP деревьев. *BSP* (*Binary Space Partition*) деревьями называют БРД, во внутренних вершинах которых используются признаковые предикаты, разделяющие исходное n-мерное пространство признаков гиперплоскостями. *BSP* с одной внутренней вершиной разделяет исходное пространство гиперплоскостью на две области; с двумя внутренними вершинами – на три области; с $\mu - 1$ внутренней вершиной – на μ областей, где μ число листьев дерева [44].

На рис.5.19 приведено разбиение прямыми с номерами $1,2,3,4$ и соответствующее этому разбиению *BSP*. Стрелки на схеме разбиения соответствуют ветвям дерева, помеченным нулями.

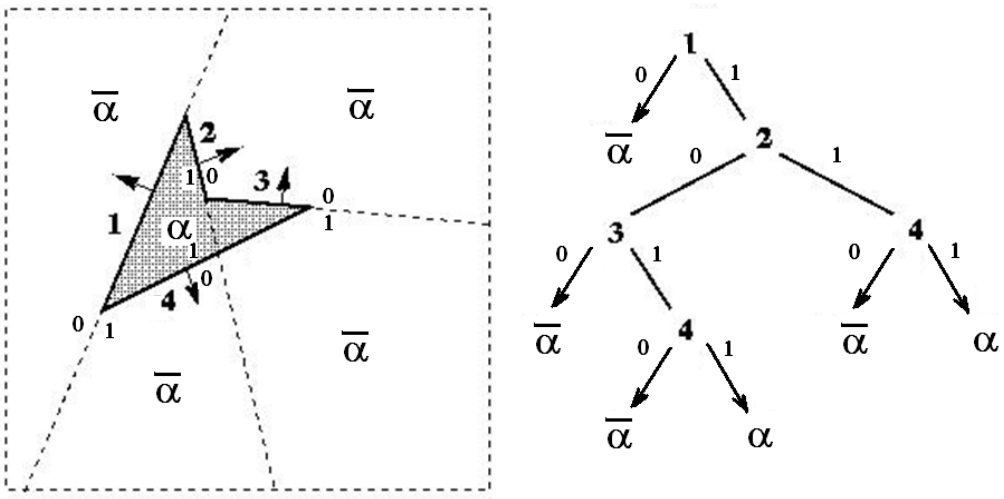

Рис. 5.19. Разбиение, определяемое *BSP*

Оценка $VCD(BSP_{n,m})$ при компьютерной реализации с использованием M бит на кодирование каждого параметра гиперплоскости может быть получена $pVCD$ методом:

$$pVCD(BSP_{n,m}) =$$
$$= (\mu - 1)(]\log n[+]\log(\mu + 3)[+(n+1)M + 2]\log((n+1)M)[).$$

Используя оценку PAC – обучаемости, основанную на вапниковской ёмкости, получаем

$$l(BSP_{n,m}, \delta, \varepsilon) \le \frac{4}{\varepsilon}(pVCD(BSP_{n,m}) \cdot \log\frac{12}{\varepsilon} + \log\frac{2}{\delta}).\square$$

Подкласс *Raw BSP* (*raw – англ. – недоработанный*) определяется специально оговоренным алгоритмом обучения, объясняющим название этого класса и состоящем в следующем.

1° Обучающая последовательность предполагается состоящей из l примеров, причем $l >> n$.

2º Первые n примеров из $(\tilde{x}_j, \alpha_j)_{j=1}^{l}$ используются для выбора предиката (гиперплоскости) для корневой вершины.

3° Следующие $k-n$ примеров от $n+1$-го до k-го используются для синтеза поддеревьев, пока не выполняется условие остановки, учитывающее ограничение: число листьев μ построенного дерева не должно превышать $\frac{k}{n}$.

4º Оставшиеся $l-k$ примеров с номерами от $k+1$ до l «попадают» в интервалы разбиения, соответствующего построенному дереву, определяя далее голосование, как завершающий этап вычисления классификации произвольного объекта. А именно, если объект попадает в терминальную вершину (лист), содержащую наибольшее число примеров из класса α, то решение применяется в пользу этого класса α (мажоритарное голосование).

Следующая теорема дает лишь подтверждение состоятельности алгоритма обучения *Raw BSP*.

Теорема5.8 [44]. Если h *– Raw BSP* классификатор, то при $l \to \infty$, $k \to \infty$ и $k/l \to 0$

$$\lim_{l\to\infty} Err_l(h) \ \to Err(h)$$

с вероятностью 1 независимо от распределения вероятностей на X^l. □

Список литературы к главе 5

1. Абезгауз Г.Г. Справочник по вероятностным расчетам / Г.Г. Абезгауз, А.П.Тронь, Ю.Н. Копенкин, И.А. Коровина. – М.: Воениздат, 1970. – 536 с.
2. Айвазян С. А., Прикладная статистика: классификация и снижение размерности / С.А. Айвазян, В.М. Бухштабер, И.С. Енюков, Л.Д. Мешалкин. – М.: Финансы и статистика, 1989.
3. Блох А. Ш. Об одном алгоритме обучения для задач по распознаванию образов / А. Ш. Блох // Вычислительная техника в машиностроении. – Минск: 1966. - №10. – С. 37 – 43.
4. Вапник В. Н., Червоненкис А. Я. Теория распознавания образов / В. Н. Вапник, А. Я. Червоненкис // Теория распознавания образов. – М.: Наука, 1974. – 416 с.

5. Воронцов К. В. Логические алгоритмы классификации (курс лекций «Машинное обучение»)[Электронный ресурс]/ К. В. Воронцов. – М.: 2012. – 53 с. – Режим доступа: http://www.machinelearning.ru/wiki/images/9/97/Voron-ML-Logic-slides.pdf
6. Гладун В.П. Составление описаний классов объектов на ЦВМ / В.П. Гладун // Кибернетика. – 1972. - №5. – С. 109 – 117.
7. Гупал А.М. Методы индуктивного вывода и их применение в экспертных системах / А.М. Гупал // Управляющие системы и машины. – 1991. – №7. – С.112–114.
8. Гупал А.М., Цветков А.М. Разработка алгоритмов индуктивного вывода знаний с использованием и листьев и деревьев решений / А.М. Гупал, А.М. Цветков // Управляющие системы и машины. – 1992. – №5/6. – С.21–26.
9. Гупал А.М.Разработка алгоритмов индуктивного вывода, основанных на построении деревьев решений / А.М. Гупал, А.М. Цветков // Кибернетика и системный анализ. – 1993. – №3. – С.174–178.
10. Гупал А.М. Об одном методе индуктивного вывода с подрезанием деревьев решений / А.М. Гупал, А.М. Цветков, А.А.Пономарев // Кибернетика и системный анализ. – 1993. – №5. – С.174– 178.
11. Донской В. И. Алгоритмы обучения, основанные на построении решающих деревьев / В. И. Донской // ЖВМ и МФ. – 1982. – Т. 22. – №4. – С. 963 – 974.
12. Донской В.И. Асимптотика числа бинарных решающих деревьев / В. И. Донской // Ученые записки Таврического национального ун-та им. В. И. Вернадского, серия «Информатика и кибернетика». – 2001. – №1. – С.36–38.
13. Донской В. И. Интеллектуализированная программная система IntMan поддержки принятия решений в задачах планирования и управления / В.И.Донской, В.Ф. Блыщик, А.А. Минин, Г.А. Махина // Искусственный интеллект. – 2002. – №2. – С.406–415.
14. Донской В.И. Исследование алгоритмов распознавания, основанных на построении решающих деревьев: автореф. дисс. на соиск. уч. степени канд. физ.-мат. наук: спец. 01.01.09 «Математическая кибернетика» / В.И. Донской. – М., 1982. – 16 с.
15. Донской В. И. Колмогоровская сложность и ее применение в машинном обучении / В. И. Донской // Таврический вестник информатики и математики. – 2012. – №2. – С. 4 – 35.
16. Донской В. И. Машинное обучение и обучаемость: сравнительный обзор [Электронный ресурс] / В.И.Донской // Intellectual Archive. – 2012. – №933. – 19 с. – Режим доступа: http://www.sciteclibrary.ru/texsts/rus/stat/st4820.pdf
17. Донской В.И. О построении программного обеспечения распознающих систем / В. И. Донской // Программирование. – 1980. – № 2. – С. 87 – 90.

18. Донской В.И. О совместном использовании абдукции, аналогии, дедукции и индукции при синтезе решений / В.И. Донской // Искусственный интеллект. – №2. – 2000. – С. 59 – 66.
19. Донской В. И., Страхов С. Б. Выбор признаков при синтезе решающих деревьев. – Симферополь: Симферопольский ун-т, 1982. – 12 с. (Рукопись деп. в ВИНИТИ, № 1765-82).
20. Донской В. И. Сложность семейств алгоритмов обучения и оценивание неслучайности извлечения эмпирических закономерностей / В. И. Донской // Кибернетика и системный анализ. – 2012. – №2. – С. 86 – 96.
21. Донской В. И. Экспертная система ДУЭЛЬ: реализация дуального подхода для IBM-совместимых компьютеров / В.И. Донской // Динамические системы. – 1994. – Вып. 13. – С. 93 – 98.
22. Донською В.И. Бінарні вирішуючи дерева у задачах інтелектуального аналізу інформації / В.И. Донською, Ю.Ю. Дюличева // Наукові вісті Національного технічного університету "Київський політехнічний Інститут". – 2001. – Вып.5. – С.12 – 18.
23. Донской В.И. Индуктивная модель *r*-корректного эмпирического леса / В.И. Донской, Ю.Ю. Дюличева // Труды международной конференции по индуктивному моделированию. – Львов, 2002. – № 2. – С. 54–58.
24. Донской В.И. Деревья решений с *k*-значными переменными / В.И. Донской, Ю.Ю. Дюличева // Труды Междунар. конф. ”Знание – Диалог – Решение”. – Том 1. – Санкт-Петербург: Изд-во ”Лань”. – 2001. – С.201 – 207.
25. Дюличева Ю. Ю. Оценка VCD r-редуцированного эмпирического леса / Ю. Ю. Дюличева //Таврический вестник информатики и математики. – 2003. – № 2. – С.35–43.
26. Дюличева Ю.Ю. Принятие решений на основе индуктивной модели эмпирического леса / Ю.Ю. Дюличева // Искусственный интеллект. – 2002. – №2. – С.110 – 115.
27. Закревский А. Д. Логика распознавания / А.Д. Закревский. – Минск: Наука и техника, 1988. – 119 с.
28. Закревский А.Д., Торопов Н.Р. Обучение распознаванию образов в булевом пространстве. – В кн.: Самообучающиеся автоматические системы. – М.: Наука, 1966. – С. 67 – 72.
29. Журавлев Ю. И. Об одном классе не всюду определенных функций алгебры логики / Юрий Иванович Журавлев //Дискретный анализ. – 1964. – Вып. 2. – С. 23 – 27.
30. Журавлев Ю. И. Об отделимости подмножеств вершин n-мерного куба / Юрий Иванович Журавлев // Науч. Труды Матеем. ин-та им. В. А. Стеклова. – 1958. – Т.1. – С. 143 – 157.
31. Журавлев Ю. И. Теоретико-множественные методы в алгебре логики / Юрий Иванович Журавлев // Проблемы кибернетики. – 1962. – Вып.2. – С. 5 – 44.

32. Лбов Г.С. Логические функции в задачах эмпирического предсказания / Геннадий Сергеевич Лбов // Вычислительные системы. – Новосибирск, 1978. – Вып. 7. – С. 34 – 64.
33. Лбов Г.С. Об одном алгоритме распознавания в пространстве разнотипных признаков / Г.С. Лбов, В.И. Котюков, А.И. Манохин // Вычислительные системы. – Новосибирск, 1973. – Вып. 55. – С. 108 – 110.
34. Лбов Г.С., Бериков В.Б., Устойчdивость решающих функций в задачах распознавания образов и анализа разнотипной информации / Г. С. Лбов, В.Б. Бериков. – Новосибирск: Изд-во Ин-та математики, 2005. – 218 с.
35. Лбов Г.С., Старцева Н.Г. Логические решающие функции и вопросы статистической устойчивости решений / Г. С. Лбов, Н.Г. Старцева. – Новосибирск: Изд-во Ин-та математики, 1999. – 212 с.
36. Орлов В.А. Применение граф-схемного метода распознавания образов: автореф. дисс. на соиск. уч. степени канд. техн. наук: спец.. 05.13.01 «Техническая кибернетика и теория информации» / В.А.Орлов. – Владивосток, 1974. – 23 с.
37. Растригин Л. А. Коллективные правила распознавания / Л. Растригин, Р. Эренштейн. – М.: Энергия,1981. – 244 с.
38. Рвачев В. Л. Методы алгебры логики в математической физике / В. Л. Рвачев. – К.: Наукова думка, 1974. – 334 с.
39. Слепян В.А. Вероятностные характеристики распределения тупиковых тестов / В.А. Слепян // Дискретный анализ. – 1968. – Вып. 12. – С. 50 – 74.
40. Сироджа И. Б. Системный синтез структурно-аналитических алгоритмов распознавания образов для автоматизации классификационной обработки данных КОД / И. Б. Сироджа // Модели и системы обработки данных. – Харьков, 1978. – Вып. 2. – С. 79 – 102.
41. Соловьев Н. А. Тесты / Н.А. Соловьев. – Новосибирск: Наука, 1978. – 190 с.
42. Breiman L. Classification and regression trees / L. Breiman, J. H. Friedman, R. A. Olshen, C. J. Stone.- Calif.: Wadsworth: 1984. – 58 p.
43. Brodley C. E., Utgoff. P. E. Multivariate decision trees / C. E. Brodley, P. E. Utgoff // Machine Learning. – 1995. – Vol. 19. – P. 45 –77.
44. Devroye L. A Probabilistic Theory of Pattern Recognition / Luc Devroye, László Györfi, Gábor Lugosi. – Springer-Verlag: New York, 1996. – 636 p.
45. Friedman J. H. Multivariate Adaptive Regression Splines / J. H. Friedman // The Annual of Statistics. – 1991. – Vol. 19. – P. 1 –141.
46. Holte R. C. Very simple classification rules perform well on most commonly used datasets / R. C. Holte // Machine Learning. – 1993. – Vol.11. – P.63-90.
47. Hunt E. B. Experiments in Induction / Earl B. Hunt, Janet Marin, Philip J. Stone. – N. Y.: Academic Press, 1966. – 247 p.
48. Hyafil L, Rivest R. L. Constructing Optimal Binary Decision Trees is NP-Complete / L. Hyafil, R.L. Rivest // Information Proc. Letters. – 1976. – Vol. 3. – №1. – P. 15 –17.

49. Kass G. V. An exploratory technique for investigating large quantities of categorical data / G. V. Kaas // Applied Statistics. – 1980. – Vol.29(2). – P.119-127.
50. Kearns M., Mansour Y. On the boosting ability of top-down decision tree learning algorithms / M. Kearns, Y. Mansour // Journal of Computer and Systems Sciences. – 1999. – Vol. 58(1). – P.109 –128.
51. Levene H. Robust tests for equality of variances / H. Levene // Contributions to Probability and Statistics / Ed. I. Olkin, Palo Alto. – Stanford University Press: 1960. – P. 278-292.
52. Li L. Data Complexity in Machine Learning and Novel Classification Algorithms / Ling Li / Doctor of Phil. Thesis. – Pasadena, California: California Institute of Technology, 2006. – 103 P.
53. Loh W.-Y., Shin Y.-S. Split Selection Methods for Classi_cation Trees / Wei-Yin Loh and Yu-Shan Shih // Statistica Sinica. – 1997. – Vol. 7. – P. 815 – 840.
54. Maimon O., Rokach L. Data Mining and Knowledge Discovery. Handbook, 2nd ed.// Oded Maimon, Lior Rokach Springer: New York, 2010. – 1285 p.
55. Marques de Sa J. P. New Results on Minimum Error Entropy Decision Trees / Joaquim P. Marques de Sa, Raquel Sebastiao, and Joao Gama, Tanja Fontes // CIAPR'11 Proceedings of the 16th Iberoamerican Congress conference on Progress in Pattern Recognition, Image Analysis, Computer vision, and Applications. Chile, Pucon. – 2011. – P. 355 – 362.
56. Marques de Sa J. P. Tree Classifiers Based on Minimum Error Entropy Decisions / Joaquim P. Marques de Sa, Raquel Sebastiao, and Joao Gama // Canadian Journal on Artificial Intelligence, Machine Learning & Pattern Recognition. – 2011. – Vol. 2. – № 3. – P. 41 – 55.
57. Mehta M., Agrawal R., Rissanen J. SLIQ: A fast scalable classifier for data mining / Manish Mehta, Rakesh Agrawal, Jorma Rissanen / In Advances in Database Technology – EDBT '96 .Avignon, France, March 1996 // Lecture Notes in Computer Science. – 1996. – Vol. 1057. – P. 18-32.
58. Muller W., Wysotzki F. Automatic construction of decision trees for classification / W. Muller, F. Wysotzki // Annals of Operations Research. – 1994. – Vol. 52. – P. 231-247.
59. Novikoff A. On convergence proofs for perceptrons / A. Novikoff // In Proc. of the Symp. on Mathematical Theory of Automata. – Brooklyn, New York: Polytechnic Institute of Brooklyn, 1962. – Vol. 12. – P. 615– 622.
60. Pearl J. Capacity and error estimates for Boolean classifiers with limited complexity / Judae Pearl // IEE Trans. on PAMI. – 1979. – Vol. 1. – No4. – 350 – 356.
61. Preparata F. P. An Estimate on the Length of Diagnostic tests / Preparata F.P. // IEEE Trans. Realiabl. – 1969. – Vol.18. – N3. – P. 131 – 136.
62. Quinlan J.R. C4.5: Programs for Machine Learning / John Ross Quinlan. – Morgan Kaufmann: 1993. – 302 c.
63. Quinlan J.R. Induction of decision trees // Machine Learning. – 1986. – Vol. 1. P. 81–106.

64. Rastogi R., Shim K. PUBLIC: A Decision Tree Classifier that Integrates Building and Pruning / Rajeev Rastogi, Kyuseok Shim // Proceedings of the 24th VLDB Conference August 1998, USA. – New York:1998. – P. 404 – 415.
65. Shih Yu-Shan. Families of splitting criteria for classification trees / Yu-Shan Shih // Statistics and Computing. – 1999. – Vol. 9. – P. 309-315.
66. Sonquist J. A. Searching for structure (alias-AID-III) // John A. Sonquist, Elizabeth Lauh Baker, James N. Morgan. – Institute for Social Research, University of Michigan: 1971. – 287 P.
67. Stasis A.C. Using decision tree algorithms as a basis for a heart sound diagnosis decision support system / A.C.Stasis, E.N.Loukis, S.A. Pavlopoulos, D.Koutsouris // Information Technology Applications in Biomedicine, 2003. 4th International IEEE EMBS Special Topic Conference, April 2003. 354 - 357
68. Taylor P. C., Silverman B. W. Block diagrams and splitting criteria for classification trees / P. C. Taylor, B. W. Silverman // Statistics and Computing. – 1993. Vol.3. – P. 147–161.
69. Vitanyi P., Li M. Ideal MDL and Its Relation to Bayesianism Bayesianism / Paul M.B. Vitanyi, Ming Li // In Proc. ISIS: Information, Statistic and Induction in Science. – Singapore: World Scientic, 1996. – P. 282 – 291.
70. Warton S.W. A Contextual Classification Method for Recognizing Land Use Pattering in High Resolution Remotely Sensed Data / S.W. Warton // Pattern Recognition/ - 1982. – Vol. 15. – No4. – P.317 – 324.

6. Оценивание точности и надежности классифицирующих алгоритмов

«В торговле, политике и мало ли где еще
оказывается порой заслугой и гениальным решением
выдать черное за белое, у нас – никогда»
Г. Гессе. Игра в бисер

6.1. Основные понятия

Оценивание классификаторов как гипотез, синтезированных по обучающей выборке различными алгоритмами обучения, связано со многими факторами. Приходится учитывать и модель генеральной совокупности используемых выборок, и способ извлечения выборки из генеральной совокупности, и особенности алгоритма обучения – синтеза гипотез. Также имеет значение поход к вычислению оценки точности. Он может осуществляться по всей заданной выборке, методом скользящего контроля или по тестовой выборке. Наконец, оценивание зависит и от того, какая модель обучения берётся за основу.

Выделим три основные группы методов оценивания классификаторов:

1. Оценивание синтезированных классификаторов по всей заданной обучающей выборке.
2. Оценивание по методу скользящего контроля.
3. Оценивание по независимой контрольной выборке.

Оценивание синтезированных классификаторов по всей выборке, представленной для обучения, приводит к получению смещенных оценок эмпирических ошибок. Это объясняется тем, что оценивание производится по той же выборке, которая использовалась для обучения. *Но именно этот препятствующий непосредственному оцениванию точности классификаторов факт и привёл к парадигме обучаемости как способности к обобщению информации, представленной обучающей выборкой.*

Основополагающие результаты в рассматриваемом направлении были получены В.Н. Вапником и А.Я. Червоненкисом [2-4]. Эти результаты базируются на важнейшем понятии теории машинного обучения – емкости классов решающих функций.

Постановку задачи обучения, предполагающую, что исходная обучающая выборка безошибочна, а в семействе используемых решающих правил H имеется то, которое является истинно правильным, называют *детерминистской* [4]. Предположим, что класс H является конечным: $|H| = N$. Обучение по методу эмпирического риска приведет к минимальной эмпирической ошибке, равной нулю, поскольку в классе H име-

ется правильный классификатор. Но этот минимум, вообще говоря, может достигаться не на единственном решающем правиле семейства H. Если правило h в действительности имеет вероятность ошибки большую ε, то «показать» безошибочный результат l раз на выборке длины l оно сможет с вероятностью $p(\varepsilon,l) < (1-\varepsilon)^l$. Тогда вероятность того, что хотя бы одно ошибочное правило семейства H доставит минимум эмпирического риска равный нулю можно оценить как

$$P(\varepsilon,l,H) < N(1-\varepsilon)^l.$$

Из уравнения $N(1-\varepsilon)^l = \delta$ получаем $l = \dfrac{\ln N - \ln\delta}{-\ln(1-\varepsilon)}$. Приведенные рассуждения являются основой доказательства следующей

Теоремы 6.1[4]. Пусть из множества, состоящего из N решающих правил, выбирается такое правило, которое на обучающей последовательности не совершает ни одной ошибки. Тогда с вероятностью $1-\delta$ можно утверждать, что вероятность ошибочной классификации при использовании этого выбранного правила на всей генеральной совокупности объектов будет меньше ε, если *только длина обучающей последовательности будет не меньше*

$$l = \frac{\ln N - \ln\delta}{-\ln(1-\varepsilon)}. \qquad (6.1)$$

Для задачи обучения классификации в общей постановке и неограниченным семейством решающих правил H для оценивания точности и надёжности, а также требуемых длин обучающих выборок используется функция роста m^H семейства H. В основе оценивания лежит неравенство

$$\Pr(\sup_{h\in H} | \nu_h^{(l)} - P(h) | > \varepsilon) \le 4m^H(2l)\exp(-\frac{1}{8}\varepsilon^2 l) \qquad (6.2)$$

где $\nu_h^{(l)} = \dfrac{n_h^{(l)}}{l}$, $n_h^{(l)}$ – число ошибок, допущенное классификатором $h \in H$, выбранным в результате обучения, на обучающей выборке длины l; $P(h)$ – вероятность ошибки классификатора h. Используя неравенство (6.2), можно получить условие для требуемой длины обучающей выборки

$$l \ge \frac{16}{\varepsilon^2}(VCD(H)\cdot\ln\frac{16\cdot VCD(H)}{\varepsilon^2} + \ln\frac{4}{\delta}) \text{ [3].}$$

Уточнение требуемых длин обучаемых выборок приведено в работе [4] на основе неравенства

$$\Pr(\sup_{h\in H} | \nu_h^{(l)} - P(h) | > \varepsilon) < 6m^H(2l)\exp(-\frac{1}{16}\varepsilon^2(l-1)).$$

Получена оценка

$$l \geq \frac{32}{\varepsilon^2} VCD(H) \cdot (1 - \frac{\ln\frac{\delta}{6}}{VCD(H)} - \frac{\varepsilon^2}{16} - \ln\frac{\varepsilon^2}{32})$$

и её уточнение [4, с. 280]

$$l \geq \frac{2 \cdot VCD(H)}{\varepsilon^2} \cdot (1 - \frac{\ln\frac{\delta}{5}}{VCD(H)} - \frac{\varepsilon^2}{2} - \ln\frac{\varepsilon^2}{2}).$$

Определение 6.1. Если обучаемость имеет место, то функцию $l = l(\varepsilon, \delta)$, которая определяет наименьшую длину выборки, достаточную для того, чтобы полученный в результате обучения классификатор h гарантировал *на этой выборке* точность ε с надежностью $1-\delta$, называют *выборочной сложностью* (*sample complexity*).□

Смысл этого определения состоит в том, *чтобы указать такую длину обучающей выборки, которая гарантирует* (ε, δ) *обучаемость при неизвестном заранее, а синтезируемом в процессе обучения классификаторе*. Известными при исследовании обучаемости являются только алгоритм (метод) синтеза, некоторый конечный набор данных – наблюдений и, как правило, семейство алгоритмов, в котором отыскивается классификатор. Причем это семейство может быть задано неявно, как, например в kNN модели. Поэтому *выборочная сложность оценивается при условии, когда априорная неопределенность гораздо выше, чем в случае оценки конкретного выбранного классификатора по тестовой выборке*.

Поскольку обучаемость всегда определяется тем, какой именно алгоритм $A: \mathrm{X}^l \to H$ используется для обучения и какова область его значений (множество гипотез H, из которого будет выбран классификатор) то пару $< A, H >$ будем называть *моделью обучения*. Модель обучения может обладать свойством обучаемости или не обладать им. Если она этим свойством обладает, то для неё имеет смысл выборочная сложность.

Для того, чтобы понять различие выборочной сложности от оценки достаточной длины выборки для достижения заданной точности единственным, уже выбранным некотором методом классификатором, рассмотрим следующую ситуацию.

Пример. Предположим, дан некоторый классификатор – алгоритм h, не являющийся обучаемым. Указать для него выборочную сложность

$l = l(\varepsilon, \delta)$ невозможно ни для каких значений (ε, δ). Пусть такой классификатор определяется следующим образом:

$$h(\tilde{x}) = \begin{cases} \alpha_j, \ \textit{если} \ \exists j : \tilde{x} = \tilde{x}_j; \\ \xi, \ \textit{если} \ \forall j : \tilde{x} \neq \tilde{x}_j, \end{cases}$$

где $(\tilde{x}_j, \alpha_j)_{j=1}^{l}$ – обучающая выборка, а ξ – псевдослучайная величина, генерируемая алгоритмом-датчиком и принимающая почти равновероятно значения 0 и 1. Классификатор h на обучающей выборке дает нулевую ошибку. Но почти всюду на множестве допустимых объектов этот классификатор даёт случайный равновероятный ответ. Можно считать, что если в контрольной выборке объекты двух классов встречаются равновероятно, то такой классификатор h будет ошибаться на любом из них с вероятностью $p = \frac{1}{2}$.

Предположим, пользователь, оценивающий качество этого классификатора h по некоторой доступной, но чересчур короткой тестовой выборке длины $l = 8$, ничего о классификаторе не знает. Он получает (случайно!) частоту ошибки ν классификатора h как черного ящика по контрольной выборке. Понятно, что с некоторой вероятностью может выпасть значение числа ошибок, например, $k = 2$, и тогда частота ошибки окажется равной $\nu = \frac{1}{4}$. Вероятность такого события при $p = (1-p) = q = \frac{1}{2}$ равна $P_n(k) = C_n^k p^k q^{n-k} \approx 0.1$, так что, несмотря на эту небольшую вероятность, оцениваемое событие действительно может произойти.

Вычисляя надежность доверительного интервала для вероятности ошибки p классификатора при помощи неравенства

$$P\{\nu - \varepsilon < p < \nu + \varepsilon\} > 1 - \delta$$

при $\varepsilon = \frac{1}{8}$ и $\delta = 0.1$, по формуле $l \geq \frac{1}{4\varepsilon^2 \delta}$ (см. ниже), пользователь получает $l \geq \frac{1}{4 \cdot (1/8)^2 \cdot 0.1} = 160$ и в итоге заключает, что неравенство $P\ \{\ \frac{1}{8} < p < \frac{3}{8}\ \} > 1 - \delta = 0.9$ будет выполняться при *требуемой длине* контрольной выборки $l > 160$.

Затем, повторяя расчеты с более длинной контрольной выборкой пользователь удивляется, что частота ошибок близка к 0.5. □

Приведенный пример позволяет сделать следующие заключения.

Следствие 6.1. Оценивание точности классификатора и требуемой длины выборки для достижения нужной точности не имеет смысла, если классификатор получен в модели, для которой нет обучаемости.

Следствие 6.2. *Выборочная сложность* модели $< A, H >$ синтеза классификатора *и длина выборки, достаточная для достижения одним единственным выбранным классификатором* $h \in H$ *нужной точности и надежности – принципиально различные понятия.* □

Так что сетовать на теорию Вапника-Червоненкиса, PAC обучаемость, $k-$сжатие и другие модели, определяющие обучаемость, не следует. Оценки выборочной сложности в большинстве случаев совпадают, «упираясь» в предельно возможное сужение области неопределенности выбора классификатора.

Широкое определение выборочной сложности может уточняться в зависимости от того, в каком смысле понимается обучаемость.

Качество классификатора зависит от *длины обучающей выборки*: $Err(h) = Err(h,l)$, поэтому сложность выборки находится в результате решения неравенств вида

$$\Pr(Err(h,l) > \varepsilon) < \sigma .$$

В случае конечного семейства гипотез H оценка длины выборки, обеспечивающей обучаемость для любого согласованного с выборкой концепта $h \in H$ (выборочной сложности), имеет вид:

$$l(H,\delta,\varepsilon) \geq \frac{1}{\varepsilon}\ln\frac{|H|}{\delta}. \qquad (6.3)$$

Оценка (6.3) получается из оценки (6.1) при использования приближенного равенства $-\ln(1-\varepsilon) \approx \varepsilon$, справедливого для малых ε.

Для равномерно обучаемого класса H конечной емкости $VCD(H)$, из которого извлекается классификатор h, известны следующие нижняя (при $0 < \varepsilon < 1/2$) и верхняя оценки выборочной сложности [13]:

$$\max\{\ \frac{1-\varepsilon}{\varepsilon}\log(\frac{1}{\delta}),\ (1-2(\varepsilon(1-\delta)+\delta))\cdot VCD(H)\ \} \leq l(\varepsilon,\delta);$$

$$l(\varepsilon,\delta) \leq \max\{\frac{8\cdot VCD(H)}{\varepsilon}\log(\frac{13}{\varepsilon}),\ \frac{4}{\varepsilon}\log(\frac{2}{\delta})\}.$$

Эти оценки незначительно отличаются от оценки выборочной сложности, требуемой для $PAC-$ обучаемости, которая основана на вапников-

ской ёмкости $VCD(H)$ семейства концептов H, из которого извлекается концепт h:

$$\max(\frac{VCD(H)-1}{32\varepsilon}, \frac{1}{\varepsilon}\ln\frac{1}{\delta}) < l(H,\delta,\varepsilon) \le \frac{4}{\varepsilon}(VCD(H)\log\frac{12}{\varepsilon} + \log\frac{2}{\delta}).$$

Оценка длины выборки, которая требуется для *PAC* обучаемости в сложностной версии *Occam's Razor* теоремы, основанной на длине описания $s(h) \le n(g)^{\alpha} l^{\beta}$ выбираемого при обучении концепта h, имеет вид [14]

$$l(\varepsilon,\delta) = \max(\frac{2}{\varepsilon}\ln\frac{1}{\delta}, (\frac{(2\ln 2)n(g)^{\alpha}}{\varepsilon})^{\frac{1}{1-\beta}});$$

здесь $n(g)$ – длина бинарного описания искомого целевого концепта g, а величины α и β являются коэффициентами сжатия целевого концепта g и обучающих данных соответственно.

При обучении сжатием

$$l(\varepsilon,\delta) = \max(\frac{2}{\varepsilon}\ln\frac{2}{\delta}, (\frac{(2\ln 2)M_h}{\varepsilon})^{\frac{1}{1-\beta}}),$$

где M_h – оценка сверху колмогоровской сложности $KP(h)$ любой выбранной гипотезы $h \in H$.

Для любой схемы компрессии, имеющей ядерный размер k, (ε,δ)-обучаемость имеет место при сложности выборки, определяемой как

$$l(\varepsilon,\delta) = \max\{\frac{2}{\varepsilon}\ln\frac{1}{\delta}, \frac{2k}{\varepsilon}\ln\frac{4k}{\varepsilon} + 2k\}. \qquad (6.4)$$

Ядерный размер k, содержащийся в оценке (6.4), характеризует *минимальную длину подвыборки*, выделенной из обучающей выборки длины l, по которой можно построить корректный на всей выборке классификатор.

Выборочная сложность (6.4) близка к полученной в [15]:

$$l(\varepsilon,\delta) = \max(\frac{4}{\varepsilon}\ln\frac{2}{\delta}, \frac{8\cdot d}{\varepsilon}\ln\frac{8d}{\varepsilon}),$$

где $d = VCD(H)$ класса функций H, использованного при обучении. Можно заметить, что эти оценки достаточно близки в случае $k \approx d$.

Для сжатия размера k, согласно теореме 4.23, выполняется двойное неравенство

$$VCD(H) < k \le VCD(H)\cdot\log l,$$

такое же, как для колмогоровской сложности этого семейства

$$VCD(H) < K_l(H) \le VCD(H)\log l.$$

Таким образом, колмогоровская сложность семейства $K_l(H)$, размер сжатия k и колмогоровская префиксная сложность $KP(h)$ оптимальной выбранной гипотезы $h \in H$ являются близкими по своему числовому значению величинами, и оценки выборочной сложности на их основе различаются незначительно. Это можно объяснить со следующей точки зрения. *Для решения задачи обучения используется только решающая выборка и определение класса гипотез в целом. На уровне этой неполной информации невозможно преодолеть порожденную ею неопределенность, и различные оптимальные оценки этой неопределенности* (*энтропии*) *просто обязаны быть близкими.*

Приведенные выше оценки выборочной сложности являются сильно завышенными. Это следует из многочисленных результатов практического применения алгоритмов машинного обучения к различным семействам гипотез, таких как нейронные сети, деревья решений, потенциальные функции и др., и подтверждается более тонкими теоретическими исследованиями.

В работах [5,7,19] с целью получения оценок точности классификаторов исследуются и учитываются особенности применяемых алгоритмов обучения и семейств гипотез, из которых выбирается классификатор, и показывается, что иногда возможно получение на порядок лучших оценок точности.

Известно, что множество вычислимых функций эффективно перечислимо. Используя минимальный номер n вычислимой функции g в соответствующей нумерации можно определить длину описания $len(g)$ этой функции как длину бинарной строки, представляющей её минимальный номер n. В работе [17] доказано существование для любой неизвестной классифицирующей вычислимой функции g, имеющей длину описания $len(g)$, и любых $\varepsilon, \delta > 0$ такого классифицирующего алгоритма h, что

$$l(\varepsilon, \delta, h) \le \max\{\ len(g)\frac{8}{\varepsilon}\log\frac{13}{\varepsilon},\ \frac{4}{\varepsilon}\log\frac{2}{\delta}\ \}.$$

Этот результат представляется особенно важным, поскольку даёт оценку выборочной сложности для класса вычислимых функций (алгоритмов), и эта оценка близка к оценкам, приведенным выше и полученным без оговорки на вычислимость целевых и применяемых для обучения концептов.

Параметрические оценки выборочной сложности и точности также представляют большой интерес, поскольку успешность машинного обучения в наибольшей степени определяется привлечением дополнительной информации и использованием моделей, адекватных решаемым задачам. В работе [1] приведена параметрическая оценка для модели минимизации среднего риска в байесовской постановке, когда случайный вектор пара-

метров – априорных вероятностей классов – подчиняется распределению Дирихле.

Приведем для представления об этих результатах только вид полученной в [1] оценки:

$$P(P_f(\tilde{\Theta}) \geq \varepsilon) \leq H(a,c;l)e^{-\varepsilon l} = \delta;$$

$$\varepsilon = \frac{1}{l}(\ln\frac{H(a,c;l)}{\delta}), \quad l = \frac{1}{\varepsilon}(\ln\frac{H(a,c;l)}{\delta}),$$

где $P_f(\tilde{\Theta})$ – вероятность среднего риска ошибки классификатора f, $H(a,c;l)$ – гипергеометрическая функция Куммера, a и c – некоторые параметры, обобщающие число ошибок, объем выборочных данных и совокупность параметров распределения Дирихле.

Известны попытки вычисления коррекции смещения оценок, полученных в результате обучения[18], но они не находят практического применения.

Оценивание по методу скользящего контроля. Метод скользящего контроля ($k-fold$ *Cross Validation*) заключается в следующем. Из заданной выборки длины L поочередно исключаются $k < L$ элементов. Получаются две выборки с длинами $l = L - k$ и l. На первой производится обучение, а по второй – как контрольной – вычисляется частота ошибок ν_i построенного в результате обучения классификатора. Такой процесс повторяется C_L^k раз (можно образно сказать, что контрольная подвыборка «скользит» по выборке длины L). В итоге получается оценка точности алгоритма обучения $\bar{\nu} = \frac{1}{C_L^k}\sum_{i=1}^{C_L^k}\nu_i$. При значении $k = 1$ скользящий контроль соответствует правилу LOO и нахождению ν_{LOO} ошибки.

Известно, что когда исходная обучающая выборка состоит из случайно и независимо выбранных из генеральной совокупности объектов, средняя ошибка скользящего контроля даёт несмещенную оценку вероятности ошибки. Однако для оценивания точности классификаторов нужно знать еще и дисперсию этой ошибки. Считается, что такие оценки неизвестны – их найти до настоящего времени не удалось. А сравнительно недавно выяснилось, что несмещенных оценок дисперсии для $k-fold$ скользящего контроля не существует [12].

Несмотря на отсутствие необходимых теоретических результатов для получения точных оценок ошибок классификации методами $k-fold$ скользящего контроля, большой интерес представляют результаты *статистического моделирования для получения приближенных оценок*

скользящего контроля [9,10]. В этом направлении выполнена работа [9], в которой в частных случаях установлена *практическая равноценность оценок LOO* и $k-fold$ скользящего контроля. *Первая из этих двух оценок оказалась немного точнее*, хотя в ряде случаев дала бóльшую погрешность [9].

В работе [16] установлена полиномиальная (по сложности) обучаемость для рекуррентных отображений (решающих функций) персептронного типа. *Интерес представляет более всего сам исследуемый класс.*

Последовательность $\vec{c} = (c_1,...,c_{n+q}) \in R^{n+q}$, где $n > 0, q \geq 0$ – целые числа, называется n-рекурсивной, если существуют вещественные числа $r_1,...,r_n$ такие, что $c_{n+j} = \sum_{i=1}^{n} c_{n+j-i} r_i$, $j = 1,...,q$. Рассматривается класс функций

$$F_{n,q} = \{ f_{\vec{c}} : \vec{c} - \textit{вектор рекурсивных параметров};\ f_{\vec{c}} : R^{n+q} \to \{-1;1\} \}.$$

В этот класс входит, например, персептронный классификатор

$$x_1,...,x_{n+q} \mapsto sign(\sum_{i=1}^{n+q} c_i x_i),$$

в котором параметры пересчитываются рекурсивно.

Теорема 6.2[16]**.**

$$\max\{\ n,\ n \cdot [\log(1 + \frac{q-1}{n})]\} \leq VCD(F_{n,q});$$

$$VCD(F_{n,q}) \leq \min\{ n+q,\ 18n + 4n\log(q+1)\}.$$

Теорема 6.3[16]**.** При любом $n > 0$ выборочная (ε, δ) сложность обучения при использовании класса $F_{n,q}$ и любой неизвестной дихотомии, представленной обучающей выборкой, полиномиальна по q, $n+q$ и L, где L – число бит для представления чисел.

6.2 Оценивание точности классификаторов в комбинаторной теории переобучения

В рамках комбинаторной теории переобучения К.В. Воронцова (обозначаемой далее ***VCT*** *– **V**orontsov' **C**ombinatorial **T**heory of overfitting*) изучается проблема надёжности синтезированных классификаторов по неполной информации в дискретной постановке. Изложим основные положения и результаты VCT, следуя работам [5-7, 19].

Пусть существует бинарная *матрица ошибок*. *Её строки соответствуют объектам, столбцы — алгоритмам; единица в матрице означает,*

что данный алгоритм ошибается на данном объекте. Требуется найти классификатор (алгоритм), число ошибок которого как можно меньше, при условии, что наблюдается не вся матрица ошибок, а только случайное подмножество её строк. Будем говорить, что в таком случае алгоритм обучается по наблюдаемым данным. Те данные, которые не наблюдаются, называются скрытыми. Предполагается, что все разбиения множества объектов на l наблюдаемых и k скрытых могут реализоваться с равной вероятностью. Способность к обобщению (обучаемость) имеет место, если частота ошибок найденного классификатора на скрытых объектах будет достаточно мала. В VCT теории *переобучением называют ситуацию, когда частота ошибок найденного классификатора на скрытых объектах существенно выше частоты его ошибок на наблюдаемых объектах*. VCT теория направлена на получение оценок вероятности переобучения и полного скользящего контроля.

В данном пункте будем применять обозначения, принятые в VCT.

$X^L = \{x_1,\ldots, x_L\}$ – генеральная выборка из L объектов.

$[X]^l$ – множество всех l элементных подмножеств из X_L.

A – класс алгоритмов – семейство гипотез.

$I : A \times X^L \to \{0,1\}$ – бинарная функция ошибок (потерь).

$n(a, X)$ – число ошибок алгоритма $a \in A$ на подвыборке $X \subset X_L$.

$\nu(a, X) = \frac{1}{|X|} n(a, X)$ – частота ошибок на подвыборке X.

$(I(a, x_i))_{i=1}^L$ – (бинарный вектор) ошибок алгоритма $a \in A$ на X_L.

$A_m = \{a \in A : n(a, X^L) = m\}$ – m -слой семейства A.

$q(a) = |\{a' \in A_{n(a,X^L)+1} : I(a,x) \le I(a',x), x \in X^L\}|$ – верхняя связность алгоритма a – *число алгоритмов в слое*, следующем за слоем, в котором находится a, *допускающих ошибки на тех же объектах, что и алгоритм* a, *плюс еще одна ошибка*; запись $I(a,x) \le I(a',x)$ соответствует отношению предшествования для пары соседних булевых векторов длины L. Это отношение порождает ориентированный граф связности множества A, и тогда $q(a)$ – число рёбер, исходящих из вершины a.

$p(a)$ – нижняя связность алгоритма a – число алгоритмов в предыдущем слое, вектор ошибок которых отличается от вектора ошибок алгоритма a только на одном объекте.

$r(a)$ – неоптимальность алгоритма a – число объектов, на которых данный алгоритм допускает ошибку, таких, что существует другой

алгоритм, не допускающий ошибки на данном объекте и на объектах, на которых не ошибается алгоритм $a \in A$. Неоптимальность оценивает возможность улучшения качества алгоритма a некоторым другим, лучшим алгоритмом из A.

$\mu : X \to a$ – метод (алгоритм) обучения, ставящий в соответствие данной выборке X алгоритм $\mu X = a \in A$.

X и $\overline{X}$ – разбиение X_L на две подвыборки: обучающую X и скрытую контрольную $\overline{X}$.

$\delta_\mu(X) = \nu(\mu X, \overline{X}) - \nu(\mu X, X)$ – отклонение частот ошибок алгоритма $\mu X = a$ на двух подвыборках X и $\overline{X}$; метод μ приводит к переобучению, если $\delta_\mu(X) \geq \varepsilon$, $\varepsilon > 0$.

$Q_\varepsilon = P[\delta_\mu(X) \geq \varepsilon] = \frac{1}{C_L^l} \sum_{X \in [X]^l} [\delta_\mu(X) \geq \varepsilon]$ – вероятность переобучения (функционал обучающей способности) , определяемая при условии, что все разбиения X_L на обучающую выборку длины l и скрытую контрольную длины $k = L - l$ равновероятны (*слабая или перестановочная вероятностная аксиоматика* VCT).

Поскольку оценка скользящего контроля, взятая по всем разбиениям $[X]^l$, является несмещенной (вследствие стабилизации усреднением), $CCV(\mu, X^L) = \mathrm{E}\,\nu(\mu X, \overline{X})$, то Q_ε оценивает отклонение вероятности ошибки от её частоты на обучающей выборке, которая, как правило, является заниженной.

Использование слабой вероятностной аксиоматики, скользящего контроля, учет свойств метода обучения и применяемых классов гипотез позволили получить *оценки обучаемости* существенно лучшие, чем оценки, основанные на применении VCD класса гипотез, из которого выбирается алгоритм классификации при обучении.

Теорема 6.4. Для любого алгоритма a, любых $X^l, \varepsilon \in (0,1)$

$$Q_\varepsilon(a, X^L) \leq H_L^{l,m}\left(\frac{l}{L}(m - \varepsilon k)\right),$$

где $m = n(a, X^L)$ – число ошибок алгоритма a на полной выборке X^L,

$H_L^{l,m}(s) = \sum_{t=0}^{|s|} \frac{C_m^s C_{L-m}^{l-t}}{C_L^l}$ – функция гипергеометрического распределения.

Теорема 6.5. Если $\mu-$ метод минимизации эмпирического риска, то для любых $X^l, \varepsilon \in (0,1)$

$$Q_\varepsilon(\mu, X^L) \le \sum_{a\in A} \frac{C_{L-q-r}^{l-q}}{C_L^l} H_{L-q-r}^{l-q,m-r}\left(\frac{l}{L}(m-\varepsilon k)\right), \qquad (6.5)$$

где $q = q(a)$ – верхняя связность алгоритма a, $r = r(a)$ – неоптимальность алгоритма a, m – число ошибок алгоритма a на полной выборке X^L, $H_L^{l,m}(s)$— функция гипергеометрического распределения. При $q = p = 0$ оценка (6.5) переходит в VC оценку.

Лемма. Если в качестве метода обучения μ взять метод максимизации отклонения частот $\mu(X) = \arg\max_{a\in A} \delta_a(X)$, то функционал обучающей способности совпадает с функционалом равномерного отклонения

$$Q_\varepsilon(\mu, X^L) = P[\delta_\mu(X) \ge \varepsilon] = \tilde{Q}_\varepsilon(A, X^L) = P[\max_{a\in A} \delta_a(X) \ge \varepsilon].$$

Теорема 6.6. Для любых $A, X^l, \varepsilon \in (0,1)$

$$\tilde{Q}_\varepsilon(A, X^L) \le \sum_{a\in A} \frac{C_{L-q-p}^{l-q}}{C_L^l} H_{L-q-p}^{l-q,m-p}\left(\frac{l}{L}(m-\varepsilon k)\right), \qquad (6.6)$$

где $q = q(a)$ – верхняя связность алгоритма a, $p = p(a)-$ нижняя связность алгоритма a, m - число ошибок алгоритма a на полной выборке X^L, $H_L^{l,m}(s)$— функция гипергеометрического распределения. При $q = p = 0$ оценка (6.6) переходит в VC оценку.

Теорема 6.7. Если векторы ошибок всех алгоритмов из A попарно различны, $n(a_0, X^L) = 0$ и Δ_{mq} – число алгоритмов в m-том слое со связностью q, то $Q_\varepsilon \le \sum_{m=]\varepsilon k[}^{k} \sum_{q=0}^{L} \Delta_{mq} \frac{C_{L-m-q}^{l-q}}{C_L^l}$.

Замечание. Нумерация гипотез семейства A по неубыванию числа ошибок на $n(a_s, X^L)$, $s = 0,1,...,$ определяет лучший алгоритм a_0. Тогда условие $n(a_0, X^L) = 0$ выражает факт существование корректного на генеральной выборке алгоритма в семействе A. □

Общий поход комбинаторной теории предполагает уточнение оценок точности в каждом отдельном классе методов и алгоритмов. В этом направлении выполнен ряд исследований. В работе [8] получены и исследованы комбинаторные оценки вероятности переобучения для логических

правил, имеющих вид пороговых конъюнкций над заданным подмножеством вещественных признаков.

Преимущества приемов оценивания, разработанных в рамках теории VCT:

а) существенно более высокая точность, чем получаемая на основе теории Вапника-Червоненкиса и других моделей обучения;

б) учитываются свойства индивидуальной модели обучения, что сужает неопределенность.

Недостатки VCT оценок:

а) сложность их вычисления;

б) нет аналитической формулы, выражающей сложность обучающей выборки; вследствие этого трудно сравнивать результаты оценивания выборочной сложности на основании различных теорий;

в) необходимость отыскания новых оценок для каждой новой исследуемой модели и трудоёмкость такой научной работы.

6.3 Оценивание по независимой контрольной выборке

При оценивании по независимой контрольной выборке не возникает никаких проблем, связанных с «подгонкой» классификатора, поскольку контрольная выборка применяется к фиксированному решающему правилу уже после того, как оно выбрано. Оценки вероятности ошибки по контрольной выборке являются несмещенными. Если (неизвестная!) вероятность ошибки выбранного в результате обучения классификатора равна p, то схема её оценивание соответствует вероятностной модели, соответствующей l независимым испытаниям с двумя исходами, которую называют схемой Бернулли [11]. Нас будет интересовать частота ошибок $\nu = \frac{k}{l}$ при независимых испытаниях единственного данного классификатора, проведенных на l примерах контрольной выборки, где k – суммарное число ошибок в l испытаниях. Известно, что математическое ожидание числа ошибок $\mathrm{E}k = np$, дисперсия $\mathrm{D}k = npq$, но поскольку p и $q = 1 - p$ неизвестны, то $\mathrm{E}k$ и $\mathrm{D}k$ невозможно определить точно, а можно только пытаться оценить.

Рассмотрим событие $\{|\nu - p| > \varepsilon\} = \{|\frac{k}{l} - p| > \varepsilon\}$.

Использование неравенства Чебышева позволяет получить оценку

$$P\{|\frac{k}{l} - p| \geq \varepsilon\} \leq \frac{\mathrm{D}(k/l)}{\varepsilon^2} = \frac{\mathrm{D}(k)}{l^2\varepsilon^2} = \frac{npq}{l^2\varepsilon^2} = \frac{pq}{l\varepsilon^2} \leq \frac{1}{4l\varepsilon^2}, \qquad (6.7)$$

что позволяет получить приемлемую оценку вероятности того, что частота ошибок классификатора на контрольной выборке на отклонится от соответствующей вероятности на величину, большую или равную ε. Приняв $\frac{1}{4l\varepsilon^2} = \delta$, получаем (ε, δ)-оценку требуемой длины выборки $l \geq \frac{1}{4\varepsilon^2\delta}$. Например, при $\varepsilon = 0{,}1$ и $\delta = 0{,}1$ понадобится выборка длины $l \geq \frac{1}{4(0.1)^2 \cdot 0.1} = 250.$

Теорема 6.8. Для любого заданного классификатора, дающего бинарные ответы $\{0,1\}$, который на l тестовых примерах, не использованных при обучении, допустил k ошибок, при любом ε: $\frac{1}{2\sqrt{l}} < \varepsilon \leq \nu$ вероятность ошибки p этого классификатора может быть оценена при помощи неравенства

$$P\{\nu - \varepsilon < p < \nu + \varepsilon\} > 1 - \delta,$$

где $\nu = \frac{k}{l}$ и $\delta = \frac{1}{4l\varepsilon^2}$.

Доказательство. Если представить неравенство (6.7) в виде

$$P\{|\frac{k}{l} - p| < \varepsilon\} > 1 - \frac{1}{4l\varepsilon^2} = 1 - \delta,$$

то его эквивалентная форма

$$P\{\nu - \varepsilon < p < \nu + \varepsilon\} > 1 - \delta \qquad (6.8)$$

определяет интервал $(\frac{k}{l} - \varepsilon, \frac{k}{l} + \varepsilon)$, в котором будет содержаться неизвестная вероятность ошибки с надежностью $1 - \frac{1}{4l\varepsilon^2}$. Но, оценка, основанная на неравенстве Чебышева, дает очень грубые результаты и во многих случаях, когда $\frac{1}{4l\varepsilon^2} \geq 1 \Leftrightarrow \varepsilon \leq \frac{1}{2\sqrt{l}}$, становится неприменимой. Поэтому требуется выполнение условия $\varepsilon \geq \frac{1}{2\sqrt{l}}$. Кроме этого, для интервального оценивания требуется выполнение условия $\nu - \varepsilon \geq 0 \Leftrightarrow \varepsilon \leq \nu$. Поэтому окончательным условием применимости рассматриваемого оценивания является одновременное выполнение неравенств $\frac{1}{2\sqrt{l}} < \varepsilon \leq \nu$. □

И все же во многих случаях оценка (6.8) может быть использована.

Например, если частоты ошибок на контрольной выборке длины $l = 900$ равна $\nu = 0.1$, то для оценивания может быть взята точность, удовлетворяющая условию $\frac{1}{2\sqrt{900}} \approx 0.017 < \varepsilon \le 0.1$. Выбрав, например, $\varepsilon = 0.06$ получим, что неравенство $0.04 < p < 0.16$ будет выполнено с надёжностью $1 - \frac{1}{4 \cdot 900 \cdot (0.06)^2} \approx 0.92$.

Известные также асимптотические результаты для оценивания вероятностей ошибок классификаторов на контрольных выборках, но их недостаток состоит в том, что никакая конечная длина выборки не оценивается.

Например, А.Н. Ширяевым получена следующая оценка[11, с. 98]:

$$P(\nu - \frac{3}{2\sqrt{l}} < p < \nu + \frac{3}{2\sqrt{l}}) > 0.8888 \quad \text{при } l \to \infty.$$

Подставляя $\varepsilon = \frac{3}{2\sqrt{l}}$ в (6.8), получаем $\delta = \frac{1}{4l\varepsilon^2} = \frac{1 \cdot 4l}{4l \cdot 9} = \frac{1}{9}$ и надежность $1 - \delta = 1 - \frac{1}{9} \approx 0{,}8888...$ – точно такой же результат.

Широкое распространение в последние десятилетия компьютеров, компьютерных сетей и электронных источников информации даёт возможность получения значительно бóльших по объёму обучающих и контрольных выборок, чем это было на начальном этапе развития теории и практики машинного обучения. Поэтому можно рассчитывать на пригодность (в числе прочих подходов) чебышевских оценок для бернуллиевской модели вероятности ошибок классификаторов.

Литература к главе 6

1. Бериков В.Б. Оценки вероятности ошибки в байесовской логико-вероятностной модели распознавания образов / В.Б. Бериков // Вычислительные технологии. – 2008. – Т. 13. – №6. – С. 28 – 39.
2. Вапник В. Н. Восстановление зависимостей по эмпирическим данным / В.Н. Вапник. – М. Наука, 1979. – 447 с.
3. Вапник В. Н., Червоненкис А. Я. О равномерной сходимости частот появления событий к их вероятностям / В.Н. Вапник, А. Я. Червоненкис // Теория вероятностей и её применения. – 1971. – Том. XVI. – С. 264 – 279.
4. Вапник В. Н., Червоненкис А. Я. Теория распознавания образов / В.Н. Вапник, А. Я. Червоненкис. – М.: Наука, 1974. – 416 с.

5. Воронцов К. В. Комбинаторные обоснования обучаемых алгоритмов / К.В. Воронцов // ЖВМиМФ. – 2004. – Т. 44. – N° 11. – С. 2099 – 2112.
6. Воронцов К. В. Обзор современных исследований по проблеме качества обучения алгоритмов / К. В. Воронцов // Таврический вестник информатики и математики, 2004. – № 1. – С. 5–24.
7. Воронцов К. В. О теоретико-множественных ограничениях и комбинаторной теории переобучения для алгоритмов классификации / К.В.Воронцов, К.В.Рудаков, Ю.В.Чехович // Труды МФТИ. – 2009. – Т.1. – №4. – С. 148 – 163.
8. Ивахненко А.А. Комбинаторные оценки вероятности переобучения пороговых конъюнкций для логических алгоритмов классификации / А.А. Ивахненко // Труды МФТИ. – 2010. – Т.2. – №3. – С. 16 – 21.
9. Неделько В.М. Исследование погрешности оценок скользящего экзамена / В.М. Неделько // Машинное обучение и анализ данных. – 2013. – Т.1. – №5. – С. 526 – 532.
10. Неделько В.М. Эмпирические интервальные оценки для вероятности ошибочной классификации / В.М. Неделько // Всеросс. конф. «Знание–Онтологии–Теории» (ЗОНТ–09). – Новосибирск: Изд-во Института математики СО РАН, 2009. – Т. 1. – С. 103–107.
11. Ширяев А.Н. Вероятность / А.Н. Ширяев. – М.: МЦНМО, 2004. – 520 с.
12. Bengio Y., Grandvalet Y. No Unbiased Estimator of the Variance of K-Fold Cross-Validation / Yoshua Bengio, Yves Grandvalet // Journal of Machine Learning Research. 2004. – No. 5. – P. 1089 –1105
13. Blumer A. Learnability and the Vapnik-Chervonenkis Dimension / A.Blumer, A.Ehrenfeucht, D. Haussler, M. Warmuth // J. Assoc. Comp. Mach., 1989. – 35. – P. 929 – 965.
14. Blumer A. Occam's Razor / A. Blumer, A. Ehrenfeucht, D. Haussler, M. Warmuth // Information Processing Letters, 1987. – Vol. 24(6). – P.377 – 380.
15. Blumer A., Littlestont N. Learning faster than promise by the Vapnik-Chervonenkis dimension / Anselm Blumer, Nick Littlestone // Discrete Applied Mathematics, 1989. – Vol. 24. – Iss. 1-3, – P. 47 – 63.
16. DasGupta B., Sontag E.D. Sample complexity for learning recurrent perceptron mappings . B. DasGupta, E.D. Sontag. – IEEE Trans. Inform. Theory. – 1996. – 42(5). – P.1479-1487.
17. Ryabko D. On computability of pattern recognition problems / Daniil Ryabko // ALT. Lecture Notes in Computer Science. – Vol. 3734. – P. 148 – 156.
18. Tibshirani R.J., Tibshirani R. A Bias Correction for the Minimum Error Rate in Cross-validation / Ryan J. Tibshirani, Robert Tibshirani // The Annals of Applied Statistics. – 2009. – Vol.3. – No2. – P. 822 – 829.
19. Vorontsov K. V. Combinatorial probability and the tightness of generalization bounds / K. V. Vorontsov // Pattern Recognition and Image Analysis. – 2008. – Vol. 18. – No. 2. – P. 243–259

7. Эмпирическое обобщение и классификация: классы задач, классы моделей и применимость теорий

«А наука … – действительно не что иное,
как "одержимость находить различия"!
Лучше нельзя определить ее суть… наука
называется искусством различения»
ГерманГессе. Нарцисс и Гольдмунд

7.1 Классы задач обучения классификации

Современное состояние теории обучения и распознавания как науки характеризуется появлением вполне обоснованных теорий, иногда базирующихся на существенно различающихся подходах и исходных положениях. Таковыми являются алгебраическая теория распознающих и классифицирующих алгоритмов Ю.И. Журавлева [9], статистическая теория обучения В.Н. Вапника и А.Я. Червоненкиса [3,4], метод потенциальных функций [1], статистическая параметрическая теория классификации, базирующаяся на байесовском подходе и ведущая свое начало от работ Р. Фишера [18], структурно-лингвистические теории [19], *MDL* [21], нейронные сети [20], *SVM* [27] и многие другие теории и их модификации [1, 2, 5,10]. Закономерно возникает вопрос о применимости каждой из рассматриваемых теорий к различным задачам и выделения в этой связи специфических классов задач обучения классификации. С указанным вопросом также связаны выбор и обоснование моделей обучения.

Обоснование выбора подхода к решению конкретной задачи обучения классификации – нетривиальная проблема. Но она, как ни странно, часто остается в тени; усилия исследователей направлены на создание алгоритмов обучения и оценивание вероятности ошибок распознавания [6,12,13,14,23,27].

Представляется целесообразным изложить и обосновать приемлемый подход к определению *областей применимости* (или неприменимости) *основных теорий обучения и классификации*. *Соответствующие области представляют собой классы задач, определяемые общностью их основных свойств*. Будем также называть такие классы задач распознавания семействами, если слово «класс» будет использоваться в контексте для обозначения выделенного множества объектов генеральной совокупности.

На рис. 7.1 приведена концептуальная схема, представляющая связь объектов и процессов, происходящих при построении решающих правил классификации. На этой схеме указаны некоторые свойства рассматриваемых объектов. В соответствии с концептуальной схемой выделен круг при-

знаков, используя которые можно описать классы решаемых задач распознавания.

Выборочное пространство $\mathbf{X}$ состоит из объектов $\tilde{x} = (x_1,...,x_n)$, называемых допустимыми, компоненты которых (переменные-признаки) принимают значения из множеств D_i, $i = \overline{1,n}$.

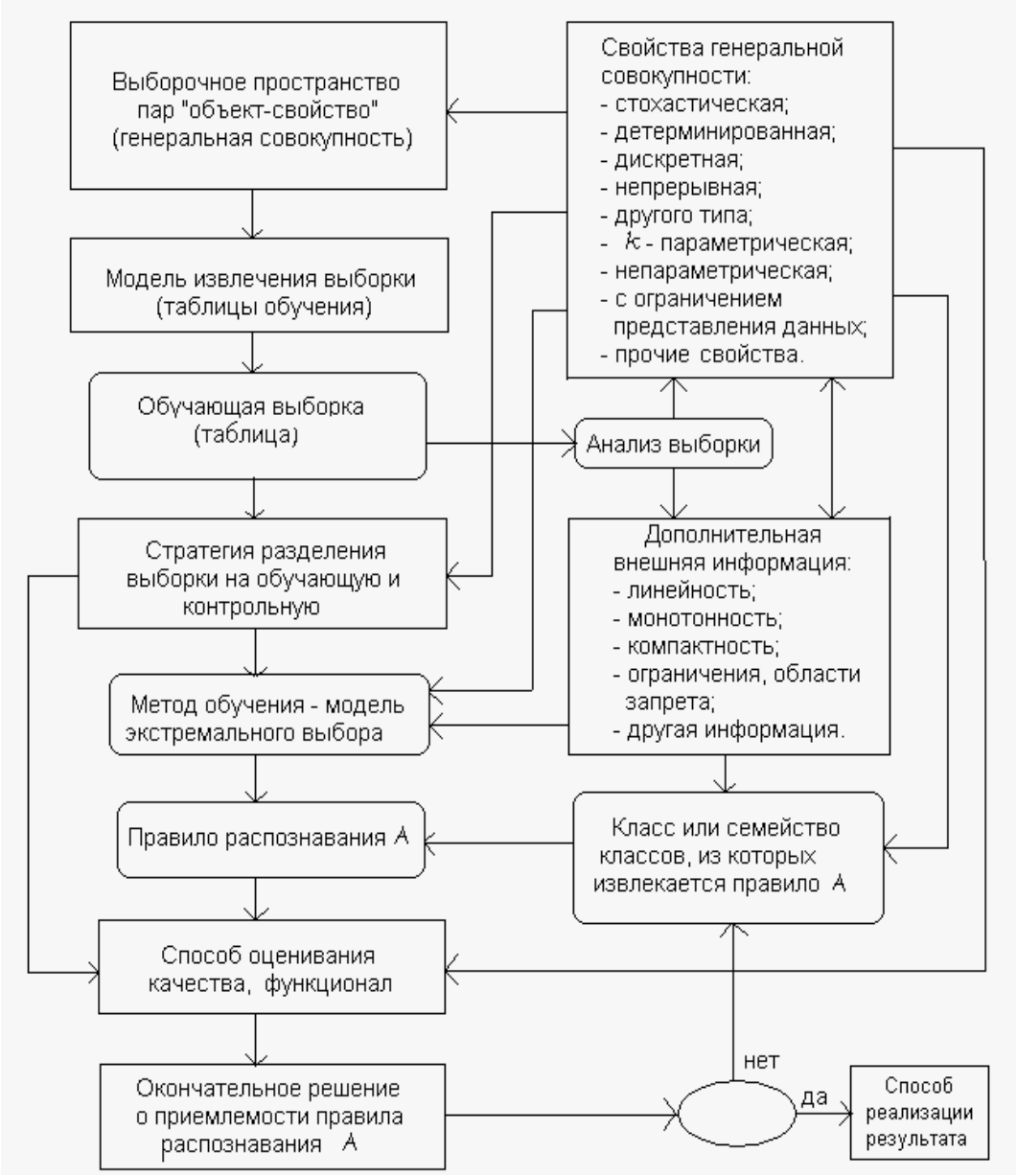

Рис.7.1. Концептуальная схема построения правил распознавания

Множества D_i могут быть непрерывными, дискретными, разнотипными. Полагается существующим некоторый набор основных свойств (предикатов) $\{\alpha_j : D_1 \times \cdots \times D_n \to \{0,1\}\}_{j=1}^{s}$. Множества $K_j = \{\tilde{x} \in \mathbf{X} : \alpha_j(\tilde{x}) = 1\}$ называют классами; $\tilde{\alpha}(X) = (\alpha_1(\tilde{x}),...,\alpha_s(\tilde{x}))$ – двоичный вектор, определяющий принадлежность объекта $\tilde{x}$ классам. Двоичные значения векто-

ра $\tilde{\alpha}(X)$ можно считать номерами классов (при пересекающихся классах – номерами комбинаций классов).

Выборочное пространство $\mathbf{X}$ является генеральной совокупностью объектов, из которой извлекается конечное подмножество объектов – выборка, которая вместе с полученными некоторым способом значениями принадлежности точек выборки классам образует таблицу обучения. Таблица обучения – это совокупность пар $(\tilde{x}_j, \alpha_j)_{j=1}^{l}$. *Выборочное пространство обладает набором свойств, которые отражаются в таблице обучения.* В общем случае не исключено, что таблица обучения может иметь пропуски в данных и ошибки – как в значениях признаков, так и в значениях принадлежности классам.

Задача обучения классификации состоит в нахождении по таблице обучения решающей функции, позволяющей правильно (или приближенно, но как можно более точно) находить для любого объекта $\tilde{x}$ из генеральной совокупности значение номера класса $\alpha(\tilde{x})$.

В таблицу 7.1 в соответствии с концептуальной схемой сведены основные свойства задач распознавания. Дадим краткое пояснение к выбору этих свойств.

В стохастических задачах все данные в таблицах обучения являются случайными величинами, извлеченными из генеральной совокупности, как правило, с неизвестными законами распределения. В некоторых случаях эти законы могут быть заданы, но неизвестными остаются их параметры.

В детерминированных задачах вся информация в таблицах обучения достоверна, существует (неизвестное) решающее правило, точно классифицирующее все допустимые объекты, но возможны пропуски и/или ошибки в данных, связанные с процессом их извлечения. Сам процесс извлечения при этом случайный, независимый и может предполагать существование соответствующих вероятностных распределений.

В недетерминированных задачах часть начальных данных не определена и нет никакой дополнительной информации об их возможных значениях. Неизвестно, существуют ли вообще какие-нибудь вероятностные распределения, в соответствии с которыми порождаются классы объектов и извлекаются обучающие выборки.

Модель извлечения обучающей выборки определяет схему её выбора из генеральной совокупности. Например, случайный и независимый выбор объектов, выбор «типичных представителей в каждом классе». Модель извлечения выборки определяет типы возможных ошибок в полученной таблице обучения.

От длины выборки зависит качество построенного решающего правила распознавания. Не всегда удаётся получить выборку, имеющую достаточную для получения желаемого качества распознавания длину. Иногда

использование длинных выборок может повлечь перенастройку (*overfitting*) *решающих правил*. Большие выборки целесообразно разделять на две части: обучающую, по которой происходит индуктивный синтез решающих правил, и контрольную, по которой оценивается качество выбранного решающего правила. Контрольная выборка не участвует в обучении и оценивает единственное решающее правило, найденное на предварительном этапе синтеза.

Метод (*алгоритм*) *обучения* определяет, как использовать таблицу обучения для выбора экстремального по качеству решающего правила распознавания из некоторого зафиксированного семейства правил. Метод может учитывать целый ряд деталей и использовать различные приёмы. В частности, может учитываться последовательность предъявления объектов таблицы обучения (если от этой последовательности может зависеть результат). Возможно исключение и добавление объектов выборки в процессе обучения и другие. Скользящий контроль также может рассматриваться как метод обучения.

Правило распознавания извлекается в процессе обучения из некоторого заранее зафиксированного семейства правил. Фиксация этого семейства происходит с учётом *внешней дополнительной информации о задаче*, например, заведомой линейности дискриминантных функций, которые являются геометрическим эквивалентом решающих правил распознавания. Качество извлечённого правила может оцениваться различными способами, например, числом (частотой) ошибок на обучающей выборке, числом ошибок на контрольной выборке, длиной алгоритмического описания найденного решающего правила.

Окончательное решение о приемлемости извлеченного правила классификации принимается на основе анализа всех указанных свойств задачи и задаваемых параметров – требуемой точности, надёжности, алгоритмической сложности извлечённого правила.

Указанное окончательное решение теоретически может приниматься автоматически, определяя завершение процесса решения задачи распознавания или продолжение поиска с возможными изменениями в выборе фиксируемого семейства решающих правил и других свойств. Однако такой *автоматический выбор в настоящее время не разработан на алгоритмическом уровне*, поэтому он реализуется исследователями на основе некоторых соображений, сформулированных в процессе решения практических задач распознавания.

Далее предлагается описывать классы задач распознавания в терминах значений свойств, которые представлены в таблице 1, по схеме [8]:

$$\{STD / VAR / SFM / SLen / ADI\}. \qquad (7.1)$$

Запись вида (7.1) называется *кодом задачи обучения классификации*.

Табл.7.1. Основные свойства задач распознавания

N	Обозначение	Наименование свойства задачи	Возможные значения свойства задачи: коды и расшифровки
1	STD	Стохастичность/ детерминированность	S – стохастическая непараметрическая; S_k – стохастическая k -параметрическая; D – детерминированная; ND – недетерминированная;
2	VAR	Дискретность/ непрерывность	D_k – дискретные k -значные переменные; C – непрерывные переменные; M – смешанные переменные;
3	SFM	Модель извлечения и формирования выборки	R_1T (R_2T) – случайный, независимый и безошибочный выбор объектов из генеральной совокупности *с безошибочным при* условии $STD = D$ определением классов (учителем), которым эти объекты принадлежат; R_1F – случайный и независимый выбор объектов из генеральной совокупности с возможными *ошибками и признаков, и классов,* которым эти объекты принадлежат; R_2T – случайный, независимый, но *безошибочный* выбор пар «объект-номер класса»; R_2F – случайный, независимый выбор пар «объект-номер класса» из генеральной совокупности пар с любыми возможными ошибками в любых их компонентах; SST – специальным образом организованное извлечение не содержащей ошибок таблицы обучения (например, выбор типичных объектов или привлечение экспертов); SF – специальным образом организованное извлечение таблицы обучения, возможно с ошибками;
4	$SLen$	Длина выборки	SS – малая выборка, не допускающая пополнение; AS – выборка средней длины; LS – большая или пополняемая выборка;
5	ADI	Дополнительная информация о задаче	L – линейность; M – монотонность; CM – компактность, определяемая в задачах обучения как свойство «близких» в каком-либо смысле объектов принадлежать одному и тому же классу; RR – наличие областей запрета в пространстве признаков; SI – другая специальная информация.

При невозможности характеризовать какое-нибудь свойство задачи, в соответствующую позицию кода задачи ставится пропуск. Пропуск означает,

что соответствующее свойство может быть любым из его перечисленных значений в таблице 7.1, и никакой информации о предпочтительном значении нет. Например, запись $\{D/D_2/R_2T/SS/-\}$ определяет детерминированную задачу с бинарными признаками и случайным, независимым и безошибочным извлечением небольшого числа пар «объект-класс» в таблицу обучения. При этом дополнительная информация для задачи отсутствует.

Кроме этого, в синтаксисе кодов будем допускать логические связки «И», «ИЛИ» и «НЕ» для комбинированного описания свойств задач. Например, $\overline{L}$ будет обозначать нелинейность; $R_2T \vee R_1T$ – безошибочный выбор объектов из генеральной совокупности с безошибочной их классификацией «учителем» или безошибочный выбор пар «объект-номер класса» из генеральной совокупности пар.

Рассмотрим некоторые семейства задач обучения классификации.

В предыдущих главах книги было показано, насколько важным понятием является емкость или $VC-$размерность класса решающих правил H, применяемых для решения задачи обучения классификации, обозначаемая $VCD(H)$. Но исследователи давно заметили [26,6], что при обучении часто используется не весь класс H, а лишь некоторая его часть, определяемая, прежде всего, особенностями алгоритма обучения, но и зависящая также и от свойств задачи, например, от вероятностных распределений. В.Н. Вапнику принадлежит следующее определение.

Определение 7.1 [26]. *Эффективной* $VC-$*размерностью* семейства H (для данной вероятностной меры P) называется минимальная $VC-$размерность подмножества функций из H, определенных на всех подмножествах $X^* \subset X$ области определения функций семейства H, совокупная мера которых почти равна единице, т.е. $P(X^*) > 1-\eta^*$, где $\eta^* > 0$ – достаточно мало́. □

Итак, с одной стороны, имеется представление о семействе используемых гипотез H как о классе применяемых классифицирующих алгоритмов. Например, нейронных сетей, распознающих автоматов или других. С другой стороны, имеет смысл представлять себе образ обучающего алгоритмического отображения $\mathrm{Im}A = H_A \subseteq H$, который может оказаться существенно у́же семейства H. Но тогда для оценивания качества обучения вместо размерности $VCD(H)$ следует использовать размерность $VCD(H_A)$. А если к тому же учесть всю информацию $\Im(D, A, \mathcal{P}, H, \Theta)$, доступную при решении конкретной задачи, которая включает данные, применяемый алгоритм обучения A, исходное семейство используемых гипотез H, свойства вероятностных распределений $\mathcal{P}$ и другие парамет-

ры Θ(если имеются), то на основе информации $\mathfrak{I}(\cdot)$ можно оценить ёмкость $VCD(\mathfrak{I}) = H_{\mathfrak{I}} \subseteq H_A \subseteq H$. Понятно, что используя дополнительную информацию, эффективную ёмкость можно ещё больше сузить.

Определение 7.2. *Действующей $VC-$размерностью решаемой задачи обучения классификации* Z, представленной начальной информацией $\mathfrak{I} = \mathfrak{I}_Z(D, A, \mathcal{P}, H, \Theta)$, которая включает как обучающую выборку D, так и дополнительные сведения о применяемом алгоритме обучения A, исходном семействе используемых гипотез H, свойствах вероятностных распределений $\mathcal{P}$ и других параметрах Θ (если таковые имеются), называется $VC-$*размерность* $VCD(H_{\mathfrak{I}})$ *сужения* $H_{\mathfrak{I}} \subseteq H$ *исходного семейства гипотез* H *за счет учета совокупной информации* $\mathfrak{I}$.□

Очевидно, что

1° $VCD(H_{\mathfrak{I}}) \le VCD(H)$;

2° в любой оценке или теореме, использующей $VCD(H)$, при наличии информации, достаточной для нахождения действующей размерности, можно заменить $VCD(H)$ меньшим значением $VCD(H_{\mathfrak{I}})$.

В представляемых далее теоремах, в частных случаях (при рассмотрении конкретных задач обучения классификации), $VC-$размерность следует *заменить действующей* $VC-$*размерностью.*

7.2. Класс задач обучения классификации $\{D/-/R_1T \vee R_2T \vee ST/S-/-\}$

Рассмотрим класс задач распознавания, определяемый кодом $\{D/-/R_1T \vee R_2T \vee ST/SS/-\}$. Значение параметра $SLen = SS$ определяет малую выборку. Малой обычно считается выборка, *при обработке которой способами, основанными на статистических методах группировки наблюдений и аппроксимации, невозможно достичь заданной точности и достоверности.* Рассматриваемые задачи – детерминированные с точной обучающей таблицей: каждый объект достоверно принадлежит одному классу (или одновременно нескольким классам, если классы пересекаются). Пересечения классов, не теряя общности, можно считать отдельно выделенными классами. Выбор методов и моделей решения таких задач определяется следующими соображениями.

Разбиение исходной выборки на обучающую и контрольную при решении задач рассматриваемого семейства нецелесообразно по следующим причинам. Точность оценивания, недостижимая на малой выборке, тем более будет недостижимой на ее части; безошибочность информации в выборке и малое число прецедентов делает нецелесообразным отказ от ис-

пользования всех начальных данных при обучении. Скользящий контроль тоже нецелесообразен по причине недостаточности начальных данных.

При удачном выборе семейства решающих правил в детерминированных задачах иногда можно указать достаточную длину обучающей выборки для получения точного и единственного решения. В таком случае некорректность задачи вовсе не будет иметь места.

Выбор решающего правила, ошибочно классифицирующего хотя бы один объект таблицы обучения, может повлечь большие ошибки при классификации произвольных допустимых объектов, поскольку для задач рассматриваемого класса это сразу же вносит значительную ошибку в результат. *Применение корректных алгоритмов и только их приемлемо для решения задач распознавания заданного класса*. Действительно, для рассматриваемых задач некорректность (наличие некоторого числа ошибок на обучающей выборке) влечёт не меньшее число ошибок на всей генеральной совокупности: кроме ошибок на обучающей выборке добавляются ошибки вне неё.

Перечисленные соображения определяют для класса задач $\{D/-/R_1T \vee R_2T \vee ST/SS/-\}$ стратегию поиска решающего правила, не допускающего ошибок на объектах таблицы обучения (корректного на выборке) с использованием всей имеющейся выборки при обучении.

При каких же условиях указанная стратегия для класса задач $\{D/-/R_1T \vee R_2T \vee ST/SS/-\}$ будет успешной: построенное решающее правило действительно будет обучено классификации объектов, не принадлежащих таблице обучения?

Предположим, решающее правило будет выбираться в процессе обучения из семейства правил H. Не теряя общности, можно рассматривать случай только двух классов с номерами $\mathbf{O}$ и 1 (с единственным основным свойством).

Используя обучающую выборку $(\tilde{x}_j, \alpha_j)_{j=1}^{l}$, составим *функциональную систему*

$$\begin{cases} f(\tilde{x}_1) = \alpha_1; \\ f(\tilde{x}_2) = \alpha_2; \\ \cdots\cdots\cdots\cdots \\ f(\tilde{x}_l) = \alpha_l; \\ f \in H. \end{cases} \qquad (7.2)$$

В системе (7.2) $\alpha_j = \alpha(\tilde{x}_j)$ принимает значения $\mathbf{O}$ и 1 соответствующие номерам классов допустимых объектов $\tilde{x}$. Решением функциональной

системы (7.2), если оно существует, является *любое корректное на обучающей выборке* решающее правило (функция) $f^* \in H$.

Процесс обучения, направленный на поиск *корректного на выборке решающего правила*, можно рассматривать как поиск решения f^* системы (7.2). При этом результат, очевидно, определяется выбором класса H, в пределах которого идет поиск.

Выбор корректного решения $f^* \in H$ (*решающего правила*) *любым способом будем называть точной настройкой на выборку.*

Предположим, *для любой таблицы обучения с произвольным столбцом номеров классов* при выбранном семействе H возможна точная настройка, но *существует не единственное корректное* (на таблице обучения) решающее правило $f^* \in H$. Предположим также, что обучение происходит по всей имеющейся выборке и оценивается функционалом эмпирического риска по этой же самой выборке. Тогда никаких гарантий правильного распознавания правилом f^* объектов, не участвовавших в обучении, нет. Действительно, при достаточно «богатом» семействе H можно построить для любого конечного множества, содержащего m допустимых объектов, не участвовавших в обучении, корректную на таблице обучения функцию $f^* \in H$, ошибающуюся на всех этих m объектах. Для этой цели каждому из них сопоставляется неправильный номер класса, и полученная таблица сливается с таблицей обучения $(\tilde{x}_j, \alpha_j)_{j=1}^{l}$. Если в семействе H найдется корректный алгоритм для такой объединенной таблицы длины $l+m$, то он будет примером случая, когда указанная стратегия обучения в рассматриваемом классе задач, несмотря на точную настройку, даёт неприемлемый результат. В таком случае обычно говорят, что *обучаемость не имеет места*.

Другая ситуация возникает, когда *точная настройка* при выбранном семействе H *возможна только для некоторого множества допустимых выборок, таких, в которых все объекты в каждом классе обладают некоторыми отличающими их от объектов другого класса свойствами*. Тогда возможность получения решения системы (7.2) с ростом длины выборки l связывается именно с проявлением в выборке указанных свойств (закономерностей) и обеспечивается существованием в классе H правила, способного «улавливать» эти свойства. Именно наличие закономерностей на генеральной совокупности в свою очередь влечет *появление в таблице обучения не любых из* 2^l *возможных двоичных столбцов системы* (7.2) $\tilde{\alpha} = (\alpha_1, \ldots, \alpha_j, \ldots, \alpha_l)^T$, *а столбцов лишь из некоторого, определенного существующей закономерностью, множества.*

Теорема 7.1. Если $VCD(H) \geq l$, то найдется такая обучающая выборка $(\tilde{x}_j, \alpha_j)_{j=1}^{l}$, что *для любого столбца* $\tilde{\alpha}$ возможна точная настройка.

Доказательство немедленно следует из определения VC-размерности. □

Теорема 7.2. Если для любой обучающей выборки $(\tilde{x}_j, \alpha_j)_{j=1}^{l}$ существует такой булев набор $\tilde{\alpha}$, что невозможна точная настройка, то $VCD(H) < l$.

Доказательство становится очевидным, если заметить, что утверждение доказываемой теоремы равносильно утверждению теоремы 7.1. □

Теорема 7.2 дает необходимое условие обучаемости для задач распознавания класса $\{D/-/R_1T \vee R_2T \vee ST/-/-\}$ при выборе стратегии, направленной на построение корректных на обучающих таблицах алгоритмов: *емкость семейства решающих правил, используемого для настройки, должна быть меньше длины выборки*. Заметим, что неотрицательная величина $l - VCD(H)$ может быть использована для получения оценки неслучайности обнаружения закономерности по обучающей выборке [17].

Условие $VCD(H) < l$ обосновывает важность знания емкости класса, используемого для решения задачи обучения классификации. В связи с представляется полезным следующий результат.

Теорема 7.3. Пусть функциональная система (7.2) при зафиксированном семействе решающих функций H для любой обучающей выборки и любых двоичных значениях $\alpha_1,\ldots,\alpha_j,\ldots,\alpha_l = \tilde{\alpha}$ может иметь не более одного решения, и при этом найдется выборка $(\tilde{x}_j, \alpha_j)_{j=1}^{l}$ такая, что для любого $\tilde{\alpha}$ существует решение $f_{\tilde{\alpha}}$. Тогда $VCD(H) = l$.

Доказательство. Поскольку существует обучающая выборка, для которой функциональная система (7.2) имеет решение при любом двоичном наборе $\alpha_1,\ldots,\alpha_l$, в семействе H найдутся функции, разбивающие эту выборку на два класса всеми способами. Поэтому $VCD(H) \geq l$.

Если к любой выборке $(\tilde{x}_j, \alpha_j)_{j=1}^{l}$ длины l добавить один произвольный не принадлежащий ей элемент $\tilde{z}$ из генеральной совокупности допустимых объектов, то функциональная система

$$\begin{cases} f(\tilde{x}_1) = \alpha_1; \\ f(\tilde{x}_2) = \alpha_2; \\ \ldots\ldots\ldots\ldots \\ f(\tilde{x}_l) = \alpha_l; \\ f(\tilde{z}) = \beta; \\ f \in H \end{cases} \tag{7.3}$$

при любом $\beta \in \{0,1\}$ будет сужением системы (7.2) и поэтому в силу условия теоремы сможет иметь не более одного решения. Если она не имеет решений, то для выборки $(\tilde{x}_j, \alpha_j)_{j=1}^{l} \cup (\tilde{z}, \beta)$ длины $l+1$ при помощи функций системы H невозможно получить разбиение, соответствующее булевому набору $\alpha_1,...,\alpha_l, \beta$. Если же решение $f_{\tilde{\alpha}}$ системы (7.3) существует для некоторого значения β, то по условию теоремы оно единственное для функциональных систем (7.2) и (7.3). Тогда $\beta = f_{\tilde{\alpha}}(\tilde{z})$, но при помощи функций системы H невозможно получить разбиение выборки $(\tilde{x}_j, \alpha_j)_{j=1}^{l} \cup (\tilde{z}, \overline{\beta})$ длины $l+1$, соответствующее булевому набору $\alpha_1,...,\alpha_l, \overline{\beta}$. Учитывая, что последнее заключение получено в результате рассмотрения любой обучающей выборки длины l, получаем неравенство $VCD(H) \le l$, которое вместе с неравенством $VCD(H) \ge l$ дает результат: $VCD(H) = l$.□

Еще раз подчеркнём, что в теории машинного обучения и классификации имеет смысл рассматривать только те задачи, в которых закономерность $\alpha = \alpha(\tilde{x})$ существует и отличается от случайной функции с равномерным распределением значений $\{0,1\}$ на множестве $\mathbf{X}$. В этом смысле закономерность – это неслучайность, и в правой части системы (7.2) должны содержаться не какие угодно столбцы, а именно те, которые связаны с объектами выборки некоторой закономерностью.

Если извлеченная из генеральной совокупности выборка является безошибочной, то выбор в процессе обучения правила классификации, которое допускает ошибки на этой выборке, как уже говорилось, представляется бессмысленным. Поэтому в этом случае требуется найти точное решение f^* функциональной системы (7.2) в некотором классе решений H. Это решение f^* называется точной настройкой на выборку. Если система имеет более чем одно решение, то при достаточно широком семействе H выбранное правило может давать большу́ю ошибку $Err(f^*)$, определяе-

мую отличием выбранного решения f^* от существующего истинного правила классификации f_0:

$$Err(f^*) = \mathrm{E}[|\, f^* - f_0\,|].$$

Ошибка оценивается по вероятностной мере P на генеральной совокупности $\mathbf{X}$.

Если же мера P не существует, не имеет смысла для некоторых задач, то можно сказать, что ошибка точного на выборке решения f^* в некоторых случаях может иметь место почти всюду на $\mathbf{X}$. Поэтому для того, чтобы выполнялось равенство $f^* = f_0$ при условии, что обучающая выборка безошибочная, решение системы (7.2) должно быть единственным. Действительно, если существуют два решения f_1^* и f_2^* – две функции, совпадающие в точках обучающей выборки, то их продолжения на $\mathbf{X}$ могут различаться почти всюду. Например, когда $f_1^*(\tilde{x}) = f_2^*(\tilde{x})$ при условии $\exists j : \tilde{x} = \tilde{x}_j$, где $\tilde{x}_j$ – элемент какой-нибудь пары из обучающей выборки $(\tilde{x}_j, \alpha_j)_{j=1}^l$, но $f_1^*(\tilde{x}) = 1 - f_2^*(\tilde{x})$ всюду на $\mathbf{X}$ кроме точек из обучающей выборки.

Продолжением выборки будем называть любую последовательность точек из $\mathbf{X}$, которая не содержит точек этой выборки.

Представим теперь, что в обучающей информации появились ошибки, которые привели к изменению столбца $\tilde{\alpha}$ в системе (7.2) и превращению его в столбец с ошибками $\tilde{\alpha}^E$. Пусть система

$$\begin{cases} f(\tilde{x}_1) = \alpha_1^E; \\ f(\tilde{x}_2) = \alpha_2^E; \\ \ldots\ldots\ldots\ldots \\ f(\tilde{x}_l) = \alpha_l^E; \\ f \in H. \end{cases}$$

имеет решение f^E. Но тогда $f^E \neq f_0$, и в таком случае представляется абсурдной точная настройка алгоритма обучения на выборку. Это приводит к следующему выводу: для любой выборки $(\tilde{x}_j, \alpha_j)_{j=1}^l$, в которой отражена некоторая закономерность, должны существовать такие двоичные столбцы $\tilde{\alpha}^E$, что по таблице обучения $(\tilde{x}_j, \alpha_j^E)_{j=1}^l$ точная настройка является невозможной. Теорему 7.2 можно усилить:

Теорема 7.4. Пусть в задаче обучения классификации решающее правило выбирается из семейства H на основе обучающей выборки $(\tilde{x}_j, \alpha_j)_{j=1}^{l}$. Тогда для любого набора точек $\{\tilde{x}_j\}_{j=1}^{l}$, который может содержаться в обучающей выборке, соответствующий булевский набор $\tilde{\alpha}$ такой, что точная настройка невозможна, найдется если и только если $VCD(H) < l$.

Следствие 7.1. Корректный на выборке длины l алгоритм классификации, выбранный из семейства H такого, что $VCD(H) \geq l$, может давать ошибке почти всюду на генеральной совокупности объектов.□

Нужно подчеркнуть, что всюду в этой статье классифицирующие алгоритмы рассматриваются *с точностью до классов функциональной эквивалентности*. Иначе говоря, одним и тем же считаются все алгоритмы (машины Тьюринга), которые для одной и той же начальной информации (слова на ленте) всегда выдают один и тот же результат.

Теорема 7.5. Для того, чтобы выбор корректного на выборке алгоритма из заданного семейства H в задаче обучения классификации из класса $\{D/-/R_1T \vee R_2T \vee ST/-/-\}$ всегда обеспечивал получение абсолютно точного решения, необходимо и достаточно, чтобы в семействе H для любой выборки существовал единственный с точностью до функциональной эквивалентности корректный на этой выборке алгоритм.

Доказательство. *Необходимость*. Действительно, если для какой-нибудь выборки в семействе H не существует корректного на ней алгоритма, то некоторое число объектов этой выборки всегда классифицируется неверно. Если же при этом корректность на выборках достигается всегда, но хотя бы для одной выборки – не единственным алгоритмом из H, а хотя бы двумя неэквивалентными алгоритмами, то их продолжения на множестве последовательностей из генеральной совокупности $\mathbf{X}$ не будут совпадать. Тогда хотя бы один из них будет давать ошибки на своем продолжении.

Достаточность. Если выбор корректного на выборке алгоритма из заданного семейства H не всегда обеспечивает получение абсолютно точного решения, то для некоторой выборки существует корректный алгоритм, не являющийся абсолютно точным решением. Зафиксируем эту выборку. Постановка задачи предполагает существование абсолютно точного решения. Это точное решение – некоторое правило f_0 – также будет корректным на зафиксированной (безошибочной в соответствие с рассматриваемой моделью) выборке. Тогда корректное на ней решение не единственно.□

Очевидно, что если в семействе H для любой конечной выборки длины l существует единственный с точностью до функциональной эквивалентности корректный на этой выборке алгоритм, то $VCD(H) < l$.

Следствии 7.2. Для того, чтобы выбор корректного на выборке длины l алгоритма из заданного семейства H в задаче обучения классификации из класса $\{D/-/R_1T \vee R_2T \vee ST/-/-\}$ всегда обеспечивал получение абсолютно точного решения, необходимо выполнение условия $VCD(H) < l$.□

Но условие $VCD(H) < l$ не является достаточным. Это сразу же видно из случая, когда при его выполнении семейство H не содержит в себе истинного решающего правила f_0. Очевидно также, что условие $f_0 \in H$ является необходимым для осуществления возможности нахождения абсолютно точного решения.

7.3 Обучение или настройка?

Детерминистская постановка задач обучения распознаванию предполагает существование точного решения f_0 (истинного решающего правила). Согласно этому правилу каждой выборке $\{\tilde{x}_j\}_{j=1}^{l}$ должен сопоставляться единственный булевский вектор $\tilde{\alpha}^*$ такой, что $\tilde{\alpha}_j^* = f_0(x_j)$, $j = \overline{1,l}$. Пары $(\tilde{x}_l, \tilde{\alpha}^*)$ для каждой выборки отражают закономерность (регулярность), выделяющую вектор $\tilde{\alpha}^*$ из всех остальных $2^l - 1$ возможных соответствий $\tilde{x}_l \mapsto \tilde{\alpha}$.

Выше установлено, что для устранения неоднозначности и обеспечения получения точного решения задачи обучения классификации необходимо и достаточно, чтобы существовала возможность нахождения точного решения системы (7.2), причем это решение должно быть единственным. Кроме этого, обязательно должно выполняться емкостное ограничение $VCD(H) < l$, поскольку в противном случае условие единственности корректного алгоритма для любой выборки длины l нарушается.

Теперь можно рассмотреть вопрос об *отличии обучения от настройки*. Этот вопрос представляется важнейшим в теории обучения классификации. Если не ограничивать емкость используемого для решения задачи обучения распознаванию семейства H, то всегда можно добиться точной настройки на непротиворечивую начальную информацию – «натянуть»

решающее правило на все точки обучающей выборки. В таком случае ни о каком обучении говорить не приходится.

Будем называть ***обучением «снизу вверх»*** такой поэтапный процесс построения решающего правила $\hat{f}_0$, на каждом этапе которого происходит *минимальное необходимое усложнение* решающего правила, обеспечивающее уменьшение числа его ошибок на обучающей выборке. При обучении «снизу вверх» происходит поэтапное усложнение решающего правила и, соответственно, расширение семейства правил, которому оно принадлежит. Обучение в рассматриваемом случае будет успешным, если для коррекций будут использованы не все точки обучающей выборки: оставшаяся часть не использованных для коррекций точек будет классифицироваться правильно и «подтверждать» построенное решающее правило. Выбор начального приближения и способы поэтапного усложнения решающего правила определяют алгоритм (метод) обучения. В качестве примера обучения методом «снизу-вверх» можно взять последовательный синтез решающего дерева по обучающей информации. Решающее правило – древообразный классификатор сначала имеет простейший вид с одной условной вершиной. На шагах обучения правило усложняется путем добавления условной вершины только в случае наличия ошибок с целью уменьшения их числа.

Для обоснования алгоритма обучения «снизу вверх» целесообразно приводить *доказательство возможности расширения* (в процессе выполнения именно этого алгоритма) семейства правил, которому принадлежит вычисляемое решающее правило, до некоторого семейства H_0, содержащего истинное решающее правило f_0 и имеющего ёмкость $VCD(H_0) < l$.

Обучением «сверху вниз» будем называть последовательный процесс нахождения решающего правила $\hat{f}_0$, принадлежащего некоторому подклассу минимальной сложности H' из выбранного изначально некоторым способом семейства H, направленный на достижение наибольшей точности правила $\hat{f}_0$ на заданной обучающей выборке. В качестве примера обучения методом «сверху вниз» можно привести оптимизационный синтез минимальной дизъюнктивной нормальной формы, частично заданной бинарной таблицей обучения, как логического классификатора. И вообще, парадигму «индукции как оптимизации» в целом [24].

Для обоснования алгоритма обучения по методу «сверху вниз» целесообразно приводить доказательство адекватности изначально заданного семейства H – наличии в нём истинного решающего правила, а также сохранении этого свойства при поэтапном сужении начального семейства.

Комбинированным обучением с возвратом будем называть процесс построения решающего правила, сочетающий оба метода обучения – «снизу вверх» и «сверху вниз». Такой процесс аналогичен поиску с возвратом (*backtracking*). Для упомянутых выше решающих деревьев соответствующим примером являются процедуры нахождения классификатора на основе оценок его текущей сложности, включающие условия возврата к более простому варианту, когда сложность становится выше заданного порога.

Адаптивным обучением будем называть процесс пошаговой коррекции параметрической модели с такими же условиями – по минимальному числу примеров, используемых для коррекции, как и в случае обучения «*снизу вверх»*. Классический пример адаптивного обучения – алгоритм линейной коррекции Розенблатта-Новикова [15], который лег в основу обучения всех параметрических моделей.

Комбинированным адаптивным обучением будем называть процесс направленного обучения с адаптацией параметров классификатора. Например, структурно-адаптивный метод обучения нейронной сети (см. главу 3), когда параметры «соединения» в структуре сети в процессе пошагового обучения с целью уменьшения ошибок могут «сбрасываться» в ноль, обеспечивая упрощение структуры и сложности классификатора.

Обучением путём сжатия данных будем называть процесс синтеза решающего правила, которое может быть определено как можно меньшим числом примеров d из заданной обучающей выборки длины l. Оставшиеся $l-d$ примеров в таком случае «безоговорочно подтверждают» построенное решение. В качестве примера можно привести машину опорных векторов (SVM) [27]. □

Любой *процесс нахождение решения системы* (7.2)*, отличающийся от обучения, является настройкой*.

7.4 Особенности класса $\{D/-/R_2F \vee SF/-/-\}$

Если выборка содержит ошибки, то можно считать, что их появление связано с искажением правильной выборки, или, говоря иначе, с переходом от правильной к ошибочной выборке. Будем обозначать такой переход следующим образом: $(\tilde{x}_l, \tilde{\alpha}^*) \xrightarrow{Err} (\tilde{x}_l^{Err}, \tilde{\alpha}^{Err})$. Переход $\tilde{x}_l \xrightarrow{Err} \tilde{x}_l^*$ можно рассматривать как изменение набора точек выборочного пространства в пределах допустимого множества и говорить в этом случае о таком же переходе безошибочной обучающей выборки – в ошибочную. Обозначим $\Delta(Err) = \| \tilde{\alpha}^* - \tilde{\alpha}^{Err} \|$ – число ошибок в векторе $\tilde{\alpha}^{Err}$.

Если $VCD(H) \geq l$, то из семейства H может быть выбрано решающее правило f^{Err}, точно настроенное на выборку $(\tilde{x}_l^{Err}, \tilde{\alpha}^{Err})$ (эмпирическая ошибка при этом – нулевая: $\hat{\varepsilon}^* = 0$), но истинная ошибка этого правила $\varepsilon = \varepsilon(f^{Err}) \geq \Delta(Err)$.

Пусть $\mathfrak{M}$ такое подмножество элементов обучающей выборки, что их удаление из этой выборки позволяет осуществить точную настройку, но удаление никакого собственного подмножества $\mathfrak{M}' \subset \mathfrak{M}$ из выборки уже не позволяет настроиться точно. Будем называть такой набор *детерминированной помехой*. Из её определения усматривается *переборный алгоритм фильтрации* (удаления) $\mathfrak{M}$ из *обучающей выборки*.

Пусть $\Delta(Err) = \Delta_{\max}$ – наибольшее возможное число ошибочных примеров, порождаемых переходом $(\tilde{x}_l, \tilde{\alpha}^*) \xrightarrow{Err} (\tilde{x}_l^{Err}, \tilde{\alpha}^{Err})$, является изначально заданным параметром задачи. Процесс обучения может состоять из следующих последовательно решаемых подзадач:

1º Выбор адекватного начального семейства H, которое должно содержать истинное решающее правило f_0, для реализации алгоритма фильтрации.

2º Выполнение переборного алгоритма фильтрации (удаления поочередно по $1,2,...,C_l^{\Delta_{\max}}$ примеров из обучающей выборки) и процедуры обучения на выборке без удалённых примеров, пока эмпирическая ошибка не станет нулевой или заданное число итераций не будет исчерпано.

Очевидно, что для решения задач из класса $\{D/-/R_2F \vee SF/-/-\}$ нецелесообразно применять корректные алгоритмы без использования фильтрации.

7.5 Особенности класса $\{ND/D_k/-/-/-\}$

Класс недетерминированных задач обучения классификации характеризуется тем, что не имеется никакой информации о законах, определяющих существование и появление той или иной обучающей выборки. Более того, неизвестно: существует ли точное решение f_0 или нет.

Заметим, что недетерминированные и стохастические задачи обучения распознаванию принципиально различаются. Информация о существовании вероятностных распределений или более – об их типах в стохастических задачах в некоторых случаях может дать возможность в явном виде выписать статистически оптимальное решающее правило. В стохастиче-

ских задачах речь идет о решениях, получаемых с точностью, определяемой заданием вероятностных мер.

Рассмотрим следующую задачу. Пусть X^{MT} – множество шифров машин Тьюринга. Каждый шифр $X(M) \in \mathrm{X}^{MT}$ машины M является натуральным числом, которое, в частности, может быть представлено двоичной строкой конечной длины. Машина Тьюринга M называется *самоприменимой*, если, начав работу над словом $p = X(M)$, являющимся шифром этой машины M, она остановится, выполнив конечное число шагов.

Пусть решающая функция, которую должен найти алгоритм обучения, задана следующим образом:

$$f_0^{SA}(X(M)) = \begin{cases} 1, & \textit{если машина } M \textit{ самоприменима}; \\ 0, & \textit{если машина } M \textit{ несамоприменима} \end{cases}$$

Известно, что такая функция f_0^{SA} не является вычислимой – не существует алгоритма (строго определенного тезисом Черча-Тьюринга), правильно вычисляющего для любого входа $X(M)$ значение $f_0^{SA}(X(M))$[11]. Тем не менее, можно сконструировать обучающую выборку $\tilde{X}$, состоящую из m_1 примеров шифров самоприменимых машин Тьюринга и m_0 примеров несамоприменимых машин.

Что же будет, если такую выборку $\tilde{X}$ взять как начальную информацию – таблицу обучения для построения алгоритма распознавания свойства самоприменимости? Такого алгоритма в принципе не существует. Тем не менее, алгоритм обучения, выбранный из подходящего для данной задачи семейства и имеющий достаточную ёмкость, может дать в качестве решения частичную функцию $\hat{f}_0^{SA}$, которая безошибочно классифицирует все примеры выборки $\tilde{X}$.

Приведенный пример принадлежит классу недетерминированных задач: неизвестно, существует ли вообще правильное решение (в данном примере – не существует алгоритмического правильного решения), и неизвестно, существует ли какой-нибудь закон появления объектов генеральной совокупности. Но предикат, определяющий основное свойство (здесь – свойство самоприменимости) и отражающий соответствующую закономерность, существует. Данный пример делает очевидной справедливость следующей теоремы.

Теорема 7.6. Существуют недетерминированные задачи обучения классификации, для которых абсолютно точное решающее правило не является вычислимым. □

Для решения задач из класса $\{ND / D_k / - / - / -\}$ целесообразно использовать алгоритмы, извлекающие закономерность, которая имеет как можно меньшую колмогоровскую сложность. Действительно, недетерминированность предполагает полное отсутствие сведений о распределении объектов генеральной совокупности и вследствие этого *допускает подход к выбору решения, которое можно обосновать как неслучайное*.

Для задач из класса $\{ND / D_k / R_1T \vee ST / - / -\}$ тоже целесообразно применение алгоритмов наименьшей колмогоровской сложности. Хотя известны и другие подходы, например, теоретико-игровой [7], часто применяемый для решения широкого класса задач в условиях априорной неопределенности.

7.6 Классификация длин выборок

Практика применения машинного обучения показала, что одна и та же длина обучающей выборки может в некоторых случаях оказаться достаточной для получения требуемой точности распознавания, а в других случаях – быть слишком короткой. Возьмём для примера детерминированную задачу с заведомо линейным, но неизвестным решающим правилом – предикатом $f_0(x_1, x_2) = [ax_1 + bx_2 = c]$. Обучающую выборку из двух точек будем полагать безошибочной, и в ней эти две точки будут принадлежать классу «лежащих на прямой». Очевидно, в этом случае можно абсолютно точно решить задачу восстановления линейного предиката.

Для обучения многослойных нейронных сетей требуются большие выборки, поскольку нейросетевые семейства решающих правил имеют большую емкость.

Классы длин выборок для рассмотренных выше задач обучения классификации должны определяться ситуативно, в зависимости от емкости семейств, из которых в процессе обучения извлекается решающее правило.

Таблица 7.2.

Параметр $SLen$	Значение параметра	Определяющее условие
SS	Малая выборка	$l < VCD(H)$
AS	Средняя выборка	$VCD(H) \le l < 1.5 \cdot VCD(H)$
LS	Большая выборка	$l > 1.5 \cdot VCD(H)$

Определять значение параметра $SLen$ в стандартных кодах задач обучения классификации предлагается в соответствии с таблицей 7.2 (l – длина обучающей выборки).

7.7 Класс $\{S_k / C / - / - / SI\}$

Параметр $ADI = SI$ (специальная информация) в стохастических параметрических задачах чаще всего определяет типы используемых вероятностных распределений и, возможно, специфические характеристики параметров (в приведенном ниже примере – равенство ковариационных матриц классов).

Рассмотрим пример k-параметрической стохастической задачи обучения распознаванию объектов двух классов, которая хорошо изучена в теории статистических решений.

Пусть согласно дополнительной информации условные вероятности появления в выборке объектов каждого из двух классов $\alpha \in \{0,1\}$ имеют многомерное нормальное распределение

$$p(\tilde{x} \mid \alpha) = \frac{1}{(2\pi)^{n/2} \mid \sum_{\alpha} \mid} \exp\{-\frac{1}{2}(\tilde{x} - \mathrm{E}_{\alpha})^T \sum_{\alpha}^{-1} (\tilde{x} - \mathrm{E}_{\alpha})\},$$

где E_{α} и $\sum_{\alpha}$ – математические ожидания и ковариационные матрицы двух классов $\alpha \in \{0,1\}$. Пусть также известны априорные вероятности появления объектов каждого из классов: p_0 и p_1.

Известно, что оптимальная (минимизирующая средний риск ошибки) дискриминантная функция в случае равных ковариационных матриц $\sum_0 = \sum_1 = \sum$ является линейной и имеет вид [15]

$$g(\tilde{x}) = \tilde{x}^T \sum^{-1} (\mathrm{E}_0 - \mathrm{E}_1) - \frac{1}{2} \mathrm{E}_0^T \sum^{-1} \mathrm{E}_0 + \frac{1}{2} \mathrm{E}_1^T \sum^{-1} \mathrm{E}_1 + \ln(\frac{p_0}{p_1}). \quad (7.4)$$

Соответствующая решающая функция имеет вид $f_g(\tilde{x}) = \begin{cases} 0, & g(x) < 0; \\ 1, & g(x) \geq 0. \end{cases}$

Решение приведенной в последнем примере задачи, когда задана обучающая выборка $(\tilde{x}_j, \alpha_j)_{j=1}^{l}$, состоит в нахождении по этой выборке статистических оценок $\hat{p}_0$, $\hat{p}_1$, $\hat{\mathrm{E}}_{\alpha}$ и $\hat{\Sigma}_{\alpha}$, $\alpha \in \{0,1\}$, и вычисления $\hat{g}(\tilde{x})$ по формуле (7.4). В рассматриваемой задаче, очевидно, не существует абсолютно точного решения f_0, но при точно заданных векторах математических ожиданий, априорных вероятностей и ковариационной матрице соответствующая статистическая задача принятия решений имеет *точное вероятностное решение* f_g – с точностью до заданной вероятностной меры. В постановке обучения классификации наилучшее решение рассматриваемой стохастической параметрической задачи является известным и

требует только вычисления необходимых моментов векторных случайных величин.

Никакой корректный алгоритм для решения приведенной статистической задачи, разумеется, не подходит. Действительно, разделяющая поверхность, соответствующая наилучшему решающему правилу, является линейной, в то же время классы пересекаются, и выборка, вообще говоря, может оказаться не разделимой линейно. Тогда корректный алгоритм построит нелинейное правило распознавания, заведомо худшее, чем f_g.

Обобщим этот вывод на случай произвольной стохастической задачи обучения распознаванию с двумя пересекающимися классами. Среди всевозможных решающих правил для такой задачи обязательно существует правило, минимизирующее вероятность ошибки или заданную функцию потерь (взвешенную функцию ошибки). Будем обозначать такое наилучшее правило f_0^H, а соответствующую ему дискриминантную функцию обозначим g_0^H. Очевидно, что *любой корректный алгоритм, примененный к рассматриваемой задаче, определит решающее правило, отличающееся от* f_0^H, *поскольку* часть точек обучающей выборки могут оказаться расположенными «по разные стороны» дискриминантной функции g_0^H произвольным образом. Следовательно, *корректные алгоритмы для решения таких задач не подходят*.

Для стохастических параметрических задач распознавания ёмкость класса, которому принадлежит дискриминантная функция, вообще говоря, не имеет значения; важно лишь то, чтобы эта функция минимизировала средний риск ошибки. Такая функция уже определена стохастическими параметрами задачи, и её не требуется отыскивать ни в каком придуманном классе.

7.8 Стохастические непараметрические задачи обучения классификации ($STD = S$)

С непараметрическими стохастическими задачами обучения классификации дело обстоит иначе. Вероятностные распределения неизвестны, их восстановление по обучающей выборке, как правило, приводит к не менее сложным задачам, чем задача обучения распознаванию в классической постановке.

Для задач рассматриваемого класса всегда можно полагать существование некоторой решающей функции f_0^H (дискриминантной функции g_0^H) наилучшей в статистическом смысле. Эта функция является неиз-

вестной, и алгоритм обучения, конечно, должен находить её наилучшее приближение. Понятно, что такой алгоритм вовсе не обязан быть корректным на выборке. Но должен ли он давать на этой выборке минимальную эмпирическую ошибку, т.е. иметь на ней как можно более близкую к точной настройку?

Учитывая результаты рассмотрения параметрических стохастических задач, можно предположить, что для рассматриваемого класса задач обучения перенастройка (выбор корректного или с очень малой эмпирической ошибкой алгоритма) может привести к большим ошибкам классификации объектов, не принадлежащих обучающей выборке. По-видимому, это связано с тем, что *неизвестная дискриминантная функция* g_0^H (если она байесовская, минимизирующая средний риск, т.е. статистически оптимальная) *должна быть полиномом невысокой степени*. Такой полином возникает вследствие неизвестных, но существующих многоэкстремальных (и, тем более, одноэкстремальных) вероятностных распределений.

Представляется целесообразным пытаться искать решающее правило как можно более близкое к байесовскому классификатору – по максимуму апостериорной условной вероятности класса. Для некоторых семейств моделей классификаторов доказаны теоремы о качестве приближения отыскиваемых решающих правил к байесовскому [22]. Именно такие модели наиболее пригодны для работы с непараметрическими стохастическими задачами обучения распознаванию.

Литература к главе 7

1. Айзерман М. А. Метод потенциальных функций в теории обучения машин / М. А. Айзерман, Э. М. Браверман, Л. И. Розоноэр – М. : Наука, 1970. – 384 с.
2. Бонгард, М. М. Проблема узнавания / М.М. Бонгард. – М.: Наука, 1967. — 320 с
3. Вапник В. Н. Восстановление зависимостей по эмпирическим данным / В.Н. Вапник. – М.: Наука, 1979. – 448 с.
4. Вапник В. Н., Червоненкис А. Я. Теория распознавания образов / В.Н. Вапник, А.Я. Червоненкис. – М.: Наука, 1974. – 416 с.
5. Васильев В. И. Распознающие системы: справочник / В. И. Васильев. – К.: Наук. думка, 1983. – 422 с.
6. Воронцов К. В. Комбинаторные оценки качества обучения по прецедентам / К.В. Воронцов // Докл. РАН. – 2004. – Т.394, №2. – с. 175-178.
7. Вьюгин В.В. Элементы математической теории машинного обучения / В.В. Вьюгин. – М.: МФТИ, 2010. – 252 с.
8. Донской В. И. Эмпирическое обобщение и распознавание: классы задач, классы математических моделей и применимость теорий. Часть I / В. И. Донской // Таврический вестник информатики и математики. – 2010. –

№1. – С.15 – 23; часть II – // Таврический вестник информатики и математики. – 2011. – №2. – С.31 – 42.
9. Журавлев Ю.И. Об алгебраическом подходе к решению задач распознавания или классификации // Проблемы кибернетики. – Вып.33. – М.: Наука, 1978, с. 5–68.
10. Ивахненко А.Г. Индуктивный метод самоорганизации моделей сложных систем / Ивахненко А. Г. – Киев: Наук. думка, 1981 –. 296 с.
11. Игошин В. И. Математическая логика и теория алгоритмов / В. И. Игошин . – М.: Академия, 2004. – 448 с.
12. Лепский А.Е., Броневич А.Г. Математические методы распознавания образов. Курс лекций / А.Е. Лепский, А.Г. Броневич. – Таганрог: Изд-во Техн. Инст-та Южного федерального университета, 2009. – 155 с.
13. Мерков А.Б. Распознавание образов: введение в методы статистического обучения / А.Б. Мерков. – М.: Едиториал УРСС, 2011. – 256 с.
14. Местецкий Л.М. Математические методы распознавания образов. Курс лекций. [Электронный ресурс] / Л.М. Местецкий.– М.: ВМиК МГУ, 2002–2004. – 85 с.
Режим доступа:
www.ccas.ru/frc/papers/mestetskii04course.pdf
15. Нильсон Н. Обучающиеся машины / Н. Нильсон. – М.:Мир, 1967. – 180 с.
16. Devroye L., Gyorfi L., Lugosi G. A Probabilistic Theory of Pattern Recognition / L. Devroye, L. Gyorfi, G.A. Lugosi. –Springer-Verlag, NY, 1996. – 636 p.
17. Donskoy V. I. The Estimations Based on the Kolmogorov Complexity and Machine Learning from Examples / V.I. Donskoy // Proceedings of the Fifth International Conference ”Neural Networks and Artificial Intelligence” (ICNNAI'2008). – Minsk: INNS. – 2008. – P. 292 – 297.
18. Fisher R.A. The Use of Multiple Measurements in Taxonomic Problems / Ronald Aylmer Fisher // Annals of Eugenics. – 1936. – 7(2). – P. 179 – 188.
19. Fu K. S. Syntactic Methods in Hattern Recognition / King Sun Fu. – N.Y.: Academic Press, 1974. – 295 p.
20. Galushkin A.I. Neural Networks Theory / Alexander I. Galushkin. – Berlin; Heidelberg : Springer, 2007. – 420 p.
21. Grünwald P.D. The Minimum Description Length Principle / Peter D. Grünwald. – Cambridge, Mass. : MIT Press, 2007. – 682 p.
22. Gyorfi L., Gyorfi Z. An Upper Bound on the Asymptotic Error Probability of the k-Nearest Neighbor Rule for Multiple Classes / Laslo Gyorfi, Zoltan Gyorfi // IEEE Trans. IT. – 1978. – Vol. IT. – No. 4. – P. 512 – 514.
23. Looney C.C. Pattern Recognition Using Neural Networks: Theory and Algorithms for Engineers and Scientists / Carl L. Looney. – Oxford University Press, 1997. – 458 p.
24. Rendell L.A. Induction as optimization / Larry A. Rendell // IEE Trans. On Syst., Man, and Cybern. – 1990. – 20(2). – P. 326 – 338.
25. Theodoridis S. Pattern Recognition / S. Theodoridis, K. Koutroumbas. – N.Y.: Academic Press, 2006. – 837 p.

26. Vapnik V.N. Measuring of VC-Dimension of a Learning Machine / V.N. Vapnik // Neural Computation. – 1994. – Vol.6. – No 5. – P. 851 – 876
27. Vapnik V.N. Support-vector networks / C. Cortes, V.N. Vapnik // Machine Learning. –1995. – Vol. 20. – Issue 3. – P. 273 – 297.

Заключение

Современный этап развития математики и информатики характеризуется возрастанием интереса к индуктивным методам, в основании которых лежит извлечение закономерностей из эмпирики. Это объясняется необходимостью оперировать с большими объёмами накопленной информации (массивами прецедентов) с целью создания самых разнообразных автоматов, способных обучаться.

Одной из центральных постановок задач в указанной области является машинное обучение классификации или, можно сказать иначе, машинное обучение распознаванию свойств. Обнаруженные эмпирически и подтверждающиеся с достаточной степенью достоверности свойства называют закономерностями. Поэтому можно говорить об обучении как о машинном извлечении закономерностей.

Задачи классификации предполагают выдачу решения, являющегося выбором из одной или нескольких альтернатив. Примечательно, что их практические приложения и многочисленные реализации развиваются и внедряются так стремительно, что практика опережает теорию. Конструирование «разумных» алгоритмов обучения машин становится коммерчески выгодным делом, завоёвывая разнообразные области приложений: бытовую технику, компьютеры, производственные автоматы, роботы, военную технику.

Но не меньший интерес представляет математическая теория обучения машин, в поле зрения которой попадают вопросы, связанные с надежностью, точностью, трудоёмкостью синтеза классификаторов. В центре этих вопросов – обучаемость, как теоретическая возможность достижения нужного качества классификаторов. Обучаемость определяется, прежде всего, тем, каким является алгоритм (метод) обучения. Здесь обучение понимается как процесс, процедура, алгоритм, а обучаемость – как возможность достижения нужной цели – получения классификатора, обладающего нужной точностью и надёжностью. В главе 2 были представлены различные подходы к определению обучаемости.

Применяя самые разнообразные разделы математики – теорию вероятностей, функциональный анализ, геометрию, – теоретики, изучающие методы машинного обучения, обычно не учитывают тот факт, что предлагаемые ими *методы будут реализованы на конечных компьютерах. А ведь это должно вносить определённые коррективы в подходы к теоретическим выводам*.

Фундаментальную роль в исследовании обучаемости моделей построения алгоритмов классификации по прецедентной информации играет теория равномерной сходимости В.Н. Вапника – А.Я. Червоненкиса и особенно – введенное ими понятие *ёмкости класса решающих правил*, в кото-

ром отыскивается классифицирующий алгоритм. Эта характеристика сложности функциональных семейств получила название $VC-$размерности или VCD. Её важность, в частности, характеризуется таким строго доказанным фактом: семейство классификаторов является PAC обучаемым тогда и только тогда, когда $VCD(H)<\infty$.

В последние годы *уточнение задач обучения*, учет свойств вероятностных распределений, особенностей обучающих алгоритмов – что является, по сути, *использованием дополнительной информации – позволили установить обучаемость в некоторых случаях даже при бесконечной* $VC-$*размерности* используемых семейств. Но при этом, конечно, изменяются определения обучаемости и добавляются свойства распределений и/или алгоритмов обучения. Так, выяснилось, что обучаемость имеет место при условии устойчивости алгоритма обучения. Например, LOO устойчивость симметричного алгоритма обучения классификации *с* ограниченной функцией потерь является достаточным условием для обеспечения универсального эмпирического обобщения. А при использовании метода асимптотической минимизации риска универсальная RO устойчивость в среднем является необходимым и достаточным условием для обеспечения универсального эмпирического обобщения.

Теория равномерной сходимости, PAC обучаемость и универсальная способность к обобщению представляют собой достаточно широко определённые модели. В них не оговариваются ни свойства распределения вероятностей, ни особенности алгоритма обучения, которые могут быть произвольными. Фиксация свойств алгоритма обучения (в частности, его заведомая устойчивость) позволяют сузить модель обучения и вследствие этого получить обучаемость даже в случае бесконечной VC размерности семейства гипотез, в которое вложен образ $\mathrm{Im}\,A$ алгоритма обучения A. Можно говорить о ёмкости образа $\mathrm{Im}\,A$ как о реально действующей VC размерности.

Конечность VC размерности также перестаёт быть необходимым условием в некоторых случаях при конкретизации вероятностной меры (например, в случае диффузных или атомарных мер).

Дополнительно выявленные фундаментальные положения дают *объяснение практически наблюдаемой обучаемости при использовании некоторых алгоритмов и моделей обучения, несмотря на кажущееся противоречие с* VC *теорией: в действительности этого противоречия нет*.

Сами классифицирующие модели могут рассматриваться как функциональные суперпозиции. Таковыми являются и нейронные сети, и машины опорных векторов, и классификаторы по методу потенциальных функций, и логические классификаторы на основе дизъюнктивных нормальных форм.

Оценки точности обученных классификаторов чаще всего содержат некоторые входящие в них параметры, характеризующие *сложность классификатора, понимаемую в том или ином смысле*. Например, число слоёв и нейронов сети, число опорных векторов, число литералов в *ДНФ* и др. В дискретной постановке, на основе аппарата частично рекурсивных функций, факт предпочтительности более простых классификаторов находит строгое объяснение на основе колмогоровской алгоритмической сложности.

В последние десятилетия интенсивно развиваются подходы к обоснованию и оцениванию методов эмпирического обобщения на основе алгоритмической сложности и случайности. Прежде всего, имеется в виду колмогоровский поход в целом и предложенный на его основе метод MDL. Предположение, что более «простые» решающие правила чаще дают правильные решения, чем «сложные», оправдалась на практике и многие годы воспринималась как «гипотеза простой структурной закономерности».

Цель исследований в направлении, связанном со сжатием и поиском как можно более коротких описаний решающих правил, – понять природу сложности и получить на основе её изучения методы нахождения оценок качества алгоритмов обучения (эмпирического обобщения). Несмотря на некоторое продвижение в теории, такие оценки до сих пор не получены для многих классов алгоритмов. Это связано, прежде всего, с математическими трудностями вывода логико-комбинаторных оценок и отсутствием общего приёма их получения.

В книге описан достаточно общий подход к оцениванию – так называемый $pVCD$ метод, – который удалось разработать, ограничив все рассматриваемые семейства моделей эмпирического обобщения до классов, реализуемых на компьютерах, и шире, – рассматривая их частично-рекурсивные представления. В рамках алгоритмического подхода введено понятие колмогоровской сложности классов алгоритмов распознавания свойств или извлечения закономерностей. На основе этого понятия предложен метод оценивания неслучайности извлечения эмпирических закономерностей.

Установлено, что колмогоровская сложность $K_l(\mathfrak{A})$ семейства алгоритмов $\mathfrak{A}$ связана с $VCD(\mathfrak{A})$ двойным неравенством

$$VCD(\mathfrak{A}) \le K_l(\mathfrak{A}) < VCD(\mathfrak{A}) \cdot \log l$$

и равна наименьшему целому, большему или равному логарифму функции роста этого семейства: $K_l(\mathfrak{A}) =]\ \log m^{\mathfrak{A}}(l)[$. Такое же неравенство для размера сжатия k обучающей выборки было получено Флойдом и Вармутом:

$$VCD(\mathfrak{A}) \le k < VCD(\mathfrak{A}) \cdot \log l.$$

Эти неравенства показывают, что действующая ёмкость используемого семейства решающих правил является неустранимой, несжимаемой неопределённостью.

Сформулировано пригодное для практического оценивания *правило* «*плюс пять*»: *Для обеспечения надёжного извлечения закономерности* (*в виде решающего правила – алгоритма*) *из используемого семейства алгоритмов длина обучающей последовательности должна быть хотя бы на 5 единиц больше, чем колмогоровская сложность этого семейства*. При этом обеспечивается, что вероятность неслучайного обнаружения закономерности будет не меньше $0{,}96$.

Для понимания и применения правила "плюс пять" нужно учитывать, что задачи синтеза закономерностей (классификаторов) по прецедентной информации являются частным случаем проблемы принятия решений в условиях неопределённости. Это означает, что решения отыскиваются в широкой области, порождённой частичной информацией. Для любой задачи из рассматриваемого класса Z с начальной информацией I эта область неопределённости $\mathfrak{D}(Z, I)$ содержит огромное количество решений, включая нужное решение g. Кроме этого, о вероятностном распределении решений в области $\mathfrak{D}(Z, I)$ ничего не известно. Поэтому представляется естественным:

а) предположить такое распределение равномерным, что соответствует случаю наибольшей неопределённости;

б) попытаться как можно больше сузить (сжать) область $\mathfrak{D}(Z, I)$ до области $\mathfrak{D}'(Z, I)$, не потеряв при этом теоретическую возможность нахождения правильного решения: $g \in \mathfrak{D}'(Z, I) \subset \mathfrak{D}(Z, I)$.

В этом смысле выше шла речь об *обучении сжатием и* $pVCD$ *методе как аппарате такого обучения и оценивания классификаторов и закономерностей, синтезированных по начальной прецедентной информации*. В этом смысле $pVCD$ метод является одним из возможных вариантов обоснования эмпирических индукторов.

Подход, связанный с синтезом классификаторов наименьшей сложности, подробно проиллюстрирован в книге на примере семейства алгоритмов обучения, основанных на построении решающих деревьев.

Оценивание классификаторов как гипотез, синтезированных по обучающей выборке различными алгоритмами обучения, связано со многими факторами. Приходится учитывать и модель генеральной совокупности используемых выборок, и способ извлечения выборки из генеральной совокупности, и особенности алгоритма обучения – синтеза гипотез. Также

имеет значение поход к вычислению оценки точности. Он может осуществляться по всей заданной выборке, методом скользящего контроля или по тестовой выборке. Наконец, оценивание зависит и от того, какая модель обучения берётся за основу.

Можно выделить три основные группы методов оценивания классификаторов:

1. Оценивание синтезированных классификаторов по всей заданной обучающей выборке.
2. Оценивание по методу скользящего контроля.
3. Оценивание по независимой контрольной выборке.

Оценивание синтезированных классификаторов по всей выборке, представленной для обучения, приводит к получению смещенных оценок эмпирических ошибок. Это объясняется тем, что оценивание производится по той же выборке, которая использовалась для обучения. *Но именно этот препятствующий непосредственному оцениванию точности классификаторов факт и привёл к парадигме обучаемости как способности к обобщению информации, представленной обучающей выборкой.*

Оценивание по методу скользящего контроля ($k-fold$ *Cross Validation*) предполагает, что из заданной выборки длины L поочередно исключаются $k < L$ элементов. Получаются две выборки с длинами $l = L - k$ и l. На первой – производится обучение, а по второй – как контрольной – вычисляется частота ошибок ν_i построенного в результате обучения классификатора. Такой процесс повторяется C_L^k раз. В итоге получается оценка точности алгоритма обучения $\overline{\nu} = \frac{1}{C_L^k}\sum_{i=1}^{C_L^k}\nu_i$. При значении $k = 1$ скользящий контроль соответствует правилу LOO и нахождению ν_{LOO} ошибки.

Когда исходная обучающая выборка состоит из случайно и независимо выбранных из генеральной совокупности объектов, средняя ошибка скользящего контроля даёт несмещенную оценку вероятности ошибки. Однако для оценивания точности классификаторов нужно знать еще и дисперсию этой ошибки. Считается, что такие оценки неизвестны – их найти до настоящего времени не удалось. А сравнительно недавно выяснилось, что несмещенных оценок дисперсии для $k-fold$ скользящего контроля не существует.

Использование слабой вероятностной аксиоматики, скользящего контроля, учет свойств метода обучения и применяемых классов гипотез позволили К.В. Воронцову получить в рамках *комбинаторной теории переобучения* (VCT) оценки обучаемости существенно лучшие, чем оценки,

основанные на применении VCD класса гипотез, из которого выбирается алгоритм классификации при обучении. Однако при отыскании оценок обучаемости для каждой вновь исследуемой модели приходится сталкиваться с существенной трудоёмкостью такой научной работы (конечно, со временем об этом недостатке говорить не придётся: все модели будут изучены в рамках VCT).

При оценивании по независимой контрольной выборке не возникает никаких проблем, связанных с «подгонкой» классификатора, поскольку контрольная выборка применяется к фиксированному решающему правилу уже после того, как оно выбрано. Оценки вероятности ошибки по контрольной выборке являются несмещенными. Если вероятность ошибки выбранного в результате обучения классификатора в действительности равна p, то схема её оценивания соответствует вероятностной модели l независимых испытаний с двумя исходами, которую называют схемой Бернулли.

Важно подчеркнуть, что достаточная для обучаемости длина выборки – сложность выборки, и длина контрольной выборки, требуемая для оценивания уже синтезированного зафиксированного классификатора – совершенно разные понятия. И сравнивать их не имеет сысла.

Широкое распространение в последние десятилетия компьютеров, компьютерных сетей и электронных источников информации даёт возможность получения значительно бóльших по объёму обучающих и контрольных выборок, чем это было на начальном этапе развития теории и практики машинного обучения. Поэтому можно рассчитывать на пригодность (в числе прочих подходов) чебышевских оценок для бернуллиевской модели вероятности ошибок синтезированных классификаторов.

Обоснование выбора подхода к решению конкретной задачи обучения классификации – нетривиальная проблема. Но она, как ни странно, часто остается в тени; усилия исследователей направлены на создание алгоритмов обучения и оценивание вероятности ошибок распознавания. В книге изложен и обоснован подход к определению *областей применимости основных теорий обучения и классификации*. *Соответствующие области представляют собой классы задач, определяемые общностью их основных свойств*.

Установлено принципиальное различие между обучением и настройкой – подбором произвольного решения функциональной системы, определяющей допустимый искомый классификатор. Выделены несколько типов или стратегий обучения классификаторов.

Обучением «снизу вверх» называется такой поэтапный процесс построения решающего правила $\hat{f}_0$, на каждом этапе которого происходит *минимальное необходимое усложнение* искомого правила, обеспечивающее

уменьшение числа его ошибок на обучающей выборке. При обучении «снизу вверх» происходит поэтапное усложнение решающего правила и, соответственно, расширение семейства, которому оно принадлежит. Обучение в рассматриваемом случае будет успешным, если для коррекций будут использованы не все точки обучающей выборки, а только часть: оставшаяся часть не использованных для коррекций точек должна классифицироваться правильно и «подтверждать» построенное решающее правило. Выбор начального приближения и способы поэтапного усложнения решающего правила определяют алгоритм (метод) обучения.

Обучением «сверху вниз» называется последовательный процесс нахождения решающего правила $\hat{f}_0$, принадлежащего некоторому подклассу минимальной сложности H' из выбранного изначально некоторым способом семейства H, направленный на достижение наибольшей точности правила $\hat{f}_0$ на заданной обучающей выборке. Для обоснования алгоритма обучения по методу «сверху вниз» целесообразно приводить доказательство адекватности изначально заданного семейства H – наличии в нём истинного решающего правила, а также сохранении этого свойства при поэтапном сужении начального семейства.

Комбинированным обучением с возвратом называется процесс построения решающего правила, сочетающий оба метода обучения – «снизу вверх» и «сверху вниз». Такой процесс аналогичен поиску с возвратом (*backtracking*). Для решающих деревьев соответствующим примером являются процедуры нахождения классификатора на основе оценок его текущей сложности, включающие условия возврата к более простому варианту, когда сложность становится выше заданного порога.

Адаптивным обучением называется процесс пошаговой коррекции параметрической модели с такими же условиями – по минимальному числу примеров, используемых для коррекции, как и в случае обучения «*снизу вверх»*. Классический пример адаптивного обучения – алгоритм линейной коррекции Розенблатта-Новикова, который лег в основу обучения всех параметрических моделей.

Комбинированным адаптивным обучением называется процесс направленного обучения с адаптацией параметров классификатора. Например, структурно-адаптивный метод обучения нейронной сети, когда параметры «соединения» в структуре сети в процессе пошагового обучения с целью уменьшения ошибок могут «сбрасываться» в ноль, обеспечивая упрощение структуры и сложности классификатора.

Обучением путём сжатия данных называется процесс синтеза решающего правила, которое может быть определено как можно меньшим числом примеров d из заданной обучающей выборки длины l. Оставшиеся $l-d$ примеров в таком случае «безоговорочно подтверждают» постро-

енное решение. В качестве примера можно привести машину опорных векторов.

В заключение можно повторить, что машинное обучение, классификация, распознавание – широчайшая область науки и приложений в кибернетике и информатике. Как давно её определил Л. Канал – это совокупность методов и совокупность задач. И ориентироваться в этой многообразной и привлекающей кажущейся простотой совокупности в действительности достаточно сложно: нужен опыт и глубокое понимание предмета.

Основные обозначения

N	– расширенное (с нулём) множество натуральных чисел.
Q	– множество рациональных чисел.
R	– множество вещественных чисел.
$\{0,1\}^{\infty}$	– множество любых конечных и бесконечных двоичных последовательностей
$\{0,1\}^{*}$	– множество любых конечных двоичных последовательностей любой длины
$\lvert A\rvert$	– число элементов в конечном множестве A
$\mathscr{B}(A)$	– булеан множества A
$]a[$	– Наименьшее целое, большее или равное a
$\log y$	– $\log_2 y$.
C_l^k	– число сочетаний из l элементов по k.
$l(x)$ и $\lvert x\rvert$	– длина строки x.
$\tilde{x}$	– $\tilde{x} = (x_1,\ldots,x_n)$ – вектор, описывающий объекты произвольной предметной области. Каждая его координата называется признаком.
D_i, $i = \overline{1,n}$	– множество допустимых значений признака x_i.
$\mathbf{X}$	– признаковое пространство всевозможных векторов $\tilde{x}$.
2^{X}	– булеан над X – множество всех подмножеств множества X.
$H \subseteq 2^{\mathrm{X}}$	– класс концептов, множество гипотез.
$G \subset 2^{\mathrm{X}}$	– класс концептов, содержащий целевой концепт $g \in G$.
$s(g)$	– Строка, являющаяся описанием функции (концепта) g.
n	– размерность признакового пространства.
l	– длина обучающей выборки.
$(\tilde{x}_j, \alpha_j)_{j=1}^{l}$	– обучающая выборка длины l; $\alpha_j = g(\tilde{x}_j)$, где $g : \mathrm{X} \to \{0,1\}$ – заранее неизвестная (целевая) функция.
X_l	– краткое обозначение обучающей выборки длины l.
X^l	– множество любых обучающих выборок длины l.
$A : \mathrm{X}^l \to H$	– алгоритм обучения (алгоритмическое отображе-

ние).

$\operatorname{Im} A \subseteq H$ – образ алгоритмического отображения A во множестве гипотез H.

P – вероятностная мера на $\mathrm{X} \times \{0,1\}$.

$\mathbf{E}$ – математическое ожидание относительно вероятностного распределения P

P^l – вероятностная мера на выборках $(\mathrm{X} \times \{0,1\})^l$

$\mathbf{E}_l$ – математическое ожидание относительно вероятностного распределения P^l.

$\mathcal{P}$ – семейство всевозможных вероятностных распределений на $\mathrm{X} \times \{0,1\}$.

$\mathcal{P}^l$ – семейство всевозможных вероятностных распределений на $(\mathrm{X} \times \{0,1\})^l$

$Err(h)$ – ошибка гипотезы h;
$Err(h) = P\{(\tilde{x}, \alpha) : h(\tilde{x}) \neq \alpha\}$

$Err_l(h)$ или $\nu_{emp} = \dfrac{\kappa}{l}$ – эмпирическая ошибка (на выборке X_l) гипотезы h;
$Err_l(h) = \dfrac{1}{l}\left|\{(\tilde{x}, \alpha) \in X_l : h(\tilde{x}) \neq \alpha\}\right|$;
κ – число примеров из l, неправильно классифицированных гипотезой h.

$\lambda(h, \tilde{x}) = \begin{cases} 1, & h(\tilde{x}) = \alpha; \\ 0, & h(\tilde{x}) \neq \alpha. \end{cases}$ – бинарная функция потерь; α – истинное значение целевой функции в точке $\tilde{x}$, а $h = A(X_l)$ – выбранная обучающим алгоритмом A по выборке $(\tilde{x}, \alpha)_{j=1}^{l}$ длины l решающая функция.

$L(h, \tilde{x}) = \begin{cases} 0, & h(\tilde{x}) = \alpha; \\ m(\tilde{x}), & h(\tilde{x}) \neq \alpha. \end{cases}$ – произвольная симметричная относительно ошибок первого и второго рода функция потерь

$KS(x)$ – колмогоровская сложность слова (строки) x, которая в статьях Колмогорова обозначалась $K(x)$. Иногда эту сложность называют простой колмогоровской сложностью, но в таком названии усматривается оксюморон, поэтому представляется предпочтительным использовать для этого исходного понятия название «колмогоровская сложность». В некоторых работах (например, у Ли и Витаньи) $K(x)$ обозначает префиксную сложность, а для исходного понятия используется обо-

значение $C(x)$.

$KS_D(x \mid y)$ – Условная колмогоровская сложность слова x при заданном слове y и при заданном способе описания – вычислимой функции (декомпрессоре) D.

$KC(x)$ – точная колмогоровская сложность, минимальная по всем декомпрессорам.

$KP(x)$ – префиксная сложность слова x.

$KPC(x \mid y)$ – точная условная префиксная сложность.

$KM(x)$ – монотонная сложность. Ли и Витаньи обозначают монотонную сложность как $Km(x)$.

$KR(x)$ – сложность разрешения.

$\mathrm{T}_C(x)$ – множество всех вычислимых функций-компрессоров, обеспечивающих сжатие слова x.

$K_T(x)$ – сжатие строки x наилучшим (для этой строки) компрессором.

$K_l(\mathfrak{A})$ – Колмогоровская сложность семейства алгоритмов (частично рекурсивных функций) $\mathfrak{A}$ относительно класса дискретных обучающих выборок длины l.

MT – машина Тьюринга.

m – максимальная перечислимая снизу полумера (универсальное вероятностное распределение), для которой имеет место равенство $-\log m(x) = KP(x) + O(1)$.

$Dom_1(\varphi)$ – подмножество области определения предиката φ, на котором этот предикат принимает значение 1.

$\Pr(E)$ – вероятность события E по соответствующей мере.

$VCD(H)$ – размерность Вапника-Червоненкиса семейства функций или концептов H.

$\boldsymbol{P}_{\boldsymbol{comp}} = P_{p.r.}$ – Класс вычислимых функций, совпадающий с классом частично рекурсивных функций.

Предметный указатель

Содержание

Донской Владимир Иосифович

АЛГОРИТМИЧЕСКИЕ МОДЕЛИ ОБУЧЕНИЯ КЛАССИФИКАЦИИ: ОБОСНОВАНИЕ, СРАВНЕНИЕ, ВЫБОР

Научное издание

Ответственный за выпуск Шторгин Д.

Формат 60x84/16. Усл. печ. листов 13,25. Тираж 300. Заказ № 14004/093

Издательство «ДИАЙПИ»
г. Симферополь, пр. Кирова, 17. Тел./факс (0652) 248-178, 711-687
dip@diprint.com.ua, www.diprint.com.ua

Свидетельство о госрегистрации ДК №1744 от 8.04.2004 г.

Отпечатано с готового оригинал-макета в полиграфцентра «КУБ»
295000, г. Симферополь, пр. Тренева, 1. Тел. 0504971790