

# Теория надёжности обучения по прецедентам (комбинаторная теория переобучения) Курс лекций

К. В. Воронцов

<http://www.MachineLearning.ru> - Участник:Vokov

[voron@forecsys.ru](mailto:voron@forecsys.ru)

15 мая 2011 г.

Материал находится в стадии разработки, может содержать ошибки и неточности. Автор будет благодарен за любые замечания и предложения, направленные по адресу [vokov@forecsys.ru](mailto:vokov@forecsys.ru), либо высказанные в обсуждении страницы «Теория надёжности обучения по прецедентам (курс лекций, К.В.Воронцов)» вики-ресурса [www.MachineLearning.ru](http://www.MachineLearning.ru).

Перепечатка фрагментов данного материала без согласия автора является плагиатом.

## Содержание

<b>1</b>	<b>Проблема переобучения и слабая вероятностная аксиоматика</b>	<b>6</b>
1.1	Обучение и переобучение . . . . .	7
1.2	Слабая вероятностная аксиоматика . . . . .	11
1.3	Вероятность переобучения . . . . .	12
1.4	Обсуждение слабой аксиоматики . . . . .	14
	Резюме . . . . .	17
	Упражнения . . . . .	17
<b>2</b>	<b>Оценивание частоты и гипергеометрическое распределение</b>	<b>18</b>
2.1	Задача оценивания (предсказания) частоты события . . . . .	18
2.2	Гипергеометрическое распределение . . . . .	19
2.3	Закон больших чисел в слабой аксиоматике . . . . .	21
2.4	Переход от ненаблюдаемой оценки к наблюдаемой . . . . .	23
2.5	Одноэлементное семейство алгоритмов . . . . .	26
	Резюме . . . . .	26
	Упражнения . . . . .	27
<b>3</b>	<b>Непараметрические критерии и доверительные оценки</b>	<b>28</b>
3.1	Доверительное оценивание . . . . .	28
3.2	Доверительные интервалы для квантилей . . . . .	29
3.3	Критерий знаков . . . . .	30
3.4	Критерий Уилкоксона–Манна–Уитни . . . . .	32
	Резюме . . . . .	34
	Упражнения . . . . .	35

<b>4</b>	<b>Эмпирические распределения и случайное блуждание</b>	<b>36</b>
4.1	Эмпирическое распределение . . . . .	36
4.2	Усечённый треугольник Паскаля . . . . .	37
4.3	Теорема Смирнова . . . . .	39
4.4	Обобщение на случай вариационного ряда со связками . . . . .	41
	Резюме . . . . .	43
<b>5</b>	<b>Теория Вапника–Червоненкиса</b>	<b>45</b>
5.1	Оценка Вапника–Червоненкиса . . . . .	45
5.2	Размерность Вапника–Червоненкиса . . . . .	48
5.3	Метод структурной минимизации риска . . . . .	49
5.4	Проблема завышенности VC-оценок . . . . .	50
	Резюме . . . . .	54
	Упражнения . . . . .	55
<b>6</b>	<b>Размерность Вапника-Червоненкиса</b>	<b>56</b>
6.1	Связь ёмкости с функцией роста . . . . .	56
6.2	Функция роста множества конъюнкций . . . . .	57
6.3	Ёмкость семейства линейных классификаторов . . . . .	58
6.4	Однопараметрическое семейство бесконечной ёмкости . . . . .	59
6.5	Другие оценки ёмкости . . . . .	60
	Резюме . . . . .	60
<b>7</b>	<b>Принцип порождающих и запрещающих множеств</b>	<b>61</b>
7.1	Простая гипотеза ПЗМ . . . . .	61
7.2	Обобщённая гипотеза ПЗМ . . . . .	63
7.3	Корректное семейство алгоритмов . . . . .	66
7.4	Функционал полного скользящего контроля . . . . .	66
	Резюме . . . . .	67
<b>8</b>	<b>Цепи алгоритмов</b>	<b>68</b>
8.1	Разновидности минимизации эмпирического риска . . . . .	68
8.2	Эксперименты с модельными семействами алгоритмов . . . . .	69
8.3	Связные семейства алгоритмов . . . . .	71
8.4	Точная оценка для монотонной цепи алгоритмов . . . . .	74
8.5	Точная оценка для унимодальной цепи алгоритмов . . . . .	77
	Резюме . . . . .	80
	Упражнения . . . . .	80
	Практикум . . . . .	81
<b>9</b>	<b>Оценки расслоения–связности</b>	<b>84</b>
9.1	Граф расслоения–связности . . . . .	84
9.2	Оценки расслоения–связности . . . . .	85
9.3	Профиль расслоения–связности . . . . .	89

Резюме . . . . .	90
Упражнения . . . . .	91
<b>10 Многомерные сети алгоритмов</b>	<b>92</b>
10.1 Двумерная монотонная сеть алгоритмов . . . . .	92
10.2 Монотонная сеть произвольной размерности . . . . .	93
10.3 Точная оценка расслоения–связности . . . . .	95
10.4 Вычислительные эксперименты . . . . .	97
Резюме . . . . .	99
Упражнения . . . . .	99
<b>11 Конъюнктивные логические закономерности</b>	<b>100</b>
11.1 Логические методы классификации . . . . .	100
11.2 Конъюнкции элементарных пороговых правил . . . . .	103
11.3 Послойное вычисление оценки расслоения–связности . . . . .	106
11.4 Эксперимент на реальных задачах классификации . . . . .	109
Резюме . . . . .	110
Упражнения . . . . .	111
<b>12 Рекуррентные оценки</b>	<b>113</b>
12.1 Добавление одного алгоритма . . . . .	113
12.2 Вычисление вероятности переобучения . . . . .	115
12.3 Упрощённая рекуррентная оценка . . . . .	117
Резюме . . . . .	118
Упражнения . . . . .	118
<b>13 Симметричные семейства алгоритмов</b>	<b>120</b>
Основы теоретико-группового подхода . . . . .	120
Резюме . . . . .	125
Упражнения . . . . .	125
<b>14 Оценки вероятности равномерного отклонения</b>	<b>126</b>
14.1 Техника порождающих и запрещающих множеств . . . . .	126
14.2 Техника цепных разложений . . . . .	128
14.3 Техника случайных блужданий . . . . .	129
14.4 Понятие радемахеровской сложности . . . . .	130
Резюме . . . . .	131
Упражнения . . . . .	131
<b>15 Точные интервальные и блочные оценки</b>	<b>132</b>
15.1 Слой булева куба . . . . .	132
15.2 Интервал булева куба и его расслоение . . . . .	133
15.3 Блочная оценка . . . . .	135
15.4 Пара алгоритмов . . . . .	137

Резюме . . . . .	140
Упражнения . . . . .	140
<b>16 Метод ближайшего соседа</b>	<b>141</b>
16.1 Профиль компактности выборки . . . . .	141
16.2 Точная оценка полного скользящего контроля . . . . .	142
16.3 Отбор эталонных объектов . . . . .	144
16.4 Эксперименты и выводы . . . . .	146
Резюме . . . . .	149
Упражнения . . . . .	149
<b>17 Монотонные классификаторы</b>	<b>150</b>
17.1 Профиль монотонности выборки . . . . .	151
17.2 Верхняя оценка полного скользящего контроля . . . . .	153
17.3 Точные оценки полного скользящего контроля . . . . .	155
Резюме . . . . .	158
Упражнения . . . . .	159
<b>18 Приложение. О комбинаторной теории переобучения</b>	<b>160</b>
18.1 Постановка задачи . . . . .	160
18.2 Долгое топтание на месте . . . . .	161
18.3 Эксперименты с переобучением . . . . .	162
18.4 «Игрушечные» частные случаи . . . . .	165
18.5 Поиск обобщений и альтернативных подходов . . . . .	166
<b>Список литературы</b>	<b>171</b>

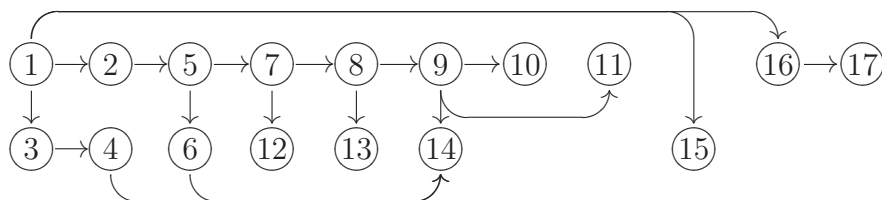
Спецкурс «Теория надёжности обучения по прецедентам» читается студентам кафедры «Математические методы прогнозирования» ВМК МГУ с 2007 года в дополнение к обязательному кафедральному курсу «Математические методы распознавания образов». Спецкурс знакомит студентов с современным состоянием теории статистического обучения (Statistical Learning Theory, SLT), которая занимается проблемами восстановления зависимостей по эмпирическим данным. Родоначальниками этой теории были советские математики В. Н. Вапник и А. Я. Червоненкис. В 80-е годы эта теория получила широкую мировую известность, и в настоящее время развивается очень активно. Один из основных вопросов SLT — как количественно оценить способность алгоритмов классификации и прогнозирования к обобщению эмпирических фактов. В каких случаях можно утверждать, что общие закономерности, выявленные по частным прецедентам, не окажутся ложными? Как избежать переобучения — ситуации, когда ответы алгоритма слишком точны на обучающей выборке, но недостаточно точны на новых данных, неизвестных на этапе обучения?

В данном пособии излагается часть курса, основанная на результатах автора, его учеников и коллег: Д. Кочедыкова, А. Ивахненко, И. Решетняка, А. Фрея, И. Толстихина, П. Ботова, М. Иванова, Г. Махиной. Другая часть курса, посвящённая обзору современных достижений SLT, не вошла в данное пособие.

Основная цель спецкурса — дать глубокое понимание явления переобучения, необходимое при решении прикладных задач интеллектуального анализа данных. Вторая цель — показать, как создаётся математическая теория, вовлечь студентов в этот увлекательный творческий процесс.

Каждый параграф соответствует одной лекции. В конце большинства лекций имеются упражнения. Для каждого упражнения в скобках указана оценка сложности. Упражнения, помеченные звёздочкой, являются открытыми проблемами на момент написания пособия. Это означает, что они наиболее интересны, хотя и не обязательно сложны — до многих задач просто не доходили руки. В конце некоторых лекций имеются задания для вычислительного практикума.

Ниже приведена схема зависимости параграфов. Лекции верхнего ряда образуют основу курса, нижние являются дополнительными.



Материал спецкурса обновляется на странице «Теория надёжности обучения по прецедентам (курс лекций, К.В.Воронцов)» вики-ресурса [www.MachineLearning.ru](http://www.MachineLearning.ru).

Автор будет рад любым замечаниям и предложениям, направленным на email [voron@forecsys.ru](mailto:voron@forecsys.ru).

*Константин Воронцов,  
апрель 2011 г.*

# 1 Проблема переобучения и слабая вероятностная аксиоматика

Машинное обучение или *обучение по прецедентам* — это наука о том, как научить компьютер решать задачи прогнозирования и принятия решений в условиях, когда знаний о предметной области не хватает для построения обоснованных математических моделей, но зато имеются значительные массивы эмпирических данных.

Типичный пример — задачи медицинской диагностики. Здесь под данными понимаются электронные истории болезни. Данные об отдельном клиническом случае включают анамнез, результаты обследований, назначенные лечебные мероприятия, показатели результативности лечения, и т. д. Требуется построить алгоритм, который на основе имеющейся информации о новом пациенте мог бы поставить диагноз, рекомендовать лечение или предсказать исход заболевания. Задача *обучения по прецедентам* заключается в том, чтобы построить такой алгоритм на основе *обучающей выборки* — совокупности прецедентов, наблюдавшихся в прошлом, для которых правильные диагнозы (решения, исходы) уже известны.

Другой пример — задача *кредитного скоринга*. Здесь прецеденты — это заявки на получение кредита в банке. Анкетные данные заявителя включают: возраст, пол, образование, профессию, стаж работы, доход семьи, размер задолженностей в других банках, наличие телефона, и т. д. Задача состоит в том, чтобы по обучающей выборке заёмщиков с уже известной кредитной историей построить алгоритм, предсказывающий, будут ли у данного заявителя проблемы с погашением кредита.

Нетрудно построить алгоритм, который выдаёт правильные решения для объектов обучающей выборки. Например, он мог бы сравнивать новый объект с каждым из обучающих объектов, и в случае полного совпадения выдавать правильное решение, записанное в таблице исходных данных, а во всех остальных случаях выбирать случайное решение. Очевидно, такой алгоритм никого не устроит. Для успешного обучения важно не только запоминать, но и обобщать. Способность алгоритмов правильно находить общие закономерности по частным эмпирическим данным называют *обобщающей способностью* (generalization ability).

Основная задача *теории статистического обучения*<sup>1</sup> (statistical learning theory, SLT) заключается в том, чтобы давать статистически обоснованные количественные оценки обобщающей способности и затем на их основе конструировать обучаемые алгоритмы, надёжно работающие вне материала обучения.

Одна из основных проблем SLT — относительно низкая точность оценок. Чрезмерно осторожные оценки, рассчитанные на худший случай, на практике могут приводить к ошибочным решениям. В данном курсе лекций рассматривается новое направление в SLT — комбинаторная теория обобщающей способности, позволяющая более точно оценивать вероятность переобучения.

---

<sup>1</sup>Второе название — *теория вычислительного обучения* (computational learning theory, COLT). Различия между COLT и SLT довольно условны. В частности, COLT включает в себя проблематику вычислительной сложности алгоритмов обучения.

## §1.1 Обучение и переобучение

Пусть задано конечное множество объектов  $\mathbb{X} = \{x_1, \dots, x_L\}$ , называемое *генеральной выборкой*; множество  $A$ , элементы которого называются *алгоритмами*, и бинарная функция  $I: A \times \mathbb{X} \rightarrow \{0, 1\}$ , называемая *индикатором ошибки*. Если  $I(a, x) = 1$ , то говорят, что алгоритм  $a$  ошибается на объекте  $x$ .

Число ошибок алгоритма  $a$  на выборке  $X \subseteq \mathbb{X}$  есть

$$n(a, X) = \sum_{x \in X} I(a, x).$$

*Частотой ошибок* алгоритма  $a$  на выборке  $X \subseteq \mathbb{X}$  называется отношение

$$\nu(a, X) = n(a, X)/|X|.$$

**Задача обучения по прецедентам.** Допустим, что генеральная выборка разбита на две подвыборки,  $\mathbb{X} = X \sqcup \bar{X}$ . Выборка  $X$  называется *наблюдаемой* или *обучающей*, для объектов  $x \in X$  известны значения индикатора ошибки  $I(a, x)$ . Выборка  $\bar{X}$  называется *скрытой* или *контрольной*, и на ней значения индикатора ошибки неизвестны. Задача состоит в том, чтобы найти алгоритм  $a \in A$  с минимальным числом ошибок на генеральной выборке  $n(a, \mathbb{X})$ , пользуясь только информацией о наблюдаемой выборке. Данная задача в общем случае не имеет точного решения, поскольку алгоритм  $a$  выбирается по неполной информации. Поэтому ставится задача поиска приближённого решения и оценивания его точности.

*Методом обучения*<sup>2</sup> называется отображение  $\mu: 2^{\mathbb{X}} \rightarrow A$ , которое произвольной обучающей выборке  $X \subset \mathbb{X}$  ставит в соответствие некоторый алгоритм  $a = \mu X$  из  $A$ .

Наиболее естественная стратегия обучения — найти алгоритм, допускающий наименьшее число ошибок на обучающей выборке. Обозначим через  $A(X)$  подмножество алгоритмов  $a$ , на которых число ошибок  $n(a, X)$  минимально:

$$A(X) = \text{Arg} \min_{a \in A} n(a, X) = \{a \in A: n(a, X) \leq n(a', X), \forall a' \in A\}. \quad (1.1)$$

Если  $\mu X \in A(X)$  для любой обучающей выборки  $X \subset \mathbb{X}$ , то метод  $\mu$  называется методом *минимизации эмпирического риска*.

*Переобученностью* алгоритма  $a$  при разбиении  $X \sqcup \bar{X} = \mathbb{X}$  называется разность частот его ошибок на контроле и обучении:

$$\delta(a, X) = \nu(a, \bar{X}) - \nu(a, X).$$

*Переобученностью* метода  $\mu$  называется переобученность алгоритма  $a = \mu X$ :

$$\delta_{\mu}(X) = \delta(\mu X, X) = \nu(\mu X, \bar{X}) - \nu(\mu X, X).$$

---

<sup>2</sup>В англоязычной литературе метод обучения принято называть *алгоритмом обучения* (learning algorithm) [71], а алгоритм  $a: \mathbb{X} \rightarrow \mathbb{Y}$  — классификатором (classifier), гипотезой (hypothesis), решающей функцией (decision function), либо просто функцией (function). Термин «алгоритм» в смысле отображения  $\mathbb{X} \rightarrow \mathbb{Y}$  употребляется в работах научной школы академика Ю. И. Журавлёва [23]. Термины «метод» и «алгоритм», обозначающие процедуру построения функции  $a$  по выборке данных употребляются в отечественной литературе попеременно [10, 1, 25, 24].

	$a_1$	$a_2$	$a_3$	$a_4$	$a_5$	$a_6$	$a_7$	$a_8$	
$x_1$	1	1	1	<b>0</b>	0	1	1	1	$X$ — наблюдаемая обучающая выборка
$x_2$	0	0	0	<b>0</b>	1	1	0	0	
$x_3$	0	1	1	<b>0</b>	0	0	0	0	
$x_4$	1	0	1	<b>0</b>	1	0	1	0	
$x_5$	0	0	1	<b>0</b>	1	1	0	0	
$x_6$	0	0	0	<b>1</b>	1	1	0	0	$\bar{X}$ — скрытая контрольная выборка
$x_7$	1	0	0	<b>1</b>	1	1	0	0	
$x_8$	0	0	0	<b>1</b>	0	0	0	1	
$x_9$	0	1	1	<b>1</b>	1	1	0	0	
$x_{10}$	0	1	1	<b>1</b>	1	1	0	0	

**Рис. 1.1.** Пример матрицы ошибок,  $L = 10$ ,  $D = 8$ ,  $\ell = k = 5$ . Показано одно из  $C_{10}^5$  разбиений выборки на наблюдаемую и скрытую подвыборки. Метод минимизации эмпирического риска выбирает алгоритм  $a_4$  и является переобученным относительно данной пары выборок, причём при любом  $\varepsilon \in (0, 1)$ .

Если  $\delta_\mu(X) \geq \varepsilon$  при некотором достаточно малом  $\varepsilon \in (0, 1)$ , то говорят, что метод  $\mu$  *переобучается* на выборке  $X$ .

**Матрица ошибок.** Бинарный вектор-столбец  $\vec{a} = (I(a, x_i))_{i=1}^L$  будем называть *вектором ошибок* алгоритма  $a$ . Совокупность всех попарно различных векторов ошибок, порождаемых алгоритмами  $a \in A$ , образует *матрицу ошибок* размера  $L \times D$ . Строки этой матрицы соответствуют объектам, столбцы — алгоритмам. Число столбцов  $D$  может быть меньше  $|A|$ , так как различные алгоритмы могут порождать одинаковые векторы ошибок. Множество алгоритмов  $A$  вполне может быть и бесконечным, однако число  $D$  попарно различных векторов ошибок всегда конечно и не превышает  $2^L$ . В дальнейшем именно матрица ошибок с попарно различными столбцами будет для нас основным объектом исследования.

**Пример 1.1.** Матрица ошибок на рис. 1.1 разбита на обучающую и контрольную выборки так, что алгоритм  $a_4$ , минимизирующий эмпирический риск, допускает ошибки на всех объектах контрольной выборки. Это и есть переобучение.

Можно предположить, что в данном примере переобучение оказалось следствием неудачного разбиения выборки на обучение и контроль. В дальнейшем мы введём *вероятность переобучения* — величину, которая характеризует выборку  $\mathbb{X}$  и метод обучения  $\mu$ , и не зависит от случайного разбиения  $X, \bar{X}$ .

**Задачи классификации и восстановления регрессии.** Допустим, что каждому объекту  $x \in \mathbb{X}$  соответствует *правильный ответ*  $y(x) \in \mathbb{Y}$ . Функция  $y: \mathbb{X} \rightarrow \mathbb{Y}$  называется *целевой зависимостью* (target function).

В качестве *алгоритмов* будем рассматривать функции того же вида  $a: \mathbb{X} \rightarrow \mathbb{Y}$ , допускающие эффективную реализацию на компьютере.

В качестве множества  $A$  чаще всего задаётся некоторое параметрическое *семейство алгоритмов*  $A = \{\varphi(x, \theta): \theta \in \Theta\}$ , где  $\varphi: \mathbb{X} \times \Theta \rightarrow \mathbb{Y}$  — фиксированная функция,  $\Theta$  — множество допустимых значений параметра  $\theta$ , называемое *пространством параметров* или *пространством поиска* (search space).



В качестве *индикатора ошибки* возьмём бинарную функцию  $I: A \times \mathbb{X} \rightarrow \{0, 1\}$ , равную 1, когда предсказание  $a(x)$  существенно отличается от правильного ответа  $y(x)$ . Индикатор ошибки может определяться по-разному в зависимости от постановки задачи, главным образом — от природы множества допустимых ответов  $\mathbb{Y}$ .

В задачах *классификации* множество классов  $\mathbb{Y}$  конечно, и индикатор ошибки чаще всего задаётся в виде<sup>3</sup>

$$I(a, x) = [a(x) \neq y(x)], \quad x \in \mathbb{X}, a \in A.$$

В задачах *восстановления регрессии* и многих задачах *прогнозирования*  $\mathbb{Y} = \mathbb{R}$ , и величину ошибки принято характеризовать непрерывной *функцией потерь* (loss function), например, квадратичной:  $\mathcal{L}(a, x) = (a(x) - y(x))^2$ . Тем не менее, можно определять и бинарные функции потерь, например,

$$I(a, x) = [|a(x) - y(x)| \geq \sigma(x)], \quad x \in \mathbb{X}, a \in A,$$

где  $\sigma(x)$  — пороговый уровень, выше которого отклонение считается ошибкой. Заметим, что бинарная функция потерь является *робастной*, то есть нечувствительной к *выбросам* — большим отклонениям ответа алгоритма  $a(x)$  от истинного  $y(x)$ .

**Пример 1.2.** Пусть объектами являются  $n$ -мерные числовые векторы,  $\mathbb{X} \subset \mathbb{R}^n$ . Обозначим через  $\langle \xi, \theta \rangle = \xi_1 \theta_1 + \dots + \xi_n \theta_n$  скалярное произведение векторов в  $\mathbb{R}^n$ . *Линейные семейства алгоритмов* определяются следующим образом:

$$A = \{a(x) = \text{sign} \langle x, \theta \rangle : \theta \in \mathbb{R}^n\} \text{ — для задач классификации, } \mathbb{Y} = \{-1, +1\};$$

$$A = \{a(x) = \langle x, \theta \rangle : \theta \in \mathbb{R}^n\} \text{ — для задач восстановления регрессии, } \mathbb{Y} = \mathbb{R}.$$

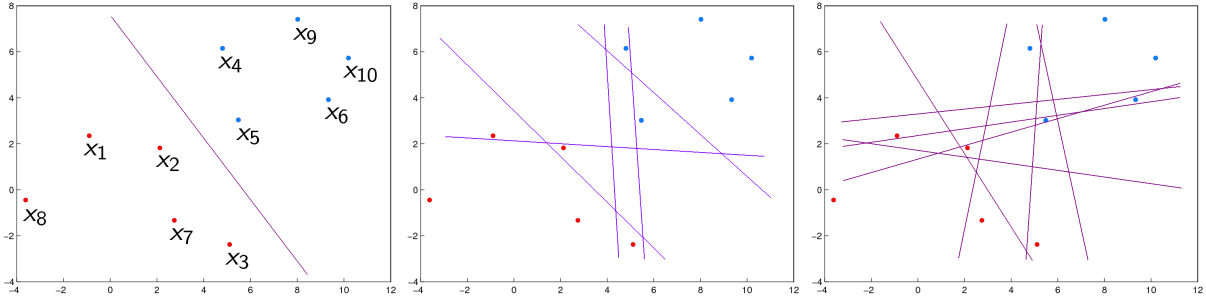
Линейный алгоритм классификации представляет собой гиперплоскость с направляющим вектором  $\theta$ , разделяющую пространство на две области — классы  $-1$  и  $+1$ .

В теории переобучения понятие «алгоритма» можно не конкретизировать, предполагая лишь, что это элементы некоторого абстрактного множества  $A$ . Главное, чтобы для любого алгоритма  $a$  была возможность определить, допускает ли он ошибку на объекте  $x$ . Такое понимание «алгоритма» с одной стороны расширяет класс рассматриваемых задач, но с другой стороны ограничивает его теми задачами, в которых важен лишь факт ошибки, но не важна величина отклонения  $|a(x) - y(x)|$ .

**Пример 1.3.** На рис. 1.2 приведён пример задачи классификации с двумя классами,  $|\mathbb{Y}| = 2$ . Объектами являются  $L = 10$  точек плоскости, по 5 объектов в каждом классе, алгоритмами — всевозможные разделяющие прямые, то есть в данном случае  $A$  — это семейство *линейных классификаторов*. В матрице ошибок содержится один нулевой столбец (объекты разделяются прямой без ошибок), 5 столбцов с одной ошибкой, 8 столбцов с двумя ошибками, и т. д.

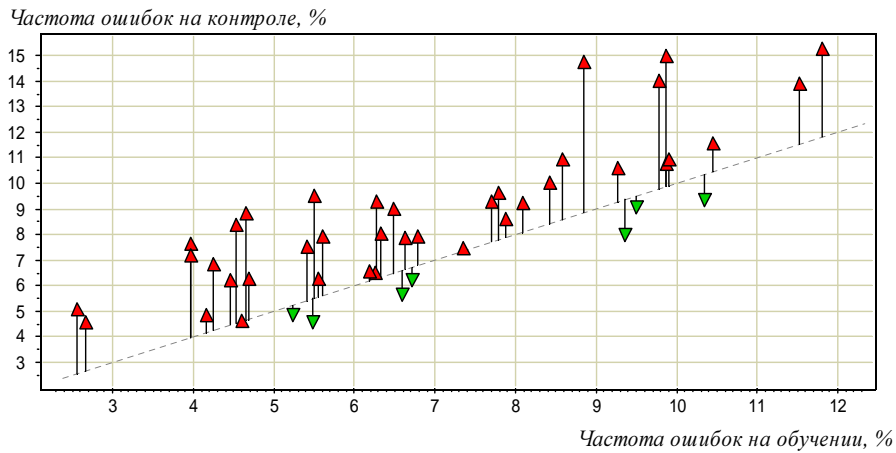
---

<sup>3</sup>Квадратные скобки переводят логическое значение в числовое: [ложь] = 0, [истина] = 1. Это очень практичное обозначение, называемое *нотацией Айверсона* [17].



$x_1$	0	1	0	0	0	0	1	0	0	0	0	1	1	0	...
$x_2$	0	0	1	0	0	0	1	1	0	0	0	0	0	0	...
$x_3$	0	0	0	1	0	0	0	1	1	0	0	0	0	1	...
$x_4$	0	0	0	0	1	0	0	0	1	1	0	0	0	0	...
$x_5$	0	0	0	0	0	1	0	0	0	1	1	1	0	0	...
$x_6$	0	0	0	0	0	0	0	0	0	0	1	0	1	0	...
$x_7$	0	0	0	0	0	0	0	0	0	0	0	0	0	1	...
$x_8$	0	0	0	0	0	0	0	0	0	0	0	0	0	0	...
$x_9$	0	0	0	0	0	0	0	0	0	0	0	0	0	0	...
$x_{10}$	0	0	0	0	0	0	0	0	0	0	0	0	0	0	...

**Рис. 1.2.** Пример матрицы ошибок, порождаемой семейством линейных алгоритмов классификации на выборке длины  $L = 10$ , содержащей 5 объектов одного класса и 5 второго. На трёх графиках сверху показаны все алгоритмы с попарно различными векторами ошибок, с числом ошибок, соответственно, 0, 1, 2.



**Рис. 1.3.** Зависимость  $\nu(\mu X, \bar{X})$  от  $\nu(\mu X, X)$  для задачи прогнозирования отдалённого результата хирургического лечения атеросклероза.

При решении прикладных задач переобучение наблюдается практически всегда. Величина переобученности может оказаться как приемлемо малой, так и неприемлемо большой. Для управления процессом обучения хотелось бы иметь точные количественные оценки переобученности.

**Пример 1.4.** На рис. 1.3 точкам соответствуют алгоритмы, построенные различными методами обучения по одной и той же выборке в задаче прогнозирования отдалённого результата хирургического лечения атеросклероза. Объектами  $x_i$  являются описания больных до проведения операции (данные гемодинамики и иммунологического обследования); ответы  $y_i \in \{0, 1\}$  кодируют результат лечения через год после

операции: 0 — успех, 1 — рестеноз шунта и повторная операция. По горизонтальной оси отложена частота ошибок на обучении, по вертикальной — на контроле. Наблюдается систематическое смещение точек графика вверх; почти все точки лежат выше биссектрисы. Это и есть переобучение.

Основная задача статистического обучения заключается в том, чтобы по наблюдаемой обучающей выборке найти в  $A$  алгоритм, который допускал бы как можно меньше ошибок на скрытой контрольной выборке. Вопрос можно ставить и так: какими свойствами должны обладать множество алгоритмов  $A$  и метод обучения  $\mu$ , чтобы вероятность переобучения была минимальной? Чтобы получить ответ, необходимо сначала уточнить, в каком смысле здесь понимается «вероятность».

## §1.2 Слабая вероятностная аксиоматика

**Основная аксиома.** Будем полагать, что объекты конечного неслучайного множества  $\mathbb{X}$  появляются в случайном порядке, причём все  $L!$  перестановок генеральной выборки  $\mathbb{X}$  равновероятны. Это предположение мы будем называть *слабой (или перестановочной) вероятностной аксиоматикой*. Другие вероятностные предположения нам не понадобятся. Далее мы увидим, что одного этого уже вполне достаточно для получения многих фундаментальных фактов теории вероятностей, математической статистики, теории статистического обучения.

Обозначим через  $S_L$  группу всех перестановок  $L$  элементов. Всевозможные перестановки генеральной выборки будем обозначать через  $\tau\mathbb{X}$ ,  $\tau \in S_L$ .

**Определение 1.1.** Пусть задан предикат  $\psi: \mathbb{X}^L \rightarrow \{0, 1\}$ . Если  $\psi(\tau\mathbb{X}) = 1$ , то будем говорить, что событие  $\psi$  произошло на перестановке  $\tau\mathbb{X}$ . Вероятностью события  $\psi$  называется доля перестановок  $\tau\mathbb{X}$ , на которых оно произошло:

$$P_\tau \psi(\tau\mathbb{X}) = \frac{1}{L!} \sum_{\tau \in S_L} \psi(\tau\mathbb{X}). \quad (1.2)$$

В слабой аксиоматике вероятность события зависит от состава объектов генеральной выборки  $\mathbb{X}$ , но не зависит от порядка их перечисления. Функция распределения и математическое ожидание также зависят от выборки.

**Определение 1.2.** Пусть  $\xi: \mathbb{X}^L \rightarrow \mathbb{R}$  — вещественная функция. Функцией распределения величины  $\xi$  на выборке  $\mathbb{X}$  будем называть функцию  $F_\xi: \mathbb{R} \rightarrow [0, 1]$  вида

$$F_\xi(z) = P_\tau [\xi(\tau\mathbb{X}) \leq z]. \quad (1.3)$$

**Определение 1.3.** Математическим ожиданием величины  $\xi: \mathbb{X}^L \rightarrow \mathbb{R}$  на выборке  $\mathbb{X}$  будем называть её среднее арифметическое по всем перестановкам  $\tau$ :

$$E_\tau \xi(\tau\mathbb{X}) = \frac{1}{L!} \sum_{\tau \in S_L} \xi(\tau\mathbb{X}). \quad (1.4)$$

Заметим, что знаки  $P_\tau$  и  $E_\tau$  можно понимать как операцию среднего арифметического по всем перестановкам:  $P_\tau \equiv E_\tau \equiv \frac{1}{L!} \sum_{\tau \in S_L}$ , то есть в слабой аксиоматике вероятность и математическое ожидание формально определяются одинаково.

**Вероятность как доля разбиений генеральной выборки.** Пусть предикат  $\varphi(X, \bar{X})$  является функцией подвыборки  $X \subset \mathbb{X}$ , её дополнения  $\bar{X} = \mathbb{X} \setminus X$ , и не зависит от порядка элементов в  $X$  и  $\bar{X}$ . Поскольку выборка  $\mathbb{X}$  фиксирована, второй аргумент договоримся опускать,  $\varphi(X) = \varphi(X, \bar{X})$ . Тогда *вероятность события*  $\varphi$  определяется как *доля разбиений* генеральной выборки:

$$P\varphi(X) = \frac{1}{C_L^\ell} \sum_{X \in [\mathbb{X}]^\ell} \varphi(X),$$

где  $[\mathbb{X}]^\ell$  — множество всех  $\ell$ -элементных подмножеств генеральной выборки  $\mathbb{X}$ .

**Эмпирические оценки вероятности.** Непосредственное вычисление величины  $P\varphi(X)$  как доли разбиений возможно только при небольших значениях  $\ell$  или  $k$ . В типичных случаях число разбиений  $C_L^\ell$  огромно. Приближённой оценкой для  $P\varphi(X)$  является среднее по некоторому подмножеству  $N \subset [\mathbb{X}]^\ell$  выборок длины  $\ell$ , не слишком большому, чтобы сумма вычислялась за приемлемое время, но и не слишком маленькому, чтобы приближение было достаточно точным:

$$\hat{P}\varphi(X) = \frac{1}{|N|} \sum_{X \in N} \varphi(X).$$

Далее символами  $\hat{P}$  и  $\hat{E}$  будет обозначаться операция усреднения по некоторому подмножеству разбиений  $N$ . Если подмножество выборок  $N$  выбирается случайно и равновероятно, то говорят об оценке *методом Монте-Карло*.

### §1.3 Вероятность переобучения

*Вероятность переобучения* определяется как доля разбиений выборки, при которых переобученность  $\delta_\mu(X)$  превышает заданный порог  $\varepsilon \in (0, 1)$ :

$$Q_\varepsilon \equiv Q_\varepsilon(\mu, \mathbb{X}) = P[\delta_\mu(X) \geq \varepsilon] = P[\nu(\mu X, \bar{X}) - \nu(\mu X, X) \geq \varepsilon]. \quad (1.5)$$

Вероятность переобучения характеризует обобщающую способность метода  $\mu$  на выборке  $\mathbb{X}$ . Её оценивание будет основной задачей на протяжении всего курса.

Наряду с  $Q_\varepsilon$  можно также оценивать *вероятность большой частоты ошибок* на скрытой контрольной выборке:

$$R_\varepsilon \equiv R_\varepsilon(\mu, \mathbb{X}) = P[\nu(\mu X, \bar{X}) \geq \varepsilon], \quad (1.6)$$

а также среднюю (по всем разбиениям) частоту ошибок на контрольной выборке, называемую *полным скользящим контролем* (complete cross-validation, CCV) [88]:

$$C \equiv C(\mu, \mathbb{X}) = E\nu(\mu X, \bar{X}). \quad (1.7)$$

Заметим, что CCV является математическим ожиданием, а  $(1 - R_\varepsilon)$  — функцией распределения случайной величины  $\nu(\mu X, \bar{X})$ .

**Обращение оценок.** Пусть  $Q_\varepsilon \leq \eta(\varepsilon)$  — верхняя оценка вероятности переобучения и существует функция  $\varepsilon(\eta)$ , обратная к  $\eta(\varepsilon)$ . Тогда  $\eta(\varepsilon(\alpha)) = \alpha$  для любого  $\alpha \in [0, 1]$ . Следовательно,  $\mathbb{P}[\delta_\mu(X) \geq \varepsilon(\eta)] \leq \eta$ , и наша верхняя оценка может быть переформулирована в эквивалентном виде: с вероятностью не менее  $(1 - \eta)$  справедлива верхняя оценка частоты ошибок на скрытой выборке:

$$\nu(\mu X, \bar{X}) \leq \nu(\mu X, X) + \varepsilon(\eta). \quad (1.8)$$

Параметр  $\varepsilon$  называют *точностью*, а  $\eta$  — *надёжностью* оценки [7].

Аналогично, если найдена верхняя оценка  $R_\varepsilon \leq \eta(\varepsilon)$  и  $\varepsilon(\eta)$  — функция, обратная к  $\eta(\varepsilon)$ , то с вероятностью не менее  $(1 - \eta)$  справедлива верхняя оценка

$$\nu(\mu X, \bar{X}) \leq \varepsilon(\eta).$$

Итак, от верхних оценок  $Q_\varepsilon$  или  $R_\varepsilon$  легко переходить к верхним оценкам частоты ошибок на контроле. Обычно из оценки (1.8) выводят новый критерий обучения, отличающийся от минимизации эмпирического риска  $\nu(a, X)$  дополнительным слагаемым  $\varepsilon(\eta)$ . Этот приём является очень важным в SLT, так как он позволяет превращать теоретические оценки в новые оптимизационные методы обучения.

**Скользкий контроль.** В машинном обучении оценку  $\hat{E}\nu(\mu X, \bar{X})$  по подмножеству разбиений используют для эмпирического измерения обобщающей способности и называют *скользящим контролем* (cross-validation, CV). Он незаменим в тех случаях, когда теоретические оценки обобщающей способности не известны или недостаточно точны. Скользящий контроль является де-факто стандартной методикой тестирования и сравнения алгоритмов машинного обучения.

**О природе переобучения.** Неформально, переобучение — это чрезмерно точная подгонка алгоритма  $a$  под конкретную обучающую выборку  $X$  в ущерб его *обобщающей способности*. Ожидается, что метод  $\mu$  обнаружит некие общие закономерности генеральной выборки  $\mathbb{X}$ , но он находит лишь частные закономерности, характерные только для  $X$ . Переобучение носит фундаментальный характер и связано с неполнотой информации в момент применения метода  $\mu$ .

Ещё одну интерпретацию переобучения даёт следующий мысленный эксперимент. Пусть задано конечное множество из  $D$  алгоритмов, которые допускают на генеральной выборке  $\mathbb{X}$  одно и то же число ошибок  $m$ , независимо друг от друга. Число ошибок любого из этих алгоритмов на обучающей выборке  $X$  подчиняется одному и тому же распределению (гипергеометрическому, как будет показано далее). Выбирая алгоритм с минимальным числом ошибок  $s$  на обучающей выборке, мы фактически находим минимум из  $D$  независимых одинаково распределённых случайных величин. Математическое ожидание минимума уменьшается с ростом числа  $D$ . Следовательно, переобученность  $\delta = \frac{m-s}{k} - \frac{s}{\ell} = \frac{m}{k} - s \frac{\ell}{\ell k}$  увеличивается с ростом  $D$ . Эти рассуждения остаются в силе и в общем случае, когда алгоритмы не являются независимыми (имеются схожие алгоритмы) и допускают различное число ошибок (имеется расслоение множества алгоритмов по уровням числа ошибок  $m$ ). Оценивать эффекты *сходства* и *расслоения* довольно трудно, но именно этим мы и займёмся в дальнейшем, так как они существенно понижают вероятность переобучения.

## §1.4 Обсуждение слабой аксиоматики

**Связь с сильной вероятностной аксиоматикой.** Классическая теоретико-мерная аксиоматика А. Н. Колмогорова (будем называть её сильной) основана на понятии вероятностного пространства  $\langle \mathcal{X}, \Omega, \mathbf{P} \rangle$ , где  $\mathcal{X}$  — множество допустимых объектов,  $\Omega$  — аддитивная  $\sigma$ -алгебра событий на  $\mathcal{X}$ ,  $\mathbf{P}$  — вероятностная мера, определённая на событиях из  $\Omega$ . В задачах статистического анализа данных обычно предполагается, что множество объектов  $\mathbb{X} = \{x_1, \dots, x_L\}$  является *простой выборкой*, то есть объекты выбираются из множества  $\mathcal{X}$  случайно и независимо согласно вероятностной мере  $\mathbf{P}$ . Независимость означает, что вероятностная мера на множестве выборок  $\mathcal{X}^L$  инвариантна относительно перестановок элементов выборки. В приложениях множество  $\mathcal{X}$ , как правило, бесконечно, а мера  $\mathbf{P}$  неизвестна.

В слабой аксиоматике множество  $\mathcal{X}$  всех гипотетически возможных объектов не вводится. Рассматривается только конечное множество объектов — *генеральная выборка*  $\mathbb{X}$ . Оно может включать в себя как объекты, наблюдавшиеся ранее, так и скрытые объекты, которые станут известны в будущем. Вероятностным пространством является конечное множество всех перестановок генеральной выборки  $\mathbb{X}$ , на котором задаётся равномерное распределение. Таким образом, случайными полагаются не сами объекты, а лишь порядок их появления, что соответствует предположению о независимости объектов выборки в сильной аксиоматике.

Следующая теорема утверждает, что для перевода оценки из слабой аксиоматики в сильную достаточно взять её математическое ожидание по выборке  $\mathbb{X}$ .

**Теорема 1.1.** Пусть в слабой аксиоматике найдено значение вероятности

$$\mathbf{P}_\tau \psi(\tau\mathbb{X}) = f(\mathbb{X}). \quad (1.9)$$

Тогда в сильной аксиоматике выполняется равенство

$$\mathbf{P}_\mathbb{X} \psi(\mathbb{X}) = \mathbf{E}_\mathbb{X} f(\mathbb{X}). \quad (1.10)$$

**Доказательство.** В силу независимости наблюдений в выборке  $\mathbb{X}$  для произвольной перестановки  $\tau$  справедливо равенство  $\mathbf{P}_\mathbb{X} \psi(\mathbb{X}) = \mathbf{P}_\mathbb{X} \psi(\tau\mathbb{X})$ . Возьмём среднее по всем перестановкам  $\tau$  от левой и правой частей этого равенства:

$$\mathbf{P}_\mathbb{X} \psi(\mathbb{X}) = \frac{1}{L!} \sum_{\tau \in S_L} \mathbf{P}_\mathbb{X} \psi(\tau\mathbb{X}) = \mathbf{P}_\tau \mathbf{E}_\mathbb{X} \psi(\tau\mathbb{X}) = \mathbf{E}_\mathbb{X} \mathbf{P}_\tau \psi(\tau\mathbb{X}) = \mathbf{E}_\mathbb{X} f(\mathbb{X}),$$

что и требовалось доказать. ■

В случаях, когда оценка  $f(\mathbb{X})$  не зависит от выборки  $\mathbb{X}$ , конечный результат — оценка в правой части (1.9) и (1.10) — будет одинаков в обеих аксиоматиках.

**Финитарные и инфинитарные вероятности.** Рассмотрим фундаментальную задачу теории вероятностей, тесно связанную с *законом больших чисел*: оценить вероятность большого отклонения частоты  $\nu(S, X)$  события  $S$  на конечной выборке  $X$  от вероятности  $P(S)$  данного события:

$$P_\varepsilon^S = \mathbf{P}_X \{ |\nu(S, X) - P(S)| > \varepsilon \}. \quad (1.11)$$

В практических задачах анализа данных вероятность события  $P(S)$  невозможно узнать точно, поскольку вероятностная мера на множестве объектов, как правило, неизвестна. Провести бесконечное число наблюдений также невозможно. В результате оказывается, что вероятность большого отклонения  $P_\varepsilon^S$  непосредственно не может быть измерена в эксперименте как частота события  $\{X : |\nu(S, X) - P(S)| > \varepsilon\}$ , поскольку само наступление этого события не может быть точно идентифицировано.

Данная проблема не возникает, если с самого начала отказаться от употребления вероятности  $P(S)$ . Она определяется как предел частоты  $\nu(S, X')$  события  $S$  на произвольной случайной выборке  $X'$  при  $|X'| \rightarrow \infty$ . В то же время, практический интерес представляет именно частота  $\nu(S, X')$ , как величина, непосредственно наблюдаемая в эксперименте. Изменим постановку задачи (1.11) и будем оценивать вероятность большого отклонения частот события  $S$  в двух различных выборках:

$$Q_\varepsilon^S = P_{X, \bar{X}} \{ |\nu(S, X) - \nu(S, X')| > \varepsilon \}. \quad (1.12)$$

Если предполагать, что выборки  $X$  и  $X'$  независимы, то для определения вероятности  $Q_\varepsilon^S$  уже не нужно ни бесконечного числа испытаний, ни знания вероятностной меры  $P$  на бесконечном множестве  $\mathcal{X}$ . Вероятность  $Q_\varepsilon^S$  является *финитарной* и может быть вычислена комбинаторными методами как доля разбиений  $\mathbb{X} = X \sqcup \bar{X}$ , при которых имеет место большое отклонение частот. Кроме того, она может быть измерена в эксперименте по подмножеству разбиений, так как идентификация события  $\{X, X' : |\nu(S, X) - \nu(S, X')| > \varepsilon\}$  не вызывает затруднений.

Таким образом, вероятности  $P(S)$  и  $P_\varepsilon^S$  в (1.11) имеют различную природу. Вероятность  $P(S)$  принципиально *инфинитарна* — для её определения требуется либо знать вероятностную меру  $P$  на бесконечном множестве  $\mathcal{X}$ , либо осуществить предельный переход  $\nu(S, X') \rightarrow P(S)$  при  $|X'| \rightarrow \infty$ , что невозможно сделать при практическом анализе данных. Вероятность  $P_\varepsilon^S$  также инфинитарна, но после замены  $P(S)$  на частоту  $\nu(S, X')$  она принимает финитарный вид  $Q_\varepsilon^S$ , допускающий и точное вычисление, и эмпирическое измерение.

Эти соображения как раз и приводят к слабой вероятностной аксиоматике, запрещающей использование инфинитарных вероятностей и «событий», которые не могут быть идентифицированы в эксперименте. Приведём ещё несколько менее формальных соображений в пользу слабой аксиоматики.

Современная теория вероятностей возникла из стремления объединить в рамках единого формализма частотное понятие вероятности, берущее начало от азартных игр, и континуальное, идущее от геометрических задач типа задачи Бюффона о вероятности попадания иглы в паркетную щель. В аксиоматике Колмогорова континуальное понятие берётся за основу как более общее. Ради этой общности в теорию вероятностей привносятся гипотезы сигма-аддитивности и измеримости — технические предположения из теории меры, имеющие довольно слабые эмпирические обоснования [2]. Таким образом, для изучения дискретных явлений, связанных со случайностью, определение вероятности как континуальной меры изначально избыточно.

Слабая аксиоматика, фактически, не выходит за рамки элементарной теории вероятностей XVII-го века, основанной на комбинаторном подсчёте вариантов.

Обратим внимание на замечание А. Н. Колмогорова в [28, стр. 252]: «представляется важной задача освобождения всюду, где это возможно, от излишних вероятностных допущений. На независимой ценности чисто комбинаторного подхода к теории информации я неоднократно настаивал в своих лекциях». Это высказывание в значительной степени относится и к математической статистике, которая также имеет дело с конечными выборками.

Один из вариантов комбинаторно-алгебраического построения теории информации можно найти в книге В. Д. Гоппы [16]. Он, в частности, пишет: «Надобность в вероятностной модели отпадает, поскольку теория информации оказывается достаточно интересной и богатой приложениями в алгебраической постановке. Одним из таких приложений является распознавание образов».

Ученик А. Н. Колмогорова Ю. К. Беляев в предисловии к книге «Вероятностные методы выборочного контроля» пишет: «возникло глубокое убеждение, что в теории выборочных методов можно получить содержательные аналоги большинства основных утверждений теории вероятностей и математической статистики, которые к настоящему времени найдены в предположении взаимной независимости результатов измерений» [3, стр. 9].

Уместно привести ещё одно высказывание А. Н. Колмогорова: «Чистая математика благополучно развивается как по преимуществу наука о бесконечном. . . Весьма вероятно, что с развитием современной вычислительной техники будет понято, что в очень многих случаях разумно изучение реальных явлений вести, избегая промежуточный этап их стилизации в духе представлений математики бесконечного и непрерывного, переходя прямо к дискретным моделям» [28, стр. 239].

**Асимптотические оценки.** Бесконечно длинные выборки не реализуются на практике просто потому, что конечна память компьютеров и время, отпущенное исследователям на эксперименты. В классической вероятностной аксиоматике данное обстоятельство не принимается во внимание, в частности, когда пишут

$$P(S) = \lim_{|X| \rightarrow \infty} \nu(S, X),$$

где  $P(S)$  — вероятность события  $S$ ,  $\nu(S, X)$  — частота события  $S$  в выборке  $X$ .

В слабой аксиоматике запись  $|X| \rightarrow \infty$  запрещена, и понятие вероятности события  $P(S)$  не определено. Мы не вправе предполагать, что выборка реальных объектов может быть сколь угодно длинной. Тем не менее, было бы нелепо отрицать массу статистических фактов, которые имеют надёжные эмпирические подтверждения, но формулируются и доказываются с явным или неявным использованием таких предельных переходов. Не исключено, что их возможно переформулировать и в финитарном виде. Столь же нелепо отказываться от преимуществ асимптотического анализа. Здесь компромисс заключается в том, чтобы применять асимптотики лишь к *численным оценкам*, как один из способов приближённых вычислений. Например, получив в слабой аксиоматике оценку  $P_\tau \psi(\tau\mathbb{X}) = f(L)$ , зависящую от длины выборки, мы можем исследовать асимптотическое поведение числовой функции  $f(L)$  при  $L \rightarrow \infty$ . При этом существование сколь угодно длинной выборки не предполагается.



## Резюме

Введены основные понятия, которые будут использоваться на протяжении всего курса: генеральная выборка, наблюдаемая обучающая выборка, скрытая контрольная выборка, метод обучения, переобученность, матрица ошибок.

Введена слабая вероятностная аксиоматика, основанная на предположении о том, что все разбиения конечной генеральной выборки имеют равные шансы реализоваться. Можно было бы вообще не вводить термин «вероятность» и говорить только о частотах и «долях разбиений выборки», однако вероятностная терминология и обозначения  $P$ ,  $E$  более привычны. Далее мы убедимся, что многие фундаментальные факты математической статистики могут быть выражены в терминах частот и не нуждаются в теоретико-мерном определении вероятности. Слабая аксиоматика адекватна многим задачам анализа данных, поскольку она имеет дело с конечными выборками и величинами, непосредственно измеримыми в эксперименте.

Поставлена основная задача — получение как можно более точных, эффективно вычисляемых оценок вероятности переобучения (1.5), либо вероятности большой частоты ошибок на контроле (1.6), либо полного скользящего контроля (1.7).

**В следующей лекции** мы дадим точное решение основной задачи для простейшего, но важного частного случая, когда семейство  $A$  состоит из единственного алгоритма. В этом случае по известной частоте ошибок данного алгоритма на наблюдаемой выборке оценивается частота его ошибок на скрытой выборке.

## Упражнения

**Задача 1.1 (1).** В задаче классификации с двумя классами  $\mathbb{Y} = \{-1, 1\}$  выборка задана точками прямой:  $\mathbb{X} = \{-\frac{L}{2}, \dots, -\frac{3}{2}, -\frac{1}{2}, \frac{1}{2}, \frac{3}{2}, \dots, \frac{L}{2}\}$ . Для семейства алгоритмов классификации  $a_\theta(x) = \text{sign}(x - \theta)$  с параметром  $\theta \in \mathbb{R}$  построить матрицу ошибок и график зависимости  $n(a_\theta, \mathbb{X})$  от  $\theta$ , если целевая зависимость имеет вид:

$$\begin{array}{ll} 1) y(x) = \text{sign } x; & 3) y(x) = \begin{cases} \text{sign } x, & |x| > M; \\ \text{sign } \sin \pi x, & |x| \leq M. \end{cases} \\ 2) y(x) = \text{sign } \sin \pi x; & \end{array}$$

Сколько различных векторов ошибок порождает данное семейство алгоритмов? Сколько из них допускают  $m$  ошибок ( $m = 0, \dots, L$ ) на генеральной выборке  $\mathbb{X}$ ?

**Задача 1.2 (2).** В задаче классификации с двумя классами  $\mathbb{Y} = \{-1, 1\}$  выборка задана точками на окружности:  $\mathbb{X} = \{(\sin \varphi_i, \cos \varphi_i) : \varphi_i = (i - \frac{1}{2})\frac{2\pi}{L}, i = 1, \dots, L\}$ . Целевая зависимость  $y(x) = \text{sign } \xi_2$ ,  $x = (\xi_1, \xi_2) \in \mathbb{R}^2$ . Для семейства линейных классификаторов  $a_\theta(x) = \text{sign}(\theta_1 \xi_1 + \theta_2 \xi_2 + \theta_0)$ , с параметром  $\theta = (\theta_1, \theta_2, \theta_3) \in \mathbb{R}^3$  описать матрицу ошибок. Сколько различных векторов ошибок порождает данное семейство? Сколько из них допускают ровно  $m$  ошибок ( $m = 0, \dots, L$ ) на выборке  $\mathbb{X}$ ?

**Задача 1.3 (5).** Каково максимальное число различных векторов ошибок, порождаемых линейными алгоритмами классификации на выборке  $\mathbb{X} = \{x_1, \dots, x_L\} \subset \mathbb{R}^n$ ?

## 2 Оценивание частоты события и гипергеометрическое распределение

Начнём с самого простого частного случая, когда множество алгоритмов состоит из единственного элемента,  $A = \{a\}$ . Тогда вероятность переобучения переходит в вероятность большого отклонения частот в двух выборках. Она тесно связана с законом больших чисел, имеющим фундаментальное значение для теории вероятностей. Поэтому мы забудем ненадолго про алгоритмы и перейдём к более общей терминологии, заодно немного упростив обозначения.

### §2.1 Задача оценивания (предсказания) частоты события

Пусть  $S \subseteq \mathbb{X}$  — некоторое множество объектов; будем называть его «событием». Событие  $S$  и вектор ошибок алгоритма  $a$  взаимно однозначно соответствуют друг другу:  $S = \{x_i: I(a, x_i) = 1\}$ ,  $I(a, x_i) = [x_i \in S]$ .

Обозначим через  $n(U) = |S \cap U|$  число элементов события  $S$  на произвольной конечной выборке  $U \subseteq \mathbb{X}$ , а через  $\nu(U) = n(U)/|U|$  — частоту события  $S$  на  $U$ .

Задача предсказания частоты события состоит в том, чтобы оценить частоту события  $S$  на скрытой выборке  $\bar{X}$  по его частоте на наблюдаемой выборке  $X$  и оценить надёжность предсказания, то есть получить оценку вида

$$\mathbb{P}[|\nu(\bar{X}) - \nu(X)| \geq \varepsilon] \leq \eta(\varepsilon); \quad (2.1)$$

В тех случаях, когда  $S$  интерпретируется как «нежелательное событие», может ставиться задача получения односторонней верхней оценки:

$$\mathbb{P}[\nu(\bar{X}) - \nu(X) \geq \varepsilon] \leq \eta(\varepsilon). \quad (2.2)$$

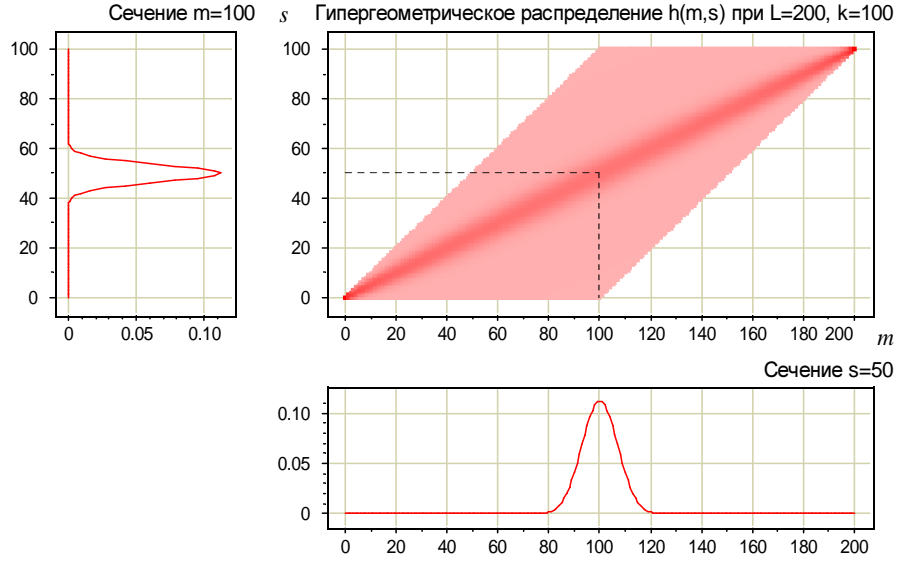
**Лемма 2.1.** Если  $n(\mathbb{X}) = m$ , то число элементов события  $S$  в наблюдаемой подвыборке  $n(X)$  и в скрытой подвыборке  $n(\bar{X})$  подчиняются гипергеометрическому распределению:

$$\mathbb{P}[n(X) = s] = \mathbb{P}[n(\bar{X}) = m - s] = h_L^{\ell, m}(s) = \frac{C_m^s C_{L-m}^{\ell-s}}{C_L^\ell}, \quad (2.3)$$

где  $s$  принимает значения от  $s_0 = \max\{0, m - k\}$  до  $s_1 = \min\{\ell, m\}$ .

**Доказательство.** Отобрать  $s$  элементов события  $S$  в наблюдаемую подвыборку можно  $C_m^s$  различными способами. Для каждого из этих способов имеется  $C_{L-m}^{\ell-s}$  способов сформировать оставшуюся часть наблюдаемой подвыборки из объектов, не принадлежащих  $S$ . Значит,  $C_m^s C_{L-m}^{\ell-s}$  — число разбиений, при которых  $s$  элементов множества  $S$  попадают в наблюдаемую подвыборку, остальные  $(m - s)$  — в скрытую. Их доля в общем числе разбиений  $C_L^\ell$  как раз и составляет  $h_L^{\ell, m}(s)$ . ■

**Замечание 2.1.** Если условие  $0 \leq s \leq m$  не выполняется, то будем полагать, что  $C_m^s = 0$ . Аналогично, если не выполняется условие  $0 \leq \ell - s \leq L - m$ , то  $C_{L-m}^{\ell-s} = 0$ . Отсюда следует, что если не выполняется условие  $s_0 \leq s \leq s_1$ , то  $h_L^{\ell, m}(s) = 0$ .



**Рис. 2.1.** Область определения гипергеометрической функции  $h(m, s) = h_L^{\ell, m}(s)$  при  $L = 200$ ,  $\ell = k = 100$ ,  $m = 30$ .

## §2.2 Гипергеометрическое распределение

*Гипергеометрическое распределение* носит фундаментальный характер и возникает во многих задачах. В данном параграфе перечисляются в справочном порядке основные свойства гипергеометрического распределения [3, 4].

1. При фиксированных  $L$  и  $\ell$  функция  $h_L^{\ell, m}(s) = \frac{C_m^s C_{L-m}^{\ell-s}}{C_L^\ell}$  определена на множестве пар целых чисел  $(m, s)$ :  $0 \leq m \leq L$ ,  $\max\{0, m - k\} = s_0 \leq s \leq s_1 = \min\{\ell, m\}$ . Это множество имеет форму параллелограмма, рис. 2.1. Вне этой области принято полагать  $h_L^{\ell, m}(s) = 0$ .

2. Введём следующие обозначения для сумм крайних левых и крайних правых членов гипергеометрического распределения:

$$H_L^{\ell, m}(z) = \sum_{s=s_0}^{\lfloor z \rfloor} h_L^{\ell, m}(s); \quad \bar{H}_L^{\ell, m}(z) = \sum_{s=\lceil z \rceil}^{s_1} h_L^{\ell, m}(s). \quad (2.4)$$

Справедлива формула полной вероятности:

$$\sum_{s=s_0}^{s_1} h_L^{\ell, m}(s) = H_L^{\ell, m}(s_1) = \bar{H}_L^{\ell, m}(s_0) = 1.$$

При фиксированных  $L$ ,  $\ell$  и  $m$  функция  $h(s) = h_L^{\ell, m}(s)$  является одномерным дискретным распределением. Для примера на рис. 2.1 слева показана функция  $h(s)$  при фиксированном  $m = 100$ . Функция  $h'(m) = h_L^{\ell, m}(s)$  распределением, вообще говоря, не является, так как не удовлетворяет условию нормировки:  $\sum_m h'(m) \neq 1$ . На рис. 2.1 снизу показана функция  $h'(m)$  при фиксированном  $s = 50$ .

3. Параметры  $\ell$  и  $m$  можно переставлять местами:  $h_L^{\ell, m}(s) = h_L^{m, \ell}(s)$ .

4. Параметры  $m$  и  $s$  можно заменять разностями:  $h_L^{\ell, m}(s) = h_L^{\ell, L-m}(\ell - s)$ .

5. Справедливы тождества:

$$h_L^{\ell, m}(s) = h_L^{\ell, L-m}(\ell - s) = h_L^{m, \ell}(s) = h_L^{m, k}(m - s) = h_L^{k, m}(m - s).$$

6. Отсюда вытекают тождества для функций  $H$  и  $\bar{H}$ :

$$H_L^{\ell, m}(s) = \sum_{j=s_0}^s h_L^{\ell, m}(j) = \sum_{j=s_0}^s h_L^{k, m}(m - j) = \bar{H}_L^{k, m}(m - s).$$

7. Распределение  $h(s)$  является унимодальным (имеет форму пика). Максимальное значение достигается при  $s^* = \frac{(m+1)(\ell+1)}{L+2}$ , с точностью до округления.

8. Таблица гипергеометрического распределения содержит  $\ell k$  ненулевых значений. Её можно эффективно вычислить с помощью рекуррентных соотношений:

$$\begin{aligned} h_L^{\ell, 0}(0) &= 1; \\ h_L^{\ell, m+1}(s) &= h_L^{\ell, m}(s) \frac{m+1}{m+1-s} \cdot \frac{k-m+s}{L-m}; \\ h_L^{\ell, m}(s+1) &= h_L^{\ell, m}(s) \frac{m-s}{s+1} \cdot \frac{\ell-s}{k-m+s+1}; \\ h_L^{\ell, m}(s-1) &= h_L^{\ell, m}(s) \frac{s}{m-s+1} \cdot \frac{k-m+s}{\ell-s+1}. \end{aligned} \tag{2.5}$$

Чтобы избежать вычислительных погрешностей, значения  $h_L^{\ell, m}(s)$  рекомендуется вычислять последовательно для всех  $m = 0, \dots, L$ , при этом для каждого  $m$  первым вычислять значение, близкое к максимальному (например, при  $s = s_{\max}$ ), затем меньшие значения вычислять через бóльшие.

9. Математическое ожидание величины  $s$  есть

$$\lambda = \sum_{s=s_0}^{s_1} s h_L^{\ell, m}(s) = \frac{\ell m}{L}.$$

10. Дисперсия величины  $s$  есть

$$\sigma^2 = \sum_{s=s_0}^{s_1} (s - \lambda)^2 h_L^{\ell, m}(s) = \lambda \frac{k(L-m)}{L(L-1)}.$$

11. При больших значениях параметров  $L, \ell, m$  предельными распределениями для  $h(s) = h_L^{\ell, m}(s)$  могут быть только распределения одного из четырёх типов:

- при  $\lambda \rightarrow \infty$  нормальное распределение  $h(s) \rightarrow \frac{1}{\sqrt{2\pi}} \exp(-\frac{(s-\lambda)^2}{2\sigma^2})$ ;
- при  $\frac{m}{L} \rightarrow p$  биномиальное распределение  $h(s) \rightarrow C_\ell^s p^s (1-p)^{\ell-s}$ ;
- при  $\frac{\ell}{L} \rightarrow p$  биномиальное распределение  $h(s) \rightarrow C_m^s p^s (1-p)^{m-s}$ ;
- при  $\frac{m}{L} \rightarrow 0$  или  $\frac{\ell}{L} \rightarrow 0$  распределение Пуассона  $h(s) \rightarrow e^{-\lambda} \lambda^s / s!$ ;
- при  $\lambda \rightarrow 0$  вырожденное распределение  $h(s) \rightarrow [s = 0]$ .

12. Гипергеометрическое распределение довольно точно приближается с помощью аппроксимации Моленара:

$$h(s) \approx C_\ell^s \tilde{p}^s (1 - \tilde{p})^{\ell-s}, \quad \tilde{p} = \frac{2m - s}{2L - \ell + 1}.$$

### §2.3 Закон больших чисел в слабой аксиоматике

Продолжим рассмотрение задач (2.1), (2.2) о предсказании частоты события.

**Теорема 2.2.** Пусть  $n(\mathbb{X}) = m$ . Для любого  $\varepsilon \in [0, 1)$  справедливы точные оценки:

$$\mathbb{P}[\nu(X) \leq \varepsilon] = H_L^{\ell, m}(\varepsilon \ell); \quad (2.6)$$

$$\mathbb{P}[\nu(\bar{X}) \geq \varepsilon] = H_L^{\ell, m}(m - \varepsilon k); \quad (2.7)$$

$$\mathbb{P}[\nu(\bar{X}) - \nu(X) \geq \varepsilon] = H_L^{\ell, m}(s_m(\varepsilon)), \quad s_m(\varepsilon) = \frac{\ell}{L}(m - \varepsilon k); \quad (2.8)$$

$$\mathbb{P}[|\nu(\bar{X}) - \nu(X)| \geq \varepsilon] = H_L^{\ell, m}(s_m(\varepsilon)) + \bar{H}_L^{\ell, m}(\bar{s}_m(\varepsilon)), \quad \bar{s}_m(\varepsilon) = \frac{\ell}{L}(m + \varepsilon k). \quad (2.9)$$

**Доказательство.** Первые две оценки являются непосредственным следствием (2.3):

$$\mathbb{P}[\nu(X) \leq \varepsilon] = \sum_{s=0}^{\ell} \left[ \frac{s}{\ell} \leq \varepsilon \right] \mathbb{P}[n(X) = s] = \sum_{s=s_0}^{[\varepsilon \ell]} h_L^{\ell, m}(s) = H_L^{\ell, m}(\varepsilon \ell);$$

$$\mathbb{P}[\nu(\bar{X}) \geq \varepsilon] = \sum_{s=0}^{\ell} \left[ \frac{m-s}{k} \geq \varepsilon \right] \mathbb{P}[n(\bar{X}) = m - s] = \sum_{s=s_0}^{[m-\varepsilon k]} h_L^{\ell, m}(s) = H_L^{\ell, m}(m - \varepsilon k).$$

Третья оценка получается аналогично первой, с той лишь разницей, что условие  $\nu(X) \frac{s}{\ell} \leq \varepsilon$  заменяется условием  $\nu(\bar{X}) - \nu(X) = \frac{m-s}{k} - \frac{s}{\ell} \geq \varepsilon$ . Отсюда элементарными преобразованиями получаем верхний предел суммирования  $s \leq \frac{\ell}{L}(m - \varepsilon k)$ .

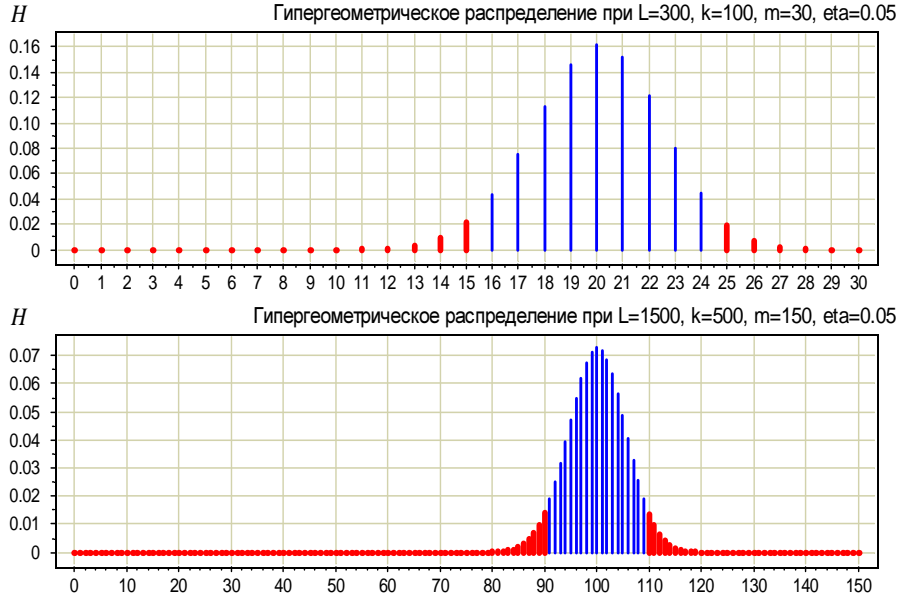
Наконец, двусторонняя оценка (2.9) получается, если представить множество разбиений в виде объединения двух непересекающихся подмножеств:

$$\begin{aligned} \mathbb{P}[|\nu(\bar{X}) - \nu(X)| \geq \varepsilon] &= \mathbb{P}[\nu(\bar{X}) - \nu(X) \geq \varepsilon] + \mathbb{P}[\nu(X) - \nu(\bar{X}) \geq \varepsilon] = \\ &= H_L^{\ell, m}(s_m(\varepsilon)) + \bar{H}_L^{\ell, m}(\bar{s}_m(\varepsilon)). \quad \blacksquare \end{aligned}$$

**О скорости сходимости в законе больших чисел.** При пропорциональном увеличении  $L$ ,  $\ell$  и  $m$  относительная ширина гипергеометрического пика уменьшается, рис. 2.2. В пределе при  $L, \ell, m \rightarrow \infty$  возможно сколь угодно точное предсказание скрытой частоты  $\nu(\bar{X})$  по наблюдаемой частоте  $\nu(X)$ . Равенство (2.9) оценивает скорость сходимости частот события в двух выборках.

Классический закон больших чисел утверждает сходимость частоты события к её вероятности. Однако в слабой аксиоматике понятие «вероятности события  $S$ » не определено. Поэтому (2.9) можно интерпретировать как *аналог закона больших чисел* в слабой аксиоматике. Основанием для такой интерпретации также служит тот факт, что два функционала — (а) вероятность большого отклонения частот события в двух выборках и (б) вероятность большого отклонения частоты события от его вероятности — оцениваются сверху друг через друга, как показано в [7]. По сути, эти две оценки отличаются не принципиально.

Классические неравенства Чебышёва, Чернова, Бернштейна, Хёфффдинга [83] оценивают скорость сходимости в законе больших чисел. Все они являются асимптотическими и дают завышенные оценки вероятности большого отклонения. Выражение (2.9) является точной (не завышенной, не асимптотической) оценкой, и потому его можно считать наиболее точным выражением закона больших чисел.

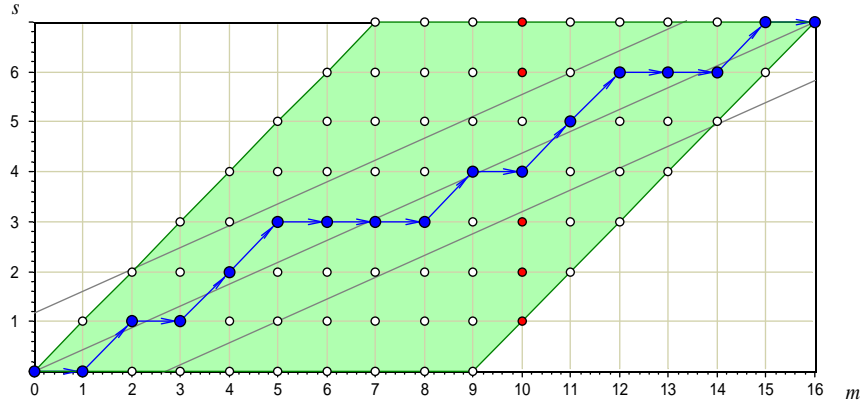


**Рис. 2.2.** Гипергеометрическая функция  $h_L^{\ell, m}(s)$ . Верхний график получен при  $L = 300$ ,  $\ell = 200$ ,  $m = 30$ . Выделены крайние левые,  $[s_0, s_m(\varepsilon)] = [0, 15]$ , и крайние правые  $[s_m(\varepsilon), s_1] = [25, 30]$ , члены распределения, соответствующие значению надёжности  $\eta = 0.05$ . Нижний график получен при значениях  $L$ ,  $\ell$ ,  $m$ , пропорционально увеличенных в 5 раз.

**Геометрическая интерпретация** соотношений (2.3), (2.8) и (2.9). Рассмотрим прямоугольную сетку  $\{0, \dots, L\} \times \{0, \dots, \ell\}$  на рис. 2.3. Положим  $b_i = [x_i \in X]$ . Тогда  $b_i = 1$  означает, что объект  $x_i$  попадает в наблюдаемую подвыборку. Договоримся отображать разбиение  $X, \bar{X}$  в виде траектории, проходящей по узлам сетки из точки  $(0, 0)$  в точку  $(L, \ell)$  согласно правилу: если  $b_i = 1$ , то смещаемся на единицу вправо-вверх; если  $b_i = 0$ , то смещаемся на единицу вправо. Все такие траектории не выходят за пределы параллелограмма, выделенного на рис. 2.3. Множество всех таких траекторий изоморфно множеству разбиений выборки  $\mathbb{X} = X \sqcup \bar{X}$ , и оба они изоморфны множеству  $L$ -мерных бинарных векторов  $(b_1, \dots, b_L)$ , содержащих ровно  $\ell$  единиц. Поэтому для подсчёта числа разбиений, удовлетворяющих некоторому свойству, достаточно найти число соответствующих траекторий.

Чтобы вывести (2.3), пронумеруем объекты выборки так, чтобы первые  $m$  объектов принадлежали множеству  $S$ . Тогда задача сведётся к подсчёту доли траекторий, проходящих через точку  $(m, s)$ . Назовём такие траектории *допустимыми*. Число всех возможных траекторий на отрезке от  $(0, 0)$  до  $(m, s)$  равно  $C_m^s$ , и для каждой траектории существует  $C_{L-m}^{\ell-s}$  вариантов её продолжения от  $(m, s)$  до  $(L, \ell)$ . Следовательно, число допустимых траекторий равно  $C_m^s C_{L-m}^{\ell-s}$ . Разделив на общее число возможных траекторий  $C_L^\ell$ , получаем требуемое  $h_L^{\ell, m}(s)$ .

Чтобы вывести (2.8), необходимо подсчитать число траекторий, проходящих через любую точку  $(m, s)$ , лежащую ниже диагонали на  $\varepsilon \frac{\ell k}{L}$  или более. Для этого суммируется число траекторий  $C_m^s C_{L-m}^{\ell-s}$  по всем  $s = s_0, \dots, s_m(\varepsilon)$ .



**Рис. 2.3.** Траектория последовательности  $\{b_i\}_{i=0}^L = 0101100010110010$ , при  $L = 16$ ,  $\ell = 7$ ,  $\varepsilon = 0.3$ . Проведены линии  $s = \frac{\ell}{L}m$  и  $s = \frac{\ell}{L}(m \pm \varepsilon k)$ . При  $m = 10$  выделены точки выше верхней линии,  $s \geq \bar{s}_m(\varepsilon)$ , и ниже нижней линии,  $s \leq s_m(\varepsilon)$ .

Для вывода двусторонней оценки (2.9) подсчитывается число траекторий, отстоящих от диагонали на  $\varepsilon \frac{\ell k}{L}$  или более. В этом случае суммирование идёт по всем  $s = s_0, \dots, s_m(\varepsilon)$ , затем по всем  $s = \bar{s}_m(\varepsilon), \dots, s_1$ .

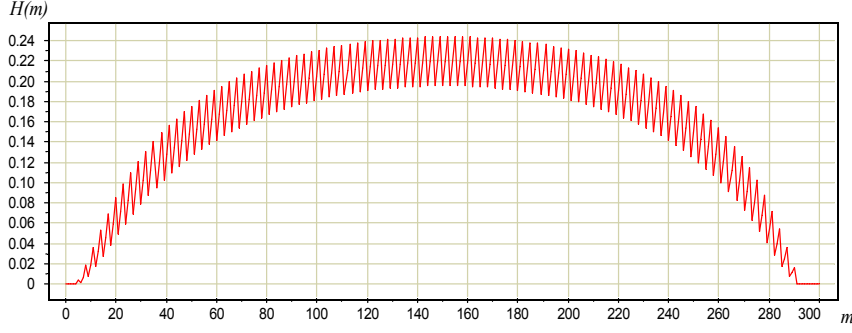
**Выборочный контроль качества** — это пример прикладной задачи, в которой оценки Теоремы 2.2 применимы непосредственно [3].

Пусть  $\mathbb{X}$  — множество изделий,  $S \subset \mathbb{X}$  — подмножество дефектных изделий. Изготовлена партия изделий  $\mathbb{X}$ , из них  $m$  оказались дефектными. Число  $m$  неизвестно. Проверить всю партию поштучно не представляется возможным. Поэтому делается *выборочный контроль качества*: случайно, независимо, без возвратов выбирается подмножество  $X \subset \mathbb{X}$ , что равносильно случайному равномерному выбору разбиения  $X \sqcup \bar{X} = \mathbb{X}$ . Зная долю дефектов в наблюдаемой выборке  $\nu(X)$ , требуется предсказать долю дефектов в скрытой выборке  $\nu(\bar{X})$ . Если при заданной точности  $\varepsilon$  и надёжности  $\eta$  имеет место оценка  $P[\nu(\bar{X}) \geq \varepsilon] < \eta$ , то партия  $\mathbb{X}$  принимается, иначе она целиком бракуется. Параметры  $\varepsilon$  и  $\eta$  подбираются из экономических соображений — с учётом стоимости контроля одного изделия и величины потерь от использования дефектного изделия.

## §2.4 Переход от ненаблюдаемой оценки к наблюдаемой

Оценки (2.6)–(2.9) зависят от числа элементов  $m$  события  $S$  в генеральной выборке  $\mathbb{X}$ , которое невозможно узнать, пока неизвестна скрытая подвыборка. В таких случаях говорят, что оценка является *ненаблюдаемой* (unobservable bound). Ситуация на первый взгляд парадоксальна. Чтобы оценить вероятность большого отклонения частот в наблюдаемой и скрытой выборке, необходимо знать число  $m$ . Однако если бы мы его знали, то по наблюдаемой частоте  $\nu(X)$  тут же вычислили бы точное значение скрытой частоты  $\nu(\bar{X})$ , поскольку  $k\nu(\bar{X}) + \ell\nu(X) = m$ .

**Верхняя оценка.** Простейшее решение проблемы неизвестного  $m$  заключается в том, чтобы взять максимум по  $m$  и получить вместо точной оценки завышенную



**Рис. 2.4.** Зависимость  $H(m) = H_L^{\ell, m}(s_m(\varepsilon))$  от  $m$  при  $L = 300$ ,  $\ell = 200$ ,  $\varepsilon = 0.05$ .

верхнюю оценку:

$$\mathbb{P}[\nu(\bar{X}) - \nu(X) \geq \varepsilon] \leq \max_{m=0, \dots, L} H_L^{\ell, m}(s_m(\varepsilon)) \equiv \Gamma_L^{\ell}(\varepsilon). \quad (2.10)$$

Здесь максимум достаточно взять по всем  $m$  от  $\lceil \varepsilon k \rceil$  до  $\lfloor L - \varepsilon \ell \rfloor$ , так как при остальных значениях  $m$  левая часть неравенства равна нулю.

К сожалению, (2.10) — довольно грубая оценка при малых  $m$ , см. рис. 2.4. По этой причине данный подход не приемлем для задач выборочного контроля качества, обучения по прецедентам, и других случаев, когда именно малые значения  $m$  представляют большой практический интерес.

Известна верхняя оценка «хвоста» гипергеометрического распределения [62, 95], с помощью которой можно оценить сверху правую часть (2.10): для любого  $\varepsilon > 0$

$$\Gamma_L^{\ell}(\varepsilon) \leq \exp\left(-2\varepsilon^2 \frac{\ell k^2}{L^2}\right).$$

Эта оценка ещё более грубая. На рис. 2.4 ей соответствует горизонтальная линия с ординатой 0.89, но она не показана, поскольку проходит много выше. Асимптотически  $\Gamma_L^{\ell}(\varepsilon) \rightarrow 0$  при одновременном стремлении  $\ell$  и  $k$  к бесконечности, что ещё раз подтверждает связь точных оценок (2.8) и (2.9) с *законом больших чисел*.

**Точная интервальная оценка.** Следующая теорема показывает, как получать точные верхние и нижние оценки для  $n(\mathbb{X})$  и  $n(\bar{X})$  по наблюдаемому значению  $s = n(X)$ .

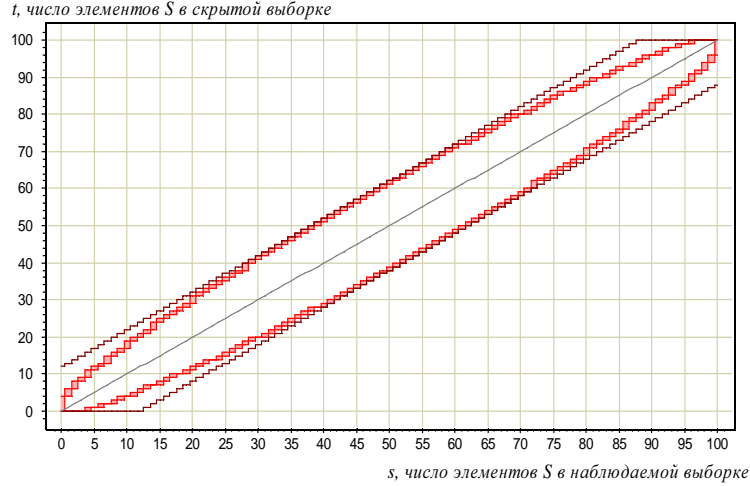
**Теорема 2.3.** Если  $s = n(X)$  — число элементов события  $S$  в наблюдаемой выборке, то для числа элементов события  $S$  в полной выборке с вероятностью  $(1 - \eta)$  справедлива верхняя оценка:

$$n(\mathbb{X}) \leq \max\{m = m_0, \dots, L \mid H_L^{\ell, m}(s) \geq \eta\}, \quad \text{где } m_0 = \lceil s \frac{L+2}{\ell+1} - 1 \rceil. \quad (2.11)$$

**Доказательство.** Рассмотрим одностороннюю точную оценку (2.7), обозначив правую её часть через  $H(\varepsilon, m)$ , где  $m = n(\mathbb{X})$  — неизвестная величина:

$$\mathbb{P}[\nu(\bar{X}) \geq \varepsilon] = H_L^{\ell, m}(m - \varepsilon k) = H(\varepsilon, m). \quad (2.12)$$





**Рис. 2.5.** Точные верхние и нижние оценки числа  $t = n(\bar{X})$  элементов события  $S$  в скрытой выборке в зависимости от их числа  $s = n(X)$  в наблюдаемой выборке. Условия эксперимента:  $L = 200$ ,  $\ell = k = 100$ ,  $\eta = 0.05$ .

Тогда с вероятностью  $(1 - \eta)$  справедлива оценка сверху  $\nu(\bar{X}) < E(\eta, m)$ , где  $E(\eta, m)$  — обратная функция от  $H(\varepsilon, m)$ . Обращение производится по первому аргументу при каждом значении второго аргумента  $m$ , который выступает в роли параметра. Поскольку функция  $E(\eta, m)$  не возрастает по первому аргументу, из оценки  $\nu(\bar{X}) < E(\eta, m)$  следует неравенство  $H(\nu(\bar{X}), m) \geq \eta$ . Подставляя  $\nu(\bar{X}) = \frac{m-s}{k}$  в функцию  $H(\nu(\bar{X}), m)$ , определяемую согласно (2.12), получаем, что с вероятностью  $(1 - \eta)$  справедливо неравенство  $H_L^{\ell, m}(s) \geq \eta$ . Чтобы разрешить данное неравенство относительно  $m$  при фиксированном  $s$ , найдём максимальное значение  $m$ , при котором оно выполнено. При максимальном значении  $m$  значение  $s$  должно находиться левее точки максимума гипергеометрического распределения  $s^* = \frac{(m+1)(\ell+1)}{L+2}$ , Следовательно,  $s(L+2) < (m+1)(\ell+1)$ . Поэтому для нахождения максимального значения  $m$  достаточно перебрать значения  $m$ , не меньшие  $s \frac{L+2}{\ell+1} - 1$ . ■

**Следствие 2.3.1.** Аналогично оценивается скрытое число  $n(\bar{X})$  по наблюдаемому числу  $s = n(X)$ : с вероятностью  $(1 - \eta)$  выполнено неравенство

$$n(\bar{X}) \leq \max\{t = t_0, \dots, k \mid H_L^{\ell, s+t}(s) \geq \eta\}, \quad t_0 = \lceil s \frac{k+1}{\ell+1} - 1 \rceil. \quad (2.13)$$

**Следствие 2.3.2.** Аналогично строятся и нижние оценки: с вероятностью  $(1 - \eta)$

$$\begin{aligned} n(\mathbb{X}) &\geq \min\{m = 0, \dots, m_0 \mid \bar{H}_L^{\ell, m}(s) \geq \eta\}; \\ n(\bar{X}) &\geq \min\{t = 0, \dots, t_0 \mid \bar{H}_L^{\ell, s+t}(s) \geq \eta\}, \end{aligned}$$

Вычисление полученных верхних и нижних оценок с использованием рекуррентных соотношений (2.5) требует порядка  $O(n(X)n(\bar{X}))$  операций.

На рис. 2.5 показаны верхние и нижние оценки числа элементов события  $S$  в скрытой выборке  $t = n(\bar{X})$  в зависимости от их числа в наблюдаемой выборке  $s = n(X)$ . Толстые ступенчатые линии — граничные области, в которых выполняется равенство  $H_L^{\ell, s+t}(s) = \eta$ . Точная верхняя оценка совпадает с верхней границей

верхней области, точная нижняя — с нижней границей нижней области. Вместе они определяют  $1 - 2\eta = 90\%$ -й доверительный интервал для числа  $t$  при каждом значении  $s$ . Тонкие ступенчатые линии — это оценки по наихудшему  $m$ , вычисленные согласно (2.10). Их точность падает по мере приближения  $m$  к 0 или к  $L$ .

**О вероятности нуль-события.** Пользуясь Теоремой 2.3, нетрудно посчитать верхнюю доверительную оценку  $n(\mathbb{X})$  для *нуль-события* [18] — такого события, которое вообще не наблюдалось,  $n(X) = 0$ . Данная задача имеет точное решение (2.13), в которое надо подставить  $s = 0$ . В частности, по графику на рис. 2.5 легко определить, что при  $s = 0$  и длине наблюдаемой выборки  $\ell = 100$  число событий в скрытой выборке длины  $k = 100$  не превзойдёт 4 с вероятностью  $1 - \eta = 95\%$ . Нижняя доверительная оценка  $n(\mathbb{X})$  для нуль-события, разумеется, равна нулю.

## §2.5 Одноэлементное семейство алгоритмов

Вернёмся к задаче оценивания вероятности переобучения и рассмотрим одноэлементное семейство алгоритмов  $A = \{a\}$ . В этом случае никакого обучения быть не может:  $\mu X = a$  для любой выборки  $X$ . Функционал  $Q_\varepsilon(\mu, \mathbb{X}) \equiv Q_\varepsilon(a, \mathbb{X})$  будем называть *вероятностью переобучения отдельного алгоритма*.

**Теорема 2.4.** Пусть алгоритм  $a$  допускает  $m$  ошибок на генеральной выборке:  $n(a, \mathbb{X}) = m$ . Тогда для любого  $\varepsilon \in [0, 1]$  справедливы точные оценки:

$$Q_\varepsilon = \mathbb{P}[\nu(a, \bar{X}) - \nu(a, X) \geq \varepsilon] = H_L^{\ell, m} \left( \frac{\ell}{L} (m - \varepsilon k) \right); \quad (2.14)$$

$$R_\varepsilon = \mathbb{P}[\nu(a, \bar{X}) \geq \varepsilon] = H_L^{\ell, m} (m - \varepsilon k); \quad (2.15)$$

$$C = \mathbb{E}\nu(a, \bar{X}) = \frac{m}{L}. \quad (2.16)$$

**Доказательство.** Равенства (2.14) и (2.15) вытекают непосредственно из Теоремы 2.2, если ввести событие  $S = \{x_i \in \mathbb{X} : I(a, x_i) = 1\}$ . Для доказательства (2.16) запишем определения  $\mathbb{E}$  и  $\nu$ , затем переставим знаки суммирования:

$$C = \frac{1}{C_L^\ell} \sum_{X \in [\mathbb{X}]^\ell} \frac{1}{k} \sum_{x_i \in \bar{X}} I(a, x_i) = \frac{1}{k C_L^\ell} \underbrace{\sum_{i=1}^L I(a, x_i)}_m \underbrace{\sum_{X \in [\mathbb{X}]^\ell} [x_i \in \bar{X}]}_{C_{L-1}^\ell} = \frac{m C_{L-1}^\ell}{k C_L^\ell} = \frac{m}{L}. \quad \blacksquare$$

## Резюме

Задача оценивания частоты события (например, ошибки фиксированного алгоритма  $a$ ) на скрытой выборке по его частоте на наблюдаемой выборке имеет точное решение, которое выражается через гипергеометрическое распределение.

В слабой вероятностной аксиоматике эта оценка описывает скорость сходимости частот в двух выборках и является аналогом закона больших чисел.

Неудобство этой оценки заключается в том, что она зависит от неизвестной частоты события на генеральной выборке  $m$ . Проблема решается либо взятием максимума по  $m$ , и тогда получается завышенная оценка, либо вычислением точной интервальной оценки, но тогда она не выражается в виде явной формулы.

**В следующей лекции** мы покажем, что в рамках слабой аксиоматики возможно получать точные решения для ряда классических задач математической статистики, таких, как доверительное оценивание и проверка статистических гипотез.

## Упражнения

**Задача 2.1 (1).** Построить графики зависимости величины  $Q_\varepsilon$ , определяемой по формуле (2.14), от  $\ell$  при фиксированном  $k$ , от  $k$  при фиксированном  $\ell$  и от  $\ell = k$ .

**Задача 2.2 (2).** Вывести точную оценку вероятности переобучения  $Q_\varepsilon$  для двух-элементного семейства  $A = \{a_1, a_2\}$ , если заданы четыре параметра

$$m_{uv} = \#\{x \in \mathbb{X}: I(a_1, x) = u, I(a_2, x) = v\}, \quad u, v \in \{0, 1\}.$$

Построить графики зависимости  $Q_\varepsilon$  от хэммингова расстояния между алгоритмами  $\rho_{12} = m_{01} + m_{10}$ , в двух случаях: 1) при  $m_{01} = m_{10}$ ; 2) при  $m_{01} = 0$ .

**Задача 2.3 (5\*).** Описать способ вычисления точной верхней оценки вероятности переобучения  $Q_\varepsilon$  для семейства из двух алгоритмов  $A = \{a_1, a_2\}$ , если при некотором разбиении  $(X, \bar{X})$ , выбранном случайно и равновероятно, известны четыре величины:

$$s_{uv} = \#\{x \in X: I(a_1, x) = u, I(a_2, x) = v\}, \quad u, v \in \{0, 1\}.$$

Построить графики зависимости  $Q_\varepsilon$  от хэммингова расстояния между алгоритмами  $\rho_{12} = s_{01} + s_{10}$ , в двух случаях: 1) при  $s_{01} = s_{10}$ ; 2) при  $s_{01} = 0$ .

Сравнить с решением предыдущей задачи.

**Задача 2.4 (3).** Вывести точную оценку вероятности переобучения  $Q_\varepsilon$  для семейства из трёх алгоритмов  $A = \{a_1, a_2, a_3\}$ , если заданы восемь параметров

$$m_{rst} = \#\{x \in \mathbb{X}: I(a_1, x) = r, I(a_2, x) = s, I(a_3, x) = t\}, \quad r, s, t \in \{0, 1\}.$$

Построить графики зависимости  $Q_\varepsilon$  от хэммингова расстояния между алгоритмами  $\rho_{12} = m_{010} + m_{100} + m_{011} + m_{101}$ , в двух случаях:

- 1) при  $m_{110} = m_{101} = m_{011} = m_{100} = m_{010} = m_{001}$ ;
- 2) при  $m_{110} = m_{101} = m_{100} = m_{010} = 0$ ,  $m_{011} = m_{001}$ .

### 3 Непараметрические критерии и доверительные оценки

Материал данной лекции не имеет прямого отношения к основной теме курса — проблеме переобучения. Он демонстрирует возможности слабой вероятностной аксиоматики на примере стандартных задач математической статистики.

Мы рассмотрим несколько задач двух типов.

Первый тип — задачи доверительного оценивания. *Доверительный интервал* для случайной величины — это отрезок, концы которого вычисляются по наблюдаемой выборке, накрывающий неизвестное истинное значение данной случайной величины с заданной *доверительной вероятностью*, близкой к единице, скажем, 0.95.

Второй тип — задачи проверки статистических гипотез. Выдвигается некоторая *гипотеза*  $H_0$ , при условии истинности которой заданная функция выборки (*статистика*) имеет известную функцию распределения. Выделяется *критическая область* наименее вероятных значений этой статистики, и фиксируется *уровень значимости* — вероятность попасть в эту область при условии истинности гипотезы  $H_0$ . Уровень значимости выбирается достаточно близким к нулю, скажем, 0.05. Если наблюдаемое значение статистики попадает в критическую область, то гипотеза  $H_0$  отвергается; в противном случае говорят, что данные не противоречат гипотезе  $H_0$ .

В слабой аксиоматике находят естественное выражение в первую очередь *непараметрические* методы статистики, не связанные с априорными предположениями о параметрическом виде плотностей распределения каких-либо случайных величин.

#### §3.1 Доверительное оценивание

Рассмотрим задачу о доверительном оценивании случайной величины.

Задана функция  $\xi: \mathbb{X} \rightarrow \mathbb{R}$ , значения которой попарно различны на элементах выборки  $\mathbb{X}$ . Требуется построить по наблюдаемой выборке  $X$  семейство вложенных *доверительных интервалов*  $\Omega_\varepsilon(X) = [\xi_\varepsilon^-(X), \xi_\varepsilon^+(X)]$  такое, что для произвольного скрытого объекта  $\bar{x}$  выполняется  $\xi(\bar{x}) \in \Omega_\varepsilon(X)$  с вероятностью не менее  $1 - \eta(\varepsilon)$ . Таким образом, длина скрытой выборки  $\bar{X} = \{\bar{x}\}$  полагается равной единице.

*Вариационным рядом* функции  $\xi$  на выборке  $U = \{u_1, \dots, u_t\} \subseteq \mathbb{X}$  называется последовательность значений  $\xi(u_1), \dots, \xi(u_t)$ , упорядоченная по возрастанию. Обозначим  $s$ -е значение вариационного ряда  $\xi$  на  $U$  через  $\xi_U^{(s)}$ , тогда  $\xi_U^{(1)} < \dots < \xi_U^{(t)}$ .

**Теорема 3.1.** *Определим семейство вложенных отрезков  $\Omega_\varepsilon(X) = [\xi_X^{(\ell-\varepsilon+1)}, \xi_X^{(\varepsilon)}]$ , где  $\varepsilon = \lceil \ell/2 \rceil, \dots, \ell$ . Тогда справедлива точная оценка:*

$$\mathbb{P}[\xi(\bar{x}) \notin \Omega_\varepsilon(X)] = 2\left(1 - \frac{\varepsilon}{L}\right), \quad \varepsilon = \lceil \ell/2 \rceil, \dots, \ell. \quad (3.1)$$

**Доказательство.** Всего имеется  $C_L^1 = L$  разбиений. Величина  $\xi(\bar{x})$  превосходит  $\xi_X^{(\varepsilon)}$  на тех разбиениях, при которых правее  $\xi(\bar{x})$  в вариационном ряду находится менее  $L - \varepsilon$  объектов. Таких разбиений ровно  $L - \varepsilon$ . Аналогично,  $\xi(\bar{x}) < \xi_X^{(L-\varepsilon+1)}$  на тех разбиениях, при которых левее  $\xi(\bar{x})$  находится менее  $L - \varepsilon$  объектов. Таких разбиений

также ровно  $L - \varepsilon$ . Итак, доля разбиений, при которых значение  $\xi(\bar{x})$  попадает вне отрезка  $\Omega_\varepsilon(X)$ , составляет  $2(L - \varepsilon)/L$ . ■

Аналогично можно доказать и точную верхнюю оценку:

$$\mathbb{P}[\xi(\bar{x}) > \xi_X^{(\varepsilon)}] = 1 - \frac{\varepsilon}{L}, \quad \varepsilon = \lceil \ell/2 \rceil, \dots, \ell. \quad (3.2)$$

Полагая в (3.1)  $\varepsilon = \ell$ , заключаем, что скрытое значение  $\xi(\bar{x})$  выходит за пределы диапазона наблюдаемых значений  $[\xi_X^{(1)}, \xi_X^{(\ell)}]$  с вероятностью  $\frac{2}{L}$ . Для предсказания  $\xi(\bar{x})$  с надёжностью  $\eta$  достаточно иметь  $\frac{1}{\eta} - 1$  объектов в случае односторонней оценки, и примерно вдвое больше,  $\frac{2}{\eta} - 1$ , для двусторонней. В частности, 19 объектов достаточно для получения верхней оценки с надёжностью 0.95.

### §3.2 Доверительные интервалы для квантилей

**Доверительные интервалы для квантилей в сильной аксиоматике** [21]. Пусть  $\xi$  — случайная величина с непрерывной строго возрастающей функцией распределения  $F(x)$ . Решение уравнения  $F(x) = \alpha$  при  $\alpha \in (0, 1)$  существует и единственно, обозначается  $\xi_\alpha$  и называется  $\alpha$ -квантилью распределения или квантилью порядка  $\alpha$ . В частности,  $\xi_{\frac{1}{2}}$  есть медиана распределения.

Пусть  $X = \{\xi_1, \dots, \xi_\ell\}$  — выборка  $\ell$  независимых случайных величин из распределения  $F$ . Построим её вариационный ряд  $\xi_X^{(1)} < \dots < \xi_X^{(\ell)}$ . С вероятностью 1 он не имеет связок. Отрезок  $[\xi_X^{(r)}, \xi_X^{(s)}]$  является доверительным интервалом для  $\alpha$ -квантили с доверительной вероятностью

$$\mathbb{P}[\xi_X^{(r)} \leq \xi_\alpha \leq \xi_X^{(s)}] = \sum_{t=r}^s C_\ell^t \alpha^t (1 - \alpha)^{\ell-t}. \quad (3.3)$$

В частности, отрезок  $[\xi_X^{(r+1)}, \xi_X^{(\ell-r)}]$  является доверительным интервалом для медианы ( $\alpha = \frac{1}{2}$ ) с доверительной вероятностью  $1 - 2 \sum_{t=0}^r C_\ell^t 2^{-\ell}$ .

**Доверительные интервалы для квантилей в слабой аксиоматике.** Постановку задачи придётся менять, поскольку понятия  $\alpha$ -квантили и функции распределения  $F(x)$  определяются через теоретико-мерную вероятность. Адекватной заменой квантили  $\xi_\alpha$  в слабой аксиоматике является  $m$ -й член вариационного ряда генеральной выборки  $\mathbb{X}$  при  $m = \alpha L$ .

Допустим, что числовая генеральная выборка  $\mathbb{X} = \{\xi_1, \dots, \xi_L\}$ , состоит из парно различных значений. Построим её вариационный ряд  $\xi_{\mathbb{X}}^{(1)} < \dots < \xi_{\mathbb{X}}^{(L)}$ .

Следующая теорема утверждает, что, имея наблюдаемую выборку  $X$ , можно предсказывать значение  $M = \xi_{\mathbb{X}}^{(m)}$  и оценивать точность предсказаний.

**Теорема 3.2.** Отрезок  $[\xi_X^{(r)}, \xi_X^{(s)}]$  является доверительным интервалом для величины  $M = \xi_{\mathbb{X}}^{(m)}$  с доверительной вероятностью

$$\mathbb{P}[\xi_X^{(r)} \leq M \leq \xi_X^{(s)}] = \sum_{t=r}^s h_L^{\ell, m}(t), \quad (3.4)$$

где вероятность  $P$  понимается в соответствии со слабой аксиоматикой, как доля разбиений генеральной выборки.

### Доказательство.

Во-первых, заметим, что  $P[\xi_X^{(r)} \leq M \leq \xi_X^{(s)}] = P[\xi_X^{(r)} \leq M] - P[\xi_X^{(s)} < M]$ .

Рассмотрим событие  $S_m = \{\xi_X^{(1)}, \dots, \xi_X^{(m)}\} \subseteq \mathbb{X}$ .

Условие  $P[\xi_X^{(r)} \leq M]$  равносильно тому, что в выборку  $X$  попадает не менее  $r$  элементов события  $S_m$ . Согласно Лемме 2.1 и определению «правого хвоста» гипергеометрического распределения (2.4) отсюда следует

$$P[\xi_X^{(r)} \leq M] = \bar{H}_L^{\ell, m}(r).$$

Аналогично,

$$P[\xi_X^{(s)} < M] = \bar{H}_L^{\ell, m}(s+1).$$

Таким образом,

$$\begin{aligned} P[\xi_X^{(r)} \leq M \leq \xi_X^{(s)}] &= \bar{H}_L^{\ell, m}(r) - \bar{H}_L^{\ell, m}(s+1) = \\ &= \sum_{t=r}^{s_1} h_L^{\ell, m}(t) - \sum_{t=s+1}^{s_1} h_L^{\ell, m}(t) = \sum_{t=r}^s h_L^{\ell, m}(t). \end{aligned}$$

Теорема доказана. ■

**Предельный переход.** Доверительная вероятность в слабой аксиоматике (3.4) выражается через гипергеометрическое распределение. Покажем, что при стремлении длины скрытой выборки к бесконечности оно стремится к биномиальному распределению, что приводит к классической доверительной вероятности (3.3).

При  $L \rightarrow \infty$  и  $k \rightarrow \infty$ , фиксированной длине наблюдаемой выборки  $\ell$ , фиксированных  $r$  и  $s$  и сохранении постоянного отношения  $\alpha = \frac{m}{L}$  имеем:

$$\begin{aligned} h_L^{\ell, m}(t) &= C_\ell^t \frac{C_k^{m-t}}{C_L^m} = C_\ell^t \frac{m!}{(m-t)!} \frac{k!}{L!} \frac{(L-m)!}{(k-m+t)!} \rightarrow C_\ell^t \frac{m^t (k-m)^{\ell-t}}{L^\ell} = \\ &= C_\ell^t \left(\frac{m}{L}\right)^t \left(\frac{k}{L} - \frac{m}{L}\right)^{\ell-t} \rightarrow C_\ell^t \alpha^t (1-\alpha)^{\ell-t}. \end{aligned}$$

Таким образом, перенос доверительных оценок в слабую аксиоматику потребовал переформулировать постановку задачи. В слабой аксиоматике доверительный интервал зависит от длины скрытой выборки, то есть от того, сколько ещё наблюдений предстоит сделать. В асимптотике, при стремлении этого числа в бесконечность, получается классический доверительный интервал в сильной аксиоматике.

## §3.3 Критерий знаков

**Классический критерий знаков** проверяет гипотезу  $H_0$  о том, что в выборке бинарных величин  $X = \{b_i\}_{i=1}^\ell$ ,  $b_i \in \{0, 1\}$ , единицы появляются с вероятностью  $p = \frac{1}{2}$ ,

или, другими словами, что выборка подчиняется биномиальному распределению с параметром  $p = \frac{1}{2}$ .

Хрестоматийный пример применения критерия знаков — проверка симметричности монеты по последовательности выпадения «орлов» и «решек».

В практических задачах анализа данных критерий знаков применяют для проверки равенства нулю медианы вещественной случайной величины  $\xi$ . Для этого переходят к бинарной выборке  $b_i = [\xi_i > 0]$ . Другое применение — проверка гипотезы сдвига  $\xi' = \xi + \delta$  в двух связанных выборках, где  $\delta$  — величина сдвига. Для этого переходят к бинарной выборке  $b_i = [\xi'_i > \xi_i + \delta]$ . В частности, при  $\delta = 0$  проверяется отсутствие сдвига, или, как ещё говорят, *отсутствие эффекта обработки*.

Для выполнения теста вычисляется статистика  $T(X) = \sum_{i=1}^{\ell} b_i$ . При условии, что гипотеза  $H_0$  верна, статистика  $T$  подчиняется биномиальному распределению:

$$P[T(X) = t] = C_{\ell}^t p^t (1-p)^{\ell-t} = C_{\ell}^t 2^{-\ell}.$$

Гипотеза  $H_0$  отвергается на уровне значимости  $\alpha$ , если значение статистики  $T(X)$  попадает в критическую область — «хвосты» биномиального распределения,

$$\frac{1}{2^{\ell}} \sum_{t=0}^{T(X)} C_{\ell}^t \notin \left[ \frac{\alpha}{2}, 1 - \frac{\alpha}{2} \right]. \quad (3.5)$$

**Критерий знаков в слабой аксиоматике** строится по сути так же, только формулировка гипотезы  $H_0$  уже не может опираться на понятие «вероятность единицы» и биномиальное распределение.

Теперь гипотеза  $H_0$  формулируется так: «наблюдаемая выборка  $X$  получена в результате случайного разбиения  $X \sqcup \bar{X} = \mathbb{X}$  генеральной выборки  $\mathbb{X} = \{b_i\}_{i=1}^L$ , содержащей ровно половину единиц,  $m = \sum_{i=1}^L b_i = \frac{L}{2}$ », где  $L$  — чётное число.

Если гипотеза  $H_0$  верна, то верно и основное предположение слабой аксиоматики о равной вероятности разбиений. Применяя Лемму 2.1 к событию  $S = \{b_i \in \mathbb{X} : b_i = 1\}$ , приходим к выводу, что статистика  $T(X)$  подчиняется гипергеометрическому распределению:

$$P[T(X) = t] = h_L^{\ell, m}(t) = C_{\ell}^t \frac{C_{L-\ell}^{m-t}}{C_L^m}, \quad m = \frac{L}{2},$$

где вероятность  $P$  понимается в соответствии со слабой аксиоматикой, как доля разбиений генеральной выборки.

Далее стандартная логика проверки статистических гипотез переносится в слабую аксиоматику без изменений. Гипотеза  $H_0$  отвергается на уровне значимости  $\alpha$ , если значение  $T(X)$  попадает в критическую область:

$$H_L^{\ell, m}(T(X)) \notin \left[ \frac{\alpha}{2}, 1 - \frac{\alpha}{2} \right],$$

**Предельный переход.** Параметр  $L$  был введён искусственно в результате мысленного достраивания наблюдаемой выборки  $X$  до генеральной выборки  $\mathbb{X}$ . Устремляя длину выборки  $\mathbb{X}$  в бесконечность при  $m = \frac{L}{2}$  и фиксированных  $\ell, t$ , получаем всё то же биномиальное распределение, приводящее к классическому критерию (3.5):

$$P[T(X) = t] = C_\ell^t \frac{m!}{(m-t)!} \frac{m!}{(m-\ell+t)!} \frac{k!}{L!} \rightarrow C_\ell^t \frac{m^t m^{\ell-t}}{L^\ell} = C_\ell^t \frac{m^\ell}{(2m)^\ell} = C_\ell^t 2^{-\ell}.$$

Таким образом, перенос критерия знаков в слабую аксиоматику потребовал лишь переформулировки нулевой гипотезы, не изменив результат по существу.

### §3.4 Критерий Уилкоксона–Манна–Уитни

**Классический критерий Уилкоксона–Манна–Уитни** — это непараметрический статистический критерий, обычно используемый для проверки *гипотезы однородности* — предположения, что две выборки  $X = (x_1, \dots, x_\ell)$  и  $\bar{X} = (x'_1, \dots, x'_k)$  взяты из одного распределения. Предполагается, что величины  $x, x'$  измеряются в количественной или порядковой шкале.

Строго говоря, нулевой гипотезой в данном критерии является более слабое предположение  $H_0: P(x < x') = \frac{1}{2}$ . Поэтому он считается не достаточно мощным для проверки гипотезы однородности.

Для простоты в качестве альтернативы рассмотрим  $H_1: P(x < x') > \frac{1}{2}$ . Проверка двусторонней альтернативы не намного сложнее.

То, что данный тест не всегда позволяет обнаружить неоднородность выборок, легко показать с помощью контрпримера. Возьмём две выборки  $X, \bar{X}$  равной длины. Пусть  $X$  сосредоточена на отрезке  $[1, 2]$ , половина выборки  $\bar{X}$  сосредоточена на отрезке  $[0, 1]$ , вторая половина — на  $[2, 3]$ . Тогда гипотеза  $H_0$  справедлива, а гипотеза однородности, очевидно, нет.

Для выполнения теста вычисляется статистика

$$U(X, \bar{X}) = \sum_{i=1}^{\ell} \sum_{j=1}^k [x_i < x'_j].$$

Обычно используется более экономный способ вычисления  $U$ . Строится вариационный ряд  $x^{(1)} < \dots < x^{(L)}$  объединённой выборки  $\mathbb{X} = X \sqcup \bar{X}$ . Для простоты будем предполагать, что вариационный ряд не содержит связок. Далее находятся ранги  $r(x_i) \in \{1, \dots, L\}$  элементов первой выборки в общем вариационном ряду, и  $U$  выражается через их суммарный ранг:

$$U(X, \bar{X}) = \ell k + \frac{\ell(\ell+1)}{2} - \sum_{i=1}^{\ell} r(x_i).$$

Таким образом, данный критерий является примером *рангового критерия*.



Распределение статистики  $U$  асимптотически нормально, причём асимптотикой рекомендуется пользоваться уже при  $\ell, k > 8$  [38, 4, 27]:

$$\tilde{U} = \frac{U - \frac{1}{2}\ell k}{\sqrt{\frac{1}{12}\ell k(\ell + k + 1)}} \sim \mathcal{N}(0, 1).$$

Если  $\tilde{U} > \Phi_{1-\alpha}$ , где  $\Phi_{1-\alpha}$  —  $(1-\alpha)$ -квантиль нормального распределения, то гипотеза  $H_0$  отвергается в пользу альтернативы  $H_1$  при уровне значимости  $\alpha$ .

**В слабой аксиоматике** аналогом гипотезы однородности является основное предположение о равной вероятности всех разбиений. Данный критерий позволяет проверить, могла ли пара выборок  $X, \bar{X}$  быть получена в результате случайного разбиения исходной выборки  $\mathbb{X}$  на две части. Соображения о недостаточной мощности этого критерия и контрпример остаются в силе.

С точки зрения слабой аксиоматики знак вероятности  $P$  в условии  $P[x < x'] = \frac{1}{2}$  употреблён некорректно, поэтому гипотезу  $H_0$  необходимо переформулировать.

Первая интерпретация гипотезы  $H_0$ . Допустим, что из подвыборок  $X, \bar{X}$  случайно и равновероятно извлекается по одному объекту,  $x$  и  $x'$  соответственно. Тогда вероятность надо понимать в смысле усреднения не только по всем разбиениям выборки  $\mathbb{X}$ , но и по всем способам взять по одному объекту из  $X$  и  $\bar{X}$ . В слабой аксиоматике вероятность  $P[x < x']$  действительно равна  $\frac{1}{2}$ :

$$\begin{aligned} P[x < x'] &= \mathbb{E} \frac{1}{\ell} \sum_{i=1}^L [x_i \in X] \frac{1}{k} \sum_{j=1}^L [x_j \in \bar{X}] [x_i < x_j] = \\ &= \frac{1}{\ell k} \sum_{i=1}^L \sum_{j=1}^L [x_i < x_j] \underbrace{P[x_i \in X] [x_j \in \bar{X}]}_{C_{L-2}^{\ell-1} / C_L^\ell} = \frac{1}{\ell k} \cdot \frac{L(L-1)}{2} \cdot \frac{C_{L-2}^{\ell-1}}{C_L^\ell} = \frac{1}{2}. \end{aligned}$$

Отсюда следует, что  $\mathbb{E}U(X, \bar{X}) = \frac{1}{2}\ell k$ . Это и есть вторая интерпретация гипотезы  $H_0$ , не требующая введения дополнительной рандомизации.

Рекуррентная формула для распределения статистики  $U$  выводится в слабой аксиоматике так же, как она была получена в исходных работах [113, 84].

**Теорема 3.3.** Пусть все значения в выборке  $\mathbb{X}$  попарно различны. Распределение статистики  $U(X, \bar{X})$ ,

$$P_{\ell,k}(u) = P[U(X, \bar{X}) = u], \quad u \in \{0, \dots, \ell k\},$$

не зависит от элементов выборки  $\mathbb{X}$  и вычисляется по рекуррентной формуле

$$P_{\ell,k}(u) = \frac{\ell}{L} P_{\ell-1,k}(u) + \frac{k}{L} P_{\ell,k-1}(u - \ell),$$

при начальных условиях

$$P_{\ell,0}(u) = P_{0,k}(u) = [u=0], \quad u \in \{0, \dots, \ell k\};$$

$$P_{\ell,k}(u) = 0, \quad u \notin \{0, \dots, \ell k\}.$$

**Доказательство.** Построим вариационный ряд элементов выборки  $x^{(1)} < \dots < x^{(L)}$ . Он не содержит связок в силу условия попарной различности.

Обозначим  $b_i = b_i(X) = [x^{(i)} \in X]$ . Бинарная последовательность  $b_1, \dots, b_L$  содержит ровно  $\ell$  единиц и  $k$  нулей. Тогда

$$U(X, \bar{X}) = \sum_{i=1}^L \sum_{j=1}^L [i < j] b_i \bar{b}_j,$$

где  $\bar{b}_j = 1 - b_j$  — отрицание бинарной величины. Значение статистики  $U$  не зависит от значений элементов выборки  $\mathbb{X}$ , а только от разбиения  $(X, \bar{X})$  и параметров длины подвыборок  $\ell, k$ . Чтобы найти рекуррентную формулу для  $U_{\ell, k} = U(X, \bar{X})$ , запишем

$$U_{\ell, k} = \sum_{i=1}^{L-1} \sum_{j=1}^{L-1} [i < j] b_i \bar{b}_j + \bar{b}_L \sum_{i=1}^{L-1} b_i.$$

Если  $b_L = 1$ , то первое слагаемое в этой сумме есть  $U_{\ell-1, k}$ , а второе равно нулю.

Если  $b_L = 0$ , то первое слагаемое в этой сумме есть  $U_{\ell, k-1}$ , а второе  $\sum_{i=1}^{L-1} b_i = \ell$ .

Таким образом,  $U_{\ell, k} = b_L U_{\ell-1, k} + \bar{b}_L (U_{\ell, k-1} + \ell)$ .

Аналогичное рекуррентное соотношение выполняется и для вероятностей:

$$\begin{aligned} P_{\ell, k}(u) &= \mathbf{P}[x^{(L)} \in X] [U_{\ell-1, k} = u] + \mathbf{P}[x^{(L)} \in \bar{X}] [U_{\ell, k-1} + \ell = u] = \\ &= \frac{\ell}{L} P_{\ell-1, k}(u) + \frac{k}{L} P_{\ell, k-1}(u - \ell). \end{aligned}$$

Начальные условия для  $P_{\ell, k}(u)$  являются очевидным следствием начальных условий для  $U$ :  $U_{\ell, 0} = U_{0, k} = 0$  и ограничений на значения аргумента  $u \in \{0, \dots, \ell k\}$ .

Теорема доказана. ■

Для проверки гипотезы  $H_0$  строится распределение Уилкоксона

$$W(U) = \sum_{u=0}^U P_{\ell, k}(u).$$

Если  $U > W_{1-\alpha}$ , где  $W_{1-\alpha}$  —  $(1-\alpha)$ -квантиль распределения Уилкоксона, то гипотеза  $H_0$  отвергается в пользу альтернативы  $H_1$  при уровне значимости  $\alpha$ .

По всей видимости, не только критерий Уилкоксона–Манна–Уитни, но и более широкий класс ранговых критериев переносится в слабую аксиоматику без значительных дополнительных усилий.

## Резюме

В слабой вероятностной аксиоматике удаётся чисто комбинаторными методами получать аналоги многих непараметрических статистических критериев и доверительных оценок. Результаты являются точными и формулируются в терминах конечных выборок. В ряде случаев меняется интерпретация исходной постановки задачи, например, формулировка нулевой гипотезы. Предельный переход  $L \rightarrow \infty$  при фиксированной длине наблюдаемой выборки  $\ell$  приводит к классическим результатам.

**В следующей лекции** мы рассмотрим ещё одну классическую задачу математической статистики — оценку равномерного отклонения двух эмпирических функций распределения. На её основе строится двухвыборочный критерий Колмогорова–Смирнова, с помощью которого можно проверять гипотезу однородности двух выборок. В слабой аксиоматике эта задача имеет точное решение. Заодно мы познакомимся с такими интересными математическими объектами, как усечённый треугольник Паскаля и одномерные случайные блуждания.

## Упражнения

В следующих упражнениях предлагается, пользуясь только слабой аксиоматикой, вывести точные формулы распределения статистики в некоторых известных непараметрических критериях.

**Задача 3.1 (3\*).** Критерий серий Вальда–Вольфовица.

**Задача 3.2 (3\*).** Критерий Краскела–Уоллиса.

**Задача 3.3 (3\*).** Критерий равенства нулю коэффициента корреляции Кенделла.

## 4 Эмпирические распределения и случайное блуждание

В Лекции 2 мы оценивали вероятность большого отклонения частот в двух выборках. В слабой вероятностной аксиоматике эта задача является естественным аналогом закона больших чисел и имеет точное решение. Оценивание вероятности большого отклонения двух функций распределения — это ещё одна классическая задача математической статистики. Её можно интерпретировать двумя способами. Во-первых, как задачу предсказания: дана эмпирическая функция распределения случайной величины на наблюдаемой выборке; требуется оценить её эмпирическую функцию распределения на скрытой выборке. Во-вторых, как задачу проверки гипотезы однородности: даны две наблюдаемые выборки; требуется определить, получены ли они из одного распределения. Для этого используются различные статистические критерии, в частности, критерий Смирнова (чаще называемый двухвыборочным критерием Колмогорова-Смирнова), основанный на сравнении двух эмпирических функций распределения.

Мы рассмотрим постановку и точное решение этих классических задач в слабой аксиоматике и покажем, что они тесно связаны с такими математическими объектами, как усечённый треугольник Паскаля и одномерные случайные блуждания.

### §4.1 Эмпирическое распределение

Определим для произвольной функции  $\xi: \mathbb{X} \rightarrow \mathbb{R}$  и произвольной конечной выборки  $U \subseteq \mathbb{X}$  эмпирическую функцию распределения  $F_\xi: \mathbb{R} \rightarrow [0, 1]$  как долю объектов  $x$  выборки  $U$ , для которых значение  $\xi(x)$  не превосходит  $z$ :

$$F_\xi(z, U) = \frac{1}{|U|} \sum_{x \in U} [\xi(x) \leq z].$$

Определим одностороннее и двустороннее *равномерное отклонение эмпирических функций распределения*:

$$D^+(X) = \max_{z \in \mathbb{R}} (F_\xi(z, \bar{X}) - F_\xi(z, X));$$

$$D^-(X) = \max_{z \in \mathbb{R}} (F_\xi(z, X) - F_\xi(z, \bar{X}));$$

$$D(X) = \max_{z \in \mathbb{R}} |F_\xi(z, \bar{X}) - F_\xi(z, X)| = \max\{D^-(X), D^+(X)\}.$$

Задача состоит в том, чтобы найти вероятность большого отклонения эмпирических функций распределения:

$$\mathbb{P}[D(X) > \varepsilon] \leq \eta(\varepsilon), \tag{4.1}$$

где вместо  $D(X)$  могут быть также подставлены  $D^-(X)$  или  $D^+(X)$ .

В классической вероятностной аксиоматике имеет место следующий факт [44].

**Теорема 4.1 (Н. В. Смирнов).** Если  $X, \bar{X} \subseteq \mathcal{X}$  — случайные, независимые, одинаково распределённые выборки;  $\xi: \mathcal{X} \rightarrow \mathbb{R}$  — случайная величина с непрерывным распределением, то справедливы асимптотические оценки

$$\lim_{\ell, k \rightarrow \infty} \mathbb{P}\{D^\pm(X) \geq \varepsilon\} = \exp\left(-2\varepsilon^2 \frac{\ell k}{\ell+k}\right); \quad (4.2)$$

$$\lim_{\ell, k \rightarrow \infty} \mathbb{P}\{D(X) \geq \varepsilon\} = 2 \sum_{i=1}^{\infty} (-1)^{i+1} \exp\left(-2\varepsilon^2 \frac{\ell k}{\ell+k} i^2\right); \quad (4.3)$$

Правая часть (4.3) представима также в виде  $1 - K\left(\varepsilon \sqrt{\frac{\ell k}{\ell+k}}\right)$ , где  $K(z)$  — функция распределения Колмогорова:

$$K(z) = \sum_{i=-\infty}^{\infty} (-1)^i e^{-2z^2 i^2}.$$

Известны и неасимптотические точные оценки, но они имеют достаточно громоздкий вид [15]. Мы покажем, что точные оценки могут быть выражены более элегантно через усечённый треугольник Паскаля [55], причём доказательство имеет прозрачный геометрический смысл.

## §4.2 Усечённый треугольник Паскаля

Пусть  $g_m^-, g_m^+$ ,  $m = 0, \dots, L$  — две неубывающие последовательности, удовлетворяющие условию  $0 \leq g_m^- \leq g_m^+ \leq m$ .

**Определение 4.1.** Усечённым треугольником Паскаля с нижней границей  $g_m^-$  и верхней границей  $g_m^+$  называется целочисленная функция  $G_m^s = G_m^s[g_m^-, g_m^+]$ , определяемая рекуррентными соотношениями

$$\begin{aligned} G_0^s &= [s = 0], \quad s \in \mathbb{Z}; \\ G_m^s &= (G_{m-1}^s + G_{m-1}^{s-1}) [g_m^- \leq s \leq g_m^+], \quad m \in \mathbb{N}, \quad s \in \mathbb{Z}. \end{aligned} \quad (4.4)$$

Усечённый треугольник Паскаля вычисляется по тому же рекуррентному правилу, что и классический треугольник Паскаля, если в нём обнулить все элементы, лежащие за пределами границ  $[g_m^-, g_m^+]$ . «Неусечённый» треугольник Паскаля совпадает с классическим и даёт биномиальные коэффициенты  $G_m^s = G_m^s[0, m]$ .

При начальном условии  $G_0^s = [s = 0]$  ненулевыми могут быть только элементы  $G_m^s$  при  $0 \leq s \leq m$ . Другие начальные условия приводят к иным неклассическим обобщениям треугольника Паскаля, которые мы здесь не рассматриваем.

Определим для произвольных  $\varepsilon > 0$  и  $m = 0, 1, 2, \dots$ , линейные границы (в дальнейшем аргумент  $\varepsilon$  иногда будем опускать):

$$\begin{aligned} g_m^+(\varepsilon) &= \frac{\ell}{L}(m + \varepsilon k); \\ g_m^-(\varepsilon) &= \frac{\ell}{L}(m - \varepsilon k). \end{aligned}$$

На рис. 4.2 и 4.4 приведены примеры четырёх вариантов усечения треугольника Паскаля с такими границами. В отличие от привычного способа изображения, треугольники «положены на бок» путём поворота на  $90^\circ$  против часовой стрелки.

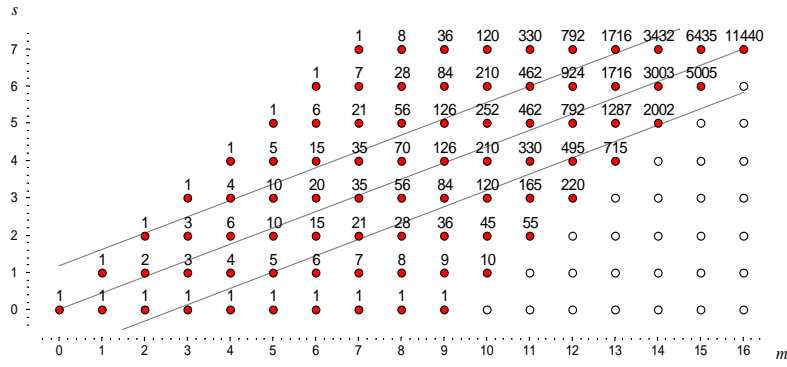


Рис. 4.1. Классический треугольник Паскаля  $C_m^s = G_m^s[0, m]$  при  $L = 16, \ell = 7, \varepsilon = 0.3$ .

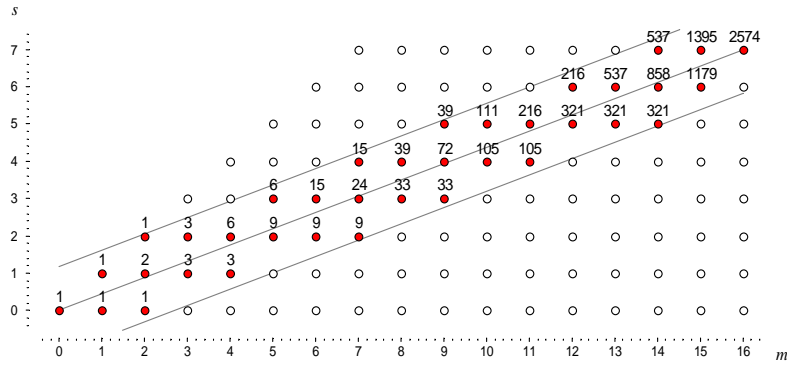


Рис. 4.2. Усечённый треугольник Паскаля  $G_m^s[g_m^-(\varepsilon), g_m^+(\varepsilon)]$  при  $L = 16, \ell = 7, \varepsilon = 0.3$ .

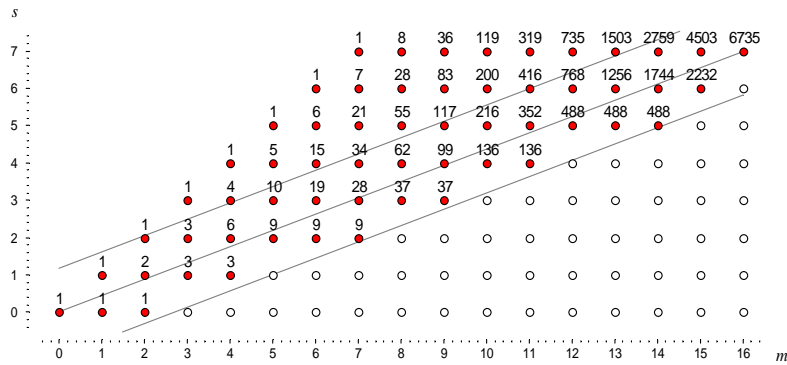


Рис. 4.3. Усечённый слева треугольник Паскаля  $G_m^s[g_m^-(\varepsilon), m]$ ,  $L = 16, \ell = 7, \varepsilon = 0.3$ .

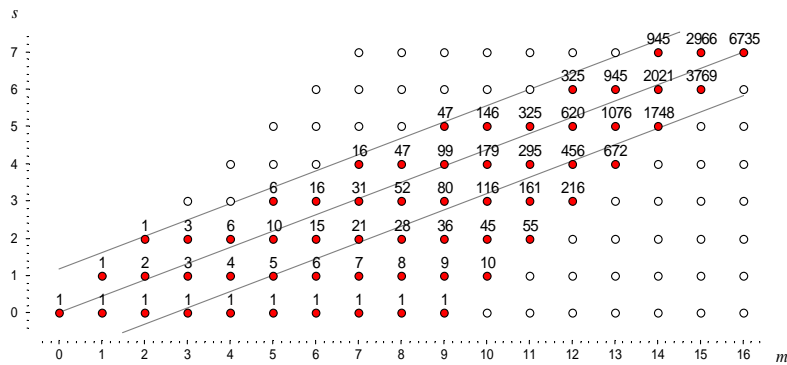


Рис. 4.4. Усечённый справа треугольник Паскаля  $G_m^s[0, g_m^+(\varepsilon)]$ ,  $L = 16, \ell = 7, \varepsilon = 0.3$ .

### §4.3 Теорема Смирнова

**Теорема 4.2.** Для произвольной конечной выборки  $\mathbb{X}$  и произвольной функции  $\xi: \mathbb{X} \rightarrow \mathbb{R}$ , значения которой попарно различны на элементах выборки  $\mathbb{X}$ , справедливы точные оценки:

$$\mathbb{P}[D^+(X) \leq \varepsilon] = G_L^\ell[0, g_L^+(\varepsilon)]/C_L^\ell; \quad (4.5)$$

$$\mathbb{P}[D^-(X) \leq \varepsilon] = G_L^\ell[g_L^-(\varepsilon), L]/C_L^\ell; \quad (4.6)$$

$$\mathbb{P}[D(X) \leq \varepsilon] = G_L^\ell[g_L^-(\varepsilon), g_L^+(\varepsilon)]/C_L^\ell. \quad (4.7)$$

**Доказательство.** 1. Составим вариационный ряд значений функции  $\xi(x)$  на элементах выборки:  $\xi(x^{(1)}) < \xi(x^{(2)}) < \dots < \xi(x^{(L)})$ . Здесь все неравенства строгие в силу условия попарной различности.

Обозначим  $b_i = b_i(X) = [x^{(i)} \in X]$ . Бинарная последовательность  $b_1, \dots, b_L$  содержит ровно  $\ell$  единиц и  $k$  нулей.

Воспользуемся определением функции распределения:

$$\begin{aligned} D(X) &= \max_{z \in \mathbb{R}} |F_\xi(z, \bar{X}) - F_\xi(z, X)| = \\ &= \max_{z \in \mathbb{R}} \left| \frac{1}{k} \sum_{i=1}^L [x_i \in \bar{X}][\xi(x_i) < z] - \frac{1}{\ell} \sum_{i=1}^L [x_i \in X][\xi(x_i) < z] \right|. \end{aligned} \quad (4.8)$$

Изменим порядок слагаемых в суммах, теперь суммируя их в порядке возрастания значений  $\xi(x_i)$ . Это равносильно тому, что в данной формуле все вхождения  $x_i$  заменятся на  $x^{(i)}$ . Тогда можно убрать сомножитель  $[\xi(x^{(i)}) < z]$ , заменив верхний предел суммирования на  $m = \max\{i: \xi(x^{(i)}) < z\}$ , и максимум брать не по действительному параметру  $z$ , а по целочисленному параметру  $m$ :

$$\begin{aligned} D(X) &= \max_{m=1..L} \left| \frac{1}{k} \sum_{i=1}^m \underbrace{[x^{(i)} \in \bar{X}]}_{1-b_i} - \frac{1}{\ell} \sum_{i=1}^m \underbrace{[x^{(i)} \in X]}_{b_i} \right| = \\ &= \max_{m=1..L} \left| \frac{m}{k} - \frac{\ell+k}{\ell k} \sum_{i=1}^m b_i \right| = \frac{L}{\ell k} \max_{m=1..L} \left| B_m - \frac{m\ell}{L} \right|, \end{aligned}$$

где  $B_m = B_m(X) = b_1 + \dots + b_m$ .

Таким образом, равномерное отклонение эмпирических распределений на выборках  $X$  и  $\bar{X}$  выражается через равномерное отклонение числа единиц в первых  $m$  членах последовательности  $b_1, \dots, b_L$  от «ожидаемого» числа единиц  $m\ell/L$ .

Теперь запишем долю разбиений выборки  $X$ , при которых равномерное отклонение эмпирических распределений не превышает пороговую точность  $\varepsilon$ :

$$\begin{aligned}
 \mathbb{P}[D(X) \leq \varepsilon] &= \\
 &= \mathbb{P}\left[\max_{m=1..L} \left|B_m - \frac{m\ell}{L}\right| \leq \frac{\varepsilon\ell k}{L}\right] = \\
 &= \mathbb{P}\left[\max_{m=1..L} \left(-B_m + \underbrace{\left(\frac{m\ell}{L} - \frac{\varepsilon\ell k}{L}\right)}_{g_m^-(\varepsilon)}\right) \leq 0\right] \left[\max_{m=1..L} \left(B_m - \underbrace{\left(\frac{m\ell}{L} + \frac{\varepsilon\ell k}{L}\right)}_{g_m^+(\varepsilon)}\right) \leq 0\right] = \\
 &= \frac{1}{C_L^\ell} \sum_{(X, \bar{X})} \prod_{m=1}^L [g_m^-(\varepsilon) \leq B_m(X) \leq g_m^+(\varepsilon)]. \tag{4.9}
 \end{aligned}$$

Последнее равенство следует из тождества  $[\max_m A_m \leq 0] = \prod_m [A_m \leq 0]$ .

2. Рассмотрим подвыборку  $X^m = \{x^{(1)}, \dots, x^{(m)}\}$ , состоящую из первых  $m$  членов вариационного ряда. Возьмём максимальное (по включению) подмножество  $N$  разбиений  $(X, \bar{X})$ , удовлетворяющих двум условиям:

- 1) они индуцируют попарно различные разбиения подвыборки  $X^m$ ;
- 2) ровно  $s$  объектов из  $X^m$  попадают в  $X$ .

Очевидно, число этих разбиений  $|N| = C_m^s$ . Представим множество разбиений  $N$  в виде объединения непересекающихся подмножеств  $N_0 = \{(X, \bar{X}) \in N : b_m(X) = 0\}$  и  $N_1 = \{(X, \bar{X}) \in N : b_m(X) = 1\}$ . Очевидно,  $|N_0| = C_{m-1}^s$ ,  $|N_1| = C_{m-1}^{s-1}$ .

Нас будет интересовать выражение  $H_m^s = \sum_{(X, \bar{X})} \prod_{r=1}^m [g_r^- \leq B_r(X) \leq g_r^+]$ , поскольку правая часть (4.9) равна  $H_L^\ell / C_L^\ell$ . Разобьём в этом выражении сумму по  $N$  на две суммы — по  $N_0$  и по  $N_1$ , и ещё заметим, что  $B_m(X) = s$  для всех  $(X, \bar{X}) \in N$ :

$$\begin{aligned}
 H_m^s &= \underbrace{\sum_{(X, \bar{X}) \in N_0} \prod_{r=1}^{m-1} [g_r^- \leq B_r(X) \leq g_r^+]}_{H_{m-1}^s} [g_m^- \leq s \leq g_m^+] + \\
 &+ \underbrace{\sum_{(X, \bar{X}) \in N_1} \prod_{r=1}^{m-1} [g_r^- \leq B_r(X) \leq g_r^+]}_{H_{m-1}^{s-1}} [g_m^- \leq s \leq g_m^+] = \\
 &= (H_{m-1}^s + H_{m-1}^{s-1}) [g_m^- \leq s \leq g_m^+].
 \end{aligned}$$

Таким образом, получена рекуррентная формула для  $H_m^s$ , формально совпадающая с формулой усечённого треугольника Паскаля (4.4). Осталось только проверить граничные случаи.

При  $m = 1$  и фиксированном  $s \in \{0, 1\}$  имеется только одно разбиение,  $|N| = 1$ , следовательно,  $H_1^s = [g_m^- \leq s \leq g_m^+]$ , что совпадает с (4.4).

При  $s = 0$  и произвольном  $m = 1, \dots, k$  имеется только одно разбиение,  $|N| = 1$ , причём ни один объект из  $X^m$  не попадает в  $X$ . Это означает, что  $B_r = 0$  при всех  $r = 1, \dots, m$ . Но тогда  $H_m^0 = \prod_{r=1}^m [g_r^- \leq s \leq g_r^+]$ , что, опять-таки, совпадает с (4.4).



Заметим также, что при  $s = 0$  запись  $G_{m-1}^{s-1} = 0$  по определению корректна, в то же время  $H_{m-1}^{s-1} = 0$ , поскольку  $N_1 = \emptyset$ . Аналогично, при  $s = m$  имеем  $N_0 = \emptyset$ , следовательно,  $H_{m-1}^s = 0 = G_{m-1}^s$ .

3. Односторонние оценки (4.5) и (4.6) выводятся аналогично. Различие в том, что для них выражение под знаком произведения в (4.9) принимает вид, соответственно, либо  $[0 \leq B_m \leq g_m^+(\varepsilon)]$ , либо  $[g_m^-(\varepsilon) \leq B_m \leq m]$ . Изменяется только форма границы в усечённом треугольнике Паскаля, соответственно, либо нижней  $g_m^-(\varepsilon) = 0$ , либо верхней  $g_m^+(\varepsilon) = m$ , и все дальнейшие рассуждения остаются в силе. ■

**Геометрическая интерпретация.** Вторую часть доказательства (после формулы (4.9)) можно провести гораздо короче и нагляднее, пользуясь следующими геометрическими соображениями.

Каждое разбиение  $X \sqcup \bar{X} = \mathbb{X}$  взаимно однозначно соответствует бинарному вектору  $b = (b_1, \dots, b_L)$ , состоящему из  $\ell$  единиц и  $k$  нулей, и, в то же время, некоторой траектории, проходящей из точки  $(0, 0)$  в точку  $(L, \ell)$  согласно правилу: если  $b_i = 1$ , то сместиться вправо и вверх на 1; если  $b_i = 0$ , то сместиться вправо на 1, см. рис. 2.3. Очевидно, траектория состоит из всех точек  $(m, B_m)_{m=0}^L$ . Выполнение совокупности условий  $[g_m^- \leq B_m(X) \leq g_m^+]$  при всех  $m = 1, \dots, L$  означает, что траектория не может проходить ниже границы  $g_m^-$  или выше границы  $g_m^+$ . На рис. 2.3 эти границы показаны линиями. Согласно (4.9) функционал  $P[D(X) \leq \varepsilon]$  в точности равен доле таких траекторий. Будем называть их *допустимыми*. Обозначим через  $H_m^s$  число допустимых траекторий, проходящих из точки  $(0, 0)$  в точку  $(m, s)$ . Допустимая траектория может прийти в  $(m, s)$  либо из  $(m-1, s-1)$ , либо из  $(m-1, s)$ . Отсюда следует рекуррентная формула для числа допустимых траекторий:  $H_m^s = H_{m-1}^{s-1} + H_{m-1}^s$ . Однако, если  $s \notin [g_m^-, g_m^+]$ , то все такие траектории уже не будут допустимыми, поэтому окончательная формула принимает вид  $H_m^s = (H_{m-1}^{s-1} + H_{m-1}^s) [g_m^- \leq s \leq g_m^+]$ , что совпадает с определением усечённого треугольника Паскаля:  $H_m^s \equiv G_m^s$ .

**Практическое вычисление** по рекуррентным соотношениям (4.4) сталкивается с проблемой переполнения: значения  $G_m^s$  выходят за пределы разрядной сетки современных компьютеров при  $L$  порядка нескольких сотен. Проблема снимается, если вывести рекуррентную формулу для отношений  $\varphi_m^s = G_m^s / C_m^s$ , которые принимают значения из отрезка  $[0, 1]$ . Применив тождества  $C_m^s = \frac{m}{m-s} C_m^{s-1} = \frac{m}{s} C_{m-1}^{s-1}$ , получим:

$$\varphi_m^s = \frac{m-s}{m} \varphi_{m-1}^s + \frac{s}{m} \varphi_{m-1}^{s-1}.$$

Усечённый треугольник Паскаля оказывается полезной концепцией не только при выводе точного выражения для критерия Смирнова, но во многих задачах, связанных со случайными блужданиями при ограничениях. Упомянем только выборочный контроль качества [3] и анализ выживаемости [55].

## §4.4 Обобщение на случай вариационного ряда со связками

В Теореме 4.1 (Смирнова) требование непрерывности функции распределения является существенным. В сильной аксиоматике оно гарантирует, что с вероятно-

стью 1 вариационный ряд  $\xi(x^{(1)}) < \xi(x^{(2)}) < \dots < \xi(x^{(L)})$  не содержит одинаковых элементов, следовательно, ранжировка определена единственным образом. Если условие непрерывности нарушается, равенства (4.2) и (4.3) могут не выполняться [4].

В Теореме 4.2 требование различности всех элементов вариационного ряда формулировалось в явном виде. Покажем, оставаясь в рамках слабой аксиоматики, что отказ от этого требования не сильно меняет вид результата — в Теореме 4.2 изменятся только границы усечения  $[g_m^-, g_m^+]$ . Следующая теорема обобщает критерий Смирнова на случай дискретных распределений.

**Теорема 4.3.** Пусть  $\xi: \mathbb{X} \rightarrow \mathbb{R}$  — произвольная функция,  $\mathbb{X}$  — произвольная конечная выборка, вариационный ряд значений  $\xi(x_i)$  состоит из  $H$  связей:

$$\underbrace{\xi(x^{(1)}) = \dots = \xi(x^{(i_1)})}_{1\text{-я связка}} < \underbrace{\xi(x^{(i_1+1)}) = \dots = \xi(x^{(i_2)})}_{2\text{-я связка}} < \dots < \underbrace{\xi(x^{(i_{H-1}+1)}) = \dots = \xi(x^{(i_H)})}_{H\text{-я связка}}.$$

Тогда в слабой аксиоматике справедливы точные оценки (4.5), (4.6), (4.7), если взять границы усечённого треугольника Паскаля  $[\tilde{g}_m^-, \tilde{g}_m^+]$ :

$$\begin{aligned} \tilde{g}_m^+(\varepsilon) &= \min\{g_{i_{h-1}}^+(\varepsilon) + m - i_{h-1}, g_{i_h}^+(\varepsilon)\}; \\ \tilde{g}_m^-(\varepsilon) &= \max\{g_{i_{h-1}}^-(\varepsilon), g_{i_h}^-(\varepsilon) + m - i_h\}; \end{aligned}$$

для всех  $m = i_{h-1}+1, \dots, i_h$ , где  $h$  пробегает значения от 1 до  $H$ ,  $i_0 = 0$ ,  $i_H = L$ .

Доказательство в целом аналогично доказательству Теоремы 4.2, поэтому остановимся только на различиях.

**Доказательство.** Рассмотрим выражение (4.8). Как и прежде, изменим порядок слагаемых, просуммировав их в порядке неубывания значений  $\xi(x_i)$ :

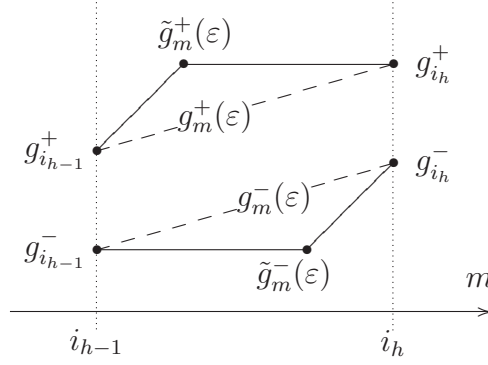
$$D(X) = \max_{z \in \mathbb{R}} \left| \frac{1}{k} \sum_{i=1}^L (1 - b_i) [\xi(x^{(i)}) < z] - \frac{1}{\ell} \sum_{i=1}^L b_i [\xi(x^{(i)}) < z] \right|.$$

Максимум достаточно брать не по всем  $z \in \mathbb{R}$ , а лишь по конечному множеству значений, которые функция  $\xi$  принимает на выборке,  $z \in \{\xi(x^{(1)}), \dots, \xi(x^{(H)})\}$ . Уберём сомножитель  $[\xi(x^{(i)}) < z]$ , заменив верхний предел суммирования  $L$  на  $m = \max\{i: \xi(x^{(i)}) < z\}$ . Заметим, что все объекты одной связки либо вместе входят, либо вместе не входят в сумму по  $i$ . Поэтому число  $m$  может принимать значения только из множества  $I_H = \{i_1, \dots, i_H\}$ :

$$D(X) = \max_{m \in I_H} \left| \frac{1}{k} \sum_{i=1}^m (1 - b_i) - \frac{1}{\ell} \sum_{i=1}^m b_i \right| = \frac{L}{\ell k} \max_{m \in I_H} \left| B_m - \frac{m\ell}{L} \right|.$$

Аналогично (4.9), получаем:

$$\mathbb{P}[D(X) \leq \varepsilon] = \mathbb{P} \prod_{m \in I_H} [g_m^-(\varepsilon) \leq B_m \leq g_m^+(\varepsilon)].$$



**Рис. 4.5.** Верхние  $\tilde{g}_m^+(\epsilon)$  и нижние  $\tilde{g}_m^-(\epsilon)$  границы усечённого треугольника Паскаля в сравнении с линейными границами  $g_m^-(\epsilon)$  и  $g_m^+(\epsilon)$  на отрезке  $m = \{i_{h-1}, \dots, i_h\}$ , соответствующем  $h$ -й связке.

Единственное отличие от (4.9) заключается в том, что прохождение допустимых траекторий  $(m, B_m)_{m=0}^L$  ограничено сверху  $g_m^+(\epsilon)$  и снизу  $g_m^-(\epsilon)$  не во всех точках  $m = 0, \dots, L$ , а только в точках  $m \in I_H$ , соответствующих концам связок.

Рассмотрим допустимые траектории на отрезке  $m = \{i_{h-1}, \dots, i_h\}$ , соответствующем  $h$ -й связке, см. рис. 4.5.

Между точками верхней границы  $(i_{h-1}, [g_{i_{h-1}}^+])$  и  $(i_h, [g_{i_h}^+])$  допустимая траектория может идти произвольным образом, следовательно, её путь ограничен сверху горизонтальной прямой  $B_m \leq g_{i_h}^+$  и наклонной прямой  $B_m \leq g_{i_{h-1}}^+ + (m - i_{h-1})$ .

Между точками нижней границы  $(i_{h-1}, [g_{i_{h-1}}^-])$  и  $(i_h, [g_{i_h}^-])$  допустимая траектория может идти произвольным образом, следовательно, её путь ограничен снизу горизонтальной прямой  $B_m \geq g_{i_{h-1}}^-$  и наклонной прямой  $B_m \geq g_{i_h}^- + (m - i_h)$ .

Таким образом, получены границы  $[\tilde{g}_m^-, \tilde{g}_m^+]$  усечённого треугольника Паскаля.

Теорема доказана. ■

**Замечание 4.1.** Если все связки одноэлементные,  $\{i_1, \dots, i_H\} \equiv \{1, \dots, L\}$ , то  $\tilde{g}_m^+(\epsilon) = g_m^+(\epsilon)$ ,  $\tilde{g}_m^-(\epsilon) = g_m^-(\epsilon)$ , и Теорема 4.3 переходит в Теорему 4.2.

**Замечание 4.2.** Полученные оценки являются точными, но ненаблюдаемыми. Модифицированные границы  $[\tilde{g}_m^-, \tilde{g}_m^+]$  существенно зависят от последовательности  $i_1, \dots, i_H$ , которая строится по всей генеральной выборке  $\mathbb{X}$ ; её невозможно знать, имея лишь наблюдаемую выборку  $X$ . Это означает, что Теорему 4.3 можно применять для проверки гипотезы однородности, однако непосредственно она не годится для предсказания эмпирической функции распределения.

## Резюме

Вероятность большого отклонения эмпирических распределений имеет асимптотическое выражение (4.2) или (4.3), которое приводится во многих учебниках и справочниках по статистике, как правило, без доказательств. В слабой аксиоматике она имеет точное выражение через усечённый треугольник Паскаля и легко обобщается

на случай вариационного ряда со связками, то есть на распределения с разрывами и дискретные распределения.

**В следующей лекции** мы вернёмся к проблеме переобучения и займёмся оцениванием вероятности большого равномерного отклонения частоты ошибок в двух выборках. Эта задача немного похожа на только что рассмотренную, но гораздо сложнее. Мы начнём с верхних оценок Вапника-Червоненкиса, которые исторически были первыми. Они легко выводятся, но сильно завышены. Проблема завышенности надолго займёт наше внимание, и только в лекции 14 мы снова вспомним про критерий Смирнова и покажем, что для одного частного случая вероятность большого равномерного отклонения всё-таки выражается через случайные блуждания и усечённый треугольник Паскаля.

## 5 Теория Вапника–Червоненкиса

Статистическая теория восстановления зависимостей по эмпирическим данным (VC-теория) была предложена В. Н. Вапником и А. Я. Червоненкисом в конце 60-х — начале 70-х годов [8, 9, 10, 7]. В середине 80-х она получила широкую мировую известность [104, 105, 106] и вместе с работами Валианта [103] на многие годы определила генеральное направление развития теории статистического обучения.

Рассмотрим основные предположения и результаты VC-теории в рамках слабой вероятностной аксиоматики, используя обозначения предыдущей главы.

### §5.1 Оценка Вапника–Червоненкиса

Напомним, что  $\mathbb{X} = \{x_1, \dots, x_L\}$  — конечное множество объектов,  $A$  — множество алгоритмов,  $I(a, x)$  — индикатор ошибки. Каждый алгоритм  $a \in A$  порождает на  $\mathbb{X}$  вектор ошибок  $\vec{a} = (I(a, x_i))_{i=1}^L$ . Введём ещё несколько понятий и обозначений.

$\vec{A} = \{\vec{a} : a \in A\}$  — множество векторов ошибок, порождаемых множеством алгоритмов  $A$  на заданной выборке  $\mathbb{X}$ . В дальнейшем мы часто будем предполагать, что множество  $A$  изначально состоит из алгоритмов с попарно различными векторами ошибок и опускать стрелку, обозначающую вектор, поскольку в этом случае алгоритмы взаимно однозначно соответствуют векторам ошибок.

$\Delta^A(\mathbb{X}) = |\vec{A}|$  — коэффициент разнообразия (shattering coefficient)<sup>4</sup> множества алгоритмов  $A$  на выборке  $\mathbb{X}$ . Он не превышает мощности множества  $A$  и не превышает  $2^L$  — числа различных булевых векторов длины  $L$ . В задачах классификации на два класса он равен числу различных дихотомий (способов разделить выборку  $\mathbb{X}$  на два класса), реализуемых всевозможными алгоритмами из  $A$ .

$\Delta^A(L) = \max_{\mathbb{X}} \Delta^A(\mathbb{X})$  — функция роста (growth function) множества алгоритмов  $A$  [10, 106]. Максимум берётся по всевозможным выборкам  $\mathbb{X} \subset \mathcal{X}$  длины  $L$  из некоторого (как правило, бесконечного) множества допустимых объектов  $\mathcal{X}$ . Функция роста является мерой сложности множества алгоритмов  $A$ . В отличие от коэффициента разнообразия, она не зависит от выборки  $\mathbb{X}$ , восстанавливаемой зависимости  $y(x)$  и метода обучения  $\mu$ . Справедлива верхняя оценка  $\Delta^A(L) \leq 2^L$ .

**Принцип равномерной сходимости** вводится в VC-теории и многих последующих работах (см. обзоры [107, 57, 12]), чтобы получать верхние оценки вероятности переобучения, не зависящие от метода  $\mu$ .

Принцип равномерной сходимости заключается в том, чтобы заменить функционал  $Q_\varepsilon$  его верхней оценкой  $\tilde{Q}_\varepsilon$  — вероятностью большого равномерного отклонения

---

<sup>4</sup>В работах В. Н. Вапника и А. Я. Червоненкиса [9, 10, 7] коэффициент разнообразия назывался индексом системы событий. Алгоритм  $a$  индуцирует событие  $S_a = \{x \in \mathbb{X} \mid I(a, x) = 1\}$ . Семейство  $A$  индуцирует систему событий  $S = \{S_a \mid a \in A\}$ . Индекс системы событий  $S$  есть число различных подмножеств вида  $S_a \cap \mathbb{X}$ , где  $a$  пробегает всё множество  $A$ , что равносильно определению через  $|\vec{A}|$ . В англоязычных работах прижился термин shattering — число разбиений всеми возможными способами, буквально «вдребезги». Другой вариант перевода — «дробление» [39].

частот в двух подвыборках:

$$Q_\varepsilon \leq \tilde{Q}_\varepsilon = \mathbf{P} \left[ \max_{\vec{a} \in \vec{A}} \delta(a, X) \geq \varepsilon \right] = \mathbf{P} \max_{\vec{a} \in \vec{A}} [\delta(a, X) \geq \varepsilon], \quad (5.1)$$

где  $\delta(a, X) = \nu(a, \bar{X}) - \nu(a, X)$ . Отметим, что здесь достаточно взять максимум по множеству векторов ошибок  $\vec{A}$  вместо максимума по самому множеству  $A$ , поскольку для любых алгоритмов  $a, a' \in A$  из  $\vec{a} = \vec{a}'$  следует  $\delta(a, X) = \delta(a', X)$ .

Термин *сходимость* означает, что если  $\tilde{Q}_\varepsilon \rightarrow 0$  при  $\ell, k \rightarrow \infty$  то переобученность также сходится к нулю:  $\delta(a, X) \rightarrow 0$ . *Равномерность* означает, что величина  $\delta(a, X)$  сходится к нулю одновременно для всех алгоритмов  $a \in A$ , в том числе для алгоритма  $a = \mu X$ , какими бы ни были метод обучения  $\mu$  и выборка  $X$ .

**Основная теорема VC-теории.** Здесь приводится доказательство, существенно более краткое, чем в исходных работах [9, 10, 7].

**Теорема 5.1.** Для любых  $\mu, \mathbb{X}$  и  $\varepsilon \in [0, 1]$  справедлива оценка

$$Q_\varepsilon \leq \Delta^A(\mathbb{X}) \max_{m=1, \dots, L} H_L^{\ell, m}(s_m(\varepsilon)), \quad s_m(\varepsilon) = \frac{\ell}{L}(m - \varepsilon k). \quad (5.2)$$

**Доказательство.** Воспользуемся принципом равномерной сходимости (5.1) и оценим сверху максимум бинарных величин их суммой<sup>5</sup>:

$$Q_\varepsilon \leq \tilde{Q}_\varepsilon \leq \mathbf{P} \sum_{\vec{a} \in \vec{A}} [\delta(a, X) \geq \varepsilon].$$

Теперь переставим местами знаки суммирования:  $\mathbf{P} \sum = \sum \mathbf{P}$  и запишем вероятность  $\mathbf{P}[\delta(a, X) \geq \varepsilon]$  для отдельного алгоритма  $a$  через гипергеометрическое распределение, согласно Теореме 2.4:

$$Q_\varepsilon \leq \tilde{Q}_\varepsilon \leq \tilde{\tilde{Q}}_\varepsilon = \sum_{\vec{a} \in \vec{A}} \mathbf{P}[\delta(a, X) \geq \varepsilon] = \sum_{\vec{a} \in \vec{A}} H_L^{\ell, m}(s_m(\varepsilon)), \quad m = n(a, \mathbb{X}). \quad (5.3)$$

Оценим  $H_L^{\ell, m}(s_m(\varepsilon))$  сверху максимумом по  $m$  и вынесем его за знак суммы. Под знаком суммы останется единица, а вся сумма будет равна  $|\vec{A}| = \Delta^A(\mathbb{X})$ . ■

Полностью аналогично доказывается верхняя оценка и для функционала  $R_\varepsilon$ .

**Следствие 5.1.1.** Для любых  $\mu, \mathbb{X}$  и  $\varepsilon \in [0, 1]$  справедлива оценка

$$R_\varepsilon \leq \sum_{\vec{a} \in \vec{A}} H_L^{\ell, m}(m - \varepsilon k) \leq \Delta^A(\mathbb{X}) \max_{m=1, \dots, L} H_L^{\ell, m}(m - \varepsilon k).$$

---

<sup>5</sup>Это можно трактовать и как оценку вероятности объединения событий  $[\delta(a, X) \geq \varepsilon]$  сверху суммой их вероятностей. Её называют также *неравенством Буля* или union bound.

Таким образом, имеется серия *VC-оценок*.

Наиболее точная оценка (5.3) — это сумма вероятностей переобучения по всем алгоритмам семейства с попарно различными векторами ошибок.

Оценка (5.2) чуть хуже — это вероятность переобучения наихудшего алгоритма (максимум  $H_L^{\ell, m}(s_m(\varepsilon))$  достигается при  $m$  порядка  $L/2$ ), помноженная на число алгоритмов с попарно различными векторами ошибок.

Оценку можно ещё немного испортить, если заменить коэффициент разнообразия  $\Delta^A(\mathbb{X})$  функцией роста  $\Delta^A(L)$ . Зато теперь оценка не будет зависеть от выборки.

Наконец, функцию гипергеометрического распределения можно заменить экспоненциальной верхней оценкой, имеющей особо простой вид при  $\ell = k$  [7]:

$$\max_m H_L^{\ell, m}(s_m(\varepsilon)) \leq \frac{3}{2} e^{-\varepsilon^2 \ell}, \quad \ell = k. \quad (5.4)$$

В итоге получается наиболее известная из VC-оценок [106]:

**Следствие 5.1.2.** Для любых  $\mu, \mathbb{X}, \varepsilon \in [0, 1]$  при  $\ell = k$  справедлива оценка:

$$Q_\varepsilon \leq \Delta^A(2\ell) \cdot \frac{3}{2} e^{-\varepsilon^2 \ell}. \quad (5.5)$$

**Корректные методы обучения.** В VC-теории [7] отдельно рассматривается так называемая *детерминистская постановка задачи*<sup>6</sup>, когда метод  $\mu$  *корректен*, то есть  $n(\mu X, X) = 0$  на любой обучающей выборке  $X$ .

**Теорема 5.2.** Если метод  $\mu$  корректен, то для любых  $\mathbb{X}, \varepsilon \in [0, 1]$

$$Q_\varepsilon \leq \Delta^A(\mathbb{X}) \frac{C_{L-\lceil \varepsilon k \rceil}^\ell}{C_L^\ell} \leq \Delta^A(L) \left( \frac{k}{L} \right)^{\varepsilon k}. \quad (5.6)$$

**Доказательство.** Начало доказательства в точности повторяет доказательство Теоремы 5.1, но к выражению для вероятности переобучения алгоритма  $a$  добавляется условие корректности:

$$\mathbb{P}[\delta(a, X) \geq \varepsilon] [n(a, X) = 0] = [m \geq \varepsilon k] \mathbb{P}[n(a, X) = 0] = [m \geq \varepsilon k] h_L^{\ell, m}(0).$$

Подставим это выражение в (5.3) и, с учётом  $h_L^{\ell, m}(0) = C_{L-m}^\ell / C_L^\ell$ , получим

$$Q_\varepsilon \leq \tilde{Q}_\varepsilon = \sum_{\vec{a} \in \vec{A}} [m \geq \varepsilon k] \frac{C_{L-m}^\ell}{C_L^\ell}, \quad m = n(a, \mathbb{X}).$$

Максимум  $C_{L-m}^\ell / C_L^\ell$  достигается при наименьшем  $m = \lceil \varepsilon k \rceil$ .

С учётом оценки  $C_{L-m}^\ell / C_L^\ell \leq \left( \frac{k}{L} \right)^m$  получаем цепочку неравенств (5.6). ■

---

<sup>6</sup>В зарубежной литературе сложилась другая терминология. Детерминистскую постановку задачи называют *реализуемым обучением* (realizable learning), имея в виду, что с помощью семейства алгоритмов  $A$  возможно реализовать истинную зависимость  $y(x)$ . Общую постановку задачи называют *нерезализуемым* или *агностическим обучением* (agnostic learning), подчёркивая принципиальную невозможность знать, находится ли истинная зависимость в семействе  $A$ , или нет.

**Следствие 5.2.1.** Если метод  $\mu$  корректен, то для любых  $\mathbb{X}$ ,  $\varepsilon \in [0, 1]$  при  $\ell = k$

$$Q_\varepsilon \leq \Delta^A(2\ell) \cdot 2^{-\varepsilon k}.$$

Итак, в случае корректности оценка  $Q_\varepsilon$  становится более точной и принимает наиболее простой вид. Отсюда совершенно не следует, что на практике надо пользоваться корректными методами обучения. Для обеспечения корректности придётся усложнять конструкцию семейства алгоритмов, что может привести к увеличению функции роста  $\Delta^A(L)$ , настолько значительному, что оно полностью скомпенсирует уменьшение комбинаторного множителя. Поэтому в VC-теории принято считать, что не следует добиваться безошибочной работы алгоритма на обучающем материале.

## §5.2 Размерность Вапника–Червоненкиса

Функция роста является мерой сложности множества  $A$  и не зависит от выборки и метода обучения. С ней тесно связана ещё одна характеристика сложности.

**Определение 5.1.** Если существует целое число  $h$  такое, что  $\Delta^A(h) = 2^h$  и  $\Delta^A(h+1) < 2^{h+1}$ , то оно называется ёмкостью или размерностью Вапника–Червоненкиса (VC-dimension) семейства алгоритмов  $A$ . Если такого числа  $h$  не существует, то говорят, что семейство  $A$  имеет бесконечную ёмкость.

Перечислим наиболее важные свойства ёмкости. Их доказательства будут даны в следующей лекции.

1. Если семейство имеет ёмкость  $h$ , то вместо тривиальной экспоненциальной оценки  $\Delta^A(L) \leq 2^L$  справедлива более точная полиномиальная оценка

$$\Delta^A(L) \leq C_L^0 + C_L^1 + \dots + C_L^h \leq \frac{3L^h}{2h!}. \quad (5.7)$$

В этом случае вероятность переобучения стремится к нулю при  $\ell \rightarrow \infty$ , поскольку гипергеометрический множитель стремится к нулю экспоненциально. Тогда говорят, что семейство  $A$  обладает свойством *обучаемости* (learnability).

2. Пусть  $X = \mathbb{R}^n$  и  $Y = \{-1, 1\}$ . Ёмкость семейства линейных классификаторов  $A = \{a(x) = \text{sign} \langle w, x \rangle \mid w \in \mathbb{R}^n\}$  равна размерности пространства  $n$ , где  $w$  — вектор параметров,  $\langle w, x \rangle$  — скалярное произведение векторов  $w$  и  $x$ . Этот факт является прямым следствием того, что в пространстве размерности  $n$  через любые  $n$  точек возможно провести гиперплоскость, а через любые  $n+1$  уже нельзя.

3. Далеко не всегда ёмкость равна числу параметров. Для любого параметрического семейства заданной ёмкости  $h$  легко построить другое семейство, которое будет иметь на один параметр больше, но при этом его ёмкость также будет равна  $h$ . Например, можно заменить один из параметров суммой двух новых параметров, что не повлияет на число порождаемых семейством векторов ошибок.

4. Пусть множество  $A$  конечно. Число алгоритмов, попарно неразличимых на выборке  $X^L$ , не превышает числа всех алгоритмов, поэтому для функции роста



справедлива оценка  $\Delta^A(L) \leq |A|$ . Ёмкость такого семейства не превышает  $\lfloor \log_2 |A| \rfloor$ , так как в противном случае функция роста оказалась бы больше  $|A|$ .

5. Множества алгоритмов, реализуемых на компьютере, всегда конечны. Если для хранения всех параметров алгоритма используется не более  $n$  бит, то число алгоритмов в таком семействе не превышает  $2^n$ , а его ёмкость не превышает  $\log_2 2^n = n$ . Чтобы эта оценка не была завышенной, для подсчёта необходимого числа бит должно использоваться *максимально экономное кодирование* параметров. Данное наблюдение лежит в основе принципа *минимума длины описания* (Minimal Description Length, MDL) [92] и принципа программируемой VC-размерности [20].

6. Существуют семейства алгоритмов классификации с одним вещественным параметром, ёмкость которых бесконечна. Это свидетельствует о нетривиальности понятия ёмкости с одной стороны, и о бесконечных выразительных способностях действительного числа с другой. С третьей стороны, все такие примеры являются искусственными и «разрушаются», как только мы возьмём вместо действительного числа его конечное представление в компьютере.

### §5.3 Метод структурной минимизации риска

Имея верхнюю оценку вероятности переобучения, нетрудно получить верхнюю оценку частоты ошибок на контрольной выборке. Она будет верна не всегда, а лишь с некоторой вероятностью, достаточно близкой к единице.

**Теорема 5.3.** При  $\ell = k$  для любого распределения на множестве  $\mathbb{X}$  с вероятностью не менее  $1 - \eta$  одновременно для всех алгоритмов  $a \in A$  справедливо неравенство

$$\nu(a, \bar{X}) < \nu(a, X) + \sqrt{\frac{h}{\ell} \ln \left( \frac{2e\ell}{h} \right) + \frac{4}{9\ell} \ln \frac{1}{\eta}}. \quad (5.8)$$

**Доказательство.** Подставим в (5.5) верхнюю оценку функции роста (6.1), оценив  $h!$  снизу по формуле Стирлинга. Полученная оценка имеет вид  $Q_\varepsilon \leq \eta(\varepsilon, \ell, h)$ . Применим к ней технику обращения (1.8): выразим  $\varepsilon$  как функцию от ёмкости  $h$ , длины обучения  $\ell$  и значения  $\eta$ . В результате получим (5.8). ■

Первое слагаемое в этой оценке — эмпирический риск. Он не возрастает с ростом ёмкости  $h$ , поскольку чем больше в семействе алгоритмов, тем более точно можно аппроксимировать выборку. Второе слагаемое возрастает с ростом ёмкости, и его можно рассматривать как *штраф за сложность* (complexity penalty). Сумма в общем случае достигает минимума при некотором  $h$ .

Для определения оптимальной сложности модели в VC-теории предлагается метод *структурной минимизации риска*. В семействе  $A$  заранее задаётся *структура* вложенных подсемейств возрастающей ёмкости  $A_1 \subset A_2 \subset \dots \subset A_h = A$ . Задача обучения решается в каждом из этих подсемейств, всего  $h$  раз. Выбирается подсемейство оптимальной ёмкости, для которого достигается минимум правой части (5.8).

Метод структурной минимизации риска является важнейшим конструктивным следствием VC-теории. Проследим ещё раз логику всего, что было проделано.

Сначала мы получили верхнюю оценку вероятности переобучения, которая справедлива для любой выборки  $X$  и любого метода обучения  $\mu$ . К этой оценке мы применили технику обращения и получили верхнюю оценку для частоты ошибок на контроле. Она справедлива для любых  $X, \bar{X}, \mu$ , с вероятностью  $(1 - \eta)$ , близкой к единице. Теперь распорядимся свободой выбора  $\mu$  так, чтобы минимизировать частоту ошибок на контроле. Полученный метод обучения называется *минимизацией оштрафованного эмпирического риска* (penalized empirical risk minimization).

Эта же общая логика эксплуатируется большинством современных подходов в теории статистического обучения [57].

## §5.4 Проблема завышенности VC-оценок

**Эксперимент 1: численные оценки требуемой длины обучения.** Основная проблема VC-оценок в том, что они чрезвычайно завышены — настолько, что их применение практически теряет смысл. Чтобы в этом убедиться, достаточно выполнить численный расчёт требуемой длины обучающей выборки  $\ell$  как функции от ёмкости  $h$ , точности  $\varepsilon$  и вероятности переобучения  $Q_\varepsilon$ .

Результаты приведены в Таблицах 5.1, 5.2, 5.3. Все данные получены при  $\ell = k$ . Первые две таблицы построены для общего случая, третья — для детерминистской постановки задачи. При построении первой таблицы использованы завышенные аппроксимации функции роста (6.1) и гипергеометрического множителя (5.4).

Основные выводы следующие.

1. Даже в детерминистском случае требуемая длина обучения на несколько порядков превышает те характерные длины выборок, с которыми обычно приходится иметь дело в прикладных задачах. Практика показывает, что хорошая обучаемость возможна при существенно меньших длинах выборки.

2. Оценки в Таблице 5.1 существенно хуже, чем в таблицах 5.2 и 5.3. В дальнейшем мы будем избегать использования завышенных аппроксимаций и стремиться к получению эффективных вычислительных методов, а не компактных формул<sup>7</sup>.

3. Правые половины таблиц соответствуют значению  $\eta = 1$  и показывают границу применимости VC-оценок. При меньших  $\ell$  верхняя оценка вероятности вырождается — становится больше 1. Сопоставление правой и левой половин таблиц показывает, что достаточная длина обучения существенно зависит от ёмкости  $h$  и точности  $\varepsilon$ , но слабо зависит от требуемой вероятности переобучения  $\eta$ .

4. Первая строка таблицы соответствует семейству из одного алгоритма,  $h = 0$ . При этом достигается наилучшая возможная оценка. Однако этот случай не интересен с точки зрения статистического обучения.

5. Учёт априорной информации о корректности метода обучения уточняет VC-оценки, но, судя по доказательствам, не устраняет ни одного из основных факторов завышенности.

---

<sup>7</sup>В наши дни стремление получать простые асимптотические формулы является скорее пережитком докомпьютерной эпохи, чем насущной необходимостью. Мнение спорное, поэтому в сноске.

$h$	$\eta = 0.01$				$\eta = 1$			
	$\varepsilon = 0.01$	$\varepsilon = 0.05$	$\varepsilon = 0.1$	$\varepsilon = 0.2$	$\varepsilon = 0.01$	$\varepsilon = 0.05$	$\varepsilon = 0.1$	$\varepsilon = 0.2$
0	60106	2404	601	150	14054	562	140	35
2	314692	9813	2149	460	265220	7786	1634	328
5	715120	21605	4631	961	665470	19565	4111	827
10	1386763	41427	8808	1806	1337061	39382	8287	1671
20	2733709	81218	17200	3504	2683987	79171	16677	3369
50	6780774	200844	42438	8616	6731042	198797	41916	8481
100	13530370	400406	84550	17149	13480635	398359	84027	17014

**Таблица 5.1.** Зависимость достаточной длины обучения  $\ell$  от ёмкости  $h$ , точности  $\varepsilon$  и надёжности  $\eta$ , вычисленная согласно оценке  $Q_\varepsilon \leq \eta = \frac{3}{2} \frac{L^h}{h!} \cdot \frac{3}{2} \exp(-\varepsilon^2 \ell)$  из Теоремы 5.1. Это наименее точная оценка, использующая аппроксимацию функции роста (6.1) и аппроксимацию гипергеометрического сомножителя (5.4).

$h$	$\eta = 0.01$				$\eta = 1$			
	$\varepsilon = 0.01$	$\varepsilon = 0.05$	$\varepsilon = 0.1$	$\varepsilon = 0.2$	$\varepsilon = 0.01$	$\varepsilon = 0.05$	$\varepsilon = 0.1$	$\varepsilon = 0.2$
0	35900	1440	360	91	506	20	10	5
2	259300	7619	1600	316	210035	5579	1089	186
5	632633	18260	3770	741	582841	16219	3250	610
10	1262928	36396	7521	1470	1213200	34320	6989	1335
20	2531001	72918	15069	2936	2481120	70820	14549	2805
50	6348132	182980	37821	7381	6298001	180900	37290	7250
100	7373100	295440	73821	14811	7373100	295440	73821	14671

**Таблица 5.2.** Зависимость достаточной длины обучения  $\ell$  от ёмкости  $h$ , точности  $\varepsilon$  и надёжности  $\eta$ , согласно оценке  $Q_\varepsilon \leq \eta = (C_L^0 + \dots + C_L^h) \Gamma_L^\ell(\varepsilon, 1)$ . Это также оценка Теоремы 5.1, но функция роста и гипергеометрический сомножитель вычисляются по точным формулам, без применения аппроксимаций.

$h$	$\eta = 0.01$				$\eta = 1$			
	$\varepsilon = 0.01$	$\varepsilon = 0.05$	$\varepsilon = 0.1$	$\varepsilon = 0.2$	$\varepsilon = 0.01$	$\varepsilon = 0.05$	$\varepsilon = 0.1$	$\varepsilon = 0.2$
0	800	140	70	35	100	20	10	5
2	2800	440	200	85	2000	300	120	45
5	6200	960	410	165	5400	800	330	125
10	11900	1820	770	310	11200	1660	690	270
20	23500	3560	1500	600	22700	3400	1420	555
50	58100	8780	3700	1465	57400	8620	3620	1425
100	107000	17480	7370	2915	107000	17320	7290	2875

**Таблица 5.3.** Зависимость достаточной длины обучения от ёмкости  $h$ , точности  $\varepsilon$  и надёжности  $\eta$ , по детерминистской оценке  $Q_\varepsilon \leq \eta = (C_L^0 + \dots + C_L^h) C_{L-\lceil \varepsilon k \rceil}^\ell / C_L^\ell$  из Теоремы 5.2, без применений аппроксимаций.

Причины завышенности VC-оценок видны из доказательства Теоремы 5.1, в котором сделаны три оценки сверху, а при выводе Следствия 5.1.2 — ещё две. Классическая VC-оценка (5.5) не зависит от конкретной выборки  $\mathbb{X}$  и метода обучения  $\mu$ , следовательно, является оценкой худшего случая (worst case bound), который, скорее всего, никогда не реализуется на практике.

**Эксперимент 2: оценки факторов завышенности.** Чтобы понять, во сколько раз увеличивается оценка на каждом знаке  $\leq$ , был проведён эксперимент [108] с логическими алгоритмами классификации на шести реальных задачах из репозитория UCI. Опуская технические детали, приведём лишь основные идеи постановки этого эксперимента и его результаты.

В каждой задаче методом Монте-Карло вычислялась оценка вероятности переобучения  $\hat{Q}_\varepsilon$ . Поделив её на функцию гипергеометрического распределения, можно получить *эффективный локальный коэффициент разнообразия* (ЭЛКР)  $\hat{\Delta}$ , показывающий, каким должно было бы быть значение коэффициента разнообразия  $\Delta^A(\mathbb{X})$ , чтобы оценка вероятности переобучения не была завышенной.

Наряду с ЭЛКР оценивались также *факторы завышенности* — коэффициенты  $r_1, r_2, r_3$ , каждый из которых является эмпирической оценкой отношения двух выражений, стоящих справа и слева от одного из знаков  $\leq$  в цепочке неравенств. Произведение факторов завышенности даёт общую завышенность VC-оценки:

$$\hat{Q}_\varepsilon \cdot r_1 \cdot r_2 \cdot r_3 = \Delta^A(L) \cdot \frac{3}{2} e^{-\varepsilon \ell^2}.$$

**Фактор**  $r_1$  оценивает завышенность неравенства (5.1), связанного с принципом равномерной сходимости. Эта завышенность имеет две составляющие — эффект локализации и эффект расслоения семейства алгоритмов.

*Эффект локализации* состоит в том, что некоторые алгоритмы из  $A$  могут не выдаваться методом обучения  $\mu$  ни на одной из обучающих подвыборок. Вместо  $A$  следовало бы рассматривать множество алгоритмов  $A_L^\ell = \{\mu X : X \subset \mathbb{X}, |X| = \ell\}$ , индуцируемых методом обучения  $\mu$  на всевозможных обучающих подвыборках  $X$  фиксированной длины  $\ell$ .

*Эффект расслоения* состоит в том, что на множестве алгоритмов  $A$  возникает существенно неравномерное распределение вероятности  $P(a) = \mathbb{P}[\mu X = a]$ . Алгоритмы с меньшим числом ошибок на генеральной выборке  $n(a, \mathbb{X})$  выдаются существенно чаще, в то же время, их существенно меньше. Эксперименты [82, 81] показывают, что основная масса алгоритмов, как правило, сосредоточена в области наилучшей частоты ошибок  $\nu(a, \mathbb{X}) \approx 0.5$ .

Эффект локализации можно считать предельным частным случаем расслоения, когда вероятность получения части алгоритмов в точности равна нулю.

**Фактор**  $r_2$  оценивает завышенность *неравенства Буля* (5.3). Она особенно велика, когда среди векторов ошибок имеется много похожих. Заметим, что неравенство

$$\mathbb{P} \max_a [\delta(a, X) \geq \varepsilon] \leq \sum_a \mathbb{P}[\delta(a, X) \geq \varepsilon]$$

Задача	класс	объектов	$r_1$	$r_2$	$r_3$	$\hat{\Delta}$	$[\hat{\Delta}_1; \hat{\Delta}_2]$
cтx	0	307	2 759	680	32.6	24	[10; 41]
	1	383	1 104	1700	11.6	12	[11; 180]
german	1	300	15 215	1500	10.9	54	[38; 530]
	2	700	44 400	9000	9.9	1.9	[1.0; 2.2]
hepatitis	0	32	308	280	9.5	83	[11; 148]
	1	123	132	680	22.5	15	[12; 27]
horse-colic	1	191	151	4500	7.2	7	[2; 9]
	2	109	504	3400	7.3	6	[3; 6]
hypothyroid	0	3012	1 964 200	400	16.5	21	[3; 220]
	1	151	581 400	460	28.7	30	[2; 44]
promoters	0	53	555	340	9.8	72	[36; 230]
	1	53	510	790	6.9	18	[9; 22]

**Таблица 5.4.** Факторы завышенности  $r_1, r_2, r_3$  и оценка  $\hat{\Delta}$  с доверительным интервалом  $[\hat{\Delta}_1; \hat{\Delta}_2]$ .

обрацается в равенство только когда события  $[\delta(a, X) \geq \varepsilon]$  несовместны. Если векторы ошибок алгоритмов  $a, a'$  схожи, то соответствующие им события будут существенно совместными. На практике часто применяются *связные семейства* алгоритмов, в которых для каждого алгоритма  $a \in A$  найдутся другие алгоритмы  $a' \in A$  такие, что векторы ошибок  $\vec{a}$  и  $\vec{a}'$  отличаются только на одном объекте [100]. Связные семейства порождаются, в частности, методами классификации с непрерывной по параметрам разделяющей поверхностью. К ним относятся линейные классификаторы, машины опорных векторов с непрерывными ядрами, нейронные сети с непрерывными функциями активации, решающие деревья с пороговыми условиями ветвления, и многие другие. Таким образом, неравенство Буля сильнее всего завышено как раз в наиболее интересных с практической точки зрения случаях.

**Фактор  $r_3$**  оценивает совокупную завышенность, возникающую в результате максимизации функции гипергеометрического распределения по  $m$  и использования его экспоненциальной аппроксимации.

Результаты представлены в Таблице 5.4. Поскольку логические закономерности строятся отдельно по классам, каждой задаче соответствует две строки таблицы (по числу классов). Основные выводы следующие.

1. Экспериментальные значения ЭЛКР имеют порядок  $10^0 \dots 10^2$ , тогда как функция роста находится в пределах  $10^5 \dots 10^{10}$ , то есть VC-оценки завышены на несколько порядков.

2. Наиболее существенны факторы завышенности  $r_1$  и  $r_2$ , связанные с эффектами расслоения и связности. Фактор  $r_3$  наименее значим.

3. Оценка ЭЛКР практически никогда не превышает длину выборки. Это означает, что если бы мы захотели определить понятие *эффективной локальной ёмкости*, то она не превышала бы единицы, то есть была бы вырождена. Отсюда возникают сомнения в состоятельности и полезности понятия ёмкости для оценок переобучения.

**В методе структурной минимизации риска** завышенность оценок (5.2) и (5.8) приводит к выбору подсемейства заниженной ёмкости, то есть к переупрощению алгоритмов, что подтверждается и в экспериментах на модельных данных [75]. Распространённой ошибкой в интерпретации результатов VC-теории является вывод о необходимости ограничивать сложность семейства алгоритмов. Такой вывод был бы справедлив, если бы VC-оценки были достаточно точными. Пример алгоритма *бустинга* [67] показывает, что обобщающая способность может улучшаться даже при практически неограниченном росте сложности семейства.

**О скользящем контроле.** При практическом применении структурной минимизации риска вместо завышенной теоретической оценки (5.8) часто рекомендуют применять оценку скользящего контроля  $\hat{E}\nu(a, \bar{X})$ . Однако такая замена ставит под сомнение ценность теоретических результатов, поскольку скользящий контроль не опирается на VC-теорию. С практической точки зрения скользящий контроль имеет ряд существенных недостатков. Во-первых, это ресурсоёмкая процедура. Во-вторых, оценка скользящего контроля имеет большую дисперсию, что может приводить к ошибкам при оптимизации сложности. В-третьих, скользящий контроль удобен для эмпирического оценивания качества метода обучения, но не удобен для конструирования новых методов. По этим причинам задача получения точных теоретических оценок не теряет актуальности.

**Когда оценка равномерной сходимости всё-таки не завышена.** Следующая теорема показывает, что в некоторых случаях неравенство (5.1) обращается в равенство.

**Теорема 5.4.** Пусть метод  $\mu$  минимизирует эмпирический риск и все алгоритмы  $\vec{a} \in \vec{A}$  имеют одинаковое число ошибок на генеральной выборке  $m = n(a, \mathbb{X})$ . Тогда верхняя оценка (5.1) обращается в равенство:  $Q_\varepsilon = \tilde{Q}_\varepsilon$ .

**Доказательство.** При фиксированном  $m$  минимизация эмпирического риска  $\nu(a, X)$  эквивалентна максимизации переобученности, поскольку

$$\delta(a, X) = \frac{m - \ell\nu(a, X)}{k} - \nu(a, X) = \frac{m}{k} - \frac{L}{k}\nu(a, X). \quad \blacksquare$$

Если же множество  $A$  расслаивается по уровням ошибок, то (5.1) может оказаться как точной, так и сильно завышенной верхней оценкой, что будет показано в дальнейшем на примерах. В общем случае требование равномерной сходимости является избыточно сильным и даёт лишь достаточное условие обучаемости.

## Резюме

VC-теория основана на принципе равномерной сходимости. Она позволяет оценивать вероятность переобучения сверху функцией от длины выборки и сложности семейства алгоритмов. Мерой сложности является коэффициент разнообразия, определяемый как число попарно различных бинарных векторов ошибок, индуцируемых всевозможными алгоритмами семейства на заданной выборке.

Конструктивным выходом VC-теории является принцип минимизации оштрафованного эмпирического риска и метод структурной минимизации риска. Вследствие завышенности оценок их непосредственное применение может приводить к неоптимальному выбору слишком простых алгоритмов.

Основной причиной завышенности VC-оценок является пренебрежение эффектами расслоения и связности, которыми на практике обладают многие семейства алгоритмов. VC-оценки зависят только от размера матрицы ошибок и полностью игнорируют её содержимое. В дальнейшем мы займёмся комбинаторными оценками переобучения, в которых эта информация не теряется, благодаря чему они могут быть гораздо точнее.

**В следующей лекции** будут рассмотрены верхние оценки функции роста и ёмкости как для общего случая, так и для некоторых конкретных семейств алгоритмов. Казалось бы, это некоторое отступление от основной цели курса — получения точных оценок вероятности переобучения. Тем не менее, оценивание функции роста — это классическая комбинаторная задача, полезная для понимания внутренней структуры семейств алгоритмов.

## Упражнения

**Задача 5.1 (1).** Доказать Теорему 5.3, используя формулу Стирлинга.

**Задача 5.2 (2).** *Степенью некорректности* метода обучения  $\mu$  на выборке  $\mathbb{X}$  называется максимальная частота ошибок на всевозможных обучающих подвыборках:  $\sigma(\mu, \mathbb{X}) = \max_{X \in [\mathbb{X}]^\ell} \nu(\mu X, X)$ . Доказать, что для любых  $\mu, \mathbb{X}$  с ограниченной некорректностью  $\sigma(\mu, \mathbb{X}) \leq \sigma$  и любого  $\varepsilon \in [0, 1]$  справедлива оценка, обобщающая (5.2) и (5.6):

$$Q_\varepsilon \leq \Delta^A(\mathbb{X}) \max_{m \in M(\varepsilon, \sigma)} H_L^{\ell, m}(s_m(\varepsilon, \sigma)), \quad (5.9)$$

где  $M(\varepsilon, \sigma) = \{m: \varepsilon k \leq m \leq k + \sigma \ell\}$ ,  $s_m(\varepsilon, \sigma) = \min\{s_m(\varepsilon), \sigma \ell\}$ .

Построить графики зависимости отношения оценки (5.2) к оценке (5.9) от степени некорректности  $\sigma$ .

**Задача 5.3 (2).** Следуя [56], показать, что если множество векторов ошибок  $\vec{A}$  кластеризуется по расстоянию Хэмминга на  $S(r)$  кластерах радиуса  $r$  каждый, то

$$\mathbb{P}[\delta_\mu(X) \geq \varepsilon + \frac{r}{\ell}] \leq S(r) \cdot \max_{m=1, \dots, L} H_L^{\ell, m}(s_m(\varepsilon)).$$

## 6 Размерность Вапника-Червоненкиса

В теории Вапника-Червоненкиса доказывается, что функция роста  $\Delta^A(L)$  либо равна  $2^L$ , либо растёт полиномиально по  $L$ , причём промежуточных вариантов не существует. В полиномиальном случае вероятность переобучения стремится к нулю при  $L \rightarrow \infty$ , следовательно, обучение асимптотически состоятельно. Разберём этот фундаментальный результат более подробно.

### §6.1 Связь ёмкости с функцией роста

**Лемма 6.1.** Функция  $\Phi_L^h = C_L^0 + C_L^1 + \dots + C_L^h$ , определённая при целых  $h$  и  $L$ , таких, что  $0 \leq h \leq L$ , однозначно задаётся рекуррентными соотношениями

$$\Phi_L^0 = 1, \quad \Phi_L^L = 2^L, \quad \Phi_L^h = \Phi_{L-1}^h + \Phi_{L-1}^{h-1}, \quad 0 \leq h \leq L.$$

Доказательство следует из того, что биномиальные коэффициенты  $C_L^h$  определяются аналогичным рекуррентным соотношением  $C_L^h = C_{L-1}^h + C_{L-1}^{h-1}$  и отличаются только граничным условием  $C_L^L = 1$ .

**Лемма 6.2 (Вапник, Червоненкис [7]).** Если для любой подвыборки  $X^{h+1}$  из  $\mathbb{X}$  выполняется  $\Delta^A(X^{h+1}) < 2^{h+1}$ , то  $\Delta^A(\mathbb{X}) \leq \Phi_L^h$ .

**Доказательство.** Доказательство проведём индукцией по  $h$ .

При  $h = 0$  из того, что  $\Delta^A(x_i) < 2$  для всех  $x_i \in \mathbb{X}$  вытекает  $\Delta^A(X^L) = 1 = \Phi_L^0$ , следовательно, утверждение леммы справедливо. Предполагая, что оно справедливо для  $h - 1$ , покажем, что оно справедливо также и для  $h$  при всех  $L$ , больших  $h$ .

Для этого при фиксированном  $h$  применим индукцию по  $L$ .

При  $L = h + 1$  имеем  $\Delta^A(X^{h+1}) \leq 2^{h+1} - 1 = \Phi_{h+1}^h$ , значит утверждение леммы выполнено. Допустим теперь, что оно выполняется для  $\Delta^A(\mathbb{X})$ . Рассмотрим выборку  $\mathbb{X}' = \mathbb{X} \cup \{x_{L+1}\}$  и оценим сверху  $\Delta^A(\mathbb{X}')$ .

Будем говорить, что алгоритм  $a$  на заданной выборке  $U$  индуцирует подвыборку  $U'$ , если  $U' = \{x \in U : a(x) = 1\}$ . Рассмотрим множество всех подвыборок, индуцируемых на  $\mathbb{X}$  всеми алгоритмами семейства  $A$ . Будем различать подвыборки двух типов:

- 1) такие подвыборки  $X^r$  из  $\mathbb{X}$ , что алгоритмы семейства  $A$  индуцируют на  $\mathbb{X}'$  как  $X^r$ , так и  $(X^r, x_{L+1})$ ;
- 2) все остальные подвыборки.

Обозначим число подвыборок первого типа  $K_1$ , а второго типа  $K_2$ . Тогда

$$\begin{aligned} \Delta^A(\mathbb{X}) &= K_1 + K_2, \\ \Delta^A(\mathbb{X}') &= 2K_1 + K_2, \end{aligned}$$

следовательно,  $\Delta^A(\mathbb{X}') = \Delta^A(\mathbb{X}) + K_1$ .

Рассмотрим подмножество алгоритмов  $A'$ , индуцирующих на  $\mathbb{X}$  только подвыборки первого типа. Тогда  $K_1 = \Delta^{A'}(\mathbb{X})$ .



Имеется две возможности.

1. Допустим, найдётся подвыборка  $X^h \subseteq \mathbb{X}$  такая, что  $\Delta^{A'}(X^h) = 2^h$ . Это означает, что алгоритмы множества  $A'$  индуцируют на  $X^h$ , а значит и на  $(X^h, x_{L+1})$ , все возможные подвыборки  $X^r \subseteq X^h$ . По определению множества  $A'$  на  $(X^h, x_{L+1})$  индуцируются также все подвыборки вида  $(X^r, x_{L+1})$ . Следовательно

$$\Delta^{A'}(X^h, x_{L+1}) = 2^h + 2^h = 2^{h+1}.$$

Но тогда  $\Delta^A(X^h, x_{L+1}) = 2^{h+1}$ , что противоречит условию леммы.

2. Допустим теперь, что  $\Delta^{A'}(X^h) < 2^h$  для любой подвыборки  $X^h \subseteq \mathbb{X}$ . По предположению индукции отсюда следует  $\Delta^{A'}(\mathbb{X}) \leq \Phi_L^{h-1}$ . Таким образом

$$\Delta^A(\mathbb{X}') = \Delta^A(\mathbb{X}) + \Delta^{A'}(\mathbb{X}) \leq \Phi_L^h + \Phi_L^{h-1} = \Phi_{L+1}^h$$

Утверждение индукции доказано для  $L + 1$ . ■

**Лемма 6.3.** *Справедлива оценка  $\Phi_L^h \leq 1.5 \frac{L^h}{h!}$ ,  $0 \leq h \leq L$ .*

Доказательство является несложным техническим упражнением [7].

Напомним определение: ёмкость множества алгоритмов  $A$  — это такое целое число  $h$ , что  $\Delta^A(h) = 2^h$  и  $\Delta^A(h + 1) < 2^{h+1}$ . Если такого числа  $h$  не существует, то говорят, что  $A$  имеет бесконечную ёмкость.

**Теорема 6.4 (Вапник, Червоненкис [7]).** *Если семейство  $A$  имеет конечную ёмкость  $h$ , то при  $L > h$  функция роста  $\Delta^A(L)$  зависит от  $L$  полиномиально:*

$$\Delta^A(L) \leq \Phi_L^h \leq 1.5 \frac{L^h}{h!}. \tag{6.1}$$

**Доказательство.** Пусть  $L \leq h$ . Тогда из условия  $\Delta^A(h) = 2^h$  вытекает, что существует выборка длины  $L$ , на которой алгоритмы семейства  $A$  индуцируют все возможные подвыборки. Значит  $\Delta^A(L) = 2^L$ .

Пусть  $L \geq h$ . Возьмём произвольную выборку  $\mathbb{X}$ . Для неё выполнено условие леммы 6.2, так как  $\Delta^A(h + 1) < 2^{h+1}$ . Следовательно  $\Delta^A(\mathbb{X}) \leq \Phi_L^h$ , и в силу произвольности выборки  $\Delta^A(L) \leq \Phi_L^h$ .

Теорема доказана. ■

## §6.2 Функция роста множества конъюнкций

Для случая, когда объекты описываются дискретными признаками  $f_j: X \rightarrow D_j$ ,  $|D_j| < \infty$ , оценим функцию роста множества всех конъюнкций ранга не выше  $K$ :

$$A = \left\{ a(x) = \bigwedge_{j \in J} [f_j(x) = d_j] \mid J \subseteq \{1, \dots, n\}, |J| \leq K, d_j \in D_j \right\}.$$

Если  $J$  — произвольное подмножество индексов из  $\{1, \dots, n\}$ , то число конъюнкций ранга  $k$ , которые можно построить по признакам из  $J$ , есть

$$H_k(J) = \sum_{\substack{J' \subseteq J \\ |J'|=k}} \prod_{j \in J'} |D_j|.$$

Если множества  $D_j$  равномошны,  $|D_j| = d$ , то  $H_k(J) = C_{|J|}^k d^k$ . В общем случае величина  $H_k(J)$  легко вычисляется по рекуррентным соотношениям:

$$\begin{aligned} H_0(J) &= 1; \\ H_k(J) &= 0, \quad k > |J|; \\ H_k(J \cup \{j\}) &= H_k(J) + |D_j| H_{k-1}(J), \quad k < |J|, \quad j = 1, \dots, n. \end{aligned}$$

Функция роста оценивается сверху числом конъюнкций ранга не выше  $K$ , которые можно построить по всем  $n$  признакам:

$$\Delta^A(L) \leq \sum_{k=1}^K H_k\{1, \dots, n\}.$$

### §6.3 Ёмкость семейства линейных классификаторов

Пусть  $X = \mathbb{R}^n$ ,  $Y = \{0, 1\}$ ,  $A$  — семейство линейных классификаторов:

$$A = \{a(x) = [\langle w, x \rangle \geq 0] \mid w \in \mathbb{R}^n\},$$

где  $\langle w, x \rangle$  — скалярное произведение векторов  $w$  и  $x$ . Каждый алгоритм этого семейства задаётся вектором  $w$  из  $\mathbb{R}^n$ .

**Теорема 6.5.** Ёмкость семейства линейных классификаторов  $A$  равна размерности пространства  $n$ .

Идея доказательства заключается в том, что в пространстве размерности  $n$  через произвольные  $n$  точек можно провести разделяющую гиперплоскость, а через некоторые  $n + 1$  — уже нельзя.

**Доказательство.** Покажем сначала, что  $\Delta^A(n) = 2^n$ . Согласно определению функции роста это равносильно следующему высказыванию:

$$\exists X^n \quad \forall (z_1, \dots, z_n) \in Y^n \quad \exists a \in A \quad \forall i = 1, \dots, n \quad a(x_i) = z_i.$$

Возьмём  $n$  векторов  $X^n = \{x_1, \dots, x_n\}$  из  $X$  таких, что у  $i$ -ого вектора  $i$ -ая компонента равна 1, а остальные равны 0. Рассмотрим алгоритм  $a \in A$ , задаваемый вектором коэффициентов  $w = (w_1, \dots, w_n)$ . Каким бы ни был бинарный вектор  $(z_1, \dots, z_n)$ , легко подобрать коэффициенты  $w_i$  так, чтобы выполнялось  $a(x_i) = [w_i \geq 0] = z_i$ . Таким образом, мы указали  $2^n$  алгоритмов, различным образом делящих выборку  $X^n$  на два класса.

Теперь покажем, что  $\Delta^A(n+1) < 2^{n+1}$ . Это равносильно высказыванию  
 $\forall X^{n+1} \exists(z_1, \dots, z_{n+1}) \in Y^{n+1} \forall a \in A \exists i = 1, \dots, n+1 \ a(x_i) \neq z_i$ .

Возьмём произвольные  $n+1$  векторов  $x_1, \dots, x_{n+1}$  из  $X$ . Число векторов превышает их размерность, поэтому среди них найдётся хотя бы один, являющийся линейной комбинацией остальных. Допустим без ограничения общности, что это  $x_{n+1}$ :

$$x_{n+1} = b_1 x_1 + \dots + b_n x_n, \quad (6.2)$$

где  $b_1, \dots, b_n$  — действительные числа.

Положим  $z_i = [b_i \geq 0]$  для всех  $i = 1, \dots, n$  и  $z_{n+1} = 0$ . Рассмотрим произвольный алгоритм  $a \in A$  с коэффициентами  $w = (w_1, \dots, w_n)$ . Допустим, что  $a(x_i) = z_i$  для всех  $i = 1, \dots, n+1$ . Умножим обе части равенства (6.2) скалярно на  $w$ :

$$\langle w, x_{n+1} \rangle = b_1 \langle w, x_1 \rangle + \dots + b_n \langle w, x_n \rangle.$$

Левая часть этого равенства строго меньше нуля, поскольку

$$[\langle w, x_{n+1} \rangle \geq 0] = a(x_{n+1}) = z_{n+1} = 0.$$

В то же время, каждое слагаемое в правой части равенства неотрицательно, так как

$$[\langle w, x_i \rangle \geq 0] = a(x_i) = z_i = [b_i \geq 0], \quad i = 1, \dots, n.$$

Таким образом, сделанное допущение приводит к противоречию. Какой бы ни была выборка  $X^{n+1}$ , алгоритмы из  $A$  не реализуют всех  $2^{n+1}$  способов поделить её на 2 класса.

Теорема доказана. ■

## §6.4 Однопараметрическое семейство бесконечной ёмкости

Существование однопараметрического семейства алгоритмов классификации, имеющего бесконечную ёмкость свидетельствует о нетривиальности понятия ёмкости с одной стороны, и о бесконечных выразительных способностях действительного числа с другой [105].

Рассмотрим семейство функций  $a: \mathbb{R} \rightarrow \{0, 1\}$  с одним параметром  $\gamma \in \mathbb{R}$ :

$$a(x; \gamma) = [\sin(\gamma x) < 0].$$

Возьмём конкретную выборку объектов  $x_i = 10^{-i}$ ,  $i = 1, \dots, \ell$ . Какова бы ни была её длина  $\ell$ , для любого вектора ответов  $(y_i)_{i=1}^{\ell}$  можно так подобрать параметр  $\gamma$ , чтобы  $a(x_i; \gamma) = y_i$ . Действительно, возьмём  $\gamma = \pi + \pi \sum_{j=1}^{\ell} y_j 10^j$ . Тогда

$$a(x_i; \gamma) = \left[ \sin \left( \pi y_i + \underbrace{\pi \sum_{j=1}^{i-1} y_j 10^{i-j}}_{\gamma_0} \right) < 0 \right] = \begin{cases} [\sin \gamma_0 < 0], & y_i = 0; \\ [\sin \gamma_0 > 0], & y_i = 1. \end{cases}$$

Из определения величины  $\gamma_0$  следует, что  $\pi 10^{-\ell} \leq \gamma_0 \leq 0.3\pi$ , поэтому значение  $\sin \gamma_0$  положительно, и правая часть равенства есть просто  $y_i$ .

Рассмотренный пример является искусственным. Если для представления числа  $\gamma$  использовать конечное число бит, ёмкость уже не будет бесконечной.

## §6.5 Другие оценки ёмкости

Оценки ёмкости были получены для нейронных сетей [50, 49, 74, 87], решающих деревьев [22], корректных полиномов над алгоритмами вычисления оценок [33], комитетных решающих правил [86], и других семейств.

Ёмкость семейств, основанных на явном хранении всей обучающей выборки, как правило, бесконечна (например, у алгоритма ближайших соседей). Ёмкость семейств, гарантирующих корректность (отсутствие ошибок) на обучающей выборке, также, как правило, бесконечна. Хотя, есть и исключения: в работах В. Л. Матросова строятся композиции алгоритмов вычисления оценок, имеющие конечную ёмкость и одновременно гарантирующие корректность [32, 33, 34, 35].

### Резюме

Ёмкость или размерность Вапника-Червоненкиса — это классическая характеристика сложности семейств алгоритмов. Для линейных семейств она равна размерности пространства параметров. Для других семейств ёмкость может нетривиальным образом зависеть от числа параметров. Существуют примеры, когда параметров много, а ёмкость равна единице, и, наоборот, когда параметр один, а ёмкость бесконечна.

Любые оценки переобучения, выражающиеся через коэффициент разнообразия или ёмкость, основаны на принципе равномерной сходимости и неравенстве Буля, и потому сильно завышены. В лекции §8.3 мы покажем, что они сильно завышены даже в самом благоприятном случае, когда ёмкость семейства равна единице.

**В следующей лекции** мы рассмотрим базовую технику для получения комбинаторных оценок вероятности переобучения и убедимся, что знания одного лишь коэффициента разнообразия категорически недостаточно для получения точных оценок.

## 7 Порождающие и запрещающие множества

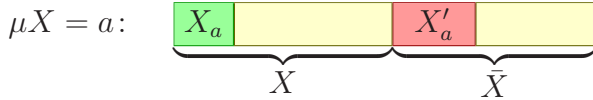
*Принцип порождающих и запрещающих множеств* (ПЗМ) позволяет получать точные (не завышенные, не асимптотические) оценки обобщающей способности [111]. Он основан на предположении, что для каждого алгоритма можно выписать необходимые и достаточные условия того, что он является результатом обучения. Если же удаётся выписать лишь необходимые условия, то получаются верхние оценки.

### §7.1 Простая гипотеза ПЗМ

**Гипотеза 7.1.** Пусть множество  $A$ , выборка  $\mathbb{X}$  и метод  $\mu$  таковы, что для каждого алгоритма  $a \in A$  можно указать пару непересекающихся подмножеств  $X_a \subset \mathbb{X}$  и  $X'_a \subset \mathbb{X}$ , удовлетворяющую условию

$$[\mu X = a] = [X_a \subseteq X][X'_a \subseteq \bar{X}], \quad \forall X \in [\mathbb{X}]^\ell. \quad (7.1)$$

Множество  $X_a$  будем называть *порождающим*, множество  $X'_a$  — *запрещающим* для алгоритма  $a$ . Гипотеза 7.1 означает, что метод  $\mu$  выбирает алгоритм  $a$  тогда и только тогда, когда в обучающей выборке  $X$  находятся все порождающие объекты и ни одного запрещающего:



Все остальные объекты  $\mathbb{X} \setminus X_a \setminus X'_a$  будем называть *нейтральными* для алгоритма  $a$ . Наличие или отсутствие нейтральных объектов в обучающей выборке не влияет на результат обучения. Далее будут приведены примеры семейств, для которых гипотеза 7.1 выполняется.

**Лемма 7.1.** Для любой выборки  $X$  справедливо тождество

$$\sum_{a \in A} [\mu X = a] = 1. \quad (7.2)$$

Доказательство с очевидностью вытекает из того, что для любой выборки  $X$  метод  $\mu$  выбирает один и только один алгоритм. Тождество (7.2) может использоваться для проверки того, что условия в правой части (7.1) сформулированы корректно.

Для произвольного  $a \in A$  обозначим через  $L_a$  число нейтральных объектов, через  $\ell_a$  — число нейтральных объектов, попадающих в обучающую выборку:

$$L_a = L - |X_a| - |X'_a|;$$

$$\ell_a = \ell - |X_a|.$$

**Лемма 7.2.** Если гипотеза 7.1 справедлива, то вероятность получить в результате обучения алгоритм  $a$  равна

$$P_a = \mathbb{P}[\mu X = a] = \frac{C_{L_a}^{\ell_a}}{C_L^\ell}. \quad (7.3)$$

**Доказательство.** Согласно гипотезе 7.1

$$P[\mu X = a] = P[X_a \subseteq X][X'_a \subseteq \bar{X}].$$

Это есть доля разбиений генеральной выборки  $\mathbb{X} = X \sqcup \bar{X}$  таких, что множество объектов  $X_a$  целиком лежит в  $X$ , а множество объектов  $X'_a$  целиком лежит в  $\bar{X}$ . Число таких разбиений равно числу способов отобрать  $\ell_a$  из  $L_a$  нейтральных объектов в обучающую подвыборку  $X \setminus X_a$ , которое, очевидно, равно  $C_{L_a}^{\ell_a}$ . Общее число разбиений равно  $C_L^\ell$ , а их отношение как раз и есть  $P_a$ . ■

Вероятность переобучения  $Q_\varepsilon$  выражается по формуле полной вероятности, если для каждого алгоритма  $a$  из  $A$  известна вероятность  $P_a$  получить его в результате обучения и условная вероятность большого отклонения частот  $P(\delta(a, X) \geq \varepsilon \mid a)$  при условии, что получен алгоритм  $a$ :

$$Q_\varepsilon = \sum_{a \in A} P_a P(\delta(a, X) \geq \varepsilon \mid a).$$

Условная вероятность даётся Теоремой 2.4, если учесть, что при фиксированном алгоритме  $a$  подмножества  $X_a$  и  $X'_a$  не участвуют в разбиениях. Рассматривая  $L_a$  нейтральных объектов и всевозможные их разбиения на  $\ell_a$  обучающих и  $L_a - \ell_a$  контрольных, получим:

$$P(\delta(a, X) \geq \varepsilon \mid a) = H_{L_a}^{\ell_a, m_a}(s_a(\varepsilon)),$$

где  $m_a$  — число ошибок алгоритма  $a$  на нейтральных объектах;  $s_a(\varepsilon)$  — наибольшее число ошибок алгоритма  $a$  на нейтральных обучающих объектах  $X \setminus X_a$ , при котором имеет место большое отклонение частот ошибок,  $\delta(a, X) \geq \varepsilon$ :

$$\begin{aligned} m_a &= n(a, \mathbb{X} \setminus X_a \setminus X'_a); \\ s_a(\varepsilon) &= \frac{\ell}{L}(n(a, \mathbb{X}) - \varepsilon k) - n(a, X_a). \end{aligned}$$

Пока это было не доказательство, а лишь наводящие соображения. Трюк с условной вероятностью может показаться не вполне очевидным. Ниже представлен строгий комбинаторный вывод точной оценки  $Q_\varepsilon$ .

**Теорема 7.3.** Если гипотеза 7.1 справедлива, то вероятность переобучения вычисляется по формуле

$$Q_\varepsilon(\mu, \mathbb{X}) = \sum_{a \in A} P_a H_{L_a}^{\ell_a, m_a}(s_a(\varepsilon)). \quad (7.4)$$

**Доказательство.** Рассмотрим функционал  $Q_\varepsilon$ . Введём в (1.5) под знак суммирования по  $X$  ещё два вспомогательных суммирования: первый — по всем алгоритмам  $a$  из  $A$  при условии  $\mu X = a$ , второй — по всем значениям  $s$  числа ошибок алгоритма  $a$  на подвыборке  $X \setminus X_a$ . Очевидно, значение  $Q_\varepsilon$  от этого не изменится:

$$Q_\varepsilon = P[\delta_\mu(X) \geq \varepsilon] = P \sum_{a \in A} [\mu X = a] \sum_{s=0}^{\ell_a} [n(a, X \setminus X_a) = s] [\delta(a, X) \geq \varepsilon]. \quad (7.5)$$

Число ошибок алгоритма  $a$  на обучающей подвыборке  $X$  равно  $s + n(a, X_a)$ , поэтому отклонение частот ошибок выражается в виде

$$\delta(a, X) = \frac{n(a, \mathbb{X}) - s - n(a, X_a)}{k} - \frac{s + n(a, X_a)}{\ell},$$

следовательно,

$$[\delta(a, X) \geq \varepsilon] = [s \leq \frac{\ell}{L}(n(a, \mathbb{X}) - \varepsilon k) - n(a, X_a)] = [s \leq s_a(\varepsilon)].$$

Подставим полученное выражение в (7.5), затем заменим  $[\mu X = a]$  правой частью равенства (7.1) и переставим знаки суммирования (очевидно,  $\mathbb{P}$  также можно рассматривать как суммирование):

$$Q_\varepsilon = \sum_{a \in A} \sum_{s=0}^{\ell_a} \underbrace{\mathbb{P}[X_a \subseteq X][X'_a \subseteq \bar{X}][n(a, X \setminus X_a) = s]}_{N(a)} [s \leq s_a(\varepsilon)]. \quad (7.6)$$

Выделенное в данной формуле выражение  $N(a)$  есть доля разбиений генеральной выборки  $\mathbb{X} = X \sqcup \bar{X}$  таких, что множество объектов  $X_a$  целиком лежит в  $X$ , множество объектов  $X'_a$  целиком лежит в  $\bar{X}$ , и в подвыборку  $X \setminus X_a$  длины  $\ell_a$  попадает ровно  $s$  объектов, на которых алгоритм  $a$  допускает ошибку.

Для наглядности представим вектор ошибок  $a$  разбитым на шесть блоков:

$$\vec{a} = \left( \underbrace{X_a; \overbrace{1, \dots, 1}^s; 0, \dots, 0}_{X \setminus X_a}; \underbrace{X'_a; \overbrace{1, \dots, 1}^{m_a - s}; 0, \dots, 0}_{\bar{X} \setminus X'_a} \right).$$

Число ошибок алгоритма  $a$  на объектах, не попадающих ни в  $X_a$ , ни в  $X'_a$ , равно  $m_a$ . Существует  $C_{m_a}^s$  способов выбрать из них  $s$  объектов, которые попадут в  $X \setminus X_a$ . Для каждого из этих способов имеется ровно  $C_{L_a - m_a}^{\ell_a - s}$  способов выбрать  $\ell_a - s$  объектов, на которых алгоритм  $a$  не допускает ошибку, и которые также попадут в  $X \setminus X_a$ . Тем самым однозначно определяется состав выборки  $X \setminus X_a$ , а, значит, и состав выборки  $\bar{X} \setminus X'_a$ . Таким образом,  $N(a) = C_{m_a}^s C_{L_a - m_a}^{\ell_a - s} / C_L^\ell$ . Подставим это выражение в (7.6) и выделим в нём формулу гипергеометрической функции вероятности:

$$Q_\varepsilon = \sum_{a \in A} \frac{C_{L_a}^{\ell_a}}{C_L^\ell} \sum_{s=s_0}^{\ell_a} [s \leq s_a(\varepsilon)] \frac{C_{m_a}^s C_{L_a - m_a}^{\ell_a - s}}{C_{L_a}^{\ell_a}} = \sum_{a \in A} P_a H_{L_a}^{\ell_a, m_a}(s_a(\varepsilon)).$$

Теорема доказана. ■

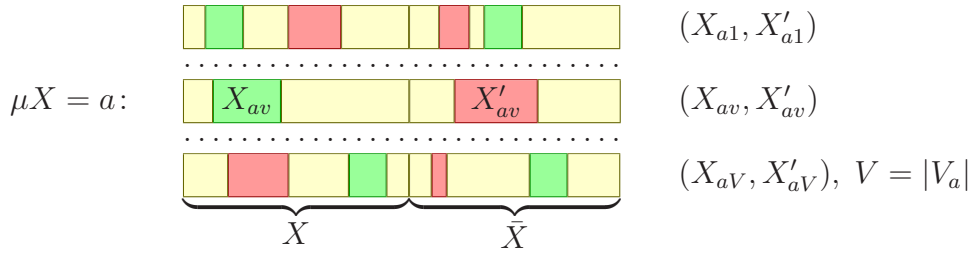
## §7.2 Обобщённая гипотеза ПЗМ

Гипотеза 7.1 накладывает слишком сильные ограничения на выборку  $\mathbb{X}$ , семейство  $A$  и метод  $\mu$ . Поэтому Теорему 7.3 удаётся применять лишь в некоторых специальных случаях. Рассмотрим естественное обобщение гипотезы 7.1. Предположим, что для каждого алгоритма  $a$  существуют различные варианты выделения порождающих и запрещающих множеств.

**Гипотеза 7.2.** Пусть множество  $A$ , выборка  $\mathbb{X}$  и метод  $\mu$  таковы, что для каждого алгоритма  $a \in A$  можно указать конечное множество индексов  $V_a$ , и для каждого индекса  $v \in V_a$  можно указать порождающее множество  $X_{av} \subset \mathbb{X}$ , запрещающее множество  $X'_{av} \subset \mathbb{X}$  и коэффициент  $c_{av} \in \mathbb{R}$ , удовлетворяющие условиям

$$[\mu X = a] = \sum_{v \in V_a} c_{av} [X_{av} \subseteq X] [X'_{av} \subseteq \bar{X}], \quad \forall X \in [\mathbb{X}]^\ell. \quad (7.7)$$

При условии  $c_{av} \equiv 1$  гипотеза 7.2 означает, что метод  $\mu$  выбирает алгоритм  $a$  тогда и только тогда, когда в обучающую выборку  $X$  попадают все объекты из  $X_{av}$  и ни одного из  $X'_{av}$ , ровно для одной из пар множеств  $(X_{av}, X'_{av})$ ,  $v \in V_a$ :



Очевидно, условие (7.7) должно быть задано так, чтобы правая часть принимала только два значения — либо 0, либо 1. Это требование накладывает определённые ограничения и на систему подмножеств  $(X_{av}, X'_{av})$ ,  $v \in V_a$ , и на коэффициенты  $c_{av}$ . В частности, при  $c_{av} \equiv 1$  условия  $[X_{av} \subseteq X] [X'_{av} \subseteq \bar{X}]$  не могут выполняться одновременно для двух индексов  $v, v' \in V_a$  ни при каком разбиении  $(X, \bar{X})$ , иначе значение в правой части окажется большим единицы.

К настоящему моменту не известны случаи, когда приходилось бы задавать коэффициенты  $c_{av}$ , отличные от +1 или -1.

Очевидно, гипотеза 7.1 является частным случаем гипотезы 7.2, когда все множества  $V_a$  одноэлементные и  $c_{av} = 1$ .

Следующая теорема утверждает, что гипотеза 7.2 верна всегда.

**Теорема 7.4.** Для любых  $\mathbb{X}$ ,  $A$  и  $\mu$  существуют множества  $V_a$ ,  $X_{av}$ ,  $X'_{av}$ , при которых справедливо представление (7.7), причём  $c_{av} = 1$  для всех  $a \in A$ ,  $v \in V_a$ .

**Доказательство.** Зафиксируем произвольный алгоритм  $a \in A$ . Возьмём в качестве индексного множества  $V_a$  множество всех подвыборок  $v \in [\mathbb{X}]^\ell$ , при которых  $\mu v = a$ . Для каждого  $v \in V_a$  положим  $X_{av} = v$ ,  $X'_{av} = \mathbb{X} \setminus v$ ,  $c_{av} = 1$ . Тогда для любого  $X \in [\mathbb{X}]^\ell$  справедливо представление, имеющее вид (7.7):

$$[\mu X = a] = \sum_{v \in V_a} [v = X] = \sum_{v \in V_a} [v = X] [\mathbb{X} \setminus v = \mathbb{X} \setminus X] = \sum_{v \in V_a} [v \subseteq X] [\mathbb{X} \setminus v \subseteq \bar{X}],$$

причём, если  $\mu X = a$ , то ровно одно слагаемое в этой сумме равно единице, остальные равны нулю; если же  $\mu X \neq a$ , то все слагаемые равны нулю. ■

Теорема 7.4 является типичной теоремой существования. Использованный при её доказательстве способ построения индексных множеств  $V_a$  требует явного перебора всех разбиений выборки, что приводит к вычислительно неэффективным оценкам



вероятности переобучения. Однако представление (7.7) в общем случае не единственно. Отдельной проблемой является поиск такого представления, в котором мощности множеств  $|V_a|$ ,  $|X_{av}|$ ,  $|X'_{av}|$  были бы как можно меньше. Хотя гипотеза 7.2 верна всегда, мы будем продолжать называть её «гипотезой», имея в виду предположение о существовании некоторого представления вида (7.7), более эффективного, чем использованное в доказательстве Теоремы 7.4.

Введём для каждого алгоритма  $a \in A$  и каждого индекса  $v \in V_a$  обозначения:

$$\begin{aligned} L_{av} &= L - |X_{av}| - |X'_{av}|; \\ \ell_{av} &= \ell - |X_{av}|; \\ m_{av} &= n(a, \mathbb{X}) - n(a, X_{av}) - n(a, X'_{av}); \\ s_{av}(\varepsilon) &= \frac{\ell}{L} (n(a, \mathbb{X}) - \varepsilon k) - n(a, X_{av}). \end{aligned}$$

В условиях гипотезы 7.2 справедливы соответствующие обобщения леммы о вероятностях получения алгоритмов и теоремы о вероятности переобучения.

**Лемма 7.5.** *Если гипотеза 7.2 справедлива, то для всех  $a \in A$  вероятность получить в результате обучения алгоритм  $a$  равна*

$$P_a = \mathbb{P}[\mu X = a] = \sum_{v \in V_a} c_{av} P_{av}; \quad (7.8)$$

$$P_{av} = \mathbb{P}[X_{av} \subseteq X][X'_{av} \subseteq \bar{X}] = \frac{C_{L_{av}}^{\ell_{av}}}{C_L^\ell}. \quad (7.9)$$

**Доказательство.** Достаточно применить операцию  $\mathbb{P}$  к левой и правой частям (7.7). Дальнейшие рассуждения аналогичны доказательству Леммы 7.2. ■

**Теорема 7.6.** *Если гипотеза 7.2 справедлива, то вероятность переобучения вычисляется по формуле*

$$Q_\varepsilon(\mu, \mathbb{X}) = \sum_{a \in A} \sum_{v \in V_a} c_{av} P_{av} H_{L_{av}}^{\ell_{av}, m_{av}}(s_{av}(\varepsilon)). \quad (7.10)$$

**Доказательство.** Аналогично доказательству Теоремы 7.3, вероятность переобучения приводится к выражению, которое отличается от (7.6) появлением знака суммирования по  $v$ , коэффициентов  $c_{av}$  и двойных индексов  $av$  вместо одинарных  $a$ :

$$Q_\varepsilon = \sum_{a \in A} \sum_{v \in V_a} \sum_{s=0}^{\ell} c_{av} \mathbb{P}[X_{av} \subseteq X][X'_{av} \subseteq \bar{X}][n(a, X \setminus X_{av}) = s][s \leq s_{av}(\varepsilon)],$$

В остальном доказательство аналогично доказательству Теоремы 7.3. ■

**Оценки функционала  $R_\varepsilon$**  (1.6) доказываются аналогично, с той лишь разницей, что выражение

$$s_{av}(\varepsilon) = \frac{\ell}{L} (n(a, \mathbb{X}) - \varepsilon k) - n(a, X_{av})$$

всюду заменяется на

$$s'_{av}(\varepsilon) = n(a, \mathbb{X}) - \varepsilon k - n(a, X_{av}).$$

### §7.3 Корректное семейство алгоритмов

Алгоритм  $a$ , не допускающий ошибок на выборке  $U \subseteq \mathbb{X}$ , называется *корректным на выборке  $U$* . Если множество  $A$  содержит алгоритм  $a_0$ , корректный на генеральной выборке  $\mathbb{X}$ , то множество  $A$  будем называть *корректным*. В этом случае формула (7.10) сильно упрощается.

**Лемма 7.7.** В случае  $m = 0$  функция гипергеометрического распределения вырождается:  $H_L^{\ell, 0}(s) = [s \geq 0]$ .

**Доказательство.** При  $s \geq 0$  сумма  $H_L^{\ell, 0}(s)$  в (2.4) состоит из одного слагаемого, равного 1. При  $s < 0$  число слагаемых равно нулю, и вся сумма равна нулю. ■

**Теорема 7.8.** Пусть гипотеза 7.2 справедлива, метод  $\mu$  является минимизацией эмпирического риска, множество  $A$  корректно. Тогда вероятность переобучения принимает более простой вид:

$$Q_\varepsilon(\mu, \mathbb{X}) = \sum_{a \in A} [n(a, \mathbb{X}) \geq \varepsilon k] P_a. \quad (7.11)$$

**Доказательство.** Рассмотрим произвольный алгоритм  $a \in A$  и произвольный индекс  $v \in V_a$ . Если некоторый объект, на котором  $a$  допускает ошибку, содержится в обучающей выборке  $X$ , то метод  $\mu$ , минимизирующий эмпирический риск, не сможет выбрать данный алгоритм, так как существует корректный алгоритм  $a_0$ , не допускающий ошибок на  $X$ . Следовательно, множество объектов, на которых алгоритм  $a$  допускает ошибку, целиком содержится в  $X'_{av}$ . Значит, алгоритм  $a$  не допускает ошибок на нейтральных объектах и  $m_{av} = 0$ . Тогда, согласно Лемме 7.7,

$$H_{L_{av}}^{\ell_{av}, 0}(s_{av}(\varepsilon)) = [s_{av}(\varepsilon) \geq 0] = \left[ \frac{\ell}{L} (n(a, \mathbb{X}) - \varepsilon k) - n(a, X_{av}) \geq 0 \right] = [n(a, \mathbb{X}) \geq \varepsilon k].$$

Подставляя это выражение в (7.10), получаем (7.11). ■

**Следствие 7.8.1.** Если в семействе  $A$  содержится алгоритм, корректный на всей генеральной выборке, то выражение (7.11) справедливо и для функционала  $R_\varepsilon(\mu, \mathbb{X})$ .

### §7.4 Функционал полного скользящего контроля

Рассмотрим функционал полного скользящего контроля (1.7). Принцип порождающих и запрещающих множеств также даёт для него точную оценку [14].

**Теорема 7.9.** Если гипотеза 7.2 справедлива, то оценка полного скользящего контроля вычисляется по формуле

$$C(\mu, \mathbb{X}) = \frac{1}{k} \sum_{a \in A} \sum_{v \in V_a} c_{av} \frac{C_{L_{av}}^{\ell_{av}}}{C_L^\ell} \left( \frac{L_{av} - \ell_{av}}{L_{av}} n(a, \mathbb{X}) + \frac{\ell_{av}}{L_{av}} n(a, X'_{av}) \right). \quad (7.12)$$

**Доказательство.** Запишем определения  $E$  и  $\nu$ , затем подставим (7.7) согласно гипотезе 7.2 и переставим знаки суммирования:

$$\begin{aligned} C(\mu, \mathbb{X}) &= \frac{1}{C_L^\ell} \sum_{X \in [\mathbb{X}]^\ell} \sum_{a \in A} [\mu X = a] \frac{1}{k} \sum_{x_i \in \bar{X}} I(a, x_i) = \\ &= \frac{1}{k} \sum_{a \in A} \sum_{v \in V_a} c_{av} \sum_{i=1}^L I(a, x_i) \underbrace{P[X_{av} \subseteq X][X'_{av} \subseteq \bar{X}][x_i \in \bar{X}]}_{p(a, v, x_i)}. \end{aligned}$$

Если  $x_i \in X'_{av}$ , то  $[X'_{av} \subseteq \bar{X}][x_i \in \bar{X}] = [X'_{av} \subseteq \bar{X}]$  и, согласно (7.9),

$$p(a, v, x_i) = P[X_{av} \subseteq X][X'_{av} \subseteq \bar{X}] = \frac{C_{L_{av}}^{\ell_{av}}}{C_L^\ell} = P_{av}.$$

Если  $x_i \notin X'_{av}$ , то  $[X'_{av} \subseteq \bar{X}][x_i \in \bar{X}] = [\{x_i\} \cup X'_{av} \subseteq \bar{X}]$  и, согласно (7.9),

$$p(a, v, x_i) = P[X_{av} \subseteq X][\{x_i\} \cup X'_{av} \subseteq \bar{X}] = \frac{C_{L_{av}-1}^{\ell_{av}}}{C_L^\ell} = P_{av} \frac{L_{av} - \ell_{av}}{L_{av}}.$$

Собирая вместе два случая,  $x_i \in X'_{av}$  и  $x_i \notin X'_{av}$ , получим

$$p(a, v, x_i) = P_{av} \left( [x_i \in X'_{av}] + [x_i \notin X'_{av}] \frac{L_{av} - \ell_{av}}{L_{av}} \right).$$

Подставляя это выражение в сумму по  $i$ , получаем

$$\sum_{i=1}^L I(a, x_i) p(a, v, x_i) = P_{av} \left( n(a, X'_{av}) + (n(a, \mathbb{X}) - n(a, X'_{av})) \frac{L_{av} - \ell_{av}}{L_{av}} \right),$$

откуда вытекает требуемое равенство (7.12). Теорема доказана. ■

## Резюме

*Принцип порождающих и запрещающих множеств* основан на гипотезе, что для каждого алгоритма можно указать множество *порождающих* объектов, которые обязаны быть в обучающей выборке, и множество *запрещающих* объектов, которых не должно быть в обучающей выборке, чтобы метод обучения выбрал именно данный алгоритм. Это довольно сильное предположение, и оно выполняется далеко не всегда. В общем случае для каждого алгоритма можно указать несколько пар порождающих и запрещающих множеств. Если они известны, то далее легко вычисляются вероятности получить каждый из алгоритмов, вероятность переобучения, и оценка полного скользящего контроля. Выбор системы порождающих и запрещающих множеств является искусством. Чем их меньше, и чем меньше их мощности, тем эффективнее будут вычисляться оценки.

**В следующей лекции** с помощью принципа порождающих и запрещающих множеств мы получим точные оценки вероятности переобучения для некоторых модельных семейств алгоритмов.

## 8 Цепи алгоритмов

Чтобы воспользоваться принципом порождающих и запрещающих множеств, необходимо конкретизировать метод обучения  $\mu$  и семейство алгоритмов  $A$ . Мы начнём с метода минимизации эмпирического риска и цепей алгоритмов — простейших модельных семейств, обладающих свойствами расслоения и связности. Чтобы подкрепить теоретические выкладки экспериментами, предлагается написать несложную программу, вычисляющую эмпирические оценки вероятности переобучения по заданной матрице ошибок.

### §8.1 Разновидности минимизации эмпирического риска

Будем полагать, что  $A$  — конечное множество, и все алгоритмы имеют попарно различные векторы ошибок. Обозначим через  $A(X)$  множество алгоритмов с минимальным числом ошибок на обучающей выборке  $X$ :

$$A(X) = \text{Arg min}_{a \in A} n(a, X). \quad (8.1)$$

**Определение 8.1.** Метод обучения  $\mu$  называется *минимизацией эмпирического риска*, МЭР (*empirical risk minimization, ERM*), если  $\mu X \in A(X)$  при всех  $X \in [\mathbb{X}]^\ell$ .

Если множество  $A(X)$  содержит более одного элемента, то возникает проблема неоднозначности выбора алгоритма. Рассмотрим сначала два крайних случая — когда выбирается наилучший или наихудший алгоритм из  $A(X)$ .

**Определение 8.2.** Метод минимизации эмпирического риска  $\mu$  называется *оптимистичным*, если  $\mu X = \arg \min_{a \in A(X)} n(a, \bar{X})$ .

**Определение 8.3.** Метод минимизации эмпирического риска  $\mu$  называется *пессимистичным*, если  $\mu X = \arg \max_{a \in A(X)} n(a, \bar{X})$ .

Оптимистичная и пессимистичная МЭР на практике не реализуемы, так как скрытую контрольную выборку  $\bar{X}$  невозможно знать на этапе обучения. Теоретически они интересны тем, что дают нижнюю и верхнюю оценки вероятности переобучения. При любых других способах разрешения неоднозначности в методе МЭР вероятность переобучения гарантированно зажата между этими двумя оценками.

Более практичным представляется способ разрешения неоднозначности, основанный на случайном выборе алгоритма из множества  $A(X)$ .

**Определение 8.4.** Метод минимизации эмпирического риска  $\mu$  называется *рандомизированным*, если  $\mu X$  — это произвольный алгоритм, выбранный случайно и равновероятно из конечного множества алгоритмов  $A(X)$ .

Рандомизация становится вторым независимым источником случайности в задаче статистического обучения. Если ранее предполагалось, что случайно только

разбиение  $X \sqcup \bar{X}$ , то теперь случаен также и выбор алгоритма  $a$  из множества  $A(X)$ . Соответствующим образом изменяется и определение вероятности переобучения:

$$Q_\varepsilon = \mathbb{E} \frac{1}{|A(X)|} \sum_{a \in A(X)} [\nu(a, \bar{X}) - \nu(a, X) \geq \varepsilon]. \quad (8.2)$$

Большинство получаемых далее оценок основаны либо на пессимистичной, либо на рандомизированной МЭР. Эксперименты показывают, что завышенность оценок пессимистичной МЭР невелика, причём оптимистичные, пессимистичные и рандомизированные оценки, как правило, сходятся друг к другу с ростом длины выборки  $L$ .

## §8.2 Эксперименты с модельными семействами алгоритмов

**Модельные семейства алгоритмов.** Напомним, что нашим основным объектом исследования является бинарная матрица ошибок размера  $L \times D$ , порождаемая выборкой  $\mathbb{X} = \{x_1, \dots, x_L\}$  и множеством алгоритмов  $A = \{a_1, \dots, a_D\}$  с попарно различными векторами ошибок.

Мы будем рассматривать *модельные семейства алгоритмов*, которые задаются непосредственно своими матрицами ошибок. Модельное семейство — это искусственный объект исследования, не связанный с какой-либо реальной выборкой. Его матрица ошибок определяется специальным образом и обладает некоторой «регулярной структурой», что облегчает вывод комбинаторных оценок. Для некоторых модельных семейств удаётся строить примеры порождающих их выборок; обычно это весьма экзотические частные случаи. Реальные семейства крайне редко обладают какой-либо регулярной структурой.

Изучение модельных семейств представляет интерес по ряду причин.

Во-первых, они позволяют исследовать влияние эффектов расслоения и связности на вероятность переобучения.

Во-вторых, на них отрабатываются математические приёмы, которые могут оказаться полезными при получении оценок более общего вида.

В-третьих, некоторые модельные семейства обладают теми же ключевыми свойствами, что и реальные — расслоением, связностью и размерностью. Они могут использоваться для аппроксимации вероятности переобучения реальных семейств. Это перспективное направление исследований в комбинаторной теории переобучения [6].

### Эмпирическое оценивание вероятности переобучения по матрице ошибок.

Идея заключается в том, чтобы оценить вероятность переобучения по некоторому подмножеству разбиений генеральной выборки  $\mathbb{X} = X_n \sqcup \bar{X}_n$ ,  $n = 1, \dots, N$ :

$$\hat{Q}_\varepsilon(\mu, \mathbb{X}) = \frac{1}{N} \sum_{n=1}^N [\delta(\mu X_n, X_n) \geq \varepsilon],$$

где  $\delta(a, X) = \nu(a, \bar{X}) - \nu(a, X)$  — переобученность алгоритма  $a$  при разбиении  $X \sqcup \bar{X}$ .

В *методе Монте-Карло* разбиения выбираются случайным образом. Число разбиений  $N$  обычно берут от нескольких сотен до десятков тысяч.

---

**Алгоритм 8.1.** Вычисление эмпирических оценок вероятности переобучения.

---

**Вход:** матрица ошибок  $A = \{a_1, \dots, a_D\}$ ; число разбиений  $N$ ; порог  $\varepsilon$ ;

**Выход:** верхняя оценка  $\bar{Q}_\varepsilon$ , нижняя оценка  $Q_\varepsilon$ .

---

- 1:  $\bar{Q}_\varepsilon := 0$ ;  $Q_\varepsilon := 0$ ;
  - 2: **для всех** разбиений  $X_n \sqcup \bar{X}_n = \mathbb{X}$ ,  $n = 1, \dots, N$
  - 3:  $A(X_n) := \underset{a \in A}{\text{Arg min}} \nu(a, X_n)$ ;
  - 4:  $\bar{a} := \arg \max_{a \in A(X_n)} \nu(a, \bar{X}_n)$ ;  $\bar{Q}_\varepsilon := \bar{Q}_\varepsilon + \frac{1}{N} [\delta(\bar{a}, X_n) \geq \varepsilon]$ ;
  - 5:  $\underline{a} := \arg \min_{a \in A(X_n)} \nu(a, \bar{X}_n)$ ;  $Q_\varepsilon := Q_\varepsilon + \frac{1}{N} [\delta(\underline{a}, X_n) \geq \varepsilon]$ ;
  - 6: **вернуть**  $\bar{Q}_\varepsilon$ ,  $Q_\varepsilon$ .
- 

**Алгоритм 8.2.** Эффективное построение графиков зависимости вероятности переобучения от числа  $d$  первых алгоритмов в семействе  $\{a_1, \dots, a_D\}$ .

---

**Вход:** матрица ошибок  $A = \{a_1, \dots, a_D\}$ ; число разбиений  $N$ ; порог  $\varepsilon$ ;

**Выход:** верхняя оценка  $\bar{Q}_\varepsilon(d)$  и нижняя оценка  $Q_\varepsilon(d)$  для подмножества  $\{a_1, \dots, a_d\}$ .

---

- 1: сгенерировать и зафиксировать  $N$  разбиений:  
 $X_n \sqcup \bar{X}_n = \mathbb{X}$ ,  $n = 1, \dots, N$ ;
  - 2: алгоритм, выбранный при  $n$ -м разбиении пессимистичной и оптимистичной МЭР:  
 $\bar{a}_n, \underline{a}_n := a_1$ ,  $n = 1, \dots, N$ ;
  - 3:  $\bar{Q}_\varepsilon(1), Q_\varepsilon(1) := \frac{1}{N} \sum_{n=1}^N [\delta(a_1, X_n) \geq \varepsilon]$ ;
  - 4: **для всех**  $d := 2, \dots, D$
  - 5: **для всех**  $n := 1, \dots, N$
  - 6: **если**  $n(a_d, X_n) < n(\bar{a}_n, X_n)$  или  
 $n(a_d, X_n) = n(\bar{a}_n, X_n)$  и  $n(a_d, \bar{X}_n) > n(\bar{a}_n, \bar{X}_n)$  **то**
  - 7:  $\bar{Q}_\varepsilon(d) := \bar{Q}_\varepsilon(d-1) - \frac{1}{N} [\delta(\bar{a}_n, X_n) \geq \varepsilon] + \frac{1}{N} [\delta(a_d, X_n) \geq \varepsilon]$ ;  
 $\bar{a}_n := a_d$ ;
  - 8: **если**  $n(a_d, X_n) < n(\underline{a}_n, X_n)$  или  
 $n(a_d, X_n) = n(\underline{a}_n, X_n)$  и  $n(a_d, \bar{X}_n) < n(\underline{a}_n, \bar{X}_n)$  **то**
  - 9:  $Q_\varepsilon(d) := Q_\varepsilon(d-1) - \frac{1}{N} [\delta(\underline{a}_n, X_n) \geq \varepsilon] + \frac{1}{N} [\delta(a_d, X_n) \geq \varepsilon]$ ;  
 $\underline{a}_n := a_d$ ;
- 

Алгоритм 8.1 вычисляет две оценки вероятности переобучения: верхняя  $\bar{Q}_\varepsilon$  соответствует пессимистичной МЭР, нижняя  $Q_\varepsilon$  — оптимистичной. Алгоритм состоит из двух вложенных циклов: внешний перебирает разбиения, внутренний выбирает из  $A$  два алгоритма, минимизирующих эмпирический риск при данном разбиении — пессимистичный  $\bar{a}$  и оптимистичный  $\underline{a}$ .

Поменяв местами два вложенных цикла, получим Алгоритм 8.2. Он эффективно (за время  $O(D)$ ) строит график зависимости вероятности переобучения от числа первых  $d$  алгоритмов в подсемействе  $\{a_1, \dots, a_d\}$ . Предполагается, что алгоритмы упорядочены по числу ошибок на генеральной выборке,  $n(a_1, \mathbb{X}) \leq \dots \leq n(a_D, \mathbb{X})$ .

### §8.3 Связные семейства алгоритмов

**Цепи алгоритмов.** Определим расстояние между алгоритмами как *расстояние Хэмминга* между их векторами ошибок:

$$\rho(a, a') = \sum_{i=1}^L |I(a, x_i) - I(a', x_i)|, \quad \forall a, a' \in A.$$

**Определение 8.5.** Конечная последовательность алгоритмов  $A = \{a_0, \dots, a_D\}$  называется *цепью алгоритмов*, если  $\rho(a_{d-1}, a_d) = 1$  для всех  $d = 1, \dots, D$ .

**Определение 8.6.** Множество алгоритмов  $A$  называется *связным*, если для любых  $a, a' \in A$  в нём существует цепь  $a = a_0, a_1, \dots, a_D = a'$ .

**Определение 8.7.** Цепь  $A = \{a_0, \dots, a_D\}$  называется *прямой*, если  $\rho(a_0, a_D) = D$ .

В прямой цепи все объекты  $x_d$ , на которых различаются два соседних алгоритма,  $I(a_{d-1}, x_d) \neq I(a_d, x_d)$ , попарно различны. Число таких объектов равно  $D$ . Отсюда следует ограничение на длину прямой цепи:  $|A| \leq L + 1$  или, что то же самое,  $D \leq L$ . Любая подпоследовательность прямой цепи  $a_u, \dots, a_v$ ,  $0 \leq u < v \leq D$ , также является прямой цепью.

**Определение 8.8.** Цепь алгоритмов  $A = \{a_0, \dots, a_D\}$  называется *монотонной*, если  $n(a_d, \mathbb{X}) = m + d$  для всех  $d = 0, \dots, D$ , при некотором  $m \geq 0$ . Алгоритм  $a_0$  называется *лучшим в цепи*.

В монотонной цепи  $I(a_{d-1}, x_i) \leq I(a_d, x_i)$  для всех  $x_i \in \mathbb{X}$ ,  $d = 1, \dots, D$ .

Монотонная цепь является прямой. Отсюда следует ограничение  $D \leq L - m$ .

В случае монотонной цепи объекты выборки можно перенумеровать так, чтобы

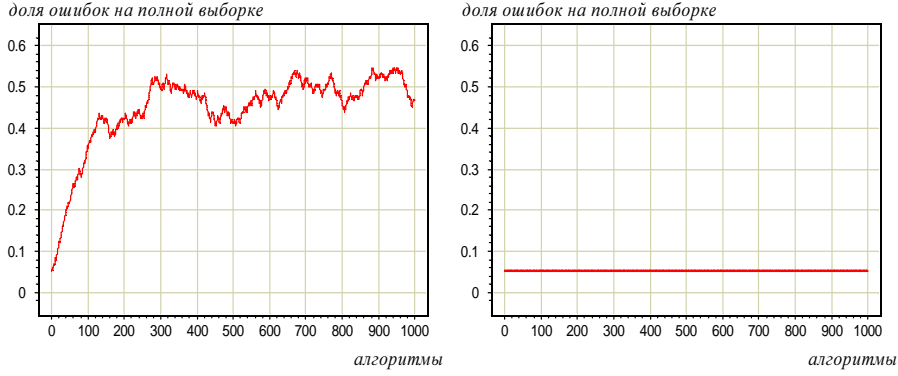
$$I(a_d, x_i) = [i \leq m + d], \quad i = 1, \dots, L, \quad d = 0, \dots, D.$$

Цепь алгоритмов может порождаться, в частности, однопараметрическим семейством классификаторов с непрерывной по параметру дискриминантной функцией. Если при непрерывном смещении параметра в сторону от оптимального значения число ошибок на полной выборке монотонно не убывает, то образуется монотонная цепь. Монотонная цепь — это одно из простейших модельных семейств, обладающее свойствами *расслоения* и *связности*.

**Пример 8.1.** Пусть  $A$  — семейство *линейных алгоритмов классификации* — параметрических отображений из  $\mathbb{X} = \mathbb{R}^n$  в  $\mathbb{Y} = \{-1, +1\}$  вида

$$a(x, w) = \text{sign}(x_1 w_1 + \dots + x_n w_n), \quad x = (x_1, \dots, x_n) \in \mathbb{R}^n,$$

где параметр  $w \in \mathbb{R}^n$  — направляющий вектор гиперплоскости, разделяющей пространство  $\mathbb{R}^n$  на два полупространства — классы  $-1$  и  $+1$ . Пусть функция потерь имеет вид  $I(a, x) = [a(x, w) \neq y(x)]$ , где  $y(x)$  — истинная классификация объекта  $x$ ,



**Рис. 8.1.** Распределение алгоритмов по частоте ошибок на генеральной выборке в модельных цепях с расслоением и без расслоения. Зависимость  $\nu(a_d, \mathbb{X})$  от  $d$  при  $\ell = k = 100$ ,  $m = 10$ .

и множество объектов  $\mathbb{X}$  линейно разделимо, т. е. существует вектор  $w^* \in \mathbb{R}^n$ , при котором алгоритм  $a(x, w^*)$  не допускает ошибок на  $\mathbb{X}$ . Тогда множество алгоритмов

$$A_\delta = \{a(x, w^* + t\delta) : t \in [0, +\infty)\}$$

порождает монотонную цепь при любом  $\delta \in \mathbb{R}^n$ , за исключением, быть может, некоторого конечного множества векторов. При этом  $m = 0$  в силу линейной делимости.

**Эксперимент с четырьмя модельными семействами.** Именно с этого эксперимента началось активное развитие комбинаторной теории переобучения. Он наглядно показал, насколько сильное влияние оказывают эффекты расслоения и связности на вероятность переобучения [109, 110].

Сгенерируем матрицы ошибок двух модельных цепей.

1. *Цепь случайных инверсий с расслоением.* Лучший алгоритм  $a_1$  допускает  $m$  ошибок на полной выборке. Каждый следующий вектор ошибок  $a_d$  получается из  $a_{d-1}$  путём инверсии одной случайно выбранной координаты. Если цепь длинная,  $D \gg L$ , то большинство алгоритмов допускают число ошибок  $m$ , близкое к  $L/2$ .

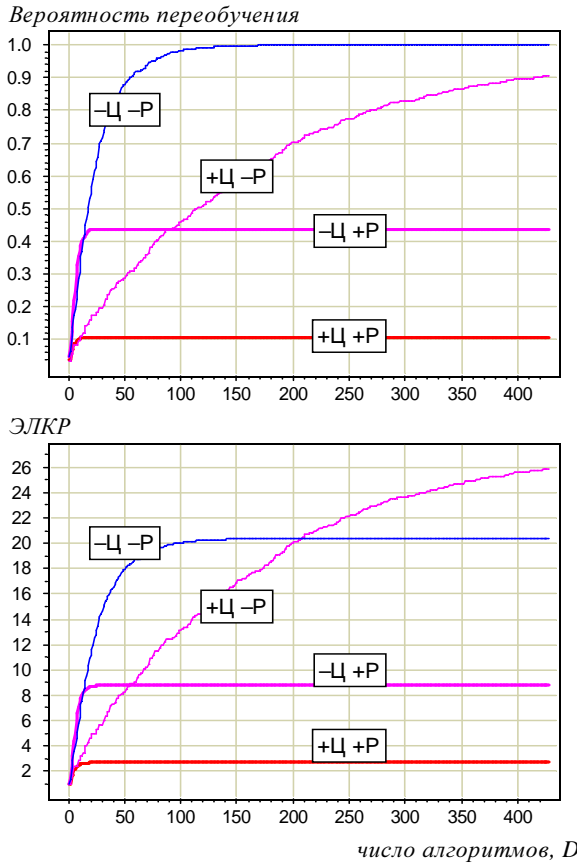
2. *Цепь случайных инверсий без расслоения.* Лучший алгоритм  $a_1$  допускает  $m$  ошибок на полной выборке. Каждый вектор ошибок  $a_d$  получается из  $a_{d-1}$  также путём инверсии одной случайно выбранной координаты, но при нечётных  $d$  производятся только инверсии  $0 \rightarrow 1$ , а при чётных  $d$  — только  $1 \rightarrow 0$ . В результате число ошибок алгоритмов на полной выборке, чередуясь, принимает значения  $m$  и  $m + 1$ .

На рис. 8.1 показаны зависимости частоты ошибок  $\nu(a_d, \mathbb{X})$  от порядкового номера  $d$  алгоритма для цепи с расслоением (слева) и без расслоения (справа).

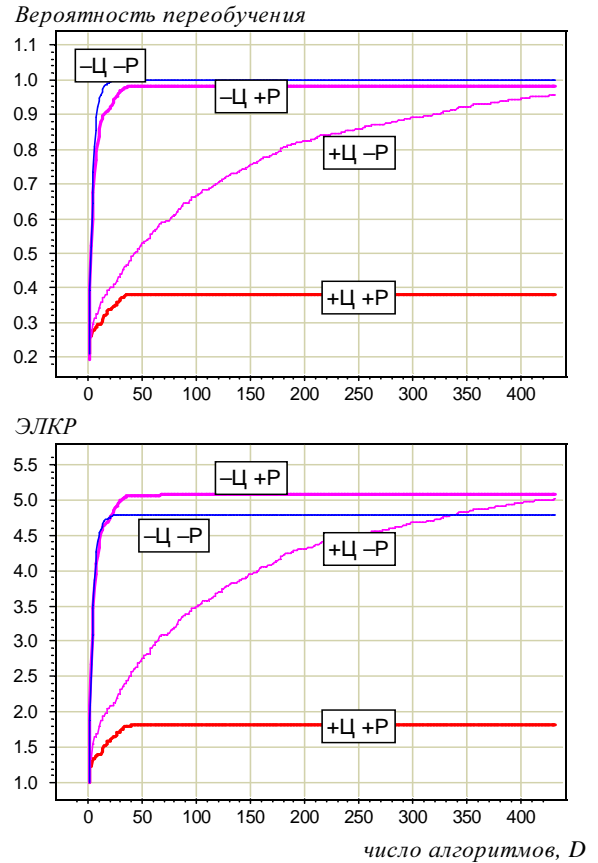
Произвольной цепи  $a_1, \dots, a_D$  можно поставить в соответствие не-цепь  $a'_1, \dots, a'_D$ , такую, что  $\nu(a'_d, \mathbb{X}) = \nu(a_d, \mathbb{X})$ , но соседние алгоритмы  $a'_{d-1}, a'_d$  существенно различаются. Для этого достаточно генерировать компоненты векторов ошибок  $\vec{a}'_d$  случайно и независимо, но так, чтобы число единиц в векторе  $\vec{a}'_d$  равнялось  $n(a_d, \mathbb{X})$ .

В данном случае строятся две не-цепи, соответствующие двум ранее построенным цепям случайных инверсий.





**Рис. 8.2.** Зависимость вероятности переобучения  $Q_\varepsilon$  и ЭЛКР  $\hat{\Delta}$  от числа алгоритмов  $D$ . Простая задача:  $\nu(a_1, \mathbb{X}) = 0.05$ .



**Рис. 8.3.** Зависимость вероятности переобучения  $Q_\varepsilon$  и ЭЛКР  $\hat{\Delta}$  от числа алгоритмов  $D$ . Трудная задача:  $\nu(a_1, \mathbb{X}) = 0.25$ .

Итого, строятся четыре модельных семейства: цепь с расслоением, цепь без расслоения и две соответствующие им не-цепи. Все они задаются матрицами ошибок размера  $L \times D$  и имеют одинаковые значения параметра  $m$ . Сопоставление этих четырёх случаев позволяет разделить влияние *связности* и *расслоения* на вероятность переобучения и ответить на вопрос — какое из этих двух свойств важнее.

На рис. 8.2 и рис. 8.3 показаны зависимости вероятности переобучения  $Q_\varepsilon$  от числа алгоритмов  $D$  для четырёх семейств, при  $\ell = k = 100$ ,  $\varepsilon = 0.05$ ,  $m = 10$  и  $m = 50$ . Число разбиений в методе Монте-Карло  $N = 10^4$ . Условные обозначения на графиках: +Ц — цепь, -Ц — не-цепь, +P — с расслоением, -P — без расслоения.

Показаны также графики зависимости ЭЛКР — *эффективного локального коэффициента разнообразия*  $\hat{\Delta}$  от числа алгоритмов  $D$ . ЭЛКР в данном случае определяется как отношение вероятности переобучения подсемейства  $\{a_1, \dots, a_D\}$  к вероятности переобучения одноэлементного подсемейства  $\{a_1\}$ .

### Основные выводы.

1. Для семейств без расслоения и связности переобучение может оказаться значительным уже при нескольких десятках алгоритмов в семействе. При больших  $D$  только одновременное наличие расслоения и связности позволяет избежать сильного переобучения (нижние кривые на графиках). Таким образом, важны оба свойства.

2. Связность снижает темп роста зависимости  $Q_\varepsilon(D)$ .

3. Расслоение понижает уровень горизонтальной асимптоты  $Q_\varepsilon(D)$ , причём тем сильнее, чем легче задача, то есть чем меньше число ошибок лучшего алгоритма, что видно из сравнения рис. 8.2 и 8.3.

4. При наличии расслоения вероятность выбрать в результате обучения алгоритм из верхних слоёв быстро падает с ростом номера слоя.

5. По мере увеличения  $D$  как вероятность переобучения, так и ЭЛКР выходят на горизонтальную асимптоту и перестают зависеть от  $D$ . В то же время, VC-оценка линейна по  $D$  и вообще не имеет горизонтальной асимптоты — на графиках ЭЛКР VC-оценке соответствует прямая  $\hat{\Delta}(D) = D$ . VC-оценка близка к точной только для не-цепей и только при малых  $D$ ; в данном эксперименте — при  $D < 10$ ; при  $D$  порядка 20 она уже превосходит единицу.

## §8.4 Точная оценка для монотонной цепи алгоритмов

**Теорема 8.1.** Пусть  $A = \{a_0, \dots, a_D\}$  — монотонная цепь алгоритмов,  $L \geq m + D$ ,  $m = n(a_0, \mathbb{X})$ , метод обучения  $\mu$  является пессимистичной минимизацией эмпирического риска. Тогда вероятность получить алгоритм  $a_d$  в результате обучения равна

$$P_d = \frac{C_{L-d-\delta}^{\ell-\delta}}{C_L^\ell}, \quad d = 0, \dots, k;$$

где  $\delta = [d < D]$ ; вероятность переобучения равна

$$Q_\varepsilon = \sum_{d=0}^{\min\{k, D\}} P_d H_{L-d-\delta}^{\ell-\delta, m} \left( \frac{\ell}{L} (m + d - \varepsilon k) \right). \quad (8.3)$$

**Доказательство.** Перенумеруем объекты таким образом, чтобы каждый из алгоритмов  $a_d$ ,  $d = 1, \dots, D$  допускал ошибку на объектах  $x_1, \dots, x_d$ . Очевидно, лучший алгоритм  $a_0$  не допускает ошибку ни на одном из этих объектов. Нумерация остальных объектов не имеет значения, так как алгоритмы не различимы на них.

Для наглядности представим выборку  $\mathbb{X}$  разбитой на три блока:

$$\begin{array}{l} \vec{a}_0 = ( 0, \quad 0, \quad 0, \quad \dots \quad 0, \quad 0, \dots, 0, \quad \overbrace{1, \dots, 1}^m ); \\ \vec{a}_1 = ( 1, \quad 0, \quad 0, \quad \dots \quad 0, \quad 0, \dots, 0, \quad 1, \dots, 1 ); \\ \vec{a}_2 = ( 1, \quad 1, \quad 0, \quad \dots \quad 0, \quad 0, \dots, 0, \quad 1, \dots, 1 ); \\ \vec{a}_3 = ( 1, \quad 1, \quad 1, \quad \dots \quad 0, \quad 0, \dots, 0, \quad 1, \dots, 1 ); \\ \dots \\ \vec{a}_D = ( 1, \quad 1, \quad 1, \quad \dots \quad 1, \quad 0, \dots, 0, \quad 1, \dots, 1 ); \end{array}$$

Зафиксируем алгоритм  $a_d$  и рассмотрим три случая.

1. Если  $d > k$ , то число ошибок алгоритма  $a_d$  на множестве объектов  $\{x_1, \dots, x_d\}$  превышает длину контрольной выборки. Часть ошибок обязательно окажется в обучающей подвыборке  $X$ , и метод  $\mu$  выберет другой алгоритм. В этом случае

$$[\mu X = a_d] = 0, \quad P_d = 0.$$

2. Если  $d \leq k$  и  $d < D$ , то метод  $\mu$  выберет алгоритм  $a_d$ , когда объекты  $x_1, \dots, x_d$  будут находиться в контрольной подвыборке  $\bar{X}$ , а объект  $x_{d+1}$  — в обучающей  $X$ :

$$[\mu X = a_d] = [x_{d+1} \in X][x_1, \dots, x_d \in \bar{X}], \quad L_d = L - d - 1, \quad \ell_d = \ell - 1.$$

3. Если  $d \leq k$  и  $d = D$ , то метод  $\mu$  выберет наихудший алгоритм в цепи  $a_D$ , когда все объекты  $x_1, \dots, x_D$  будут находиться в контрольной подвыборке  $\bar{X}$ :

$$[\mu X = a_d] = [x_1, \dots, x_d \in \bar{X}], \quad L_d = L - d, \quad \ell_d = \ell.$$

Таким образом, справедлива простая гипотеза о порождающих и запрещающих множествах (7.1). Чтобы применить Теорему 7.3, найдём для каждого алгоритма  $a_d$  значения  $L_d, \ell_d, m_d, s_d$ , объединив случаи 2 и 3 с помощью поправки  $\delta = [d < D]$ :

$$L_d = L - d - \delta, \quad \ell_d = \ell - \delta, \quad m_d = m + d - d = m, \quad s_d(\varepsilon) = \frac{\ell}{L}(m + d - \varepsilon k).$$

Подставляя эти выражения в (7.3) и (7.4), получим утверждение теоремы. ■

**Замечание 8.1.** Верхний предел суммирования в (8.3) можно положить равным  $D$ , поскольку при  $d > k$  имеем  $C_{L-d-\delta}^{\ell-\delta} = 0$ , следовательно,  $P_d = 0$ .

**Замечание 8.2.** Полезно проверить, что совокупность условий  $\mu X = a$  определена корректно, доказав тождества  $\sum_{a \in A} [\mu X = a] = 1$  либо  $\sum_{a \in A} P_a = 1$ . Для монотонной цепи обе эти проверки вынесены в Упражнения.

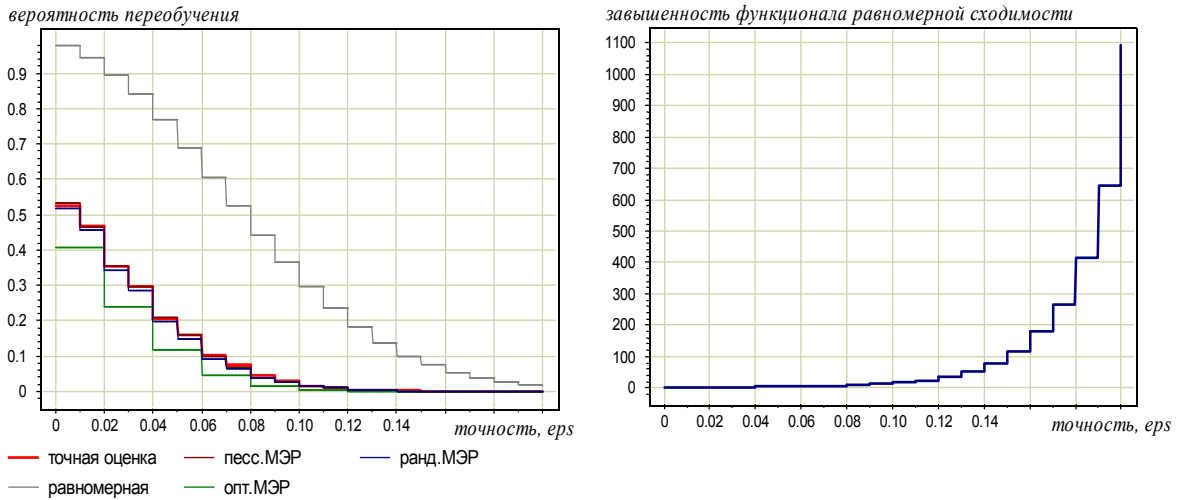
**Вычислительный эксперимент.** Построим зависимость вероятности переобучения  $Q_\varepsilon$  от точности  $\varepsilon$  и длины цепи  $D$ . Заодно проверим полученную формулу, сравнив результат с эмпирической оценкой  $\hat{Q}_\varepsilon$ , вычисленной *методом Монте-Карло* по  $N = 1000$  случайных разбиений. Графики на рис. 8.4 построены при  $\ell = k = 100$  и  $m = 20$ , то есть когда лучший алгоритм допускает 10% ошибок на полной выборке.

Пессимистичная и рандомизированная МЭР дают почти одинаковые оценки  $Q_\varepsilon$ , оптимистичная — заметно заниженную оценку, см. левый график на рис. 8.4.

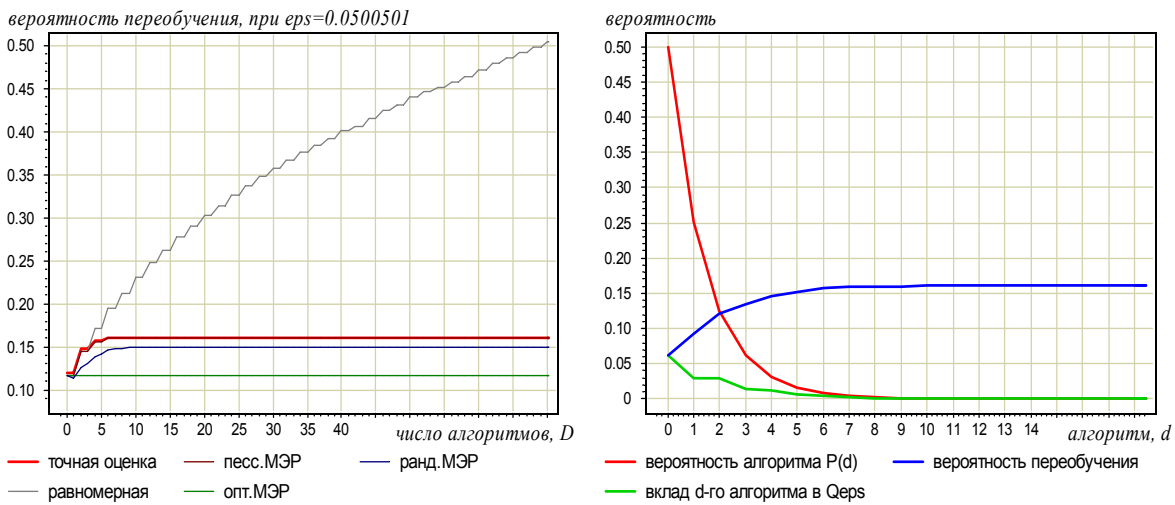
В эксперименте вычислялась также эмпирическая оценка функционала равномерной сходимости,

$$\hat{P}_\varepsilon = \hat{P} \left[ \max_{a \in A} \delta(a, X) \geq \varepsilon \right].$$

Он является завышенной верхней оценкой вероятности переобучения,  $Q_\varepsilon \leq P_\varepsilon$ , см. стр. 46. Правый график на рис. 8.4 показывает, что эта оценка может быть завышенной в сотни раз.



**Рис. 8.4.** Слева: зависимость оценок вероятности переобучения  $Q_\varepsilon$  от  $\varepsilon$ : точная оценка из Теоремы 8.1 и четыре оценки, вычисленные методом Монте-Карло по 1000 случайных разбиений: для пессимистичной, оптимистичной и рандомизированной МЭР. Верхняя кривая соответствует оценке по функционалу равномерной сходимости  $\hat{P}_\varepsilon$ . Справа: степень завышенности функционала равномерной сходимости  $\hat{P}_\varepsilon/Q_\varepsilon$ . Все графики построены при  $\ell = k = 100$ ,  $t = 20$ .



**Рис. 8.5.** Слева: зависимость вероятности переобучения  $Q_\varepsilon$  от числа алгоритмов  $D$ . Справа: вероятность получения каждого из алгоритмов  $P_d = P[\mu X = a_d]$ , вклад каждого алгоритма в вероятность переобучения  $Q_\varepsilon$ , значение  $Q_\varepsilon$  для пессимистичной МЭР по подмножеству алгоритмов  $\{a_0, \dots, a_d\}$  как функция от числа алгоритмов  $d$ . Все графики построены при  $\ell = k = 100$ ,  $t = 20$ ,  $\varepsilon = 0.05$ .

Левый график на рис. 8.5 показывает, что с ростом числа алгоритмов в монотонной цепи функционал равномерной сходимости  $P_\varepsilon$  продолжает возрастать, тогда как вероятность переобучения  $Q_\varepsilon$  после 5–8 алгоритмов выходит на горизонтальную асимптоту. Согласно Теореме 5.4, оценка равномерной сходимости завышена из-за того, что она не учитывает эффект расслоения. Поэтому кривую  $\hat{P}_\varepsilon$  (верхняя кривая

на левом графике рис. 8.5) можно рассматривать как оценку вероятности переобучения для *цепи без расслоения*. Только совместное проявление эффектов расслоения и связности понижает вероятность переобучения до приемлемо малых значений. Этот же вывод был сделан выше в экспериментах со случайными цепями.

Вкладом  $Q_\varepsilon(a)$  алгоритма  $a$  в вероятность переобучения  $Q_\varepsilon$  будем называть слабое под знаком суммы  $\sum_{a \in A}$  в общей формуле вероятности переобучения (7.10):

$$Q_\varepsilon = \sum_{a \in A} \underbrace{\sum_{v \in V_a} c_{av} P_{av} H_{L_{av}}^{\ell_{av}, m_{av}}(s_{av}(\varepsilon))}_{Q_\varepsilon(a)}.$$

Рис. 8.5 (справа) показывает, что существенные вклады в вероятность переобучения вносят только алгоритмы нескольких нижних слоёв. Это справедливо не только для монотонных цепей, но и для многих других семейств алгоритмов.

## §8.5 Точная оценка для унимодальной цепи алгоритмов

Унимодальная цепь является более реалистичной моделью однопараметрического *связного семейства*, по сравнению с монотонной цепью. Она предполагает, что отклонение вещественного параметра от оптимального значения, как в большую, так и в меньшую сторону, приводит к увеличению числа ошибок.

**Определение 8.9.** Прямая цепь  $A = \{a_D, \dots, a_1, a_0, a'_1, \dots, a'_D\}$  называется *унимодальной*, если её левая ветвь  $a_0, a_1, \dots, a_D$  и правая ветвь  $a_0, a'_1, \dots, a'_D$  являются монотонными цепями. Алгоритм  $a_0$  называется *лучшим в цепи*.

**Пример 8.2 (продолжение примера 8.1).** Пусть множество объектов  $\mathbb{X} \subset \mathbb{R}^n$  линейно разделимо, т. е. существует *линейный классификатор*  $a(x, w^*)$ , не допускающий ошибок на  $\mathbb{X}$ . Тогда множество алгоритмов  $\{a(x, w^* + t\delta) : t \in \mathbb{R}\}$  порождает унимодальную цепь для почти любого направляющего вектора  $\delta \in \mathbb{R}^n$ .

Обозначим через  $m = n(a_0, \mathbb{X})$  число ошибок лучшего алгоритма.

Рассмотрим унимодальную цепь с ветвями равной длины,  $D = D'$ . Перенумеруем объекты так, чтобы каждый из алгоритмов  $a_d$ ,  $d = 1, \dots, D$  допускал ошибку на объектах  $x_1, \dots, x_d$ ; а каждый из алгоритмов  $a'_d$ ,  $d = 1, \dots, D$  допускал ошибку на объектах  $x'_1, \dots, x'_d$ . Множества объектов  $\{x_1, \dots, x_D\}$  и  $\{x'_1, \dots, x'_D\}$  не пересекаются в силу того, что унимодальная цепь является прямой. Очевидно, лучший алгоритм  $a_0$  не допускает ошибку ни на одном из этих объектов. Нумерация остальных объектов не имеет значения, так как алгоритмы не различимы на них.

Для наглядности представим выборку  $X$  разбитой на четыре блока:

$$\begin{array}{l}
 \vec{a}_0 = ( 0, 0, 0, \dots, 0, 0, 0, 0, \dots, 0, 0, \dots, 0, \overbrace{1, \dots, 1}^m ); \\
 \vec{a}_1 = ( 1, 0, 0, \dots, 0, 0, 0, 0, \dots, 0, 0, \dots, 0, \overbrace{1, \dots, 1}^m ); \\
 \vec{a}_2 = ( 1, 1, 0, \dots, 0, 0, 0, 0, \dots, 0, 0, \dots, 0, \overbrace{1, \dots, 1}^m ); \\
 \vec{a}_3 = ( 1, 1, 1, \dots, 0, 0, 0, 0, \dots, 0, 0, \dots, 0, \overbrace{1, \dots, 1}^m ); \\
 \dots \\
 \vec{a}_D = ( 1, 1, 1, \dots, 1, 0, 0, 0, \dots, 0, 0, \dots, 0, \overbrace{1, \dots, 1}^m ); \\
 \vec{a}'_1 = ( 0, 0, 0, \dots, 0, 1, 0, 0, \dots, 0, 0, \dots, 0, \overbrace{1, \dots, 1}^m ); \\
 \vec{a}'_2 = ( 0, 0, 0, \dots, 0, 1, 1, 0, \dots, 0, 0, \dots, 0, \overbrace{1, \dots, 1}^m ); \\
 \vec{a}'_3 = ( 0, 0, 0, \dots, 0, 1, 1, 1, \dots, 0, 0, \dots, 0, \overbrace{1, \dots, 1}^m ); \\
 \dots \\
 \vec{a}'_D = ( 0, 0, 0, \dots, 0, 1, 1, 1, \dots, 1, 0, \dots, 0, \overbrace{1, \dots, 1}^m ).
 \end{array}$$

Будем полагать, что если минимум (8.1) достигается на двух алгоритмах с одинаковым числом ошибок и на обучающей, и на контрольной выборке,  $A(X) = \{a_d, a'_d\}$ , то метод  $\mu$  выбирает алгоритм  $a_d$  из левой ветви.

**Теорема 8.2.** Пусть  $A = \{a_D, \dots, a_1, a_0, a'_1, \dots, a'_D\}$  — унимодальная цепь,  $\ell \geq 2$ ,  $D \geq 1$ ,  $L \geq 2D + m$ . Пусть  $\mu$  — метод пессимистичной МЭР. Тогда вероятность получить каждый из алгоритмов цепи в результате обучения есть

$$\begin{aligned}
 P_0 &= \mathbb{P}[\mu X = a_0] = \frac{C_{L-2}^{\ell-2}}{C_L^\ell}; \\
 P_d &= \mathbb{P}[\mu X = a_d] = \frac{C_{L-d-\delta}^{\ell-\delta} - \delta C_{L-2d-2}^{\ell-1}}{C_L^\ell}, \quad d = 1, \dots, \min\{k, D\}; \\
 P'_d &= \mathbb{P}[\mu X = a'_d] = \frac{C_{L-d-\delta}^{\ell-\delta} - C_{L-2d-\delta}^{\ell-\delta}}{C_L^\ell}, \quad d = 1, \dots, \min\{k, D\};
 \end{aligned}$$

где  $\delta = [d < D]$ . Положим  $s_d(\varepsilon) = \frac{\ell}{L}(m + d - \varepsilon k)$ . Тогда вероятность переобучения

$$\begin{aligned}
 Q_\varepsilon &= \frac{C_{L-2}^{\ell-2}}{C_L^\ell} H_{L-2}^{\ell-2, m}(s_0(\varepsilon)) + \sum_{d=1}^{\min\{k, D\}} \left( 2 \frac{C_{L-d-\delta}^{\ell-\delta}}{C_L^\ell} H_{L-d-\delta}^{\ell-\delta, m}(s_d(\varepsilon)) - \right. \\
 &\quad \left. - \delta \frac{C_{L-2d-2}^{\ell-1}}{C_L^\ell} H_{L-2d-2}^{\ell-1, m}(s_d(\varepsilon)) - \right. \\
 &\quad \left. - \frac{C_{L-2d-\delta}^{\ell-\delta}}{C_L^\ell} H_{L-2d-\delta}^{\ell-\delta, m}(s_d(\varepsilon)) \right). \tag{8.4}
 \end{aligned}$$

**Доказательство.** Рассмотрим лучший алгоритм  $a_0$ . Если оба объекта  $x_1, x'_1$  лежат в обучающей выборке, то никакой другой алгоритм уже не может быть получен, поскольку  $\mu$  минимизирует эмпирический риск. Если хотя бы один из объектов  $x_1, x'_1$

не лежит в обучающей выборке, то  $\mu$  выберет другой алгоритм в силу пессимистичности. Таким образом,

$$[\mu X = a_0] = [x_1, x'_1 \in X].$$

Рассмотрим произвольный алгоритм  $a_d$  из левой ветви,  $d < D$ . Если бы не было правой ветви, то условие получения алгоритма  $a_d$  было бы таким же, как в монотонной цепи:  $[\mu X = a_d] = [x_{d+1} \in X][x_1, \dots, x_d \in \bar{X}]$ . При наличии правой ветви алгоритм  $a_d$  будет результатом обучения при тех же разбиениях  $X \sqcup \bar{X}$ , за исключением случаев, когда один из алгоритмов правой ветви  $a'_{d+1}, \dots, a'_D$  также минимизирует эмпирический риск. Это происходит, когда объекты  $x'_1, \dots, x'_{d+1}$  лежат в контрольной выборке. Следовательно,

$$[\mu X = a_d] = [x_{d+1} \in X][x_1, \dots, x_d \in \bar{X}](1 - [x'_1, \dots, x'_{d+1} \in \bar{X}]), \quad d = 1, \dots, D - 1.$$

Условия для правой ветви выписываются аналогично. Отличие состоит в том, что при  $\{a_d, a'_d\} \subseteq A(X)$  предпочтение отдаётся алгоритму  $a_d$  из левой ветви:

$$[\mu X = a'_d] = [x'_{d+1} \in X][x'_1, \dots, x'_d \in \bar{X}](1 - [x_1, \dots, x_d \in \bar{X}]), \quad d = 1, \dots, D - 1.$$

Для последних алгоритмов  $a_D, a'_D$  те же рассуждения приводят к условиям

$$\begin{aligned} [\mu X = a_D] &= [x_1, \dots, x_D \in \bar{X}]; \\ [\mu X = a'_D] &= [x'_1, \dots, x'_D \in \bar{X}](1 - [x_1, \dots, x_D \in \bar{X}]). \end{aligned}$$

Таким образом, справедлива обобщённая гипотеза ПЗМ (7.7). Каждому алгоритму  $a \in A$  соответствует одна или две пары ПЗМ, причём если вторая пара есть, то её коэффициент отрицателен,  $c_{av} = -1$ .

ПЗМ и параметры  $L_{av}, \ell_{av}, c_{av}$  всех алгоритмов удобно свести в таблицу:

алгоритм $a$	ПМ	ЗМ	$L_{av}$	$\ell_{av}$	$c_{av}$
$a_0$	$\{x_1, x'_1\}$	$\emptyset$	$L - 2$	$\ell - 2$	$+1$
$a_d, d = 1, \dots, D - 1$	$\{x_{d+1}\}$	$\{x_1, \dots, x_d\}$	$L - d - 1$	$\ell - 1$	$+1$
	$\{x_{d+1}\}$	$\{x_1, \dots, x_d, x'_1, \dots, x'_{d+1}\}$	$L - 2d - 2$	$\ell - 1$	$-1$
$a'_d, d = 1, \dots, D - 1$	$\{x'_{d+1}\}$	$\{x'_1, \dots, x'_d\}$	$L - d - 1$	$\ell - 1$	$+1$
	$\{x'_{d+1}\}$	$\{x'_1, \dots, x'_d, x_1, \dots, x_d\}$	$L - 2d - 1$	$\ell - 1$	$-1$
$a_D$	$\emptyset$	$\{x_1, \dots, x_D\}$	$L - D$	$\ell$	$+1$
$a'_D$	$\emptyset$	$\{x'_1, \dots, x'_D\}$	$L - D$	$\ell$	$+1$
	$\emptyset$	$\{x'_1, \dots, x'_D, x_1, \dots, x_D\}$	$L - 2D$	$\ell$	$-1$

Для всех алгоритмов и всех пар ПЗМ  $m_{av} = m$ ,  $s_{av}(\varepsilon) = \frac{\ell}{L}(m + d - \varepsilon k)$ .

Теперь легко записать вероятности  $P_d, P'_d$  по формулам (7.8), (7.9) и вероятность переобучения по формуле (7.10). Объединяя случаи  $d < D$  и  $d = D$  с помощью поправки  $\delta = [d < D]$ , получим (8.4). ■

**Замечание 8.3.** При  $d > k$  запрещающее множество не умещается целиком в контрольную выборку, и тогда  $[\mu X = a_d] = [\mu X = a'_d] = 0$ . Это обстоятельство можно учесть явно, ограничив верхний индекс суммирования в (8.4) минимумом  $\min\{k, D\}$ . Но можно и оставить верхний индекс  $D$ , поскольку биномиальные коэффициенты  $C_{L-d-\delta}^{\ell-\delta}$  в этом случае обращаются в нуль.

**Замечание 8.4.** Проверка корректности условий  $\mu X = a$  вынесена в Упражнения.

## Резюме

Модельные семейства алгоритмов задаются непосредственно матрицами ошибок, а не реальными выборками. Монотонная цепь алгоритмов — это простейшее модельное семейство, обладающее свойствами *расслоения* и *связности*. Унимодальная цепь алгоритмов состоит из двух монотонных цепей и является более адекватной моделью семейства алгоритмов с одним вещественным параметром.

Метод порождающих и запрещающих множеств даёт точную оценку вероятности переобучения для монотонных и унимодальных цепей алгоритмов.

Хорошо расслоенные цепи алгоритмов почти не переобучаются. Этот факт служит косвенным обоснованием для процедур одномерной оптимизации, которые часто применяются в машинном обучении для выбора одного критически важного параметра по отложенной выборке (hold-out model selection), например, константы регуляризации, ширины окна сглаживания, и т. п.

**В следующей лекции** мы введём понятие графа расслоения–связности семейства алгоритмов и получим верхнюю оценку вероятности переобучения, применимую к произвольным семействам, как модельным, так и реальным. В общем случае это именно верхняя оценка, однако для монотонных цепей, монотонных сетей и некоторых других модельных семейств алгоритмов она является точной.

## Упражнения

**Задача 8.1 (1).** Доказать, что цепь алгоритмов  $A = \{a_0, \dots, a_D\}$  является прямой тогда и только тогда, когда все объекты  $x_d$ , на которых различаются два соседних алгоритма,  $I(a_{d-1}, x_d) \neq I(a_d, x_d)$ , попарно различны.

**Задача 8.2 (1).** Проверить, что в случае монотонной цепи условия  $[\mu X = a]$  в Теореме 8.1 определены корректно, доказав тождество  $\sum_{a \in A} [\mu X = a] = 1$ .

Подсказка: преобразовать условия вида  $[x \in X]$  к виду  $1 - [x \in \bar{X}]$ .

**Задача 8.3 (1).** Проверить, что в случае унимодальной цепи условия  $[\mu X = a]$  в Теореме 8.2 определены корректно, доказав тождество  $\sum_{a \in A} [\mu X = a] = 1$ .

Подсказка: сначала решить предыдущую задачу.



**Задача 8.4 (1).** Для монотонной цепи, используя выражения для  $P_a$  и комбинаторные тождества, доказать равенство  $\sum_{a \in A} P_a = 1$ .

**Задача 8.5 (1).** Для унимодальной цепи, используя выражения для  $P_a$  и комбинаторные тождества, доказать равенство  $\sum_{a \in A} P_a = 1$ .

В следующих упражнениях предлагается вывести точную оценку вероятности переобучения  $Q_\varepsilon$  для некоторого модельного семейства алгоритмов  $A$ , в котором все векторы ошибок попарно различны. Предполагается что метод обучения  $\mu$  является пессимистичной минимизацией эмпирического риска.

**Задача 8.6 (1).**  $A = \{a_0, a_1, \dots, a_D\}$  — единичная окрестность лучшего алгоритма  $a_0$ ,  $n(a_0, \mathbb{X}) = m$ . Содержит  $D$  алгоритмов с попарно различными векторами ошибок, каждый из которых допускает на одну ошибку больше, чем  $a_0$ .

**Задача 8.7 (3).**  $A = \{a_{-D'}, \dots, a_0, \dots, a_D\}$  — несимметричная унимодальная цепь алгоритмов с ветвями различной длины,  $D \neq D'$ .

**Задача 8.8 (5\*).**  $A = \{a_0, \dots, a_D\}$  — прямая цепь без расслоения; число ошибок, чередуясь, принимает два значения:  $n(a_d, \mathbb{X}) = m + (d \bmod 2)$ ,  $d = 0, \dots, D$ .

**Задача 8.9 (7\*).**  $A = \{a_0, \dots, a_C, \dots, a_D\}$  — прямая цепь, составленная из цепи без расслоения  $a_0, \dots, a_C$  и монотонной цепи  $a_C, \dots, a_D$  с общим алгоритмом  $a_C$ .

**Задача 8.10 (8\*).**  $A = \{a_{-D'}, \dots, a_{-C'}, \dots, a_C, \dots, a_D\}$  — прямая цепь, составленная из монотонной цепи  $a_{-C'}, \dots, a_{-D'}$ , цепи без расслоения  $a_{-C'}, \dots, a_C$  и монотонной цепи  $a_C, \dots, a_D$ , с общими алгоритмами  $a_{-C'}$  и  $a_C$  соответственно.

**Задача 8.11 (5\*).** Матрица ошибок  $m$ -диагональна, то есть алгоритм  $a_d$  допускает ошибки на объектах  $x_{d+1}, \dots, x_{d+m}$ ,  $d = 0, \dots, D$ .

**Задача 8.12 (10\*).** Произвольная цепь алгоритмов.

**Задача 8.13 (5).** Пучок из  $h$  монотонных цепей  $\{a_0, a_{j1}, \dots, a_{jH}\}$ ,  $j = 1, \dots, h$ , с общим лучшим алгоритмом  $a_0$ ,  $n(a_0, \mathbb{X}) = m$ . Предполагается, что множества объектов  $X(j) = \{x_i : I(a_0, x_i) = 0 \text{ и } I(a_{jH}, x_i) = 1\}$ ,  $j = 1, \dots, h$ , попарно не пересекаются.

## Практикум

В практическом задании предлагается реализовать Алгоритм 8.2 и использовать его для проверки теоретических оценок и исследования новых семейств.

**Задача 8.14 (5).** Написать программу, позволяющую:

- генерировать модельные семейства алгоритмов  $A = \{a_1, \dots, a_D\}$  в виде бинарной матрицы ошибок размера  $L \times D$  (в матрице ошибок не должно быть одинаковых векторов ошибок); легко заменять генераторы данных;

- вычислять точные верхние и нижние оценки вероятности переобучения, если соответствующие формулы известны;
- вычислять методом Монте-Карло, т. е. по  $N$  случайным разбиениям:
  - верхнюю оценку  $\bar{Q}_\varepsilon$  для пессимистичного  $\mu_{\text{пес}}$  (худший из лучших);
  - нижнюю оценку  $\underline{Q}_\varepsilon$  для оптимистичного  $\mu_{\text{опт}}$  (лучший из лучших);
  - среднюю оценку  $\hat{Q}_\varepsilon$  для рандомизированного  $\mu_{\text{ран}}$  (случайный из лучших);
  - оценку  $\tilde{Q}_\varepsilon$  вероятности равномерного отклонения частот (5.1).
- строить графики, откладывая по оси  $X$  число первых  $d$  алгоритмов (либо, как вариант, номер слоя  $m$ ), по оси  $Y$ :
  - эмпирические оценки  $\bar{Q}_\varepsilon, \underline{Q}_\varepsilon, \hat{Q}_\varepsilon, \tilde{Q}_\varepsilon$  для подмножества  $A(d) = \{a_1, \dots, a_d\}$ ;
  - точные значения  $\bar{Q}_\varepsilon, \underline{Q}_\varepsilon, \hat{Q}_\varepsilon, \tilde{Q}_\varepsilon$  для подмножества  $A(d)$  (если известны);
 если по оси  $X$  откладывается  $m$ , то по оси  $Y$  дополнительно откладывать:
  - число алгоритмов в  $m$ -м слое;
  - доля разбиений, на которых  $n(\mu_{\text{пес}}X, \mathbb{X}) = m$ ;
  - доля разбиений, на которых  $n(\mu_{\text{опт}}X, \mathbb{X}) = m$ ;
- строить графики, в которых по оси  $Y$  откладываются точные (если известны) и эмпирические значения  $\bar{Q}_\varepsilon, \underline{Q}_\varepsilon, \hat{Q}_\varepsilon, \tilde{Q}_\varepsilon$ , по оси  $X$ :
  - число ошибок лучшего алгоритма  $m$ ;
  - длина обучения  $\ell$ , при одновременном росте длины контроля  $\ell = k$ ;
  - длина обучения  $\ell$  при фиксированной длине контроля  $k$ ;
  - длина контроля  $k$  при фиксированной длине обучения  $\ell$ .

Следующая серия задач направлена на экспериментальную проверку оценок и исследование зависимостей  $\bar{Q}_\varepsilon, \underline{Q}_\varepsilon, \hat{Q}_\varepsilon, \tilde{Q}_\varepsilon$  от параметров модельного семейства.

**Задача 8.15 (3).** Монотонная цепь с параметрами  $m$  и  $D$ .

**Задача 8.16 (3).** Унимодальная цепь с параметрами  $m$  и  $D$ .

**Задача 8.17 (3).** Единичная окрестность с параметрами  $m$  и  $D$  (см. Задачу 8.6).

**Задача 8.18 (3).** Проверить гипотезу, что монотонная цепь и цепь случайных инверсий [110] переобучаются практически одинаково. Сравнить, представив на одном графике, точные значения  $\bar{Q}_\varepsilon$  для монотонной цепи, их эмпирические оценки и эмпирические оценки  $\bar{Q}_\varepsilon$  для цепи случайных инверсий при одинаковых  $m$ . Как меняются различия между  $\bar{Q}_\varepsilon$  монотонной цепи и цепи случайных инверсий с ростом  $\ell$  и  $m$ ?

**Задача 8.19 (3).** Проверить гипотезу, что вероятность переобучения унимодальной сети (приблизительно) в два раза больше, чем у монотонной сети. Сравнить, представив на одном графике, точные значения  $\bar{Q}_\varepsilon$  и их эмпирические оценки для монотонной и унимодальной цепей при одинаковых  $L, \ell, m, \varepsilon$ .

**Задача 8.20 (3).** Проверить гипотезу, что разреженное подмножество прямой цепи переобучается практически так же, как сама цепь. Сравнить, представив на одном графике, эмпирические оценки  $\bar{Q}_\varepsilon$  прямой цепи  $\{a_0, \dots, a_D\}$  и её подмножества  $\{a_0, a_t, a_{2t}, a_{3t}, \dots\}$  при  $t = 1, \dots, D$ .

## 9 Оценки расслоения–связности

Первые оценки вероятности переобучения для произвольного множества алгоритмов  $A$ , учитывавшие расслоение и связность, были получены независимо Д. Кочедыковым и И. Решетняком. В [111] была предложена оценка, основанная на рекуррентном вычислении порождающих и запрещающих множеств при поочерёдном добавлении алгоритмов в  $A$ . Её доказательство было довольно громоздким; кроме того, требовалось, чтобы в  $A$  существовал корректный алгоритм, не допускающий ошибок на генеральной выборке. Позже оценка была улучшена, ограничение корректности снято, и найдено простое доказательство, которое приводится ниже.

### §9.1 Граф расслоения–связности

Напомним основные обозначения:

$\mathbb{X} = \{x_1, \dots, x_L\}$  — конечное множество объектов;

$A$  — множество алгоритмов;

$I(a, x)$  — индикатор ошибки алгоритма  $a \in A$  на объекте  $x \in \mathbb{X}$ ;

$n(a, X)$  — число ошибок алгоритма  $a$  на выборке  $X \subseteq \mathbb{X}$ ;

$\rho(a, b)$  — хэммингово расстояние между векторами ошибок алгоритмов  $a$  и  $b$ ;

$A_m = \{a \in A : n(a, \mathbb{X}) = m\}$  —  $m$ -й слой множества  $A$ .

Будем полагать, что все векторы ошибок  $\vec{a} = (I(a, x_i))_{i=1}^L$ , порождаемые алгоритмами  $a$  из  $A$ , попарно различны. Если это не так, то алгоритмы, соответствующие дублирующим векторам ошибок, исключим из множества  $A$ .

Введём на  $A$  естественное отношение порядка:  $a \leq b$  тогда и только тогда, когда  $I(a, x) \leq I(b, x)$  для всех  $x \in \mathbb{X}$ . Определим  $a < b$  если  $a \leq b$  и  $a \neq b$ .

Если  $a < b$  и при этом  $\rho(a, b) = 1$ , то будем говорить, что  $a$  предшествует  $b$  и записывать  $a \prec b$ . Очевидно, что  $n(a, \mathbb{X}) + 1 = n(b, \mathbb{X})$ .

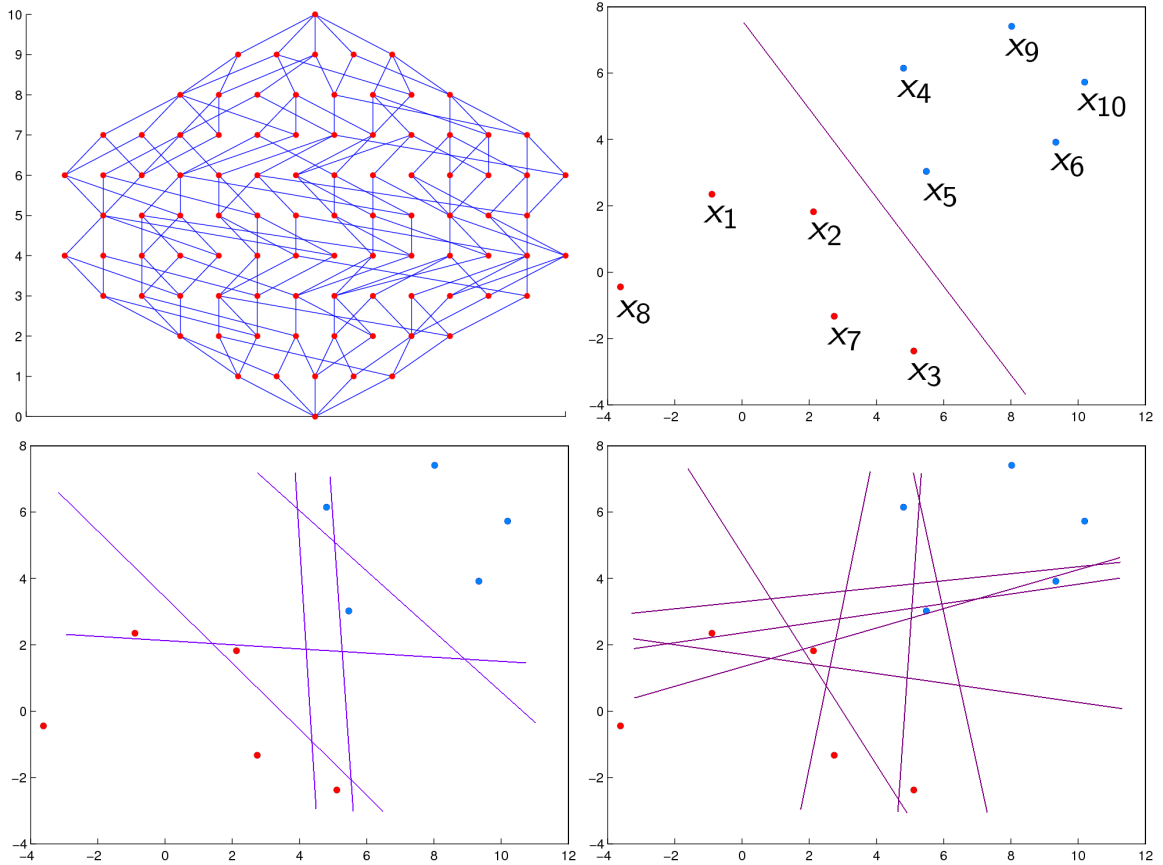
**Определение 9.1.** Графом расслоения–связности множества алгоритмов  $A$  будем называть направленный граф  $\langle A, E \rangle$  с множеством рёбер  $E = \{(a, b) : a \prec b\}$ .

Граф расслоения–связности является многодольным, доли соответствуют слоям  $A_m$ , рёбрами могут соединяться только алгоритмы соседних слоёв.

Каждому ребру  $a \prec b$  графа расслоения–связности соответствует один и только один объект  $x_{ab} \in \mathbb{X}$ , такой, что  $I(a, x_{ab}) = 0$  и  $I(b, x_{ab}) = 1$ .

**Пример 9.1.** На рис. 9.1 показан граф расслоения–связности, порождаемый семейством линейных алгоритмов классификации на выборке длины  $L = 10$ . Начальный фрагмент его матрицы ошибок приводился на рис. 1.2, стр. 10. Выборка линейно разделима, поэтому в графе имеется нулевой слой, состоящий из единственной вершины, соответствующей нулевому вектору ошибок. Первый слой образуется 5 алгоритмами с одной ошибкой, второй слой — 8 алгоритмами с двумя ошибками, и т. д.

В типичном случае граф расслоения–связности изоморфен графу транзитивной редукции отношения порядка  $\leq$ , называемому также *диаграммой Хассе*. Отличие



**Рис. 9.1.** Пример графа расслоения–связности (вверху слева; по вертикальной оси отложены номера слоёв), порождаемого семейством линейных алгоритмов классификации на выборке из 10 объектов, по 5 объектов каждого класса (вверху справа). Первый слой образуется 5 алгоритмами с одной ошибкой (внизу слева), второй слой — 8 алгоритмами с двумя ошибками (внизу справа), и т. д.

в том, что в графе  $\langle A, E \rangle$  рёбрами соединяются только алгоритмы, отличающиеся на одном объекте, тогда как в диаграмме Хассе рёбрами соединяются также и алгоритмы  $a, b$ , отличающиеся более чем на одном объекте, если не существует такого  $c \in A$ , что  $a < c < b$ . В общем случае граф расслоения–связности является подграфом диаграммы Хассе естественного отношения порядка на множестве бинарных векторов ошибок всех алгоритмов семейства  $A$ .

## §9.2 Оценки расслоения–связности

**Ослабление метода порождающих и запрещающих множеств.** Напомним, что для получения точных оценок обобщающей способности мы записывали необходимое и достаточное условие (7.7) того, что алгоритм  $a \in A$  выдаётся методом  $\mu$  в результате обучения по выборке  $X \in [\mathbb{X}]^\ell$ :

$$[\mu X = a] = \sum_{v \in V_a} c_{av} [X_{av} \subseteq X] [X'_{av} \subseteq \bar{X}].$$

Ослабим это условие.

Во-первых, ограничимся необходимым условием, что равносильно замене равенства оценкой сверху.

Во-вторых, чтобы упростить оценки, для каждого алгоритма будем искать только одну пару из порождающего и запрещающего множества, и полагать  $c_{av} = 1$ .

**Гипотеза 9.1.** Пусть множество  $A$ , выборка  $\mathbb{X}$  и метод  $\mu$  таковы, что для каждого алгоритма  $a \in A$  можно указать порождающее множество  $X_a \subset \mathbb{X}$  и запрещающее множество  $X'_a \subset \mathbb{X}$ , удовлетворяющие условиям

$$[\mu X = a] \leq [X_a \subseteq X] [X'_a \subseteq \bar{X}], \quad \forall X \in [\mathbb{X}]^\ell. \quad (9.1)$$

Если гипотеза 9.1 верна, то справедлива верхняя оценка, аналогичная точной оценке из Теоремы 7.3 (доказательство тривиально воспроизводится шаг за шагом):

$$Q_\varepsilon \leq \bar{Q}_\varepsilon = \sum_{a \in A} \bar{P}_a H_{L_a}^{\ell_a, m_a}(s_a(\varepsilon)); \quad \bar{P}_a = \frac{C_{L_a}^{\ell_a}}{C_L^\ell}; \quad (9.2)$$

$$L_a = L - |X_a| - |X'_a|;$$

$$\ell_a = \ell - |X_a|;$$

$$m_a = n(a, \mathbb{X} \setminus X_a \setminus X'_a);$$

$$s_a(\varepsilon) = \frac{\ell}{L} (n(a, \mathbb{X}) - \varepsilon k) - n(a, X_a).$$

Величина  $\bar{P}_a$  является верхней оценкой вероятности  $P_a = \mathbb{P}[\mu X = a]$  получить алгоритм  $a$  в результате обучения. Поскольку  $\sum_{a \in A} P_a = 1$  и  $\bar{P}_a \geq P_a$ , величина  $\sum_{a \in A} \bar{P}_a$ , скорее всего, окажется большей единицы. Она будет приблизительно равна степени завышенности оценки вероятности переобучения  $\bar{Q}_\varepsilon$ .

**Верхняя оценка вероятности переобучения.** Для каждого алгоритма  $a \in A$  определим два множества объектов:  $X_a$  — множество объектов  $x_{ab}$ , соответствующих всевозможным рёбрам графа  $(a, b) \in E$ , исходящим из  $a$ :

$$X_a = \{x \in \mathbb{X} \mid \exists b \in A: a \prec b, I(a, x) < I(b, x)\}, \quad (9.3)$$

и  $X'_a$  — множество объектов  $x \in \mathbb{X}$ , таких, что  $a$  ошибается на  $x$  и существует лучший алгоритм  $b$ , который не ошибается на  $x$ :

$$X'_a = \{x \in \mathbb{X} \mid \exists b \in A: b \prec a, I(b, x) < I(a, x)\}. \quad (9.4)$$

**Лемма 9.1.** Если  $\mu$  — метод пессимистичной минимизации эмпирического риска, то множества  $X_a$  (9.3) и  $X'_a$  (9.4) являются, соответственно, порождающим и запрещающим для алгоритма  $a$  в смысле Гипотезы 9.1.

**Доказательство.** Докажем от противного, что если  $\mu X = a$ , то  $X_a \subseteq X$  и  $X'_a \subseteq \bar{X}$ .

Допустим, что найдётся  $x_{ab} \in X_a$ , не лежащий в  $X$ . Тогда  $n(a, X) = n(b, X)$ , поскольку векторы ошибок  $a$  и  $b$  отличаются только на объекте  $x_{ab}$ . В то же время,

$n(a, \mathbb{X}) + 1 = n(b, \mathbb{X})$ , поэтому для метода  $\mu$ , в силу его пессимистичности, выбор алгоритма  $b$  по выборке  $X$  будет предпочтительнее, чем  $a$ , что противоречит условию  $\mu X = a$ . Значит,  $X_a \subseteq X$ .

Допустим теперь, что найдётся  $x \in X'_a$ , лежащий в  $X$ . Тогда существует  $b \in A$ , для которого  $n(b, X) < n(a, X)$ . Поскольку метод  $\mu$  минимизирует эмпирический риск, выбор алгоритма  $b$  будет предпочтительнее, чем  $a$ , что противоречит условию  $\mu X = a$ . Значит,  $X'_a \subseteq \bar{X}$ .

Лемма доказана. ■

**Определение 9.2.** *Верхней связностью  $q(a)$  (upper connectivity) алгоритма  $a \in A$  будем называть число рёбер графа, исходящих из вершины  $a$ :*

$$q(a) = |X_a| = \#\{x_{ab} \in \mathbb{X} : a \prec b\}.$$

**Определение 9.3.** *Нижней связностью  $d(a)$  (lower connectivity) алгоритма  $a \in A$  будем называть число рёбер графа, входящих в вершину  $a$ :*

$$d(a) = |D_a| = \#\{x_{ba} \in \mathbb{X} : b \prec a\}.$$

Связность  $q(a)$  (или  $d(a)$ ) есть реализуемое семейством  $A$  число способов изменить алгоритм  $a$  так, чтобы он стал делать на одну ошибку больше (или меньше). Связность можно интерпретировать как число степеней свободы семейства  $A$  в локальной окрестности алгоритма  $a \in A$ .

**Определение 9.4.** *Неоптимальностью  $r(a)$  (inferiority) алгоритма  $a \in A$  будем называть число объектов  $x \in \mathbb{X}$ , на которых алгоритм  $a$  ошибается, при том, что существует алгоритм  $b \in A$ , лучший, чем  $a$  (то есть  $b < a$ ), не ошибающийся на  $x$ :*

$$r(a) = |X'_a|.$$

В терминах графа расслоения–связности  $r(a)$  есть число различных объектов  $x_{bc}$ , соответствующих всевозможным рёбрам  $(b, c)$  на путях, ведущих к вершине  $a$ .

Справедливо неравенство  $d(a) \leq r(a) \leq n(a, \mathbb{X})$ .

Равенство  $r(a) = d(a)$  достигается на всех алгоритмах двух самых нижних слоёв.

Равенство  $r(a) = n(a, \mathbb{X})$  достигается в случае, когда существует корректный алгоритм  $a_0 \in A$ :  $n(a_0, \mathbb{X}) = 0$ .

**Теорема 9.2 (оценка расслоения–связности).** *Пусть  $\mu$  — метод пессимистичной минимизации эмпирического риска, векторы ошибок всех алгоритмов из  $A$  попарно различны. Тогда имеет место верхняя оценка вероятности переобучения:*

$$Q_\varepsilon(\mu, \mathbb{X}) \leq \sum_{a \in A} \frac{C_{L-q-r}^{\ell-q}}{C_L^\ell} H_{L-q-r}^{\ell-q, m-r} \left( \frac{\ell}{L} (m - \varepsilon k) \right), \quad (9.5)$$

где  $q = q(a)$  — верхняя связность,  $r = r(a)$  — неоптимальность,  $m = n(a, \mathbb{X})$  — число ошибок алгоритма  $a$  на генеральной выборке.

**Доказательство.** Данная оценка следует непосредственно из общей оценки (9.2) и Леммы 9.1, если заметить, что произвольный алгоритм  $a \in A$  ошибается на всех объектах запрещающего множества  $X'_a$  и не ошибается на всех объектах порождающего множества  $X_a$ :  $|X_a| = q(a)$ ,  $n(a, X_a) = 0$ ,  $|X'_a| = n(a, X'_a) = r(a)$ . ■

### Некоторые свойства оценки расслоения–связности.

1. Благодаря комбинаторному сомножителю  $C_{L-q-r}^{\ell-q}/C_L^\ell$  вклад каждого алгоритма  $a$  в оценку  $Q_\varepsilon$  экспоненциально убывает с ростом неоптимальности  $r$  и связности  $q$ . Отсюда следуют два важных для практики вывода.

Во-первых, связные семейства менее подвержены переобучению.

Во-вторых, только нижние слои вносят существенный вклад в переобучение.

Благодаря последнему обстоятельству становится возможным эффективное вычисление приближённой (нижней) оценки  $\bar{Q}_\varepsilon$  по слоям снизу вверх.

2. Если пренебречь расслоением и связностью, положив  $r = q = 0$  для каждого  $a \in A$  в формуле (9.5), то получится лучшая из VC-оценок (5.3):

$$Q_\varepsilon(\mu, \mathbb{X}) \leq \sum_{a \in A} H_L^{\ell, m} \left( \frac{\ell}{L} (m - \varepsilon k) \right).$$

3. Оценка расслоения–связности является достижимой. Неравенство  $Q_\varepsilon \leq \bar{Q}_\varepsilon$  переходит в равенство, когда условие (9.1) является равенством. Другими словами, когда условие  $[\mu X = a]$ , выраженное с помощью единственной пары множеств  $(X_a, X'_a)$  для любого алгоритма  $a \in A$ , является не только необходимым, но и достаточным. Это так, в частности, для монотонной цепи алгоритмов (стр. 74) и её многомерного обобщения — монотонной сети алгоритмов, которая будет рассмотрена ниже.

4. Оценка расслоения–связности принимает особенно простой вид, когда в семействе  $A$  существует корректный алгоритм  $a_0$ :  $n(a_0, \mathbb{X}) = 0$ . Такие семейства будем называть *корректными* относительно выборки  $\mathbb{X}$ .

**Теорема 9.3.** *Если  $\mu$  — метод пессимистичной минимизации эмпирического риска и семейство  $A$  корректно, то*

$$Q_\varepsilon(\mu, \mathbb{X}) \leq \sum_{a \in A} \frac{C_{L-q-m}^{\ell-q}}{C_L^\ell} [m \geq \varepsilon k], \quad (9.6)$$

где  $q = q(a)$ ,  $m = n(a, \mathbb{X})$ .

**Доказательство.** Если в  $A$  существует корректный алгоритм  $a_0$ , то неоптимальность любого алгоритма  $a \in A$  равна числу его ошибок,  $r(a) = n(a, \mathbb{X})$ . Поэтому в (9.5) можно подставить  $m - r = 0$ . Тогда, согласно Лемме 7.7 (стр. 66), гипергеометрическое распределение вырождается:  $H_{L-q-m}^{\ell-q, 0} \left( \frac{\ell}{L} (m - \varepsilon k) \right) = [m \geq \varepsilon k]$ . Отсюда вытекает утверждение теоремы. ■



**Снова монотонная цепь алгоритмов.** Ранее уже была получена точная оценка вероятности переобучения (8.3) для монотонной цепи алгоритмов  $a_0 \prec a_1 \prec \dots \prec a_D$ , в которой лучший алгоритм допускает  $m$  ошибок,  $n(a_0, \mathbb{X}) = m$ :

$$Q_\varepsilon = \sum_{d=0}^{\min\{k, D\}} \frac{C^{L-d-\delta}}{C^L} H_{L-d-\delta}^{\ell-\delta, m} \left( \frac{\ell}{L} (m + d - \varepsilon k) \right), \quad \delta = [d < D].$$

Легко убедиться, что она совпадает с общей оценкой расслоения–связности (9.5). Это следует из того, что

- 1) связность равна 1 для всех алгоритмов, кроме последнего,  $q(a_d) = [d < D]$ ;
- 2) неоптимальность и число ошибок связаны равенством  $n(a_d, \mathbb{X}) = r(a_d) + m$ ;
- 3) числитель (9.5) обращается в нуль при  $\ell - \delta \leq L - d - \delta$ , то есть  $d \leq k$ , следовательно, ненулевой вклад в сумму могут давать только первые  $k$  алгоритмов.

Таким образом, монотонная цепь алгоритмов — это пример семейства, для которого общая оценка расслоения–связности (9.5) является точной.

### §9.3 Профиль расслоения–связности

В ряде случаев общая оценка (9.5) приводится к более удобному виду.

**Определение 9.5.** Профилем расслоения–связности множества алгоритмов  $A$  называется матрица  $(\Delta_{mq})$  размера  $(L+1) \times (L+1)$ , где  $\Delta_{mq}$  — число алгоритмов в  $m$ -м слое со связностью  $q$ :

$$\Delta_{mq} = \sum_{a \in A} [n(a, \mathbb{X}) = m] [q(a) = q].$$

**Теорема 9.4.** Пусть граф расслоения–связности имеет исток — единственный алгоритм  $a_0$ , от которого можно добраться по рёбрам до любого другого алгоритма, и  $m_0 = n(a_0, \mathbb{X})$ . Тогда справедлива верхняя оценка вероятности переобучения

$$Q_\varepsilon \leq \sum_{m=m_0}^L \sum_{q=0}^L \Delta_{mq} \frac{C^{L-q}}{C^L} H_{L-q-m+m_0}^{\ell-q, m_0} \left( \frac{\ell}{L} (m - \varepsilon k) \right). \quad (9.7)$$

**Доказательство.** Возьмём произвольный алгоритм  $a \in A$ . В силу единственности истока все запрещающие объекты из  $X'_a$  находятся на рёбрах графа, составляющих всевозможные пути между вершинами  $a_0$  и  $a$ . Поскольку все эти пути имеют общее начало  $a_0$  и общий конец  $a$ , число объектов в  $X'_a$  равно в точности длине пути  $n(a, \mathbb{X}) - m_0$ . С другой стороны, оно же равно  $r(a)$ . Значит,

$$Q_\varepsilon \leq \sum_{m=m_0}^L \sum_{a \in A_m} \frac{C^{\ell-q(a)}}{C^L} H_{L-q(a)-m+m_0}^{\ell-q(a), m_0} \left( \frac{\ell}{L} (m - \varepsilon k) \right).$$

Перегруппировав слагаемые и воспользовавшись определением профиля расслоения–связности, получим утверждение теоремы. ■

**Теорема 9.5.** Пусть верны предположения предыдущей теоремы и исток является корректным алгоритмом,  $m_0 = n(a_0, \mathbb{X}) = 0$ . Тогда справедлива верхняя оценка вероятности переобучения

$$Q_\varepsilon \leq \sum_{m=\lceil \varepsilon k \rceil}^L \sum_{q=0}^L \Delta_{mq} \frac{C_{L-q-m}^{\ell-q}}{C_L^\ell}. \quad (9.8)$$

**Доказательство.** В оценку (9.7) подставим  $m_0 = 0$ . Гипергеометрическое распределение вырождается согласно Лемме 7.7, откуда и получаем оценку (9.8). ■

Существование истока — довольно сильное предположение, и в большинстве практических ситуаций оно не выполняется. Предположение корректности ещё сильнее, поскольку вводится дополнительное требование, чтобы исток был корректным алгоритмом,  $m_0 = n(a_0, \mathbb{X}) = 0$ .

**Гипотеза о сепарабельности профиля расслоения–связности.** На рис. 9.2 показаны графики зависимости<sup>8</sup>  $\Delta_{mq}$  от  $m$  и  $q$ , для множества линейных алгоритмов классификации и линейно разделимых двумерных выборок длины  $L = 20, 50, 100$ .

Видно, что профиль связности концентрируется в точке  $q = 2$ , что совпадает с размерностью пространства. С увеличением длины выборки доминирование данной компоненты профиля только усиливается.

Глядя на графики, можно выдвинуть гипотезу, что профиль расслоения–связности  $\Delta_{mq}$  является с высокой точностью *сепарабельным*:

$$\Delta_{mq} \lesssim \Delta_m \lambda_q,$$

где  $\Delta_m$  — коэффициент разнообразия  $m$ -го слоя,  $\lambda_q$  — доля алгоритмов, имеющих связность  $q$ . Очевидно, выполняется условие нормировки  $\lambda_0 + \dots + \lambda_L = 1$ .

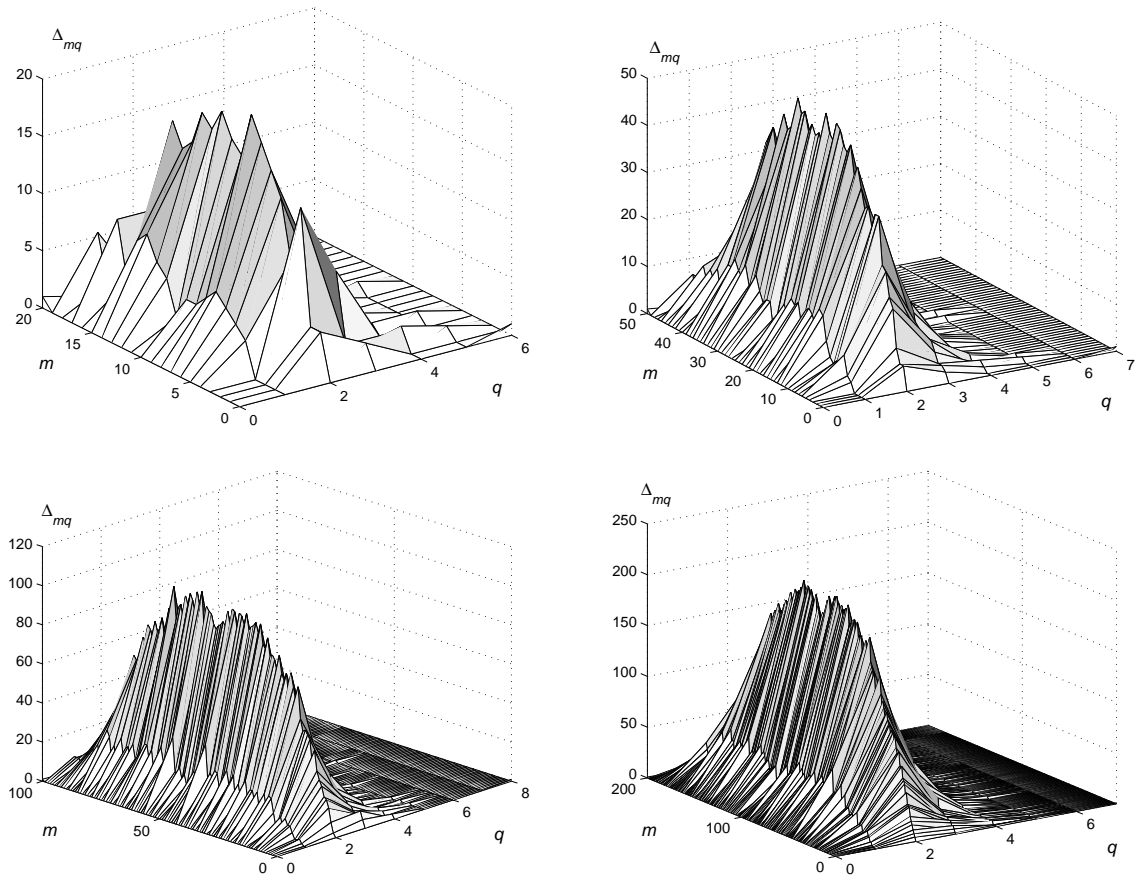
Вектор  $(\Delta_m)_{m=0}^L$  предлагается называть *профилем расслоения*, а вектор  $(\lambda_q)_{q=0}^L$  — *профилем связности* множества алгоритмов  $A$ .

Доказательство гипотезы сепарабельности пока остаётся открытой проблемой.

## Резюме

Метод порождающих и запрещающих множеств существенно упрощается, если для каждого алгоритма записать лишь необходимые условия того, что он будет получен в результате обучения. При этом вместо точных оценок вероятности переобучения получаются верхние оценки. Они существенно точнее VC-оценок, поскольку учитывают структуру графа расслоения–связности. Для некоторых семейств, в частности, для монотонной цепи алгоритмов, оценка расслоения–связности является точной. Вклад алгоритма в вероятность переобучения убывает экспоненциально с ростом связности и номера слоя, в котором находится алгоритм. Отсюда следует, что связные семейства менее склонны к переобучению, а для приближённого вычисления вероятности переобучения, возможно, будет достаточно взять алгоритмы из нескольких нижних слоёв.

<sup>8</sup>Вычислительные эксперименты выполнены Ильёй Решетняком.



**Рис. 9.2.** Профили расслоения–связности для двумерных выборок длины  $L = 20, 50, 100, 200$ . Профиль  $\Delta_{mq}$  — это количество алгоритмов с числом ошибок  $m$  на генеральной выборке и связностью  $q$ .

**В следующей лекции** мы покажем, что оценка расслоения–связности является точной для одного весьма нетривиального модельного семейства, обладающего наряду со свойствами расслоения и связности ещё и свойством размерности.

### Упражнения

**Задача 9.1 (3\*).** Какие ограничения необходимо наложить на семейство  $A$ , чтобы условие (9.1) было не только необходимым, но и достаточным, следовательно, оценка расслоения–связности (9.5) обращалась бы в равенство? Возможно ли сформулировать эти ограничения в терминах графа расслоения–связности?

**Задача 9.2 (5\*).** Обобщить оценку расслоения–связности (9.5) на случай, когда в графе расслоения–связности имеются «дырки», то есть для некоторых пар алгоритмов  $(a, b)$ ,  $a < b$ , не существует алгоритма  $c \in A$ , такого, что  $a \prec c \prec b$ .

**Задача 9.3 (10\*).** Обосновать гипотезу сепарабельности для линейных алгоритмов классификации.

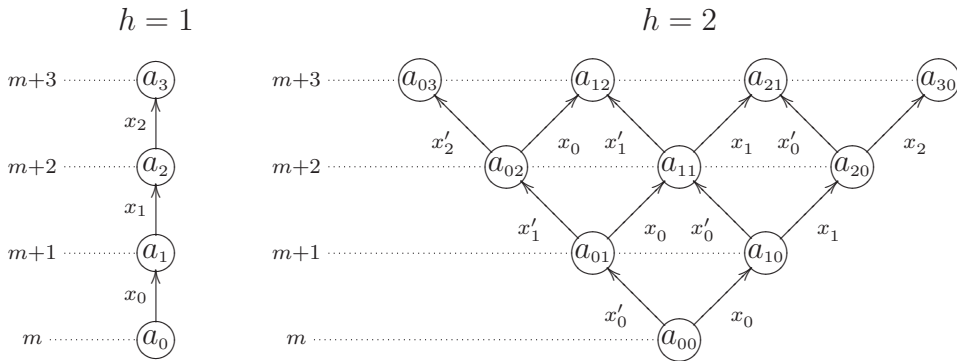
## 10 Многомерные сети алгоритмов

Существуют примеры нетривиальных модельных семейств алгоритмов, для которых оценки расслоения–связности являются точными. В частности, это многомерные монотонные сети алгоритмов. Эти семейства интересны ещё и тем, что они обладают не только расслоением и связностью, но и размерностью, то есть всеми ключевыми свойствами реальных семейств.

### §10.1 Двумерная монотонная сеть алгоритмов

В §8.4 мы рассмотрели монотонную цепь алгоритмов, которая является моделью однопараметрического семейства алгоритмов. Естественным обобщением этой модели на многомерный случай представляется семейство, граф расслоения–связности которого изоморфен прямоугольной решётке заданной размерности  $h$  или некоторому её невырожденному фрагменту той же размерности.

На рисунке показаны четыре нижних слоя графов расслоения–связности размерностей 1 (слева) и 2 (справа). Горизонтальные линии соответствуют слоям,  $m$  — число ошибок лучшего алгоритма на генеральной выборке. Объекты вдоль стрелок означают, что при переходе от нижнего алгоритма к верхнему ошибка появляется именно на данном объекте.



Рассмотрим произвольный алгоритм  $a_{uv}$  из слоя  $m+t = n(a_{uv}, \mathbb{X})$ . Число алгоритмов в этом слое равно  $t+1$ . Верхняя связность  $q(a_{uv}) = 2$  совпадает с размерностью  $h = 2$ . Неоптимальность  $r(a_{uv})$  равна  $t$ , поскольку  $a_{uv}$  отличается от лучшего алгоритма  $a_{00}$  ровно на  $t$  объектах. Подставив все эти данные в оценку (9.5) и заменив сумму по алгоритмам суммой по слоям, получим оценку расслоения–связности для  $H$  нижних слоёв двумерной сети алгоритмов:

$$Q_\varepsilon \leq \sum_{t=0}^H (t+1) \frac{C_{L-t-2}^{\ell-2}}{C_L^\ell} H_{L-t-2}^{\ell-2, m} \left( \frac{\ell}{L} (m+t-\varepsilon k) \right).$$

Пока это лишь предварительные соображения, показывающие, что иногда оценки расслоения–связности получаются очень просто. К тому же, эта оценка некорректно учитывает последний слой. Далее мы введём необходимый формализм, обобщим

оценку на случай произвольной размерности и аккуратно разберёмся с верхним словом. Мы также убедимся, что оценка расслоения–связности для  $h$ -мерной монотонной сети является точной. Первым эту оценку получил Павел Ботов [6].

## §10.2 Монотонная сеть произвольной размерности

Пусть  $J = (j_1, \dots, j_h)$  — целочисленный вектор индексов.

Положим  $|J| = j_1 + \dots + j_h$ .

Введём на векторах индексов отношение частичного порядка: для произвольных  $J = (j_1, \dots, j_h)$  и  $K = (k_1, \dots, k_h)$  положим  $J \leq K$ , если  $j_d \leq k_d$  для всех  $d = 1, \dots, h$ . Положим  $J < K$ , если  $J \leq K$  и  $J \neq K$ .

**Определение 10.1.** *Монотонной сетью алгоритмов размерности  $h$  и высоты  $H$  называется множество алгоритмов  $A = \{a_J : |J| \leq H\}$  с попарно различными векторами ошибок, такое, что*

- 1) если  $J < K$ , то  $a_J < a_K$ ;
- 2)  $n(a_J, \mathbb{X}) = t + |J|$  при некотором  $t$ .

Алгоритм  $a_0 = a_{(0, \dots, 0)}$  называется *лучшим* в  $A$ .

Данное определение может показаться неконструктивным, так как оно не описывает матрицу ошибок в явном виде. Однако следующая лемма показывает, что на самом деле матрица ошибок определена однозначно.

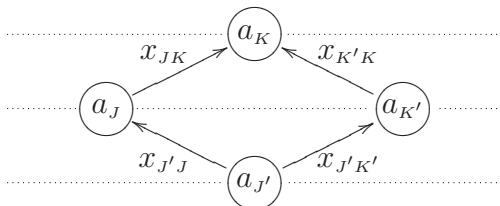
**Лемма 10.1.** *Множество объектов  $\mathbb{X}$  распадается на объекты трёх типов:*

- 1)  $t$  объектов, на которых ошибаются все алгоритмы;
- 2)  $Hh$  объектов  $\{x_t^d : t = 0, \dots, H-1, d = 1, \dots, h\}$  таких, что  $I(a_J, x_t^d) = [t < j_d]$ ;
- 3) все остальные объекты, на которых ни один из алгоритмов не ошибается.

**Доказательство.** 1. Множество объектов, на которых ошибаются все алгоритмы, совпадает с множеством объектов, на которых ошибается лучший алгоритм  $a_0$ , и их число равно  $t = n(a_0, \mathbb{X})$ . Что и утверждается в первом пункте леммы.

Введём на векторах индексов покомпонентное сложение и умножение на число:  $J + K = (j_1 + k_1, \dots, j_h + k_h)$  и  $cJ = (cj_1, \dots, cj_h)$  для произвольных  $J = (j_1, \dots, j_h)$  и  $K = (k_1, \dots, k_h)$ . Определим единичные векторы  $E_d = ([d=s])_{s=1}^h$ ,  $d = 1, \dots, h$ .

2. Возьмём произвольный алгоритм  $a_J \in A$  и две размерности  $\{d, s\} \subseteq \{1, \dots, h\}$  такие, что  $j_d < H$ ,  $j_s > 0$ . Построим векторы индексов  $K = J + E_d$ ,  $J' = J - E_s$ ,  $K' = K - E_s$ . Четыре алгоритма  $a_J, a_K, a_{J'}, a_{K'}$  образуют граф расслоения–связности (напомним, что если алгоритм  $a$  предшествует алгоритму  $b$ ,  $a \prec b$ , то через  $x_{ab}$  обозначается тот единственный объект, для которого  $I(a, x_{ab}) = 0$  и  $I(b, x_{ab}) = 1$ ):



Алгоритмы  $a_{J'}$  и  $a_K$  отличаются только на двух объектах, поэтому среди четырёх объектов  $x_{J'J}$ ,  $x_{J'K'}$ ,  $x_{JK}$ ,  $x_{K'K}$  имеются только два различных. Единственный допустимый вариант совпадений — когда  $x_{JK} = x_{J'K'}$  и  $x_{J'J} = x_{K'K}$ . Продолжая по очереди уменьшать в векторах  $J$  и  $K$  все координаты  $j_1, \dots, j_h$  кроме  $j_d$ , придём в итоге к тому, что два алгоритма  $a_{j_d E_d} \prec a_{(j_d+1)E_d}$  также отличаются на объекте  $x_{JK}$ . Таким образом, объект  $x_{JK}$  определяется только координатой  $d \in \{1, \dots, h\}$  и значением индекса  $t = j_d \in \{0, \dots, H-1\}$ , и не зависит от остальных координат вектора  $J$ . Обозначим этот объект через  $x_t^d$ .

Докажем от противного, что среди объектов  $x_t^d$  нет совпадающих. Пусть  $x_t^d = x_{t'}^{d'}$  и для определённости  $t < t'$ . Тогда  $a_{tE_d} < a_{t'E_{d'}}$ , следовательно  $I(a_{t'E_{d'}}, x_t^d) = 1$ . С другой стороны,  $I(a_{t'E_{d'}}, x_{t'}^{d'}) = 0$ . Поэтому объекты  $x_t^d$  и  $x_{t'}^{d'}$  не могут совпадать. Предположим теперь, что  $x_t^d = x_{t'}^{d'}$  при  $d \neq d'$ . Возьмём алгоритм  $a_J$ ,  $J = tE_d + t'E_{d'}$ . Алгоритмы  $a_{J+E_d}$  и  $a_{J+E_{d'}}$  отличаются от  $a_J$  на объектах  $x_t^d$  и  $x_{t'}^{d'}$  соответственно и должны иметь различные векторы ошибок по определению монотонной сети. Поэтому объекты  $x_t^d$  и  $x_{t'}^{d'}$ , опять-таки, не могут совпадать.

Итак, произвольный алгоритм  $a_J \in A$  допускает ошибки на тех и только тех объектах  $x_t^d$ , у которых  $t < j_d$ , что и утверждается во втором пункте леммы.

3. В выборке могут оставаться объекты  $x$ , на которых алгоритм  $a_0$  не допускает ошибку, и которые не совпадают ни с одним из  $x_t^d$ . Если бы один из алгоритмов  $a \in A$  допускал ошибку на таком объекте  $x$ , то существовали бы алгоритмы  $b, c \in A$ , такие, что  $b \prec c \leq a$  и  $x = x_{bc}$ . Но тогда объект  $x$  был бы одним из объектов  $x_t^d$ , как было показано выше. Таким образом, ни один из алгоритмов  $a \in A$  не может допускать ошибку на таком объекте  $x$ . Что и утверждается во третьем пункте леммы.

Лемма доказана. ■

**Следствие 10.1.1.** Для любого  $a_J \in A$ ,  $J = (j_1, \dots, j_d, \dots, j_h)$  и любого  $d \in \{1, \dots, h\}$  увеличение индекса  $t = j_d$  на единицу,  $K = (j_1, \dots, j_d+1, \dots, j_h)$  приводит к появлению ошибки на объекте  $x_t^d$ :

$$a_J \prec a_K; \quad I(a_J, x_t^d) = 0, \quad I(a_K, x_t^d) = 1.$$

**Интерпретация.** Монотонная сеть размерности  $h$  и высоты  $H$  — это модель семейства алгоритмов с  $h$  непрерывными параметрами. Модельное предположение заключается в том, что по мере увеличения  $d$ -го параметра ошибки возникают последовательно на объектах  $x_0^d, \dots, x_{H-1}^d$ , независимо от значений остальных параметров. В реальных задачах это требование как правило, не выполняется. В то же время, известны примеры, когда монотонная сеть порождается искусственной выборкой и реальным семейством алгоритмов (упражнения 10.1, 10.2).

**Размерные характеристики.** Из Леммы 10.1 следует, что необходимым условием существования монотонной сети является ограничение  $m + Hh \leq L$ .

Число алгоритмов в слое  $m + t$  равно числу неотрицательных целочисленных векторов индексов  $J = (j_1, \dots, j_h)$  таких, что  $|J| = j_1 + \dots + j_h = t$ . Оно же есть

число способов выбрать  $t$  предметов из  $h$  с повторениями,  $\bar{C}_h^t = C_{h+t-1}^{h-1}$ . Просуммировав число алгоритмов по слоям и воспользовавшись известным комбинаторным тождеством, нетрудно найти число алгоритмов во всей сети:

$$|A| = C_{h-1}^{h-1} + \dots + C_{h+H-1}^{h-1} = C_{H+h}^h.$$

### §10.3 Точная оценка расслоения–связности

Из формул (9.3), (9.4) и графа расслоения–связности следует, что для любого алгоритма  $a_J$  из монотонной сети  $A$

$$\begin{aligned} X_J &= \{x_t^d : t = j_d, d = 1, \dots, h\} \text{ — порождающее множество;} \\ X'_J &= \{x_t^d : t < j_d, d = 1, \dots, h\} \text{ — запрещающее множество.} \end{aligned}$$

Лемма 9.1 даёт лишь необходимое условие того, что алгоритм  $a_J$  будет получен как результат обучения:  $[\mu X = a_J] \leq [X_J \subseteq X][X'_J \subseteq \bar{X}]$ . Однако в случае многомерной монотонной сети необходимое условие является также и достаточным.

**Лемма 10.2.** *Если  $\mu$  — пессимистичная минимизация эмпирического риска, то*

$$[\mu X = a_J] = [X_J \subseteq X][X'_J \subseteq \bar{X}] \text{ для каждого } a_J \in A.$$

**Доказательство.** Докажем, что если  $X_J \subseteq X$  и  $X'_J \subseteq \bar{X}$ , то только алгоритм  $a_J$  может быть результатом обучения.

Алгоритм  $a_J$  допускает ошибки только на объектах из  $X'_J$ , но все они лежат в контроле  $\bar{X}$ , поэтому  $n(a_J, X) = 0$ .

Рассмотрим произвольный алгоритм  $a_K$  такой, что  $K \not\leq J$ . Это означает, что  $k_d > j_d$  для некоторой координаты  $d$ . Но тогда  $a_K$  допускает ошибку на объекте  $x_{j_d}^d$ , который лежит в  $X_J$ , следовательно, и в обучении  $X$ . Поэтому алгоритм  $a_K$  не может быть выбран методом  $\mu$ , минимизирующим эмпирический риск.

Рассмотрим произвольный алгоритм  $a_K$  такой, что  $K < J$ . Тогда оба алгоритма,  $a_J$  и  $a_K$  не допускают ошибок на обучении  $X$ . Допустим, что  $k_d < j_d$  для некоторой координаты  $d$ . Тогда на контрольном объекте  $x_{k_d}^d$  из двух алгоритмов только  $a_J$  допускает ошибку, поэтому метод  $\mu$ , будучи пессимистичным, выберет алгоритм  $a_J$ .

Таким образом, только алгоритм  $a_J$  может быть результатом обучения. ■

**Теорема 10.3 (О монотонной сети алгоритмов).** *Пусть  $A$  — монотонная сеть размерности  $h$  и высоты  $H$ , метод  $\mu$  является пессимистичной минимизацией эмпирического риска и выполнено условие  $m + Hh \leq L$ . Тогда*

$$Q_\varepsilon(\mu, \mathbb{X}) = \sum_{t=0}^{\min\{H, k\}} \frac{C_{h+t-1}^t C_{L-q-t}^{\ell-q}}{C_L^\ell} H_{L-q-t}^{\ell-q, m} \left( \frac{\ell}{L} (m + t - \varepsilon k) \right), \quad q = [t < H]h.$$

**Доказательство.** Возьмём произвольный алгоритм  $a_J$  и рассмотрим три случая.

1. Если  $|J| > k$ , то число ошибок алгоритма  $a_J$  на запрещающих объектах  $X'_J$  превышает длину контрольной выборки. Часть ошибок обязательно окажется в обучающей подвыборке  $X$ , и метод  $\mu$  выберет другой алгоритм. В этом случае

$$[\mu X = a_J] = 0.$$

2. Если  $|J| \leq k$  и  $|J| = H$ , то есть алгоритм  $a_J$  находится в последнем слое, то порождающих объектов для  $a_J$  не существует, следовательно,

$$[\mu X = a_J] = [X'_J \subseteq \bar{X}].$$

В этом случае верхняя связность  $q(a_J) = 0$ , неоптимальность  $r(a_J) = |J|$ .

3. Если  $|J| \leq k$  и  $|J| < H$ , то реализуется основной случай из Леммы 10.2:

$$[\mu X = a_J] = [X_J \subseteq X][X'_J \subseteq \bar{X}].$$

В этом случае верхняя связность  $q(a_J) = |X_J| = h$ , неоптимальность  $r(a_J) = |J|$ .

Применим оценку расслоения–связности (9.5), которая в данном случае является точным равенством:

$$Q_\varepsilon = \sum_{J: |J| \leq H} P_J H_{L-q-r}^{\ell-q, m+|J|-r} \left( \frac{\ell}{L} (m + |J| - \varepsilon k) \right), \quad P_J = \frac{C_{L-q-r}^{\ell-q}}{C_L^\ell},$$

где  $P_J = \mathbf{P}[\mu X = a_J]$  — вероятность получить алгоритм  $a_J$  в результате обучения,  $q = q(a_J)$  — верхняя связность,  $r = r(a_J) = |J|$  — неоптимальность алгоритма  $a_J$ .

Верхняя связность и неоптимальность в каждом слое одинаковы у всех алгоритмов. Это позволяет выразить вероятность того, что методом  $\mu$  будет получен некоторый алгоритм  $a_J$  из слоя  $m + t$ , где  $t = |J|$ :

$$P_t = \mathbf{P}[n(\mu X, \mathbb{X}) = m+t] = C_{h+t-1}^t P_J = \frac{C_{h+t-1}^t C_{L-q-t}^{\ell-q}}{C_L^\ell}, \quad (10.1)$$

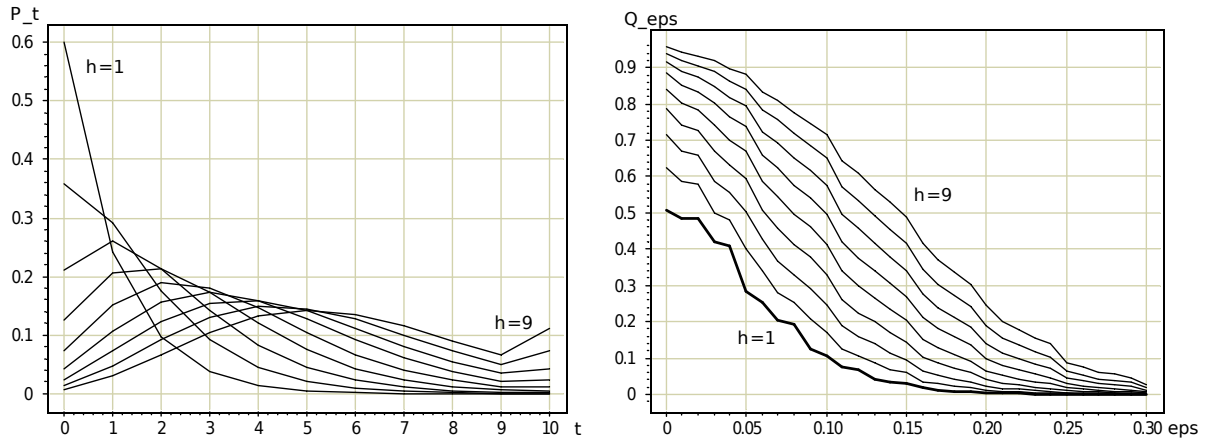
и затем заменить сумму по алгоритмам суммой по слоям  $t = 0, \dots, H$ . Наконец, остаётся учесть, что при  $t > k$  алгоритмы вообще не вносят вклад в оценку, а при  $t = H$  обнуляется верхняя связность  $q$ , которая в остальных случаях равна  $h$ .

Таким образом,

$$Q_\varepsilon = \sum_{t=0}^{\min\{H, k\}} P_t H_{L-q-t}^{\ell-q, m} \left( \frac{\ell}{L} (m + t - \varepsilon k) \right), \quad q = [t < H]h,$$

что и требовалось доказать. ■





**Рис. 10.1.** Зависимости  $P_t$  от  $t$  (слева) и  $Q_\epsilon$  от  $\epsilon$  (справа) для монотонных сетей размерностей  $h = 1, \dots, 9$  при  $L = 100$ ,  $\ell = 60$ ,  $m = 10$ ,  $H = 10$ .

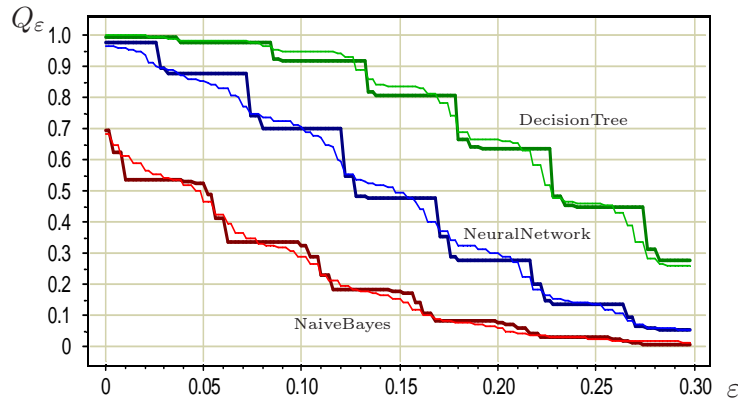
### §10.4 Вычислительные эксперименты

**Эксперимент 1: вычисление оценки расслоения–связности.** На рис. 10.1 слева показан график зависимости вероятности  $P_t$  получить алгоритм из слоя  $m + t$ , вычисленной по формуле (10.1), от числа  $t$ . Справа показан график зависимости вероятности переобучения  $Q_\epsilon$  от порога  $\epsilon$ . На каждом графике изображены 9 кривых, соответствующих различным значениям размерности  $h = 1, \dots, 9$ .

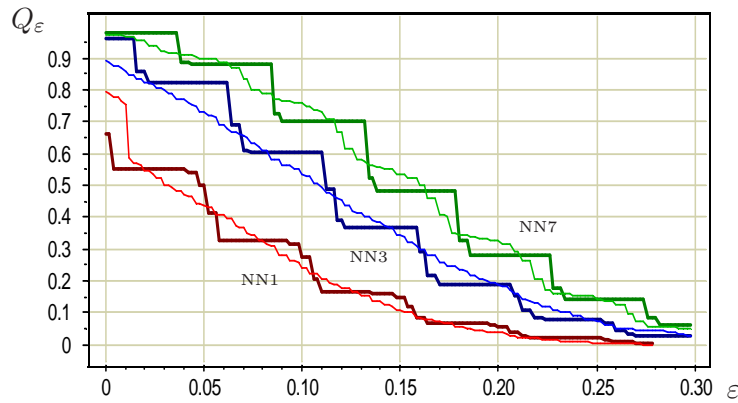
Зависимость  $P_t$  многомерной сети на отрезке  $[0, H - 1]$  унимодальна. Вероятность последнего слоя немного выше, так как для него верхняя связность равна нулю. При росте размерности  $h$  положение максимума  $P_t$  смещается вправо и растёт вероятность переобучения. Это означает, что рост связности, который приводит к экспоненциальному уменьшению вероятностей  $P_J$  отдельных алгоритмов  $a_J$ , всё же компенсируется ростом мощности слоёв.

**Эксперимент 2: сравнение монотонных сетей с реальными семействами.** На основе платформы RapidMiner были получены экспериментальные зависимости  $Q_\epsilon$  от  $\epsilon$  для задач из репозитория UCI (sonar, breast-cancer-wisconsin) на методах классификации NaiveBayes, SVM, DecisionTree, NeuralNetwork [6]. Экспериментальные кривые переобученности  $\hat{Q}_\epsilon$ , полученные методом Монте-Карло, аппроксимировались кривыми переобученности монотонных сетей путём подбора размерности  $h$  при фиксированных  $L$ ,  $\ell$  и  $m$ . Оказалось, что чем сложнее семейство алгоритмов, тем выше проходит кривая переобученности и тем выше размерность  $h$  аппроксимирующей монотонной сети, см. рис. 10.2, 10.3, 10.4 («более ступенчатые» кривые соответствуют монотонным сетям, «более гладкие» — реальным семействам).

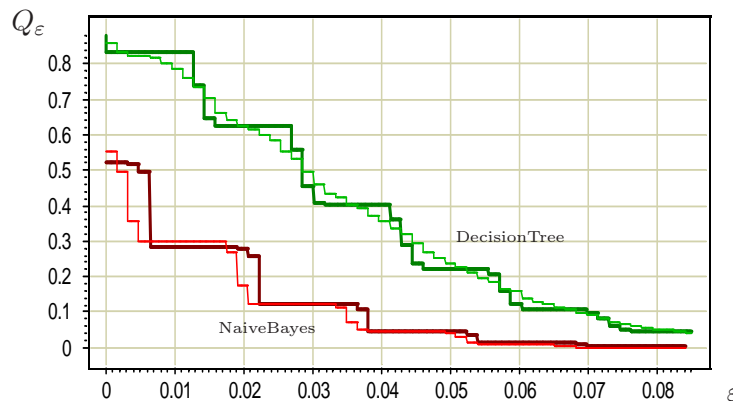
Примечателен тот факт, что на задаче breast-cancer-wisconsin метод NaiveBayes показал кривую переобученности, эквивалентную монотонной сети размерности 1, рис. 10.4. Эффективная размерность данной задачи и в самом деле близка к 1, так как очень хорошим разделяющим признаком является сумма всех 9 признаков. DecisionTree на тех же данных не смог найти этот признак и показал большую размерность, большую переобученность и большую ошибку на контроле.



**Рис. 10.2.** Вероятность переобучения  $Q_\epsilon$  алгоритмов NaiveBayes, NeuralNetwork, DecisionTree и монотонных сетей размерностей, соответственно,  $h = 8, 28, 46$ . Средняя частота ошибок на обучении/контроле, соответственно: 0.28/0.32; 0.05/0.20; 0.02/0.26. Задача sonar, 20 признаков,  $L = 208$ ,  $\ell = 187$ .



**Рис. 10.3.** Вероятность переобучения  $Q_\epsilon$  алгоритмов NeuralNetwork с числом нейронов в скрытом слое 1, 3, 7 и монотонных сетей размерностей, соответственно,  $h = 8, 22, 30$ . Средняя частота ошибок на обучении/контроле, соответственно: 0.29/0.34; 0.12/0.23; 0.03/0.19. Задача sonar, 20 признаков,  $L = 208$ ,  $\ell = 187$ .



**Рис. 10.4.** Вероятность переобучения  $Q_\epsilon$  алгоритмов NaiveBayes, DecisionTree и монотонных сетей размерности  $h = 1, 20$ . Средняя частота ошибок на обучении/контроле, соответственно: 0.04/0.04; 0.03/0.06. Задача breast-cancer-wisconsin, 9 признаков,  $L = 700$ ,  $\ell = 630$ .

## Резюме

Точные оценки вероятности переобучения к данному моменту известны для ряда модельных семейств алгоритмов. В их числе многомерные монотонные сети алгоритмов, обладающие свойствами расслоения, связности и размерности, что сближает их с реальными семействами. Оценка расслоения–связности для многомерной монотонной сети является точной.

В экспериментах кривые вероятности переобучения реальных семейств неплохо приближаются с помощью многомерных монотонных сетей при соответствующем подборе размерности, которую можно интерпретировать как «эффективную размерность» задачи.

**В следующей лекции** мы применим оценку расслоения–связности к семейству конъюнктивных логических правил над вещественными признаками. Это первый пример не модельного семейства. Оно широко используется в логических алгоритмах классификации и реально применяется при решении прикладных задач.

## Упражнения

**Задача 10.1 (1).** Привести пример выборки в  $\mathbb{R}^2$ , для которой множество линейных классификаторов (не обязательно всех возможных) порождает монотонную сеть алгоритмов.

**Задача 10.2 (3\*).** Привести пример выборки в  $\mathbb{R}^h$ , для которой множество линейных классификаторов (не обязательно всех возможных) порождает монотонную сеть алгоритмов.

**Задача 10.3 (5\*).** Сформулировать требования к семейству алгоритмов, при которых оценка расслоения–связности является точной.

**Задача 10.4 (5).** Получить оценку вероятности переобучения для многомерной монотонной сети, когда по каждой размерности  $d$  задана своя высота  $H_d$ .

**Задача 10.5 (8).** Получить оценку вероятности переобучения для унимодальной  $h$ -мерной сети.

**Задача 10.6 (5).** Получить оценку вероятности переобучения для пучка  $h$  монотонных цепей длины  $H$ .

## 11 Конъюнктивные логические закономерности

Мы добрались, наконец, до первого практического применения комбинаторной теории переобучения. А именно, оценки расслоения–связности будут применены для улучшения *логических методов классификации*.

### §11.1 Логические методы классификации

Рассмотрим задачу классификации. Допустим, что каждому объекту  $x_i \in \mathbb{X}$  соответствует *правильный ответ*  $y_i \in \mathbb{Y}$ , где  $\mathbb{Y}$  — конечное множество имён классов. Объекты описываются набором  $n$  числовых *признаков*  $f_j: \mathbb{X} \rightarrow \mathbb{R}$ ,  $j = 1, \dots, n$ .

Логические методы классификации основаны на построении композиций информативных, хорошо интерпретируемых логических закономерностей.

**Логические правила и требование интерпретируемости.** *Предикатом* будем называть произвольное отображение вида  $r: \mathbb{X} \rightarrow \{0, 1\}$ . Если  $r(x) = 1$ , то будем говорить, что предикат *выделяет* объект  $x$ .

*Правилом* (rule) будем называть предикат из некоторого фиксированного семейства предикатов  $R$ . Правила отличаются от предикатов тем, что обладают свойством *интерпретируемости* — допускают запись на естественном языке в терминах предметной области, достаточно просты и понятны прикладным специалистам. Эти требования формализуются в самой конструкции семейства  $R$ . Мы рассмотрим только один вид правил, пожалуй, самый распространённый на практике — конъюнкции (логическое И) элементарных пороговых предикатов:

$$r(x; \theta) = \prod_{j \in J} [f_j(x) \lesseqgtr_j \theta^j], \quad (11.1)$$

где  $J \subseteq \{1, \dots, n\}$  — подмножество признаков,  $\theta^j \in \mathbb{R}$  — *порог* по  $j$ -му признаку,  $\theta = (\theta^j)_{j \in J}$  — *вектор порогов*,  $\lesseqgtr_j$  — один из знаков сравнения  $\{\leq, \geq, =\}$ . Число  $|J|$  называется *рангом конъюнкции*. Обычно оно ограничивается сверху в угоду всё той же интерпретируемости. Как утверждают психологи, людям трудно помнить и понимать правила, содержащие более 7 условий. На практике ограничение на  $|J|$  устанавливается прикладными специалистами исходя из специфики задачи.

**Примеры закономерностей.** Из задачи медицинского прогнозирования: если возраст пациента выше 60 лет И ранее он перенёс инфаркт, то операцию делать не стоит. Из задачи кредитования физических лиц: если заёмщик указал в анкете свой домашний телефон И его зарплата превышает \$2000 в месяц И сумма кредита не превышает \$5 000, то кредит можно выдать. Из задачи распознавания спама: если в письме присутствует слово «бесплатно» И указан московский телефонный номер И домен отправителя находится в Китае, то это спам.

**Понятия закономерности и информативности.** *Закономерностью* класса  $y \in \mathbb{Y}$  будем называть правило  $r \in R$ , выделяющее на заданной выборке  $X \subseteq \mathbb{X}$  достаточно

много объектов класса  $y$  и мало объектов всех остальных классов. Для формализации этого требования вводят два критерия:  $p(r, X)$  — число *положительных примеров* — объектов класса  $y$ , выделяемых правилом  $r$ , и  $n(r, X)$  — число *отрицательных примеров* — объектов всех остальных классов, выделяемых правилом  $r$ . Для поиска закономерностей в семействе правил  $R$  по обучающей выборке  $X$  естественно ставить задачу двухкритериальной оптимизации:

$$\begin{aligned} p(r, X) &= \#\{x_i \in X \mid r(x_i) = 1, y_i = y\} \rightarrow \max; \\ n(r, X) &= \#\{x_i \in X \mid r(x_i) = 1, y_i \neq y\} \rightarrow \min. \end{aligned}$$

На практике два критерия часто сворачивают в один *критерий информативности* и решают задачу оптимизации  $\mathcal{H}(p, n) \rightarrow \max$ . Примерами критериев информативности являются: простые свёртки типа  $p - \gamma n$  [26], критерий бустинга  $\sqrt{p} - \sqrt{n}$  [98], энтропийный критерий, индекс Джини, точный тест Фишера, тест  $\chi^2$ , тест  $\omega^2$ , и другие [85]. В обзорной работе [68] перечислено около двух десятков различных критериев информативности. Многие критерии оценивают степень неслучайности разбиения выборки  $X$  на подмножества положительных и отрицательных примеров относительно исходного разбиения выборки  $X$  на классы. Однако ни один из критериев нельзя назвать наиболее обоснованным или безусловно предпочтительным. Выбор критерия информативности является эвристикой.

Обозначим число положительных и отрицательных примеров в выборке  $X$  через

$$\begin{aligned} P(X) &= \#\{x_i \in X : y_i = y\}; \\ N(X) &= \#\{x_i \in X : y_i \neq y\}. \end{aligned}$$

**Пример 11.1 (точный тест Фишера).** Если принять в качестве нулевой гипотезы предположение, что предикаты  $r(x)$  и  $[y(x) = y]$  являются независимыми случайными величинами, то при фиксированном числе выделяемых значений  $(p + n)$  число выделяемых отрицательных примеров  $n$  подчиняется гипергеометрическому распределению  $h_{P+N}^{p+n, N}(n)$ . Достижимый уровень значимости даётся функцией гипергеометрического распределения. Он равен вероятности чисто случайной реализации наблюдаемых значений  $p$  и  $n$ . Чем он меньше, тем менее правдоподобна нулевая гипотеза. Поэтому он может служить мерой неслучайности величины  $r(x)$ , а неслучайность можно трактовать как закономерность. В качестве максимизируемого критерия информативности берут минус логарифм достигаемого уровня значимости:

$$\mathcal{H}(p, n) = -\log H_{P+N}^{p+n, N}(n).$$

**Пример 11.2 (энтропийный критерий).** Асимптотическим приближением гипергеометрического критерия является *энтропийный критерий информативности* или *выигрыш информации* (information gain, IGain) [89]:

$$\text{IGain}(p, n) = h\left(\frac{P}{P+N}\right) - \frac{p+n}{|X|} h\left(\frac{p}{p+n}\right) - \frac{|X| - p - n}{|X|} h\left(\frac{P-p}{|X| - p - n}\right),$$

где  $h(q) = -q \log_2 q - (1 - q) \log_2(1 - q)$  — функция энтропии пары исходов с вероятностями  $q$  и  $1 - q$ .

**Пример 11.3 (критерий Джини).** На практике используют также *индекс Джини* (Gini impurity), отличающийся от IGain только функцией  $h(q) = 2q(1-q)$ , которая на самом деле очень близка к функции энтропии [58].

**Композиции закономерностей.** Каждая закономерность выделяет лишь часть выборки и относит её к одному из классов. Поэтому закономерности можно рассматривать как «неполноценные» классификаторы, а «полноценный» классификатор строить как композицию закономерностей. Одним из наиболее распространённых типов композиций является *взвешенное голосование*:

$$a(x) = \arg \max_{y \in \mathbb{Y}} \sum_{r \in R_y} w_r r(x),$$

где  $R_y$  — множество закономерностей класса  $y$ ,  $w_r \geq 0$  — вес закономерности  $r$ .

Построение таких композиций можно вести по-разному. Одна из стратегий заключается в том, чтобы сначала построить списки закономерностей  $R_y$ , затем, рассматривая закономерности  $r(x)$  как новые признаки, построить на них линейный классификатор любым из известных методов, например, логистической регрессией или методом опорных векторов [69]. Другая стратегия заключается в том, чтобы строить закономерности по очереди, и для каждой закономерности  $r$  сразу определять вес  $w_r$ . Так, в частности, работают алгоритмы бустинга [64].

**Переобученность закономерностей.** Итак, имеется три типа параметров, обучаемых по выборке, и, соответственно, три источника переобучения:

- 1) пороги  $\theta^j$ ,  $j \in J$  для каждого набора признаков  $J$ ;
- 2) набор признаков  $J$  для каждой закономерности  $r$ ;
- 3) вес  $w_r$  для каждой закономерности  $r$ .

Будем предполагать, что хороший алгоритм  $a(x)$  возможно построить только из непереобученных закономерностей  $r(x)$ . Поэтому нашей целью будет уменьшение переобучения отдельных закономерностей. Проблемой переобучения весов  $w_r$  мы заниматься не будем, отчасти потому, что она лучше исследована [51, 97, 79].

Обычные критерии информативности не учитывают переобучение порогов  $\theta^j$ . Поэтому в результате оптимизации  $p(r, X)$  и  $n(r, X)$  по обучающей выборке  $X$  соответствующие значения  $p' = p(r, \bar{X})$  и  $n' = n(r, \bar{X})$  на контрольной выборке  $\bar{X}$  могут сильно отличаться от оптимальных. Наш замысел заключается в том, чтобы

- 1) получить оценку расслоения–связности для семейства правил (11.1);
- 2) обратить эту оценку, чтобы оценить  $p', n'$  на контроле через  $p, n$  на обучении;

$$\begin{aligned} \frac{1}{k} p(r, \bar{X}) &\geq \frac{1}{\ell} p(r, X) - \varepsilon_p(\eta), \quad \text{с вероятностью } 1 - \eta; \\ \frac{1}{k} n(r, \bar{X}) &\leq \frac{1}{\ell} n(r, X) + \varepsilon_n(\eta), \quad \text{с вероятностью } 1 - \eta; \end{aligned}$$

- 3) подставить оценки  $p', n'$  в критерий информативности вместо  $p, n$ :

$$\mathcal{H}'(p, n) = \mathcal{H}(p - \ell\varepsilon_p(\eta), n + \ell\varepsilon_n(\eta));$$

- 4) использовать модифицированный критерий  $\mathcal{H}'$  для отбора закономерностей.

Модифицированный критерий  $\mathcal{H}'$  оценивает не сами правила, а наборы признаков  $J$  с учётом возможного переобучения порогов.

Минус предлагаемой схемы в том, что она не учитывает переобучение при выборе подмножеств  $J$  и весов  $w_r$ , значит, не даёт окончательного решения проблемы переобучения для алгоритма  $a(x)$ . Плюс же в том, что она ничего не меняет в методах поиска логических закономерностей, за исключением критерия информативности, поэтому её легко встраивать в имеющиеся библиотеки алгоритмов.

Начнём осуществление нашего замысла с описания множества векторов ошибок, индуцируемых конъюнкциями вида (11.1).

## §11.2 Конъюнкции элементарных пороговых правил

Оценки вероятности переобучения, полученные в общем случае для алгоритмов, легко переносятся на логические правила, если определить индикатор ошибки как  $I(r, x_i) = [r(x_i) \neq [y_i=y]]$ , где  $y \in \mathbb{Y}$  — фиксированный класс. После этого для правил, как и ранее для алгоритмов, вводятся векторы ошибок  $\vec{r} = (I(r, x_i))_{i=1}^L$ , отношения порядка и предшествования.

Будем полагать, что в правилах вида (11.1) множество признаков  $J$  фиксировано, и все знаки сравнения суть  $\leq$ . Наша задача заключается в том, чтобы оценить вероятность переобучения, возникающего в результате оптимизации *вектора порогов*  $\theta = (\theta^j)_{j \in J}$  по обучающей выборке  $X$ .

**Описание классов эквивалентности правил** в терминах стандартных представителей и граничных подмножеств, рассматриваемое ниже, предложено Андреем Ивахненко. Им же подготовлены все иллюстрации, выполнена программная реализация и эксперименты на реальных задачах классификации.

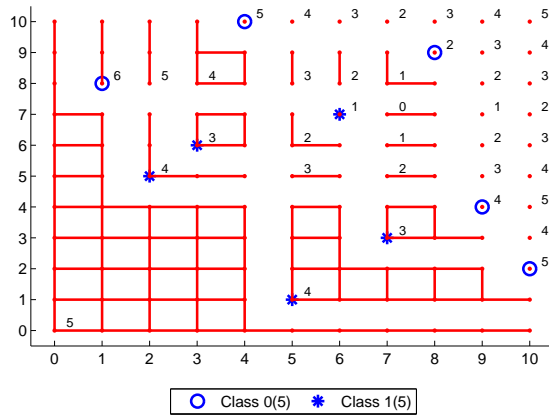
Допустим, что значения  $x_i^j = f_j(x_i)$  каждого признака  $j \in J$  на всех объектах  $x_i \in \mathbb{X}$  попарно различны. Без ограничения общности будем предполагать, что все признаки принимают целые значения  $1, \dots, L$ . Значения порогов  $\theta^j$  в правилах (11.1) также выбираются из целых значений  $0, \dots, L$ . Таким образом, и объекты, и правила описываются элементами одного и того же дискретного множества  $\Theta = \{0, \dots, L\}^{|J|}$ . Будем полагать, что  $R = \Theta$  и  $\mathbb{X} \subset \Theta$ .

Индикатор ошибки индуцирует на множестве правил  $R$  *классы эквивалентности*. Два правила эквивалентны, если их векторы ошибок совпадают. Векторы порогов  $\theta, \theta' \in \Theta$  эквивалентных правил  $r(x, \theta), r(x, \theta')$  могут не совпадать.

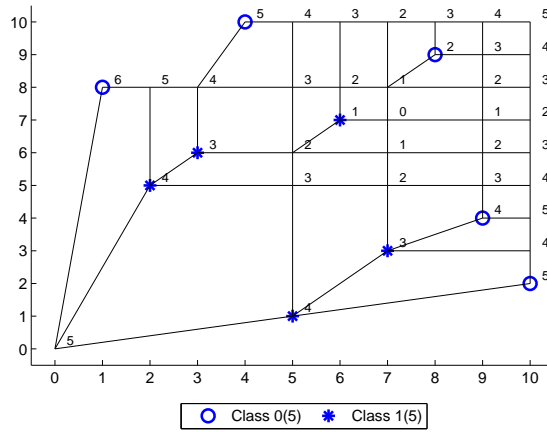
Для произвольных векторов  $u = (u^j)_{j \in J}, v = (v^j)_{j \in J}$  из  $\Theta$  введём естественное отношение порядка:  $(u \leq v) \leftrightarrow \forall j \in J (u^j \leq v^j)$ . Положим  $(u < v) \leftrightarrow (u \leq v \text{ и } u \neq v)$ .

**Пример 11.4.** На рис. 11.1 показан пример задачи с  $n = 2$  признаками,  $L = 10$  объектами и семейство правил  $r(x; \theta) = [x \leq \theta] = [x^1 \leq \theta^1] [x^2 \leq \theta^2]$ . Каждому правилу соответствует узел целочисленной прямоугольной сетки  $(\theta^1, \theta^2) \in \Theta = \{0, \dots, L\}^2$ .

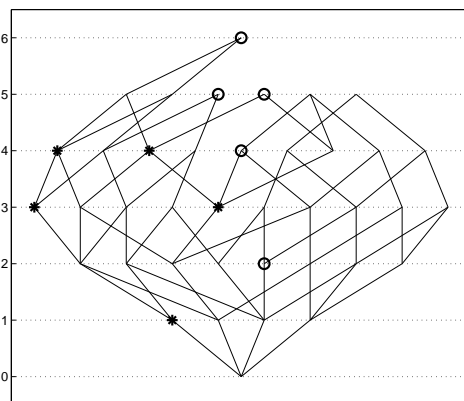
**Лемма 11.1.** Пусть  $E \subseteq R$  — класс эквивалентности правил. Тогда классу  $E$  принадлежит правило  $r(x; \theta_E)$ , где  $\theta_E^j = \min_{\theta \in E} \theta^j$  для всех  $j \in J$ .



**Рис. 11.1.** Двумерная выборка из  $L = 10$  объектов, по 5 объектов в каждом классе. Объекты отмечены крупными точками. Правилам соответствуют мелкие точки в узлах сетки. Эквивалентные правила соединены отрезками. Рядом с каждым классом эквивалентности указано число ошибок на генеральной выборке  $n(r, \mathbb{X})$ .



**Рис. 11.2.** Граф связей между стандартными представителями классов эквивалентных правил для выборки, представленной на Рис. 11.1.



**Рис. 11.3.** Граф расслоения–связности, изоморфный графу связей на Рис. 11.2. По вертикальной оси отложено число ошибок правил  $n(r, \mathbb{X})$ .



**Доказательство.** В силу эквивалентности и бинарности правил из  $E$ , предикат  $r_E(x) = \prod_{\theta \in E} r(x; \theta)$  принимает на всех объектах  $x \in \mathbb{X}$  те же значения, что и любое правило  $\theta$  из  $E$ ,  $r_E(x) = r(x; \theta)$ . Кроме того, предикат  $r_E$  представим в виде (11.1):

$$r_E(x) = \prod_{\theta \in E} \prod_{j \in J} [x^j \leq \theta^j] = \prod_{j \in J} [x^j \leq \min_{\theta \in E} \theta^j] = \prod_{j \in J} [x^j \leq \theta_E^j] = r(x; \theta_E).$$

Таким образом, правило  $r(x; \theta_E)$  также принадлежит  $E$ . Лемма доказана.  $\blacksquare$

Будем называть правило  $r_E(x) = r(x; \theta_E)$  *стандартным представителем* класса эквивалентности  $E$ . На рис. 11.1 стандартные представители соответствуют левым нижним точкам каждого класса эквивалентности:  $(0, 0)$ ,  $(1, 8)$ ,  $(2, 5)$ ,  $(5, 1)$ , и т. д.

*Граничной точкой* подмножества  $S \subseteq \mathbb{X}$  назовём вектор  $\theta_S$  с координатами  $\theta_S^j = \max_{x \in S} x^j$ ,  $j \in J$ . Заметим, что  $r(x, \theta_S) = 1$  для любого  $x \in S$ .

*Граничным объектом* подмножества  $S$  назовём объект  $x \in S$ , такой, что  $x^j = \theta_S^j$  при некотором  $j$ .

*Граничным подмножеством* назовём такое подмножество  $S \subseteq \mathbb{X}$ , все элементы которого являются его граничными объектами.

Пустое множество будем считать граничным с граничной точкой  $\theta_\emptyset = 0$ ,  $j \in J$ .

**Теорема 11.2.** *Каждому классу эквивалентности  $E$  взаимно однозначно соответствует граничное подмножество  $S$ , причём  $\theta_E = \theta_S$ .*

**Доказательство.** Рассмотрим произвольный класс эквивалентности  $E$  со стандартным представителем  $r(x; \theta_E)$ . Зафиксируем произвольный признак  $j \in J$ . В силу Леммы 11.1 уменьшение порога  $\theta_E^j$  на единицу приводит к уменьшению значения  $r(x; \theta_E)$  с 1 до 0 на некотором объекте  $x \in \mathbb{X}$ , таком, что  $x \leq \theta_E$  и  $x^j = \theta_E^j$ . Иначе мы не вышли бы за пределы класса эквивалентности  $E$ , и  $\theta_E$  не мог бы быть стандартным представителем. Из предположения о попарной различности значений каждого признака на объектах из  $\mathbb{X}$  следует, что такой объект  $x$  единственный при каждом  $j$ . Множество всех таких объектов  $S = \{x \in \mathbb{X} \mid x \leq \theta_E \text{ и } \exists j \in J: x^j = \theta_E^j\}$  определено единственным образом и является граничным, причём  $\theta_S = \theta_E$ .

Верно и обратное. Произвольному граничному подмножеству  $S$  соответствует граничная точка  $\theta_S$ . Правило  $r(x; \theta_S)$  лежит в некотором классе эквивалентности  $E$  и является его стандартным представителем,  $\theta_E = \theta_S$ , поскольку уменьшение порога  $\theta_S^j$  по любой из координат приведёт к изменению значения  $r(x; \theta_S)$  на одном из граничных объектов множества  $S$ .

Теорема доказана.  $\blacksquare$

Из теоремы следует, что  $|S| \leq |J|$ , в силу предположения, что значения каждого признака попарно различны на всей выборке  $\mathbb{X}$ .

**Свойства граничных подмножеств.** Обозначим через  $M_q$  множество всех граничных подмножеств мощности  $q$  и перечислим их основные свойства.

1.  $M_1$  состоит из всех  $L$  одноэлементных подмножеств,  $M_1 = \{\{x_1\}, \dots, \{x_L\}\}$ .

2.  $M_2$  состоит из всех пар несравнимых объектов из  $\mathbb{X}$ .
3. Мощность граничных подмножеств не может превышать ранга конъюнкции:  $M_q = \emptyset$  при  $q > |J|$ .
4. Граничное подмножество может состоять только из попарно несравнимых объектов. Однако не всякое подмножество из трёх и более попарно несравнимых объектов является граничным. Например, объекты  $(1, 3, 4)$ ,  $(3, 1, 1)$ ,  $(2, 2, 2)$  попарно несравнимы, но последний из них не является граничным.
5. Любое подмножество  $S' \subset S$  граничного подмножества  $S$  также граничное.

Исходя из этих свойств, можно предложить следующий алгоритм перебора всех граничных подмножеств.

На первом шаге строится множество  $M_1$  всех  $L$  одноэлементных подмножеств. Далее на каждом шаге  $q = 2, \dots, |J|$  к каждому подмножеству  $S' \in M_{q-1}$  добавляется всеми возможными способами ещё один объект  $x \in \mathbb{X} \setminus S'$ , и если полученное подмножество  $S = S' \cup \{x\}$  граничное (что легко проверяется по определению), то оно включается в  $M_q$ .

Для вычисления оценки расслоения–связности (9.5) этот алгоритм не достаточно эффективен, так как он не перебирает правила по слоям снизу вверх и не подсчитывает характеристики связности и неоптимальности каждого правила.

**Графические представления классов эквивалентности.** На рис. 11.2 изображен *граф связей* между классами эквивалентности. Вершины графа соответствуют стандартным представителям классов эквивалентности. Рёбрами соединены правила, векторы ошибок которых различаются на одном объекте. Граф связей зависит только от объектов  $x_i$ , но не зависит от их классификаций  $y_i$ .

Если учесть классификации объектов, то для каждого класса эквивалентности  $E$  можно вычислить число ошибок  $n(r_E, \mathbb{X})$ . На Рис. 11.2 оно указано рядом с каждой точкой — стандартным представителем класса эквивалентности. Граф расслоения–связности данного семейства правил изоморфен графу связей и является многодольным. Его  $m$ -й слой образуется всеми правилами, для которых  $n(r_E, \mathbb{X}) = m$ , см. Рис. 11.3. Слои располагаются в порядке возрастания числа ошибок снизу вверх.

### §11.3 Послойное вычисление оценки расслоения–связности

Рассмотрим теперь алгоритм перебора правил, позволяющий эффективно вычислять характеристики связности и неоптимальности каждого правила и, двигаясь по слоям снизу вверх, вовремя прерывать вычисление оценки расслоения–связности.

**Построение окрестности правила.** Рассмотрим следующую вспомогательную задачу. Задано правило  $r(x; \theta)$  с вектором порогов  $\theta = (\theta^j)_{j \in J}$ . Требуется построить его *окрестность*  $V_\theta$  — множество всех правил  $r(x; \theta')$ , векторы ошибок которых отличаются от его вектора ошибок только на одном объекте.

При построении окрестности будут заодно формироваться и характеристики правила  $r(x, \theta)$ , необходимые для вычисления оценки расслоения–связности:

$X_\theta$  — порождающее множество,  $q(\theta) = |X_\theta|$  — верхняя связность;

$X'_\theta$  — запрещающее множество,  $r(\theta) = |X'_\theta|$  — неоптимальность;

$D_\theta$  — множество объектов входящих рёбер,  $d(\theta) = |D_\theta|$  — нижняя связность;

$m(\theta) = n(a, \mathbb{X})$  — число ошибок правила на генеральной выборке.

Будем строить только такие векторы порогов  $\theta'$ , которые являются граничными точками и, согласно Теореме 11.2, одновременно стандартными представителями классов эквивалентности правил. Построение разбивается на два этапа, см. Алгоритм 11.1.

На первом этапе (шаги 1–5) строятся все соседние правила  $r(x; \theta')$ , получаемые из  $\theta$  путём уменьшения некоторых порогов,  $\theta' \leq \theta$ . Для этого из граничного подмножества  $S(\theta) = \{x \in S \mid \exists j \in J: x^j = \theta^j\}$  поочерёдно исключается один из объектов и соответствующие пороги  $\theta^j$ , по которым он и был граничным, уменьшаются. Число получаемых таким способом соседних правил не превышает  $|S(\theta)|$ .

На втором этапе (шаги 6–11) строятся все соседние правила  $r(x; \theta')$ , получаемые путём увеличения некоторых порогов,  $\theta' \geq \theta$ . Это более сложный случай, и здесь приходится прибегать к рекурсивной процедуре. Сначала делается подготовительная работа (шаги 6, 7): по каждой координате  $j \in J$  определяется максимальная граница  $\bar{\theta}^j$ , выше которой соседних правил быть не может. Это необязательное построение, но оно позволит в дальнейшем сократить поиск. Каждый объект выборки  $x \in \mathbb{X}$  может находиться в одном из трёх статусов:  $x.\text{проверен} \in \{\text{нет}, \text{плохой}, \text{хороший}\}$ . Сначала все объекты не проверены. Затем просматриваются все объекты  $x \in \mathbb{X}$ , покрываемые правилом  $r(x; \bar{\theta})$ , но не покрываемые правилом  $r(x; \theta)$ . Это означает, что  $\theta < x \leq \bar{\theta}$ , и хотя бы по одной координате объект  $x$  лежит выше порога:  $\theta^j < x^j$ . Для каждого такого объекта вызывается процедура Проверить( $x$ ). Она устанавливает статус объекта  $x.\text{хороший}$ , если существует правило  $r(x; \theta')$ , покрывающее только объект  $x$  и все объекты, покрываемые правилом  $r(x; \theta)$ . Если же правило  $r(x; \theta')$ , наряду с  $x$ , покрывает ещё один объект  $\tilde{x}$ , не покрываемый правилом  $r(x; \theta)$ , то устанавливается статус объекта  $x.\text{плохой}$ , и процедура Проверить( $\tilde{x}$ ) вызывается рекурсивно для объекта  $\tilde{x}$ . Каждый «хороший» объект  $x$  индуцирует соседнее правило  $\theta' = \max\{\theta, x\}$ , которое и добавляется в список  $V_\theta$ . Введение статусов объектов позволяет избежать их повторного перебора в основном цикле второго этапа.

Этими двумя этапами поиск окрестных правил ограничивается. Другие случаи рассматривать не нужно, так как если порог уменьшается по одной координате,  $\theta'^j < \theta^j$ , и увеличивается по другой,  $\theta'^t > \theta^t$ , то граничные точки  $\theta$  и  $\theta'$  не могут быть соседними, поскольку векторы ошибок соответствующих им правил отличаются, как минимум, на двух объектах.

При эффективной реализации Алгоритма 11.1 по каждому признаку  $f_j$  заранее строится индекс — массив номеров объектов, упорядоченных по возрастанию значений признака. Тогда на шагах 4, 7, 9, 14 можно будет просматривать только нужные объекты, не перебирая всю выборку.

---

**Алгоритм 11.1.** Построение окрестности  $V_\theta$  правила  $r(x; \theta)$ .

---

**Вход:**

$J$  — набор признаков,  $\theta = (\theta^j)_{j \in J}$  — правило,  $y$  — класс правил,  $\mathbb{X}$  — выборка.

**Выход:**

$V_\theta, X_\theta, X'_\theta, D_\theta, q(\theta), r(\theta), d(\theta), m(\theta)$ .

---

*Этап 1 — построение окрестных правил путём уменьшения порогов:*

- 1:  $V_\theta := \emptyset$ ;
- 2: **для всех**  $x \in \mathbb{X}$  таких, что  $r(x; \theta) = 1$  и  $\exists j \in J: \theta^j = x^j$
- 3:     **для всех**  $j \in J$  таких, что  $x^j = \theta^j$
- 4:          $\theta'^j := \max\{x_i^j \mid x_i \in \mathbb{X}, x_i < \theta^j\}$ ;
- 5:     Добавить( $\theta, \theta', x$ );

*Подготовка к этапу 2 — поиск максимальной граничной точки  $\bar{\theta}$ :*

- 6: **для всех**  $j \in J$
- 7:      $\bar{\theta}^j := \max\{L, x^j \mid x \in \mathbb{X}, x^j > \theta^j, x^t \leq \theta^t, t \neq j\}$ ;

*Этап 2 — построение окрестных правил путём увеличения порогов:*

- 8: **для всех**  $j \in J$
  - 9:     **для всех**  $x$  таких, что  $\theta^j < x^j \leq \bar{\theta}^j$
  - 10:     **если**  $x < \bar{\theta}$  и  $x$ .проверен = нет **то**
  - 11:         Проверить( $x$ );
- 

- 12: **ПРОЦЕДУРА** Проверить( $x$ )
  - 13: **для всех**  $j \in J$  таких, что  $\theta^j < x^j$
  - 14:     **для всех**  $\tilde{x}$  таких, что  $\theta^j < \tilde{x}^j < x^j$
  - 15:         **если**  $\theta < \tilde{x} < x$  **то**
  - 16:              $x$ .проверен := плохой;
  - 17:         **если**  $\tilde{x}$ .проверен = нет **то** Проверить( $\tilde{x}$ );
  - 18:     **выход**;
  - 19:  $x$ .проверен := хороший;
  - 20:  $\theta'^j := \max\{\theta^j, x^j\}$ , для всех  $j \in J$ ;
  - 21: Добавить( $\theta, \theta', x$ );
- 

- 22: **ПРОЦЕДУРА** Добавить( $\theta, \theta', x_i$ )
  - 23: добавить  $\theta'$  в список  $V_\theta$ ;
  - 24: **если**  $r(x_i; \theta) = [y_i=y]$  **то**  
          правило  $\theta'$  находится слоем выше, чем  $\theta$ :
  - 25:      $X_\theta := X_\theta \cup \{x_i\}$ ;    $q(\theta) := |X_\theta|$ ;
  - 26: **иначе**  
          правило  $\theta'$  находится слоем ниже, чем  $\theta$ :
  - 27:      $D_\theta := D_\theta \cup \{x_i\}$ ;    $d(\theta) := |D_\theta|$ ;
  - 28:      $X'_\theta := X'_\theta \cup X'_{\theta'} \cup \{x_i\}$ ;    $r(\theta) := |X'_\theta|$ ;
  - 29:      $m(\theta) := m(\theta') + 1$ ;
-

---

**Алгоритм 11.2.** Вычисление оценки расслоения–связности для семейства пороговых конъюнкций.

---

**Вход:**  $J$  — набор признаков,  $y$  — класс правил,  $\mathbb{X}$  — выборка.

**Выход:**  $Q_\varepsilon$  — оценка вероятности переобучения (9.5).

---

- 1:  $\Theta := \text{Arg min}_\theta n(\theta, \mathbb{X}); Q_\varepsilon := 0;$
  - 2: **повторять**
  - 3:  $Q_{\varepsilon,m} := 0; \Theta' := \emptyset;$
  - 4: **для всех**  $\theta \in \Theta$
  - 5: построить окрестность  $V_\theta$  с помощью Алгоритма 11.1;
  - 6:  $Q_{\varepsilon,m} := Q_{\varepsilon,m} + \frac{1}{C_L^\ell} C_{L-q(\theta)-r(\theta)}^{\ell-q(\theta)} H_{L-q(\theta)-r(\theta)}^{\ell-q(\theta), m(\theta)-r(\theta)} \left( \frac{\ell}{L} (m(\theta) - \varepsilon k) \right);$
  - 7:  $\Theta' := \Theta' \cup \{ \theta' \in V_\theta : m(\theta') = m(\theta) + 1 \};$
  - 8:  $Q_\varepsilon := Q_\varepsilon + Q_{\varepsilon,m}; \Theta := \Theta';$
  - 9: **пока** вклад слоя  $Q_{\varepsilon,m}$  не станет мал.
- 

**Послойный перебор классов эквивалентности.** Алгоритм 11.2 вычисляет  $Q_\varepsilon$  — оценку вероятности переобучения (9.5), перебирая правила по слоям снизу вверх. Сначала формируется нижний слой. На каждом шаге слой  $\Theta$  состоит из правил  $\theta$  с одинаковым числом ошибок  $m(\theta)$  на генеральной выборке. Для каждого правила  $\theta$  вычисляется его вклад в вероятность переобучения, строится окрестность, и объединение верхних частей всех полученных окрестностей становится следующим слоем  $\Theta'$ . Шаги повторяются до тех пор, пока не будет набрано достаточное число слоёв и вклад очередного слоя не окажется пренебрежимо мал.

## §11.4 Эксперимент на реальных задачах классификации

Мы не будем здесь описывать детали реализации логических алгоритмов классификации, использованных в этом эксперименте. Скажем лишь, что информативность правил оценивалась с помощью точного теста Фишера [85]; для формирования информативных наборов признаков применялся поиск в ширину и адаптивный случайный поиск [31, 30]; для построения композиции закономерностей применялись три алгоритма: LR, DL, WV.

*Логистическая регрессия* (LR, logistic regression) — это линейный классификатор, на вход которому вместо обычных бинарных признаков подаются закономерности, построенные независимо друг от друга [72];

*Решающий список* (DL, decision list) — это жадный алгоритм, в котором каждая следующая закономерность обучается по выборке объектов, ещё не покрытых предыдущими закономерностями [93].

*Взвешенное голосование* (WV, weighted voting) — это бустинг закономерностей, похожий на метод SLIPPER [64]. В нём перед построением каждой следующей закономерности производится переоценка весов обучающих объектов, в результате которой закономерность настраивается преимущественно на тех объектах, которые оказались наиболее трудными для всех предыдущих закономерностей.

	алгоритмы	задачи					
		australian	echo-card	heart dis.	hepatitis	labor	liver
1	RIPPER-opt	15.5	<b>2.9</b>	19.7	20.7	18.0	32.7
2	RIPPER+opt	15.2	5.5	20.1	23.2	18.0	<b>31.3</b>
3	C4.5 (Tree)	14.2	5.5	20.8	18.8	14.7	37.7
4	C4.5 (Rules)	15.5	6.8	20.0	18.8	14.7	37.5
5	C5.0	<b>14.0</b>	4.3	21.8	20.1	18.4	31.9
6	SLIPPER	15.7	4.3	<b>19.4</b>	<b>17.4</b>	<b>12.3</b>	32.2
7	LR	14.8	4.3	19.9	18.8	14.2	32.0
8	WV	14.9	4.3	20.1	19.0	14.0	32.3
9	DL	15.1	4.5	20.5	19.5	14.7	35.8
10	<b>WV+MC</b>	<b>13.9</b>	<b>3.0</b>	<b>19.5</b>	<b>18.3</b>	<b>13.2</b>	<b>30.7</b>
11	<b>DL+MC</b>	14.5	3.5	19.8	18.7	13.8	32.8
12	<b>WV+SC</b>	<b>14.1</b>	<b>3.2</b>	<b>19.3</b>	<b>18.1</b>	<b>13.4</b>	<b>30.2</b>
13	<b>DL+SC</b>	14.4	3.6	19.5	18.6	13.6	32.3

**Таблица 11.1.** Результаты экспериментов на 6 реальных задачах классификации из репозитория UCI. Для каждой пары ⟨задача, алгоритм⟩ в таблице выведена средняя ошибка (в процентах) по 10-кратному скользящему контролю. Для каждой задачи жирным выделены три лучших результата. Алгоритмы 1–7 выступают в роли эталонных. Обозначения наших алгоритмов: WV — взвешенное голосование, DL — решающий список, +SC и +MC — модификации на основе оценки расслоения–связности и метода Монте-Карло соответственно.

Были реализованы две модификации критерия информативности  $\mathcal{H}'(p, n)$ . SC-модификация основана на оценке расслоения–связности. MC-модификация основана на непосредственной оценке функционала  $Q_\varepsilon$  методом Монте-Карло по 100 случайным разбиениям  $\mathbb{X} = X \sqcup \bar{X}$ . Для обеих модификаций полагалось  $\ell = k$ .

В роли эталонов для сравнения выступали известные логические алгоритмы классификации: C4.5 [91], C5.0 [90], RIPPER [63] и SLIPPER [64].

Таблица 11.1 показывает, что немодифицированные версии алгоритмов WV, DL имеют качество классификации, сопоставимое с эталонными алгоритмами. Взвешенное голосование WV немного лучше решающего списка DL, что согласуется с результатами других исследований. Обе модификации, SC- и MC-, во всех случаях опережают свои немодифицированные версии и значимо снижают переобучение. Различия между SC- и MC-модификациями не значимы. Отсюда следует вывод, что оценка расслоения–связности, несмотря на некоторую завышенность, хорошо справляется с ролью критерия отбора правил.

## Резюме

Логические алгоритмы классификации представляют собой композиции информативных логических правил — закономерностей. В качестве правил часто используются конъюнкции пороговых условий над числовыми признаками. Данное семейство правил описывается системой всех граничных подмножеств мощности, не пре-

вышающей ранга конъюнкции. Для практического вычисления оценки расслоения–связности предлагается перебирать правила по слоям снизу вверх, и прекращать перебор в тот момент, когда новые слои перестанут вносить существенный вклад в оценку. С помощью этой методики был построен модифицированный критерий отбора признаков. Заменяв стандартный критерий модифицированным, и ничего более не меняя в методе обучения логических классификаторов WV и DL, во всех 6 экспериментах удалось пронаблюдать значимое увеличение обобщающей способности.

**В следующей лекции** мы рассмотрим одну весьма общую комбинаторную технику, с помощью которой можно вычислять как точные, так и верхние или нижние оценки, причём именно вычислять, а не выводить — оценки будут записываться не в виде формулы, а в виде итерационного процесса.

## Упражнения

**Задача 11.1 (1).** Доказать, что если вектор  $\theta$  является граничной точкой подмножества  $S \subset \mathbb{X}$ , то в  $S$  можно единственным образом выделить граничное подмножество  $S(\theta)$  с той же граничной точкой  $\theta$ . Выписать выражение  $S(\theta)$  через  $S$  и  $\theta$ .

**Задача 11.2 (2).** Доказать, что каждый класс эквивалентности правил представляет собой диаграмму Юнга.

**Задача 11.3 (2).** Рассмотрим граф со множеством вершин  $\Theta$ , в котором рёбрами соединены все соседние вершины (отличающиеся на 1 только по одной из координат). Объекты генеральной выборки  $x = (x^j)_{j \in J} \in \mathbb{X}$  соответствуют некоторым вершинам этого графа,  $\mathbb{X} \subset \Theta$ . Для каждого  $x \in \mathbb{X}$  удалим из графа все рёбра  $(\theta, \bar{\theta})$ , такие, что

- 1)  $\bar{\theta}^j = \theta^j + 1$  для некоторого  $j \in J$  и  $\bar{\theta}^s = \theta^s$  для всех  $s \in J \setminus \{j\}$ ;
- 2)  $r(x, \theta) = 0$ ,  $r(x, \bar{\theta}) = 1$ .

Доказать, что после этого связанные компоненты графа будут взаимно однозначно соответствовать классам эквивалентности правил.

**Задача 11.4 (2).** Описать структуру классов эквивалентности и предложить эффективный алгоритм их послойного перебора для семейства правил

$$r(x; \theta_1, \theta_2) = [\theta_1 \leq x \leq \theta_2],$$

где все значения  $x_i \in \mathbb{X} \subset \mathbb{R}$  попарно различны. Подсказка: рассмотреть эквивалентное семейство двухпризнаковых конъюнкций  $r(x; \theta_1, \theta_2) = [-x \leq \theta_1][x \leq \theta_2]$ .

**Задача 11.5 (5\*).** Описать структуру классов эквивалентности и предложить эффективный алгоритм их послойного перебора для случая, когда индикатор ошибки определяется отдельно для положительных и отрицательных примеров:

$$\begin{aligned} I_p(r, x_i) &= [r(x_i) = 0][y_i = y]; \\ I_n(r, x_i) &= [r(x_i) = 1][y_i \neq y]. \end{aligned}$$

**Задача 11.6 (5).** Описать структуру классов эквивалентности и предложить эффективный алгоритм их послойного перебора для случая, когда каждый признак может принимать одинаковые значения на различных объектах.

**Задача 11.7 (3\*).** Обобщить Алгоритм 11.2 на случай, когда на шаге 1 вместо множества всех правил, доставляющих глобальный минимум числу ошибок, находится лишь одно локально оптимальное правило. Предусмотреть возможность переходов от правил  $m$ -го слоя к правилам не только  $(m + 1)$ -го слоя, но и  $(m - 1)$ -го слоя.



## 12 Рекуррентные оценки

Ещё одна идея вычисления оценок обобщающей способности заключается в том, чтобы взять сначала только один алгоритм, например, лучший — для него оценка известна. Затем добавлять алгоритмы по одному, на каждом шаге соответствующим образом корректируя оценку. Если добавляемый алгоритм похож на предыдущие, то, скорее всего, коррекция будет незначительной. Если добавлять алгоритмы по слоям, в порядке возрастания числа ошибок  $n(a, \mathbb{X})$ , то, скорее всего, можно будет ограничиться перебором нескольких нижних слоёв.

С помощью этого подхода были получены первые оценки расслоения–связности [111]. Это был громоздкий способ, и, к тому же, он требовал существования в семействе корректного алгоритма. Гораздо более изящные и общие оценки по графу расслоения–связности появились несколько позже [112].

**Постановка задачи.** Допустим, что векторы ошибок всех алгоритмов из множества  $A$  известны, попарно различны, и в  $A$  есть корректный на  $\mathbb{X}$  алгоритм. Пусть  $\mu$  — пессимистичный метод минимизации эмпирического риска.

Рассмотрим две задачи.

Первая задача — найти для каждого алгоритма  $a \in A$  всю информацию, необходимую для вычисления вероятности переобучения  $Q_\varepsilon$  по Теореме 7.6:

$$\mathfrak{I}(a) = \langle X_{av}, X'_{av}, c_{av} \rangle_{v \in V_a},$$

где  $V_a$  — индексное множество,  $X_{av}$  — множество порождающих объектов,  $X'_{av}$  — множество запрещающих объектов,  $c_{av} \in \mathbb{R}$ .

Вторая задача — вывести формулы для коррекции текущей оценки вероятности переобучения при добавлении ещё одного алгоритма.

### §12.1 Добавление одного алгоритма

Перенумеруем алгоритмы в порядке неубывания  $n(a, \mathbb{X})$  — числа ошибок на полной выборке:  $A = \{a_0, \dots, a_D\}$ . В силу предположения о корректности,  $n(a_0, \mathbb{X}) = 0$ .

Обозначим через  $\mu_d$  метод обучения, удовлетворяющий определению 8.3 и выбирающий алгоритмы только из подмножества  $A_d = \{a_0, \dots, a_d\}$ . Рассмотрим процедуру последовательного добавления алгоритмов, на каждом шаге которой осуществляется переход от метода  $\mu_{d-1}$  к методу  $\mu_d$ .

Допустим, что для всех алгоритмов  $a_t$ ,  $t < d$ , информация  $\mathfrak{I}(a_t)$  относительно метода  $\mu_{d-1}$  уже вычислена. Зададимся целью вычислить информацию  $\mathfrak{I}(a_d)$  и скорректировать информацию  $\mathfrak{I}(a_t)$ ,  $t < d$ , относительно метода  $\mu_d$ . Такая корректировка необходима, поскольку алгоритм  $a_d$  может отбирать некоторые разбиения у каждого из предыдущих алгоритмов  $a_t$ .

**Лемма 12.1.** *Метод  $\mu_d$  выбирает алгоритм  $a_d$  тогда и только тогда, когда все объекты, на которых  $a_d$  допускает ошибку, попадают в контрольную выборку:*

$$[\mu_d X = a_d] = [X'_d \subseteq \bar{X}], \quad X'_d = \{x_i \in \mathbb{X} : I(a_d, x_i) = 1\}.$$

**Доказательство.** Если хотя бы один объект, на котором  $a_d$  допускает ошибку, окажется в обучающей выборке  $X$ , то метод  $\mu_d$  выберет алгоритм с меньшим числом ошибок на  $X$ . Такой алгоритм точно есть, например,  $a_0$ . Итак, условие  $X'_d \subseteq \bar{X}$  является необходимым для того, чтобы метод  $\mu_d$  выбрал алгоритм  $a_d$ . Покажем, что оно является также и достаточным. Для этого достаточно показать, что если алгоритмов из  $A_d$ , не допускающих ошибок на обучающей выборке  $X$ , окажется несколько, то из них будет выбран именно  $a_d$ . В силу упорядоченности множества  $A_d$  алгоритм  $a_d$  допускает максимальное число ошибок на  $\mathbb{X}$ , а среди алгоритмов с таким же числом ошибок на  $\mathbb{X}$  имеет максимальный номер. Поэтому, согласно определению 8.3, алгоритм  $a_d$  будет выбран методом  $\mu_d$  из  $A_d$  всегда, когда  $X'_d \subseteq \bar{X}$ . ■

Допустим, что непосредственно перед добавлением алгоритма  $a_d$  условия выбора каждого из предыдущих алгоритмов  $a_t$  были записаны в виде (7.7):

$$[\mu_{d-1}X=a_t] = \sum_{v \in V_t} c_{tv} \underbrace{[X_{tv} \subseteq X][X'_{tv} \subseteq \bar{X}]}_{J_{tv}(d-1)}, \quad t < d.$$

После добавления алгоритма  $a_d$  эти условия изменятся. К ним добавится требование, чтобы множество  $X'_d$  не лежало целиком в контрольной выборке  $\bar{X}$ ; иначе вместо алгоритма  $a_t$  метод  $\mu_d$  выберет алгоритм  $a_d$ :

$$\begin{aligned} [\mu_d X=a_t] &= [\mu_{d-1}X=a_t][X'_d \not\subseteq \bar{X}] = \\ &= \sum_{v \in V_t} c_{tv} \underbrace{[X_{tv} \subseteq X][X'_{tv} \subseteq \bar{X}][X'_d \not\subseteq \bar{X}]}_{J_{tv}(d)}, \quad t < d. \end{aligned} \quad (12.1)$$

Индукцией по  $d$  легко доказывается, что тождество (7.2) выполняется на каждом шаге, то есть что условия (12.1) определены корректно. Действительно, предполагая

$$\sum_{t=0}^{d-1} [\mu_{d-1}X=a_t] = 1,$$

получаем

$$\sum_{t=0}^d [\mu_d X=a_t] = \sum_{t=0}^{d-1} [\mu_{d-1}X=a_t][X'_d \not\subseteq \bar{X}] + [X'_d \subseteq \bar{X}] = 1.$$

Чтобы получить правила корректировки информации  $\mathfrak{I}(a_t)$ , достаточно привести выражение (12.1) к виду (7.7), что и будет сделано в следующей лемме.

**Лемма 12.2.** *Корректировка информации  $\mathfrak{I}(a_t)$ ,  $t < d$  при добавлении алгоритма  $a_d$  сводится к проверке для каждого  $v \in V_t$  такого, что  $X_{tv} \cap X'_d = \emptyset$ , трёх условий:*

- 1) если  $X'_d \setminus X'_{tv} = \{x_i\}$  — одноэлементное множество, то  $x_i$  присоединяется к  $X_{tv}$ ;
- 2) если  $|X'_d \setminus X'_{tv}| > 1$ , то множество индексов  $V_t$  пополняется ещё одним элементом (обозначим его  $w$ ) и полагается  $c_{tw} = -c_{tv}$ ,  $X_{tw} = X_{tv}$ ,  $X'_{tw} = X'_{tv} \cup X'_d$ ;

3) если  $|X'_d \setminus X'_{tv}| = 0$ , то из множества индексов  $V_t$  удаляется индекс  $v$ ; соответственно, из  $\mathfrak{I}(a_t)$  удаляется вся тройка  $\langle X_{tv}, X'_{tv}, c_{tv} \rangle$ .

**Доказательство.** Если  $X_{tv} \cap X'_d \neq \emptyset$ , то из  $X_{tv} \subseteq X$  следует, что множество  $X'_d$  не лежит целиком в контрольной выборке  $\bar{X}$ . Следовательно, никакая корректировка для тройки  $\langle X_{tv}, X'_{tv}, c_{tv} \rangle$  не нужна:

$$J_{tv}(d) = [X_{tv} \subseteq X][X'_{tv} \subseteq \bar{X}] = J_{tv}(d-1).$$

Если всё-таки  $X_{tv} \cap X'_d = \emptyset$ , то возможны три случая, в зависимости от мощности множества  $X'_d \setminus X'_{tv}$ .

Первый случай:  $X'_d \setminus X'_{tv} = \{x_i\}$  — одноэлементное множество. Тогда справедлива цепочка равенств  $[X'_d \not\subseteq \bar{X}] = [x_i \notin \bar{X}] = [x_i \in X]$ . Подставляя в (12.1), получим

$$J_{tv}(d) = [X_{tv} \sqcup \{x_i\} \subseteq X][X'_{tv} \subseteq \bar{X}].$$

Второй случай:  $|X'_d \setminus X'_{tv}| > 1$ . Тогда

$$\begin{aligned} J_{tv}(d) &= [X_{tv} \subseteq X][X'_{tv} \subseteq \bar{X}](1 - [X'_d \subseteq \bar{X}]) = \\ &= J_{tv}(d-1) - [X_{tv} \subseteq X][X'_{tv} \cup X'_d \subseteq \bar{X}]. \end{aligned} \quad (12.2)$$

Таким образом, в выражении для  $[\mu_d X = a_t]$  появится ещё одно слагаемое, что равносильно добавлению во множество  $V_t$  ещё одного индекса (обозначим его  $w$ ), для которого полагается  $c_{tw} = -c_{tv}$ ,  $X_{tw} = X_{tv}$ ,  $X'_{tw} = X'_{tv} \cup X'_d$ .

Наконец, третий случай:  $|X'_d \setminus X'_{tv}| = 0$ . Тогда представление (12.2) остаётся в силе, однако разность  $J_{tv}(d)$  оказывается равной нулю, поскольку  $X'_{tv} \cup X'_d = X'_{tv}$ . Обнуление  $J_{tv}(d)$  равносильно удалению индекса  $v$  из множества индексов  $V_t$  вместе с удалением соответствующей тройки  $\langle X_{tv}, X'_{tv}, c_{tv} \rangle$  из информации  $\mathfrak{I}(a_t)$ . ■

## §12.2 Вычисление вероятности переобучения

Леммы 12.1, 12.2 и Теорема 7.8 позволяют рекуррентно вычислять вероятность переобучения  $Q_\varepsilon$ . На каждом  $d$ -м шаге,  $d = 0, \dots, D$ , добавляется алгоритм  $a_d$ , вычисляется информация  $\mathfrak{I}(a_d)$ ; затем для каждого  $t = 0, \dots, d-1$  корректируется информация  $\mathfrak{I}(a_t)$  и вероятности  $P_{tv}$ . На основе скорректированной информации обновляется текущая оценка  $Q_\varepsilon$ . По окончании последнего  $D$ -го шага текущая оценка  $Q_\varepsilon$  должна совпадать с точным значением вероятности переобучения. Эта рекуррентная процедура записана в виде псевдокода Алгоритма 12.1.

Некоторые обозначения, использованные в Алгоритме 12.1, требуют дополнительных пояснений.

Во-первых, вместо парного индекса  $a_{tv}$  для алгоритма  $a_t$  всюду используется сокращённое обозначение  $tv$ .

Во-вторых, предполагается  $L_{tv} = L - |X_{tv}| - |X'_{tv}|$ ,  $\ell_{tv} = \ell - |X_{tv}|$ , и вычисление величин  $L_{tv}$  и  $\ell_{tv}$  в явном виде не записывается, чтобы не перегружать псевдокод.

---

**Алгоритм 12.1.** Рекуррентное вычисление вероятности переобучения

---

**Вход:**

матрица ошибок  $I(a_d, x_i)$ ,  $d = 0, \dots, D$ ,  $i = 1, \dots, L$ ;

**Выход:**

информация  $\mathfrak{J}(a_d) = \langle X_{dv}, X'_{dv}, c_{dv} \rangle_{v \in V_d}$  для всех  $d = 0, \dots, D$ ,  
вероятность переобучения  $Q_\varepsilon$ ;

---

1:  $Q_\varepsilon := 0$ ; упорядочить  $A = \{a_0, \dots, a_D\}$  по возрастанию  $n(a_d, \mathbb{X})$ ;

2: **для всех**  $d := 1, \dots, D$

3:   добавить алгоритм  $a_d$ :

$$V_d := \{\emptyset\}; \quad X_d := \emptyset; \quad X'_d := \{x \in \mathbb{X} : I(a_d, x) = 1\}; \quad c_d := 1; \quad P_d := \prod_{j=0}^{|X'_d|-1} \binom{k-j}{L-j};$$

**если**  $n(a_d, \mathbb{X}) \geq \varepsilon k$  **то**  $Q_\varepsilon := Q_\varepsilon + P_d$ ;

4: **для всех**  $t := 0, \dots, d-1$

5:   **для всех**  $v \in V_t$  таких, что  $X_{tv} \cap X'_d = \emptyset$

6:      $\Delta := |X'_d \setminus X'_{tv}|$ ;

7:     **если**  $\Delta = 1$  **то**

8:       скорректировать множество  $X_{tv}$  и вероятность  $P_{tv}$ :

$$X_{tv} := X_{tv} \sqcup (X'_d \setminus X'_{tv}); \quad P'_{tv} := P_{tv}; \quad P_{tv} := P_{tv} \ell_{tv} / L_{tv};$$

**если**  $n(a_t, \mathbb{X}) \geq \varepsilon k$  **то**  $Q_\varepsilon := Q_\varepsilon + c_{tv}(P_{tv} - P'_{tv})$ ;

---

9:   **иначе если**  $\Delta > 1$  **то**

10:     добавить в  $V_t$  новый индекс  $w$ ;

$$X_{tw} := X_{tv}; \quad X'_{tw} := X'_{tv} \cup X'_d; \quad c_{tw} := -c_{tv}; \quad P_{tw} := P_{tv} \prod_{j=0}^{\Delta-1} \left(1 - \frac{\ell_{tv}}{L_{tv}-j}\right);$$

**если**  $n(a_t, \mathbb{X}) \geq \varepsilon k$  **то**  $Q_\varepsilon := Q_\varepsilon + c_{tw} P_{tw}$ ;

11:   **иначе**

12:     удалить из  $V_t$  индекс  $v$ ;

**если**  $n(a_t, \mathbb{X}) \geq \varepsilon k$  **то**  $Q_\varepsilon := Q_\varepsilon - c_{tv} P_{tv}$ ;

---

В-третьих, при добавлении алгоритма  $a_d$  на шаге 3 создаётся тройка  $\langle X_d, X'_d, c_d \rangle$  и вычисляется вероятность  $P_d$  получения алгоритма  $a_d$  методом  $\mu_d$ . При этом в индексное множество заносится «пустой элемент», обозначаемый  $\emptyset$ . Это позволяет в дальнейшем сокращать запись индексов, полагая, что если индекс не двойной, а одинарный, то имеется в виду первый элемент индексного множества  $V_d$ , то есть  $X_d \equiv X_{d\emptyset}$ ,  $X'_d \equiv X'_{d\emptyset}$ ,  $c_d \equiv c_{d\emptyset}$ ,  $P_d \equiv P_{d\emptyset}$ .

Вычисление вероятностей  $P_{tw} = \mathbb{P}[X_{tw} \subseteq X][X'_{tw} \subseteq \bar{X}]$  по формуле (7.9) также требует пояснений.

На шаге 3 вычисляется вероятность получения алгоритма  $a_d$  методом  $\mu_d$ :

$$P_d = \frac{C_{L-|X'_d|}^\ell}{C_L^\ell} = \prod_{j=0}^{|X'_d|-1} \binom{k-j}{L-j}.$$

Значения  $P_d$  одинаковы для всех алгоритмов с равным числом ошибок  $n(a_d, \mathbb{X})$ . Поэтому их можно вычислить один раз заранее для всех  $n = 0, \dots, L$ .

На шаге 8 к порождающему множеству  $X_{tv}$  добавляется один объект, что приводит к очень простой корректировке предыдущего значения вероятности  $P'_{tv}$ :

$$P_{tv} = \frac{C_{L_{tv}-1}^{\ell_{tv}-1}}{C_L^\ell} = \frac{C_{L_{tv}}^{\ell_{tv}}}{C_L^\ell} \frac{\ell_{tv}}{L_{tv}} = P'_{tv} \frac{\ell_{tv}}{L_{tv}}.$$

На шаге 10 из тройки  $\langle X_{tv}, X'_{tv}, c_{tv} \rangle$  формируется новая тройка  $\langle X_{tw}, X'_{tw}, c_{tw} \rangle$  путём добавления  $\Delta$  элементов к запрещающему множеству  $X'_{tv}$ . Следовательно,

$$P_{tw} = \frac{C_{L_{tw}-\Delta}^{\ell_{tw}}}{C_L^\ell} = \frac{C_{L_{tv}}^{\ell_{tv}} (L_{tv} - \ell_{tv} - \Delta + 1) \cdots (L_{tv} - \ell_{tv})}{C_L^\ell (L_{tv} - \Delta + 1) \cdots L_{tv}} = P_{tv} \prod_{j=0}^{\Delta-1} \left( 1 - \frac{\ell_{tv}}{L_{tv} - j} \right).$$

Алгоритм 12.1 может оказаться вычислительно неэффективным, если шаг 10 будет выполняться слишком часто. Каждое его выполнение приводит к добавлению ещё одного слагаемого в сумму (7.10). Следующая теорема позволяет сокращать время вычисления за счёт понижения точности верхней оценки  $Q_\varepsilon$ .

**Теорема 12.3.** *Если в Алгоритме 12.1 иногда пропускать шаг 10 при  $c_{tv} = 1$ , то вычисляемое в результате значение  $Q_\varepsilon$  будет верхней оценкой вероятности переобучения.*

**Доказательство.** Выполнение шага 10 при  $c_{tv} = 1$ ,  $c_{tw} = -1$  приводит к уменьшению текущего вычисленного значения  $Q_\varepsilon$  на величину  $P_{tw} \geq 0$ . Соответственно, невыполнение шага 10 приводит к исключению из суммы (7.11) отрицательного слагаемого  $-P_{tw}$ , и, возможно, ещё некоторого числа положительных и отрицательных слагаемых, которые появятся в этой сумме в результате последующих корректировок тройки  $\langle X_{tw}, X'_{tw}, c_{tw} \rangle$  при выполнении шагов 10 и 12. Каждая такая корректировка возникает в результате добавления некоторого алгоритма  $a_d$ ,  $d > t$ , который «отнимает» часть разбиений у алгоритма  $a_t$ , уменьшая слагаемое  $P_{tw}$  до величины  $\tilde{P}_{tw}$ :

$$\tilde{P}_{tw} = \mathbf{P}[X_{tw} \subseteq X][X'_{tw} \subseteq \bar{X}][X'_d \not\subseteq \bar{X}] \leq P_{tw}.$$

Исключение из суммы (7.11) отрицательного слагаемого  $-P_{tw}$  вместе со всеми последующими слагаемыми, корректирующими тройку  $\langle X_{tw}, X'_{tw}, c_{tw} \rangle$ , может только увеличить значение  $Q_\varepsilon$ , вычисляемое Алгоритмом 12.1. ■

## §12.3 Упрощённая рекуррентная оценка

Уберём из Алгоритма 12.1 шаги 9–12. В этом случае шаг 10 пропускается всегда, тройки  $\langle X_{tw}, X'_{tw}, c_{tw} \rangle$  с отрицательными значениями  $c_{tw}$  никогда не создаются, каждому алгоритму  $a_d$  соответствует только одна тройка, и все индексные множества  $V_d$ ,  $d = 0, \dots, D$  одноэлементны. Согласно Теореме 12.3, значение  $Q_\varepsilon$ , вычисляемое упрощённым рекуррентным Алгоритмом 12.1, будет верхней оценкой вероятности переобучения. В следующей теореме эта оценка выписывается в явном виде через профиль расслоения и связности множества  $A$ .

**Теорема 12.4.** Пусть векторы ошибок всех алгоритмов множества  $A$  попарно различны, в  $A$  есть корректный на  $\mathbb{X}$  алгоритм,  $\Delta_{mq}$  — число алгоритмов в  $m$ -м слое со связностью  $q$ . Тогда справедлива верхняя оценка вероятности переобучения

$$Q_\varepsilon \leq \sum_{m=\lceil \varepsilon k \rceil}^L \sum_{q=0}^L \Delta_{mq} \frac{C_{L-m-q}^{\ell-q}}{C_L^\ell}. \quad (12.3)$$

**Доказательство.** Рассмотрим упрощённый рекуррентный Алгоритм 12.1, из которого удалены шаги 9–12. Он даёт верхнюю оценку вероятности переобучения. Для каждого алгоритма  $a \in A$  он строит единственную тройку  $\langle X_a, X'_a, 1 \rangle$ , в которой за-прещающее множество  $X'_a = \{x_i \in \mathbb{X} : I(a, x_i) = 1\}$ , порождающее множество состоит из всех объектов, добавленных на шаге 8. Это те и только те объекты  $x_i$ , для которых существует алгоритм  $a' \in A$ , допускающий на одну ошибку больше, чем  $a$ . Очевидно, при этом  $X'_{a'} \setminus X'_a = \{x_i\}$  — одноэлементное множество. Число таких объектов  $x_i$  совпадает со значением связности  $q(a)$ . Таким образом,  $|X'_a| = n(a, \mathbb{X})$ ,  $|X_a| = q(a)$  для произвольного алгоритма  $a \in A$ . Следовательно, оценка (7.11) принимает вид:

$$\begin{aligned} Q_\varepsilon &\leq \sum_{a \in A} [n(a, \mathbb{X}) \geq \varepsilon k] \frac{C_{L-a}^{\ell-a}}{C_L^\ell} = \sum_{a \in A} [n(a, \mathbb{X}) \geq \varepsilon k] \frac{C_{L-n(a, \mathbb{X})-q(a)}^{\ell-q(a)}}{C_L^\ell} = \\ &= \sum_{m=\lceil \varepsilon k \rceil}^L \sum_{q=0}^L \underbrace{\sum_{a \in A} [n(a, \mathbb{X}) = m] [q(a) = q]}_{\Delta_{mq}} \frac{C_{L-m-q}^{\ell-q}}{C_L^\ell}. \quad \blacksquare \end{aligned}$$

Полученная оценка совпадает с оценкой расслоения–связности (9.8), стр. 90. Она показывает, что эффекты расслоения и связности снижают переобучение, поскольку комбинаторный множитель  $\frac{C_{L-m-q}^{\ell-q}}{C_L^\ell}$  убывает экспоненциально по  $q$  и по  $m$ .

## Резюме

Рекуррентное вычисление вероятности переобучения основано на том, что алгоритмы добавляются в семейство по одному, и при каждом добавлении пересчитывается оценка вероятности переобучения. Пропуская некоторые шаги в рекуррентной процедуре, можно сэкономить время вычислений и получить либо верхнюю, либо нижнюю оценку. Верхняя оценка совпадает с оценкой расслоения–связности, которая была получена ранее.

**В следующей лекции** мы рассмотрим теоретико-групповой подход, который позволяет выводить точные оценки вероятности переобучения для модельных семейств, обладающих свойством симметрии.

## Упражнения

**Задача 12.1 (2).** Доказать, что если в Алгоритме 12.1 иногда пропускать шаг 10 при  $c_{tv} = -1$ , то вычисляемое в результате значение  $Q_\varepsilon$  будет нижней оценкой вероятности переобучения.

Свяжем с каждым индексом  $v \in V_a$  целое значение  $s_v$ , назовём его *номером смены*. На шаге 3 положим  $s_\emptyset = 0$ . На шаге 10 положим  $s_w = s_v + 1$ . Обозначим через  $Q_\varepsilon^S$  оценку вероятности переобучения, которая получится, если пропускать шаг 10 при  $s_v \geq S$ .

**Задача 12.2 (2).** Доказать, что последовательность  $Q_\varepsilon^S$  является верхней оценкой при всех чётных  $S$  и нижней оценкой при всех нечётных  $S$ .

**Задача 12.3 (3).** Доказать, что последовательность  $Q_\varepsilon^0, Q_\varepsilon^1, Q_\varepsilon^2, \dots$  образует знакопеременный ряд, сходящийся к  $Q_\varepsilon$ .

**Задача 12.4 (3\*).** Обобщить рекуррентную процедуру на случай, когда в графе расслоения–связности множества  $A$  имеется единственный наилучший (но не обязательно корректный) алгоритм.

**Задача 12.5 (10\*).** Обобщить рекуррентную процедуру на случай произвольного множества  $A$  (в общем случае не содержащего корректного алгоритма).

## 13 Симметричные семейства алгоритмов

Получение точных оценок вероятности переобучения в общем случае является сложной комбинаторной задачей. Иногда её удаётся упростить, воспользовавшись особенностями конкретного семейства алгоритмов. Регулярный способ такого упрощения даёт теоретико-групповой подход, предложенный Александром Фреем [45, 65]. Подход основан на эксплуатации свойства симметрии семейства алгоритмов. Он подходит, главным образом, для модельных семейств. Реальные семейства, как правило, не являются симметричными. Тем не менее, теоретически, реальные семейства могут заменяться симметричными семействами с «похожим переобучением» для получения оценок обобщающей способности.

Ещё одна особенность теоретико-группового подхода состоит в том, что оценки выводятся для рандомизированного метода минимизации эмпирического риска.

### Основы теоретико-группового подхода

Нам понадобится совсем немного сведений из теории групп. Напомним, что группой  $G$  называется непустое множество с определённой на нём бинарной операцией  $\circ$ , удовлетворяющей трём аксиомам:

- 1) ассоциативность:  $(a \circ b) \circ c = a \circ (b \circ c)$  для любых  $a, b, c \in G$ ;
- 2) существование единичного элемента  $e \in G$ :  $e \circ a = a$  для любого  $a \in G$ ;
- 3) существование обратного элемента  $a^{-1} \in G$ :  $a^{-1} \circ a = e$  для любого  $a \in G$ .

Классическим примером является симметрическая группа  $S_L$ . Это множество всех  $L!$  перестановок (биекций) конечного множества мощности  $L$  в себя, с бинарной операцией суперпозиции перестановок. Единичный элемент в  $S_L$  — это тождественная перестановка, оставляющая все элементы множества на месте.

Подгруппа — это подмножество  $H$  группы  $G$ , которое является группой относительно операции, определённой в  $G$ . Любая конечная группа является подгруппой некоторой симметрической группы (теорема Кэли).

**Рандомизированная минимизация эмпирического риска.** Пусть  $A$  — множество алгоритмов с попарно различными векторами ошибок. Обозначим через  $A(X)$  множество алгоритмов с минимальным числом ошибок на обучающей выборке  $X$ :

$$A(X) = \text{Arg min}_{a \in A} n(a, X). \quad (13.1)$$

*Рандомизированный метод минимизации эмпирического риска* выбирает произвольный алгоритм из множества  $A(X)$  случайно и равновероятно (стр. 68). Поскольку в задаче статистического обучения появляется второй независимый источник случайности, определение вероятности переобучения  $Q_\varepsilon$  приходится модифицировать. Наиболее естественный вариант модификации — усреднение по множеству  $A(X)$ :

$$Q_\varepsilon(A) = \mathbb{E} \frac{1}{|A(X)|} \sum_{a \in A(X)} [\delta(a, X) \geq \varepsilon]. \quad (13.2)$$



Переставим местами знаки суммирования,  $E \sum = \sum E$ . Получим сумму по всем алгоритмам  $a \in A$ , каждое слагаемое которой обозначим через  $Q_\varepsilon(a, A)$  и назовём вкладом алгоритма  $a$  в вероятность переобучения:

$$Q_\varepsilon(A) = \sum_{a \in A} Q_\varepsilon(a, A), \quad Q_\varepsilon(a, A) = E \frac{[a \in A(X)]}{|A(X)|} [\delta(a, X) \geq \varepsilon].$$

Аналогичным образом введём вероятность реализации алгоритма  $a$ :

$$P(a, A) = E \frac{[a \in A(X)]}{|A(X)|}. \quad (13.3)$$

Если бы множество  $A(X)$  при любом  $X$  состояло из единственного алгоритма  $a = \mu X$ , то вероятность переобучения (13.2) и вероятность реализации (13.3) имели бы привычный вид:

$$Q_\varepsilon(A) = P[\delta(\mu X, X) \geq \varepsilon]; \quad P(a, A) = P[a = \mu X].$$

**Перестановки объектов.** Рассмотрим действие симметрической группы  $S_L$  на выборке  $\mathbb{X} = (x_1, \dots, x_L)$ . Возьмём произвольную перестановку  $\pi \in S_L$ . Обозначим через  $\pi x$  тот объект, в который перестановка  $\pi$  переводит объект  $x \in \mathbb{X}$ . Действие перестановок на объектах естественным образом переносится на подмножества объектов, на алгоритмы как бинарные векторы ошибок длины  $L$  и на множества алгоритмов:

$$\begin{aligned} \text{действие перестановки } \pi \text{ на подмножество объектов: } \pi X &= \{\pi x : x \in X\}; \\ \text{действие перестановки } \pi \text{ на алгоритм: } \pi \vec{a} &= (I(\pi a, x_i))_{i=1}^L = (I(a, \pi^{-1} x_i))_{i=1}^L; \\ \text{действие перестановки } \pi \text{ на множество алгоритмов: } \pi A &= \{\pi a : a \in A\}. \end{aligned}$$

Заметим, что действие одной и той же перестановки  $\pi$  сначала на выборку  $\mathbb{X}$ , затем на алгоритм  $a$  восстанавливает исходный вектор ошибок алгоритма  $a$ . Благодаря такому определению действие на алгоритм обладает рядом полезных свойств.

**Лемма 13.1.** Свойства действия произвольной перестановки  $\pi \in S_L$  на алгоритмы:

- 1)  $I(\pi a, \pi x) = I(a, x)$  для любых  $a \in A$  и  $x \in \mathbb{X}$ ;
- 2)  $n(\pi a, \mathbb{X}) = n(a, \mathbb{X})$  для любого  $a \in A$ ;
- 3)  $n(\pi a, \pi X) = n(a, X)$  для любых  $a \in A$  и  $X \subseteq \mathbb{X}$ ;
- 4)  $\delta(\pi a, \pi X) = \delta(a, X)$  для любых  $a \in A$  и  $X \subseteq \mathbb{X}$ .
- 5)  $a \in A(X) \leftrightarrow \pi a \in (\pi A)(\pi X)$  для любых  $a \in A$  и  $X \subseteq \mathbb{X}$ .
- 6)  $|A(X)| = |(\pi A)(\pi X)|$  для любых  $A$  и  $X \subseteq \mathbb{X}$ .

**Доказательство.** Свойство 1) следует из определения:

$$I(\pi a, \pi x) = I(a, \pi^{-1} \pi x) = I(a, x).$$

Свойство 2) следует из свойства 1):

$$n(\pi a, \mathbb{X}) = \sum_{i=1}^L I(\pi a, x_i) = \sum_{i=1}^L I(\pi a, \pi x_i) = \sum_{i=1}^L I(a, x_i) = n(a, \mathbb{X}).$$

Свойство 3) также следует из свойства 1):

$$n(\pi a, \pi X) = \sum_{x \in \pi X} I(\pi a, x) = \sum_{x \in X} I(\pi a, \pi x) = \sum_{x \in X} I(a, x) = n(a, X).$$

Свойство 4) следует из свойства 3):

$$\delta(a, X) = \frac{L - n(a, X)}{k} - \frac{n(a, X)}{\ell} = \frac{L - n(\pi a, \pi X)}{k} - \frac{n(\pi a, \pi X)}{\ell} = \delta(\pi a, \pi X).$$

Свойство 5) следует из определения (13.1) и свойства 3):

$$\begin{aligned} a \in A(X) &\leftrightarrow \forall a' \in A \ n(a, X) \leq n(a', X) \leftrightarrow \\ &\leftrightarrow \forall a' \in A \ n(\pi a, \pi X) \leq n(\pi a', \pi X) \leftrightarrow \\ &\leftrightarrow \forall a' \in \pi A \ n(\pi a, \pi X) \leq n(a', \pi X) \leftrightarrow \pi a \in (\pi A)(\pi X). \end{aligned}$$

Свойство 6) следует из свойства 5):

$$\begin{aligned} |A(X)| &= \sum_{a \in A} [a \in A(X)] = \sum_{a \in A} [\pi a \in (\pi A)(\pi X)] = \\ &= \sum_{a' \in \pi A} [a' \in (\pi A)(\pi X)] = |(\pi A)(\pi X)|. \end{aligned}$$

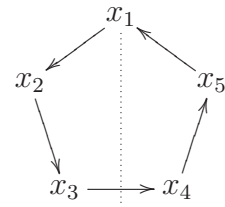
**Группа симметрий множества алгоритмов.** Рассмотрим подмножество всех перестановок, действие которых на множество  $A$  не меняет его:

$$\text{sym}A = \{\pi \in S_L : \pi A = A\}.$$

Если подействовать любой из перестановок  $\pi \in \text{sym}A$  на строки матрицы ошибок множества  $A$ , то получится ровно то же самое множество столбцов; переставив столбцы, можно получить исходную матрицу ошибок. Очевидно, множество  $\text{sym}A$  является группой. Будем называть её *группой симметрий* множества алгоритмов  $A$ .

**Пример 13.1.** Рассмотрим множество алгоритмов, заданное матрицей ошибок

$$\begin{matrix} & a_1 & a_2 & a_3 & a_4 & a_5 \\ \begin{matrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \end{matrix} & \begin{pmatrix} 1 & 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 & 1 \\ 1 & 0 & 0 & 1 & 1 \\ 1 & 1 & 0 & 0 & 1 \end{pmatrix} \end{matrix}$$



Группа симметрий данного множества алгоритмов совпадает с группой симметрий правильного пятиугольника и называется *диэдральной группой*. Образующими элементами группы являются циклическая перестановка  $\pi_1 = (x_1, x_2, x_3, x_4, x_5)$  и осевая симметрия  $\pi_2 = (x_2, x_5)(x_3, x_4)$ .

Пусть далее  $G \subseteq \text{sym}A$  — произвольная подгруппа группы  $\text{sym}A$ .

Для любой перестановки  $\pi \in G$  и любого алгоритма  $a \in A$  алгоритм  $\pi a$  снова лежит в  $A$ . В таких случаях говорят, что группа  $G$  *действует* на множестве  $A$ .

*Орбитой* алгоритма  $a \in A$  называется множество алгоритмов  $Ga = \{\pi a : \pi \in G\}$ . Орбита целиком лежит в  $A$ . Орбиты двух различных алгоритмов  $Ga, Ga'$  либо совпадают, либо не пересекаются. Следовательно, множество  $A$  разбивается на непересекающиеся подмножества — орбиты:

$$A = \bigsqcup_{\omega \in \Omega(A)} \omega = \bigsqcup_{\omega \in \Omega(A)} Ga_\omega,$$

где  $\Omega(A)$  — множество всех орбит в  $A$ ,  $a_\omega$  — произвольный представитель орбиты  $\omega$ .

Из свойства 2) леммы 13.1 следует, что алгоритмы одной орбиты обязательно лежат в одном слое. Обратное, вообще говоря, неверно.

**Лемма 13.2.** *Алгоритмы из одной орбиты имеют равные вероятности реализации и равные вклады в вероятность переобучения: для любой перестановки  $\pi$  из  $G$*

$$P(a, A) = P(\pi a, A); \quad Q_\varepsilon(a, A) = Q_\varepsilon(\pi a, A).$$

**Доказательство.** Воспользуемся определением вероятности реализации (13.3), свойствами 5), 6) из леммы 13.1 и свойством симметрии  $A = \pi A$ :

$$P(a, A) = \mathbf{E} \frac{[a \in A(X)]}{|A(X)|} = \mathbf{E} \frac{[\pi a \in (\pi A)(\pi X)]}{|(\pi A)(\pi X)|} = \mathbf{E} \frac{[\pi a \in A(\pi X)]}{|A(\pi X)|}.$$

Под знаком  $\mathbf{E}$  можно всюду заменить  $\pi X$  на  $X$ , так как результат не зависит от порядка суммирования разбиений:

$$P(a, A) = \mathbf{E} \frac{[\pi a \in A(X)]}{|A(X)|} = P(\pi a, A).$$

Воспользуемся определением вклада алгоритма, свойствами 4), 5), 6) из леммы 13.1 и свойством симметрии  $A = \pi A$ :

$$\begin{aligned} Q_\varepsilon(a, A) &= \mathbf{E} \frac{[a \in A(X)]}{|A(X)|} [\delta(a, X) \geq \varepsilon] = \\ &= \mathbf{E} \frac{[\pi a \in (\pi A)(\pi X)]}{|(\pi A)(\pi X)|} [\delta(\pi a, \pi X) \geq \varepsilon] = \\ &= \mathbf{E} \frac{[\pi a \in A(\pi X)]}{|A(\pi X)|} [\delta(\pi a, \pi X) \geq \varepsilon]. \end{aligned}$$

Вновь заменяя  $\pi X$  на  $X$  под знаком  $\mathbf{E}$ , получим:

$$Q_\varepsilon(a, A) = \mathbf{E} \frac{[\pi a \in A(X)]}{|A(X)|} [\delta(\pi a, X) \geq \varepsilon] = Q_\varepsilon(\pi a, A). \quad \blacksquare$$

### Разложение вероятности переобучения по орбитам множества алгоритмов.

Из леммы 13.2 о равном вкладе алгоритмов одной орбиты немедленно следует формула разложения вероятности переобучения по орбитам. Она является основным инструментом получения точных оценок для рандомизированного метода минимизации эмпирического риска.

**Теорема 13.3.** Для любой генеральной выборки  $\mathbb{X}$ , любого множества алгоритмов  $A$  с попарно различными векторами ошибок и любого  $\varepsilon \in [0, 1]$  справедлива формула разложения вероятности переобучения по орбитам множества  $A$ :

$$Q_\varepsilon(A) = \sum_{\omega \in \Omega(A)} |\omega| \mathbb{E} \frac{[a_\omega \in A(X)]}{|A(X)|} [\delta(a_\omega, X) \geq \varepsilon], \quad (13.4)$$

где  $\Omega(A)$  — множество всех орбит в  $A$ ,  $a_\omega$  — произвольный представитель орбиты  $\omega$ .

**Доказательство.** Перегруппируем слагаемые в (13.2) по орбитам множества  $A$ , затем применим лемму 13.2 о равном вкладе алгоритмов одной орбиты:

$$\begin{aligned} Q_\varepsilon(A) &= \sum_{\omega \in \Omega(A)} \sum_{a \in \omega} \mathbb{E} \frac{[a \in A(X)]}{|A(X)|} [\delta(a, X) \geq \varepsilon] = \\ &= \sum_{\omega \in \Omega(A)} |\omega| \mathbb{E} \frac{[a_\omega \in A(X)]}{|A(X)|} [\delta(a_\omega, X) \geq \varepsilon]. \end{aligned}$$

■

### Разложение вероятности переобучения по орбитам множества разбиений.

В некоторых случаях удобнее делать группировку слагаемых не по орбитам множества алгоритмов, а по орбитам множества разбиений. Напомним, что через  $[\mathbb{X}]^\ell$  мы обозначаем множество всех  $\ell$ -элементных подмножеств генеральной выборки  $\mathbb{X}$ .

Представим вероятность переобучения в виде суммы вкладов разбиений:

$$Q_\varepsilon(A) = \frac{1}{C_L^\ell} \sum_{X \in [\mathbb{X}]^\ell} Q_\varepsilon(X, A), \quad Q_\varepsilon(X, A) = \frac{1}{|A(X)|} \sum_{a \in A(X)} [\delta(a, X) \geq \varepsilon],$$

где  $Q_\varepsilon(X, A)$  — вклад разбиения  $X \sqcup \bar{X}$  в вероятность переобучения. Поскольку разбиениям  $X \sqcup \bar{X}$  взаимно однозначно соответствуют выборки  $X \in [\mathbb{X}]^\ell$ , далее будем говорить также о вкладе выборки  $X$  в вероятность переобучения.

Орбитой выборки  $X \in [\mathbb{X}]^\ell$  называется множество выборок  $GX = \{\pi X : \pi \in G\}$ . Множество всех выборок длины  $\ell$  разбивается на непересекающиеся орбиты:

$$[\mathbb{X}]^\ell = \bigsqcup_{\tau \in \Omega[\mathbb{X}]^\ell} \tau = \bigsqcup_{\tau \in \Omega[\mathbb{X}]^\ell} GX_\tau,$$

где  $\Omega[\mathbb{X}]^\ell$  — множество всех орбит в  $[\mathbb{X}]^\ell$ ,  $X_\tau$  — произвольный представитель орбиты  $\tau$ .

**Лемма 13.4.** Выборки из одной орбиты дают равные вклады в вероятность переобучения:  $Q_\varepsilon(X, A) = Q_\varepsilon(\pi X, A)$  для любой перестановки  $\pi$  из  $G$ .

**Доказательство.** Запишем определение вклада выборки, затем воспользуемся свойствами 4), 5), 6) из леммы 13.1 и свойством симметрии  $A = \pi A$ :

$$\begin{aligned} Q_\varepsilon(X, A) &= \sum_{a \in A(X)} \frac{[\delta(a, X) \geq \varepsilon]}{|A(X)|} = \sum_{a' \in (\pi A)(\pi X)} \frac{[\delta(a', \pi X) \geq \varepsilon]}{|(\pi A)(\pi X)|} = \\ &= \sum_{a' \in A(\pi X)} \frac{[\delta(a', \pi X) \geq \varepsilon]}{|A(\pi X)|} = Q_\varepsilon(\pi X, A). \end{aligned}$$

■

**Теорема 13.5.** Для любой генеральной выборки  $\mathbb{X}$ , любого множества алгоритмов  $A$  с попарно различными векторами ошибок и любого  $\varepsilon \in [0, 1]$  справедлива формула разложения вероятности переобучения по орбитам множества  $[\mathbb{X}]^\ell$ :

$$Q_\varepsilon(A) = \frac{1}{C_L^\ell} \sum_{\tau \in \Omega[\mathbb{X}]^\ell} \frac{|\tau|}{|A(X_\tau)|} \sum_{a \in A(X_\tau)} [\delta(a, X_\tau) \geq \varepsilon]. \quad (13.5)$$

где  $\Omega[\mathbb{X}]^\ell$  — множество всех орбит в  $[\mathbb{X}]^\ell$ ,  $X_\tau$  — произвольный представитель орбиты  $\tau$ .

Доказательство аналогично доказательству теоремы 13.3.

## Булев куб

ToDo<sup>1</sup>

## Центральный слой хэммингова шара

## Пучок монотонных цепей

## Резюме

Теоретико-групповой подход упрощает получение точных оценок вероятности переобучения для сложных модельных семейств алгоритмов, обладающих свойством симметрии. Он применим только к методу рандомизированной минимизации эмпирического риска. Чтобы его применить, необходимо описать группу симметрий семейства алгоритмов и в явном виде построить все орбиты действия этой группы на множестве алгоритмов, либо (если это окажется проще) на множестве всех разбиений генеральной выборки. Если вся группа симметрий окажется слишком сложной, можно взять произвольную её подгруппу. В таком случае оценка останется точной, но, скорее всего, возрастёт её сложность (увеличится число слагаемых).

Теоретико-групповая оценка вероятности переобучения для пучка монотонных цепей является точной и учитывает свойства расслоения и связности.

**В следующей лекции** мы вернёмся к функционалу равномерного отклонения, который используется в VC-теории, и покажем, что связность для него учесть можно, а расслоение — нельзя.

## Упражнения

ToDo<sup>2</sup>

## 14 Оценки вероятности равномерного отклонения

Функционал вероятности *равномерного отклонения* частот ошибок в двух выборках вводится в VC-теории и берётся за основу во многих последующих исследованиях (см. обзоры [107, 57, 12]). Он является верхней оценкой вероятности переобучения, не зависящей от метода обучения  $\mu$ , что даёт ему определённое практическое преимущество. С помощью обращения из него получается верхняя оценка частоты ошибок на контрольной выборке. Её минимизация, в свою очередь, приводит к новому методу обучения  $\mu$ , который, конечно же, отличается от обычной минимизации эмпирического риска. По идее, новый метод  $\mu$  должен быть менее подвержен переобучению, поскольку он строился из соображений оптимизации обобщающей способности. С другой стороны, завышенность использованной оценки может повлечь за собой неоптимальность или даже неадекватность нового метода.

Получив несколькими способами верхние оценки вероятности большого равномерного отклонения, мы выясним, что все они учитывают связность, но не учитывают расслоение. Именно в этом проявляется завышенность оценок вероятности переобучения через вероятность равномерного отклонения.

### §14.1 Техника порождающих и запрещающих множеств

Напомним, что *переобученностью* алгоритма  $a$  на выборке  $X$  мы называем разность частоты его ошибок на двух выборках, контрольной и обучающей:

$$\delta(a, X) = \nu(a, \bar{X}) - \nu(a, X).$$

Функционал вероятности переобучения

$$Q_\varepsilon(\mu, \mathbb{X}) = \mathbb{P}[\delta(\mu X, X) \geq \varepsilon]$$

оценивается сверху вероятностью большого равномерного (по множеству алгоритмов  $A$ ) отклонения частот в двух подвыборках:

$$Q_\varepsilon(\mu, \mathbb{X}) \leq \tilde{Q}_\varepsilon(A, \mathbb{X}) = \mathbb{P} \max_{a \in A} [\delta(a, X) \geq \varepsilon] = \mathbb{P} \left[ \max_{a \in A} \delta(a, X) \geq \varepsilon \right].$$

**Метод максимизации переобученности.** Принцип порождающих и запрещающих множеств может быть применён к функционалу  $\tilde{Q}_\varepsilon(A, \mathbb{X})$ , если специальным образом ввести метод обучения. Эта идея принадлежит Илье Толстихину.

**Определение 14.1.** Метод обучения  $\mu$  называется *максимизацией переобученности (discrepancy maximization)*, если  $\mu X = \arg \max_{a \in A} \delta(a, X)$ .

Из определения следует, что если метод  $\mu$  максимизирует переобученность, то вероятность переобучения совпадает с вероятностью равномерного отклонения,

$$Q_\varepsilon(\mu, \mathbb{X}) = \tilde{Q}_\varepsilon(A, \mathbb{X}).$$

**Лемма 14.1.** Если метод  $\mu$  максимизирует переобученность, то для каждого алгоритма  $a \in A$  множество  $X_a = \{x_{ab} \in \mathbb{X} : a \prec b\}$  является порождающим, а множество  $X'_a = \{x_{ba} \in \mathbb{X} : b \prec a\}$  — запрещающим в смысле Гипотезы 9.1:

$$[\mu X = a] \leq [X_a \subseteq X][X'_a \subseteq \bar{X}].$$

**Доказательство.** Из того, что  $\mu X = a$  следует

$$\delta(a, X) \geq \delta(b, X) \quad \text{для всех } b \in A. \quad (14.1)$$

Возьмём произвольный  $x_{ab} \in X_a$ . Тогда  $I(a, x_{ab}) = 0$ ,  $I(b, x_{ab}) = 1$ . Если допустить, что  $x_{ab} \in \bar{X}$ , то получим

$$\delta(b, X) = \nu(b, \bar{X}) - \nu(b, X) = \nu(a, \bar{X}) + \frac{1}{k} - \nu(a, X) > \delta(a, X),$$

что противоречит (14.1), значит,  $x_{ab} \in X$ . В силу его произвольности  $X_a \subseteq X$ .

Возьмём произвольный  $x_{ba} \in X'_a$ . Тогда  $I(b, x_{ba}) = 0$ ,  $I(a, x_{ba}) = 1$ . Если допустить, что  $x_{ba} \in X$ , то получим

$$\delta(b, X) = \nu(b, \bar{X}) - \nu(b, X) = \nu(a, \bar{X}) - \nu(a, X) + \frac{1}{\ell} > \delta(a, X),$$

что противоречит (14.1), значит,  $x_{ba} \in \bar{X}$ . В силу его произвольности  $X'_a \subseteq \bar{X}$ . ■

**Оценка связности.** Из Леммы 14.1 и общей оценки расслоения–связности (9.2) следует оценка связности для вероятности равномерного отклонения.

**Теорема 14.2.** Пусть векторы ошибок всех алгоритмов из  $A$  попарно различны. Тогда имеет место верхняя оценка вероятности равномерного отклонения:

$$\tilde{Q}_\varepsilon(A, \mathbb{X}) \leq \sum_{a \in A} \frac{C_{L-q-d}^{\ell-q}}{C_L^\ell} H_{L-q-d}^{\ell-q, m-d} \left( \frac{\ell}{L} (m - \varepsilon k) \right), \quad (14.2)$$

где  $q = q(a) = |X_a|$  — верхняя связность,  $d = d(a) = |X'_a|$  — нижняя связность алгоритма  $a$ ,  $m = n(a, \mathbb{X})$  — число ошибок алгоритма  $a$  на генеральной выборке.

Эта оценка гораздо хуже аналогичной по структуре оценки расслоения–связности (9.5) для пессимистичной МЭР. Единственное различие заключается в том, что неоптимальность  $r(a)$  заменяется на нижнюю связность  $d(a)$ . Неоптимальность растёт с номером слоя линейно, соответственно, вклады алгоритмов в  $Q_\varepsilon$  убывают экспоненциально. Нижняя связность  $d(a)$  не превышает  $r(a)$  и, как показывают эксперименты [29], концентрируется вокруг одного и того же значения во всех слоях, поэтому алгоритмы всех слоёв вносят примерно равный вклад в оценку (14.2). Таким образом, эта оценка учитывает связность, но не учитывает расслоение.

## §14.2 Техника цепных разложений

Метод оценивания вероятности равномерного отклонения, основанный на *цепных разложениях* (chain expansion), предложен Денисом Кочедыковым в [29].

**Теорема 14.3.** Для любых  $\mathbb{X}$ ,  $A$  и любого  $\varepsilon \in [0, 1]$

$$\tilde{Q}_\varepsilon(A, \mathbb{X}) \leq \sum_{a \in A} \left[ \{s_a(\varepsilon)\} < \frac{\ell}{L} \right] \frac{C_{L-d}^\ell h_{L-d}^{\ell, m-d}(s_a(\varepsilon))}{C_L^\ell}. \quad (14.3)$$

где  $d = d(a)$  — нижняя связность алгоритма  $a$ ,  $m = n(a, \mathbb{X})$  — число ошибок алгоритма  $a$  на генеральной выборке,  $s_a(\varepsilon) = \frac{\ell}{L}(m - \varepsilon k)$ ,

**Доказательство.** Обозначим через  $U_a(X, \varepsilon)$  или просто  $U_a$  индикатор переобученности алгоритма  $a$  на выборке  $X$ :

$$U_a = [\delta(a, X) \geq \varepsilon] = [n(a, X) \leq s_a(\varepsilon)].$$

Обозначим через  $\bar{U}_a$  отрицание  $U_a$ , то есть  $\bar{U}_a = 1 - U_a$ .

Воспользуемся тем, что на множестве алгоритмов  $A$  задано отношение частичного порядка как естественное отношение порядка на булевых векторах ошибок алгоритмов. Дополним это отношение до линейного порядка произвольным образом. Представим вероятность максимума бинарных величин как сумму вероятностей, воспользовавшись *цепным разложением*:

$$\tilde{Q}_\varepsilon = \mathbb{P} \max_a U_a = \sum_{a \in A} \mathbb{P} U_a \prod_{\substack{b \in A \\ b < a}} \bar{U}_b.$$

Оставим в цепном разложении только такие сомножители  $\bar{U}_b$ , что  $b \prec a$ . Остальные сомножители тривиально оценим сверху единицей,  $\bar{U}_b \leq 1$ ,  $b \not\prec a$ . Тогда

$$\tilde{Q}_\varepsilon \leq \sum_{a \in A} \mathbb{P} \prod_{\substack{b \in A \\ b \prec a}} U_a \bar{U}_b.$$

Рассмотрим произведение  $U_a \bar{U}_b$  для произвольных  $b$  и  $a$  таких, что  $b \prec a$ :

$$\begin{aligned} U_a \bar{U}_b &= [n(a, X) \leq s_a(\varepsilon)] [n(b, X) > s_b(\varepsilon)] = \\ &= [s_a(\varepsilon) - \frac{\ell}{L} = s_b(\varepsilon) < n(b, X) \leq n(a, X) \leq s_a(\varepsilon)] = \\ &= [s_a(\varepsilon) - \frac{\ell}{L} < n(a, X) \leq s_a(\varepsilon)] [n(b, X) = n(a, X)] = \\ &= [n(a, X) = \lfloor s_a(\varepsilon) \rfloor] [\{s_a(\varepsilon)\} < \frac{\ell}{L}] [x_{ba} \in \bar{X}], \end{aligned}$$

где  $\lfloor s \rfloor$  — целая часть  $s$ ,  $\{s\}$  — дробная часть  $s$ . Тогда

$$\tilde{Q}_\varepsilon \leq \sum_{a \in A} \mathbb{P} \prod_{\substack{b \in A \\ b \prec a}} U_a \bar{U}_b = \sum_{a \in A} \mathbb{P} \prod_{\substack{b \in A \\ b \prec a}} [n(a, X) = \lfloor s_a(\varepsilon) \rfloor] [\{s_a(\varepsilon)\} < \frac{\ell}{L}] [x_{ba} \in \bar{X}].$$



Сомножители, не зависящие от  $b$ , вынесем за знак произведения. Введём множество  $X'_a = \{x_{ba} \in \mathbb{X} : b \prec a\}$  и заметим, что

$$\prod_{\substack{b \in A \\ b \prec a}} [x_{ba} \in \bar{X}] = [X'_a \subseteq \bar{X}].$$

Таким образом,

$$\tilde{Q}_\varepsilon \leq \sum_{a \in A} [\{s_a(\varepsilon)\} < \frac{\ell}{L}] \mathbf{P}[n(a, X) = \lfloor s_a(\varepsilon) \rfloor] [X'_a \subset \bar{X}].$$

Учитывая, что  $|X'_a| = d(a)$ , получим оценку (14.2). Теорема доказана.  $\blacksquare$

Оценка (14.3) по своей структуре аналогична оценке (14.2). В ней есть два улучшения: во-первых, суммирование производится не по всем слоям семейства алгоритмов (при  $\ell = k$  учитывается только каждый второй слой), во-вторых, вместо «левого хвоста» гипергеометрического распределения  $H$  в ней фигурирует значение  $h$  гипергеометрического распределения в одной точке. С другой стороны, оценка (14.3) учитывает только нижнюю связность, что в итоге делает её более слабой по сравнению с оценкой (14.2). При этом обе оценки не учитывают расслоение.

### §14.3 Техника случайных блужданий

Вероятность равномерного отклонения частот в случае монотонной цепи может совпадать с вероятностью равномерного отклонения эмпирических распределений. Этот любопытный факт был замечен Иваном Шаниным.

Для произвольного семейства  $A$  выразим равномерное отклонение частоты ошибок в выборках  $X$  и  $\bar{X}$  через индикаторы  $b_i = b_i(X) = [x_i \in X]$ :

$$\begin{aligned} D(X) &= \max_{a \in A} \delta(a, X) = \max_{a \in A} (\nu(a, \bar{X}) - \nu(a, X)) = \\ &= \max_{a \in A} \sum_{i=1}^L I(a, x_i) \left( \underbrace{\frac{1}{k} [x_i \in \bar{X}]}_{1-b_i} - \underbrace{\frac{1}{\ell} [x_i \in X]}_{b_i} \right) = \\ &= \frac{L}{\ell k} \max_{a \in A} \sum_{i=1}^L I(a, x_i) \left( \frac{\ell}{L} - b_i \right). \end{aligned} \quad (14.4)$$

**Монотонная цепь алгоритмов** — это модельное семейство  $A = \{a_0, a_1, \dots, a_D\}$ , в котором алгоритм  $a_0$  допускает  $m$  ошибок на генеральной выборке, а каждый последующий алгоритм хуже предыдущего на одном объекте (определение 8.8, стр. 71). Перенумеруем объекты  $x_1, \dots, x_L$  таким образом, чтобы

$$I(a_d, x_i) = [i \leq m + d], \quad i = 1, \dots, L, \quad d = 0, \dots, D. \quad (14.5)$$

**Теорема 14.4.** Для монотонной цепи алгоритмов  $A$ , произвольной генеральной выборки  $\mathbb{X}$  и произвольного  $\varepsilon \in [0, 1]$  справедлива точная оценка вероятности того, что равномерное отклонение не превысит  $\varepsilon$ :

$$\mathbf{P} \left[ \max_{a \in A} \delta(a, X) \leq \varepsilon \right] = \frac{G_L^\ell [g_L^-(\varepsilon), L]}{C_L^\ell},$$

где  $g_t^-(\varepsilon)$  — левая граница усечённого треугольника Паскаля:

$$g_t^-(\varepsilon) = \frac{\ell}{L}(t - \varepsilon k) [m \leq t \leq m + D].$$

**Доказательство.** В случае монотонной цепи представление (14.4) упрощается после подстановки в него (14.5):

$$D(X) = \frac{L}{\ell k} \max_{d=0..D} \sum_{i=1}^{m+d} \left( \frac{\ell}{L} - b_i \right) = \frac{L}{\ell k} \max_{d=0..D} \left( \frac{\ell(m+d)}{L} - B_{m+d} \right),$$

где  $B_{m+d} = b_1 + \dots + b_{m+d}$ . Таким образом, равномерное отклонение частоты ошибок в выборках  $\bar{X}$  и  $X$  выражается через равномерное отклонение числа единиц среди первых  $m + d$  членов бинарной последовательности  $b_1, \dots, b_L$  от «ожидаемого» числа единиц  $\frac{\ell}{L}(m + d)$ .

Найдём теперь вероятность того, что  $D(X)$  не превышает  $\varepsilon$ :

$$\begin{aligned} \mathbb{P}[D(X) \leq \varepsilon] &= \mathbb{P} \left[ \max_{d=0..D} \left( \frac{\ell}{L}(m+d) - B_{m+d} \right) \leq \frac{\ell}{L}\varepsilon k \right] = \\ &= \mathbb{P} \prod_{d=0}^D \left[ \frac{\ell}{L}(m+d - \varepsilon k) \leq B_{m+d} \right] = \mathbb{P} \prod_{t=1}^L [g_t^-(\varepsilon) \leq B_t]. \end{aligned}$$

Полученное выражение в точности совпадает с вероятностью равномерного отклонения эмпирических распределений (4.9), для которой уже найдено точное выражение (4.6) через усечённый слева треугольник Паскаля. Теорема доказана. ■

## §14.4 Понятие радемахеровской сложности

*Радемахеровская сложность* (Rademacher complexity) введена в теорию статистического обучения В. Кольчинским [77, 76, 78]. Это понятие сложности семейства алгоритмов отличается математическим изяществом и удобством использования в приложениях [52, 54, 53]. Считается, что оно даёт наиболее точные оценки вероятности большого равномерного отклонения частот.

**Определение 14.2.** *Радемахеровская сложность (Rademacher complexity) множества алгоритмов  $A$  на выборке  $\mathbb{X}$  есть*

$$\mathcal{R}(A, \mathbb{X}) = \mathbb{E} \max_{a \in A} \delta(a, X) = 2 \mathbb{E} \max_{a \in A} \frac{1}{L} \sum_{i=1}^L \sigma_i(X) I(a, x_i),$$

где  $\sigma_i(X)$  — радемахеровская случайная величина,

$$\sigma_i(X) = \frac{L}{2k} [x_i \in \bar{X}] - \frac{L}{2\ell} [x_i \in X] = \begin{cases} -\frac{L}{2\ell}, & x_i \in X, \\ +\frac{L}{2k}, & x_i \in \bar{X}. \end{cases}$$

Обычно рассматривается случай  $\ell = k$ . Тогда радемахеровская случайная величина  $\sigma_i(X)$  принимает одной из двух значений  $-1, +1$ , каждое с вероятностью  $\frac{1}{2}$ .

Смысл этой величины легко понять, если заметить, что  $r(a, \sigma) = \frac{1}{L} \sum_{i=1}^L \sigma_i I(a, x_i)$  есть выборочная ковариация индикатора ошибок и случайного радемахеровского шума. Супремум  $\sup_{a \in A} |r(a, \sigma)|$  выделяет в семействе  $A$  алгоритм, вектор ошибок которого наиболее похож на случайный. Математическое ожидание  $E$  по вектору шума необходимо для того, чтобы величина сложности  $\mathcal{R}$  не зависела от выбора конкретного вектора шума. Степень близости вектора потерь и вектора шума, выраженная с помощью ковариации, вполне соответствует интуитивным представлениям о сложности: чем сложнее семейство, тем выше шансы найти в нём алгоритм, «похожий» на произвольный шум. Благодаря линейности ковариации радемахеровская сложность обладает удобными алгебраическими свойствами, что упрощает её оценивание.

**Радемахеровская оценка вероятности переобучения**

ToDo<sup>3</sup>

**Радемахеровская сложность монотонной цепи**

**Экспериментальное сравнение оценок вероятности равномерного отклонения**

## Резюме

Функционал равномерного отклонения, введённый в теории Вапника-Червоненкиса, учитывает связность семейства алгоритмов, но не учитывает его расслоение. Из-за этого он может давать сильно завышенные оценки вероятности переобучения. Тем не менее, он активно используется в теории статистического обучения. Его достоинство в том, что он не зависит от метода обучения, что позволяет разрабатывать новые методы обучения путём минимизации его обращённых верхних оценок.

При получении верхних оценок вероятности равномерного отклонения могут накапливаться дополнительные погрешности. Среди трёх рассмотренных техник первые две дают завышенные оценки, третья даёт точную оценку, но только для частного случая монотонной цепи алгоритмов.

**В следующей лекции** мы вернёмся к модельным семействам алгоритмов, чтобы продолжить поиски альтернативных способов оценивания вероятности переобучения. Будет получено несколько точных оценок, не использующих принцип порождающих и запрещающих множеств или граф расслоения–связности.

## Упражнения

**Задача 14.1 (5\*).** Получить точную оценку вероятности равномерного отклонения  $\tilde{Q}_\varepsilon(A, \mathbb{X})$  для унимодальной цепи алгоритмов.

**Задача 14.2 (10\*).** Получить точную оценку  $\tilde{Q}_\varepsilon(A, \mathbb{X})$  для произвольного множества алгоритмов, заданного графом расслоения–связности.

## 15 Точные интервальные и блочные оценки

Продолжим рассмотрение точных оценок вероятности переобучения для модельных семейств алгоритмов, обладающих свойствами расслоения и связности. Начнём с семейства, вообще не обладающего расслоением и покажем, что в этом случае оценка становится вырожденной.

### §15.1 Слой булева куба

Рассмотрим множество  $A$ , состоящее из всех  $C_L^m$  алгоритмов, допускающих ровно  $m$  ошибок на полной выборке  $\mathbb{X}$ , и имеющих попарно различные векторы ошибок. Поскольку все возможные векторы ошибок образуют булев куб размерности  $L$ , то векторы ошибок множества  $A$  — это  $m$ -й слой булева куба.

**Теорема 15.1.** Пусть  $\mu$  — произвольный метод минимизации эмпирического риска,  $A$  —  $m$ -й слой булева куба. Тогда для любого  $\varepsilon \in [0, 1]$

$$Q_\varepsilon = [\varepsilon k \leq m \leq L - \varepsilon \ell].$$

**Доказательство.** Согласно Теореме 5.4, в случае одного слоя алгоритмов минимизация эмпирического риска эквивалентна принципу равномерной сходимости. Следовательно,

$$Q_\varepsilon = \mathbb{P} \left[ \max_{a \in A} (\nu(a, \bar{X}) - \nu(a, X)) \geq \varepsilon \right].$$

Если  $m \leq k$ , то максимум достигается на таком  $a \in A$ , у которого в контроль попадают все  $m$  ошибок, а в обучение — ни одной. Тогда  $\nu(a, \bar{X}) = \frac{m}{k}$ ,  $\nu(a, X) = 0$  и

$$Q_\varepsilon = \mathbb{P} \left[ \frac{m}{k} - 0 \geq \varepsilon \right] = [m \geq \varepsilon k] = [\varepsilon k \leq m \leq k].$$

Если  $m > k$ , то максимум достигается на таком  $a \in A$ , который допускает ошибки на всех контрольных объектах. Тогда

$$Q_\varepsilon = \mathbb{P} \left[ 1 - \frac{m-k}{\ell} \geq \varepsilon \right] = [m \leq L - \varepsilon \ell] = [k < m \leq L - \varepsilon \ell].$$

Объединяя два несовместных случая  $m \leq k$  и  $m > k$ , получаем требуемое. ■

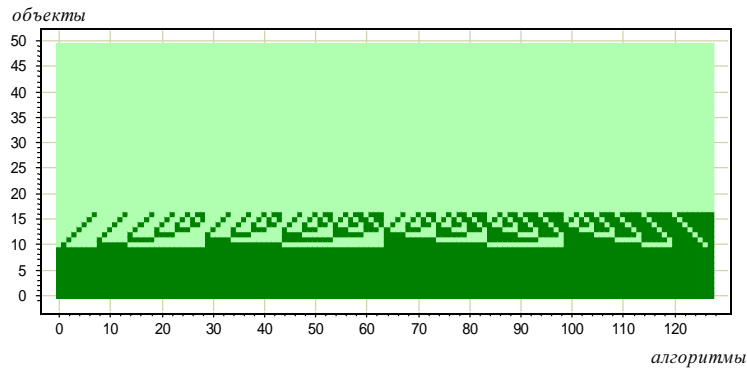
Таким образом, точная оценка вероятности переобучения является вырожденной и принимает значения либо 0, либо 1. Хотя этот результат тривиальный и в определённом смысле отрицательный, он позволяет сделать несколько важных выводов.

Во-первых, алгоритмы самых нижних слоёв,  $m < \lceil \varepsilon k \rceil$ , не вносят вклад в переобучение.

Во-вторых, нижний слой семейства алгоритмов, содержащий алгоритмы с числом ошибок не менее  $\lceil \varepsilon k \rceil$ , не должен содержать все такие алгоритмы. Не имеет смысла применять метод МЭР к слишком богатым семействам алгоритмов.

## §15.2 Интервал булева куба и его расслоение

Предположим, что векторы ошибок всех алгоритмов из  $A$  попарно различны и образуют интервал ранга  $m$  в  $L$ -мерном булевом кубе. Это означает, что объекты делятся на три группы:  $m_0$  «внутренних» объектов, на которых ни один из алгоритмов не допускает ошибок;  $m_1$  «шумовых» объектов, на которых все алгоритмы допускают ошибки; и  $m$  «пограничных» объектов, на которых реализуются все  $2^m$  вариантов допустить ошибки. Других объектов нет:  $m_0 + m_1 + m = L$ . Алгоритмы допускают от  $m_1$  до  $m_1 + m$  ошибок. Число алгоритмов в  $A$  равно  $2^m$ . Интервал булева куба обладает свойствами расслоения и связности. На рис. 15.1 показан пример матрицы ошибок для интервала булева куба ранга  $m = 7$ .



**Рис. 15.1.** Матрица ошибок интервала булева куба при  $L = 50$ ,  $m = 7$ ,  $m_1 = 10$ . Число алгоритмов в семействе  $|A_m| = 2^m = 128$ . Алгоритмы пронумерованы по слоям слева направо, например,  $A_2$  — это первые 29 алгоритмов,  $A_3$  — первые 64.

Для большей общности рассмотрим множество алгоритмов  $A_t$ , образованное нижними  $t$  слоями интервала булева куба. Это алгоритмы, допускающие не более  $t$  ошибок на пограничных объектах,  $|A_t| = C_m^0 + C_m^1 + \dots + C_m^t$ . Параметр  $t$  может принимать значения  $0, \dots, m$ . При  $t = 0$  имеем единственный алгоритм, при  $t = m$  — полный интервал ранга  $m$ . Это модельное семейство интересно тем, что оно позволяет исследовать зависимость вероятности переобучения  $Q_\varepsilon$  от числа слоёв.

**Теорема 15.2.** Пусть  $\mu$  — пессимистичный метод минимизации эмпирического риска,  $A = A_t$  — нижние  $t$  слоёв интервала булева куба с  $m$  пограничными и  $m_1$  шумовыми объектами. Тогда для любого  $\varepsilon \in [0, 1]$  вероятность переобучения есть

$$Q_\varepsilon = \sum_{s=0}^m \sum_{s_1=0}^{m_1} \frac{C_m^s C_{m_1}^{s_1} C_{L-m-m_1}^{\ell-s-s_1}}{C_L^\ell} [s_1 \leq \frac{\ell}{L}(m_1 + \min\{t, m - s\} - \varepsilon k)].$$

**Доказательство.** Обозначим через  $X_0, X_1, S$  соответственно множества всех внутренних, шумовых и пограничных объектов; а через  $s_0, s_1, s$  соответственно — число внутренних, шумовых и пограничных объектов, попавших в обучающую выборку  $X$ .

Поскольку метод  $\mu$  пессимистичный, он всегда будет выбирать из  $A$  алгоритм, который не ошибается на всех обучающих пограничных объектах, но ошибается

на всех контрольных пограничных объектах. Поэтому

$$\nu(\mu X, X) = \frac{s_1}{\ell}; \quad \nu(\mu X, \bar{X}) = \frac{(m_1 - s_1) + \min\{t, m - s\}}{k}.$$

Число разбиений  $X \sqcup \bar{X}$ , при которых  $|X_0 \cap X| = s_0$ ,  $|X_1 \cap X| = s_1$ ,  $|S \cap X| = s$ , равно  $C_{m_0}^{s_0} C_{m_1}^{s_1} C_m^s$ . Следовательно, вероятность переобучения представима в виде

$$Q_\varepsilon = \sum_{\substack{s_0=0 \\ s_0+s_1+s=\ell}}^{m_0} \sum_{s_1=0}^{m_1} \sum_{s=0}^m \frac{C_{m_0}^{s_0} C_{m_1}^{s_1} C_m^s}{C_L^\ell} \left[ \frac{(m_1 - s_1) + \min\{t, m - s\}}{k} - \frac{s_1}{\ell} \geq \varepsilon \right].$$

Чтобы получить утверждение теоремы, достаточно воспользоваться соотношениями  $m_0 + m_1 + m = L$  и  $s_0 + s_1 + s = \ell$  и преобразовать неравенство в квадратных скобках к виду  $s_1 \leq \frac{\ell}{L}(m_1 + \min\{t, m - s\} - \varepsilon k)$ . ■

Интервал булева куба является уникальным примером семейства, для которого эффект расслоения практически не способствует снижению переобучения.

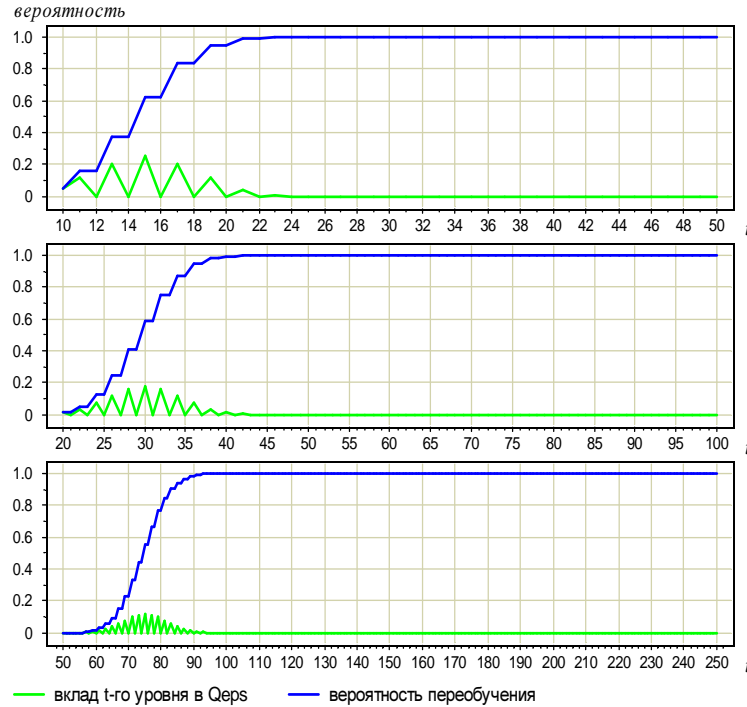
**Следствие 15.2.1.** *Если  $\mu$  — пессимистичная минимизация эмпирического риска,  $A_m$  — интервал ранга  $t$  булева куба, то вероятность переобучения  $Q_\varepsilon$  совпадает с вероятностью равномерного отклонения:*

$$Q_\varepsilon = \tilde{Q}_\varepsilon = \mathbb{P} \left[ \max_{a \in A} \delta(a, X) \geq \varepsilon \right].$$

**Доказательство.** Пессимистичный метод  $\mu$  всегда выбирает из  $A_m$  алгоритм, который не ошибается на всех обучающих пограничных объектах, но ошибается на всех контрольных пограничных объектах, что эквивалентно максимизации  $\delta(a, X)$ . ■

**Вычислительный эксперимент.** На рис. 15.2 представлены графики зависимости вероятности переобучения  $Q_\varepsilon$  от числа нижних слоёв  $t = m_1, \dots, m_1 + m$ . Три эксперимента отличались длиной генеральной выборки (200, 400, 1000), при этом сохранялись пропорции  $\frac{m}{L} = 0.2$  и  $\frac{m_1}{L} = 0.05$ , то есть генеральная выборка всегда содержала 20% пограничных и 5% шумовых объектов. На графиках также показаны вклады слоёв в значение функционала  $Q_\varepsilon$ . Только нижние слои дают ненулевые вклады (каждый второй слой вообще не вносит вклад в  $Q_\varepsilon$  в силу отношения  $\frac{\ell}{L} = \frac{1}{2}$ , этим объясняется зубчатость графиков вкладов). Оказывается, что 20% пограничных объектов — это настолько мощный интервал, что при всех трёх значениях  $L$  вероятность переобучения быстро достигает значения 1. Вероятность переобучения близка к нулю только если брать самые нижние слои интервала (не более 2% от длины выборки).

**Интерпретации и выводы.** Приходится констатировать отрицательный, по сути, результат. Интервал булева куба с «разумными» на первый взгляд параметрами  $m_1, t$  оказывается слишком богатым семейством алгоритмов. Оценки вероятности переобучения  $Q_\varepsilon$  для интервала близки к нулю лишь при очень низких значениях  $m_1, t$ . Отсюда следуют два вывода.



**Рис. 15.2.** Зависимость вероятности переобучения  $Q_\varepsilon$  от числа  $t$  нижних слоёв при  $\varepsilon = 0.05$ . Верхний график:  $\ell = k = 100$ ,  $m_1 = 10$ ,  $m = 40$ . Средний график:  $\ell = k = 200$ ,  $m_1 = 20$ ,  $m = 80$ . Нижний график:  $\ell = k = 500$ ,  $m_1 = 50$ ,  $m = 200$ .

Во-первых, хорошая обобщающая способность вряд ли возможна, если в выборке есть значительное количество пограничных объектов, на которых алгоритмы семейства допускают ошибки всеми возможными способами. Фактически, доля таких объектов добавляется к величине переобученности. Может возникнуть подозрение, что это происходит из-за пессимистичности МЭР, однако оценка для рандомизированного МЭР практически столь же плоха (упражнение 15.2).

Во-вторых, использовать интервал булева куба как модель реальных семейств вряд ли целесообразно. Гипотеза о существовании слоя пограничных объектов представляется достаточно разумной. Однако в реальных задачах алгоритмами семейства реализуются, по всей видимости, далеко не все способы допустить ошибки на пограничных объектах. Возможно, более адекватной была бы модель, в которой тем или иным способом вводится характеристика «степени граничности» объектов и оценивается распределение этой характеристики в выборке.

### §15.3 Блочная оценка

Допустим, что векторы ошибок всех алгоритмов  $A = \{a_1, \dots, a_D\}$  попарно различны, метод  $\mu$  — пессимистичная минимизация эмпирического риска. Значения  $I(a_d, x_i)$  образуют бинарную  $L \times D$ -матрицу ошибок, столбцы которой являются векторами ошибок алгоритмов, строки соответствуют объектам. Обозначим через  $b = (b_1, \dots, b_D)$  произвольный бинарный вектор размерности  $D$ . Выборка  $\mathbb{X}$  разбивается на непересекающиеся блоки  $U_b \subseteq \mathbb{X}$  так, что всем объектам в блоке соответствует

одна и та же строка  $b = (b_1, \dots, b_D)$  в матрице ошибок:

$$U_b = \{x_i \in \mathbb{X} \mid I(a_d, x_i) = b_d, d = 1, \dots, D\}.$$

Обозначим через  $B$  множество бинарных векторов  $b$ , которым соответствуют непустые блоки  $U_b$ . Очевидно,  $|B| \leq \min\{L, 2^D\}$ .

Обозначим  $m_b = |U_b|$ .

Каждой обучающей выборке  $X \in [\mathbb{X}]^\ell$  поставим в соответствие целочисленный вектор  $(s_b)_{b \in B}$  такой, что  $s_b = |X \cap U_b|$  — число объектов из блока  $U_b$ , попадающих в обучающую выборку. Множество всех таких векторов, соответствующих всевозможным обучающим выборкам, обозначим через  $S$ . Очевидно,  $S$  можно также определить и другим способом:

$$S = \left\{ s = (s_b)_{b \in B} \mid s_b = 0, \dots, m_b, \sum_{b \in B} s_b = \ell \right\}.$$

Запишем число ошибок алгоритма  $a_d$  на обучающей выборке  $X$  и контрольной выборке  $\bar{X}$  в виде суммы по блокам:

$$\begin{aligned} n(a_d, X) &= \sum_{b \in B} b_d |X \cap U_b| = \sum_{b \in B} b_d s_b; \\ n(a_d, \bar{X}) &= \sum_{b \in B} b_d |\bar{X} \cap U_b| = \sum_{b \in B} b_d (m_b - s_b). \end{aligned}$$

Таким образом, выбор алгоритма методом  $\mu$  зависит только от того, сколько объектов  $s_b$  из каждого блока попадёт в обучающую выборку, но не зависит от того, какие именно это будут объекты. Определим функцию  $d^* : S \rightarrow \{1, \dots, D\}$  как номер алгоритма, выбранного методом  $\mu$  по обучающей выборке. Если минимум  $n(a, X)$  достигается на нескольких алгоритмах, то пессимистичный метод  $\mu$  выбирает алгоритм с бóльшим  $n(a, \bar{X})$ . Если же и таких алгоритмов несколько, будем полагать, что выбирается алгоритм с бóльшим порядковым номером:

$$\begin{aligned} A(s) &= \text{Arg min}_{d=1, \dots, D} \sum_{b \in B} b_d s_b, \\ A'(s) &= \text{Arg max}_{d \in A(s)} \sum_{b \in B} b_d (m_b - s_b), \\ d^*(s) &= \max\{d : d \in A'(s)\}, \end{aligned} \tag{15.1}$$

где через  $\text{Arg min}_{d=1, \dots, D} f(d)$  обозначается множество значений  $d$ , при которых функция  $f(d)$  достигает минимального значения.

**Теорема 15.3.** Пусть  $\mu$  — пессимистичная минимизация эмпирического риска, векторы ошибок всех алгоритмов  $a \in A$  попарно различны. Тогда вероятность получить алгоритм  $a_d$  в результате обучения:

$$\mathbb{P}[\mu X = a_d] = \frac{1}{C_L^\ell} \sum_{s \in S} \left( \prod_{b \in B} C_{m_b}^{s_b} \right) [d^*(s) = d]; \tag{15.2}$$



вероятность переобучения:

$$Q_\varepsilon = \frac{1}{C_L^\ell} \sum_{s \in S} \left( \prod_{b \in B} C_{m_b}^{s_b} \right) \left[ \sum_{b \in B} b_{d^*(s)} (m_b \ell - s_b L) \geq \varepsilon k \ell \right]. \quad (15.3)$$

**Доказательство.**

Произвольному набору значений  $(s_b)_{b \in B}$  из  $S$  соответствует множество выборов  $X \in [\mathbb{X}]^\ell$  таких, что  $|X \cap U_b| = s_b$ . Число таких выборов равно произведению  $\prod_{b \in B} C_{m_b}^{s_b}$ , так как для каждого блока  $U_b$  существует  $C_{m_b}^{s_b}$  способов отобрать  $s_b$  объектов в подвыборку  $X \cap U_b$ .

Поскольку условия  $\mu X = a_d$  и  $d^*(s) = d$  равносильны, вероятность получить алгоритм  $a_d$  в результате обучения выражается в следующем виде:

$$\mathbb{P}[\mu X = a_d] = \frac{1}{C_L^\ell} \sum_{X \in [\mathbb{X}]^\ell} [d^*(s) = d] = \frac{1}{C_L^\ell} \sum_{s \in S} \left( \prod_{b \in B} C_{m_b}^{s_b} \right) [d^*(s) = d].$$

Теперь запишем вероятность переобучения:

$$Q_\varepsilon = \mathbb{P}[\delta_\mu(X) \geq \varepsilon] = \mathbb{P} \sum_{d=1}^D [\mu X = a_d] [\delta(a_d, X) \geq \varepsilon].$$

Распишем отклонение частот ошибок алгоритма  $a_d$  в виде суммы по блокам:

$$\delta(a_d, X) = \frac{1}{k} \sum_{b \in B} b_d (m_b - s_b) - \frac{1}{\ell} \sum_{b \in B} b_d s_b = \frac{1}{\ell k} \sum_{b \in B} b_d (m_b \ell - s_b L).$$

Тогда выражение для вероятности переобучения примет вид:

$$Q_\varepsilon = \frac{1}{C_L^\ell} \sum_{s \in S} \left( \prod_{b \in B} C_{m_b}^{s_b} \right) \sum_{d=1}^D [d^*(s) = d] \left[ \sum_{b \in B} b_d (m_b \ell - s_b L) \geq \varepsilon \ell k \right].$$

Отсюда немедленно вытекает требуемое выражение (15.3). ■

Объём вычислений по формулам (15.2) и (15.3) экспоненциален по длине выборки  $L$ . В худшем случае, когда все блоки  $U_b$  одноэлементные, множество  $S$  состоит из всевозможных булевых векторов длины  $L$ , содержащих ровно  $\ell$  единиц. Тогда число слагаемых в (15.2) и (15.3) равно  $C_L^\ell$ .

Вычисления по Теореме 15.3 эффективны только когда число блоков  $|B|$  невелико, в частности, при малом числе алгоритмов. Ещё один недостаток блочной оценки в том, что она в явном виде не учитывает свойства расслоения и связности.

## §15.4 Пара алгоритмов

Рассмотрим семейство из двух алгоритмов  $A = \{a_1, a_2\}$ . Чтобы воспользоваться блочной оценкой, положим  $B = (11, 10, 01, 00)$ .

Пусть в выборке  $\mathbb{X}$  имеется  $m_{11}$  объектов, на которых оба алгоритма допускают ошибку;  $m_{10}$  объектов, на которых только  $a_1$  допускает ошибку;  $m_{01}$  объектов,

на которых только  $a_2$  допускает ошибку;  $m_{00} = L - m_{11} - m_{10} - m_{01}$  объектов, на которых оба алгоритма дают верный ответ:

$$\begin{aligned}\vec{a}_1 &= (1, \dots, 1, 1, \dots, 1, 0, \dots, 0, 0, \dots, 0); \\ \vec{a}_2 &= (\underbrace{1, \dots, 1}_{m_{11}}, \underbrace{0, \dots, 0}_{m_{10}}, \underbrace{1, \dots, 1}_{m_{01}}, \underbrace{0, \dots, 0}_{m_{00}}).\end{aligned}$$

**Теорема 15.4.** Пусть  $\mu$  — пессимистичная минимизация эмпирического риска, и семейство состоит из двух алгоритмов  $A = \{a_1, a_2\}$ . Тогда для любого  $\varepsilon \in [0, 1)$

$$\begin{aligned}Q_\varepsilon &= \sum_{s_{11}=0}^{m_{11}} \sum_{s_{10}=0}^{m_{10}} \sum_{s_{01}=0}^{m_{01}} \frac{C_{m_{11}}^{s_{11}} C_{m_{10}}^{s_{10}} C_{m_{01}}^{s_{01}} C_{L-m_{11}-m_{10}-m_{01}}^{\ell-s_{11}-s_{10}-s_{01}}}{C_L^\ell} \times \\ &\times \left( [s_{10} < s_{01}] [s_{11} + s_{10} \leq \frac{\ell}{L}(m_{11} + m_{10} - \varepsilon k)] + \right. \\ &\left. + [s_{10} \geq s_{01}] [s_{11} + s_{01} \leq \frac{\ell}{L}(m_{11} + m_{01} - \varepsilon k)] \right).\end{aligned}\quad (15.4)$$

**Доказательство.** Воспользуемся Теоремой 15.3. Множество  $S$  состоит из целочисленных векторов  $s = (s_{11}, s_{10}, s_{01}, s_{00})$ , для которых  $s_{11} + s_{10} + s_{01} + s_{00} = \ell$ . Поэтому сумма  $\sum_{s \in S}$  преобразуется в тройную сумму  $\sum_{s_{11}=0}^{m_{11}} \sum_{s_{10}=0}^{m_{10}} \sum_{s_{01}=0}^{m_{01}}$ , при этом  $s_{00}$  выражается через остальные компоненты вектора  $s$ . Номер  $d^*(s)$  алгоритма, выбранного методом  $\mu$  по обучающей выборке, равен 1 при  $s_{10} < s_{01}$  и 2 при  $s_{10} \geq s_{01}$ . Теперь подставим значения  $m_b, s_b, d^*(s)$  в (15.3):

$$\begin{aligned}\left[ \sum_{b \in B} b_{d^*(s)} (m_b \ell - s_b L) \geq \varepsilon k \right] &= [d^*(s) = 1] [(m_{10} + m_{11})\ell - (s_{10} + s_{11})L \geq \varepsilon k] + \\ &+ [d^*(s) = 2] [(m_{01} + m_{11})\ell - (s_{01} + s_{11})L \geq \varepsilon k].\end{aligned}$$

Отсюда следует требуемая оценка (15.4). ■

**Вычислительный эксперимент.** На Рис. 15.3 и 15.4 показаны зависимости вероятности переобучения  $Q_\varepsilon$  и оценки ЭЛКР  $\hat{\Delta}$  от различности алгоритмов  $\rho(a_1, a_2) = m_{01} + m_{10}$  при  $\ell = k = 100$ ,  $\varepsilon = 0.05$ . Тонкими линиями показаны оценки ЭЛКР, вычисленные методом Монте-Карло по 1000 случайных разбиений.

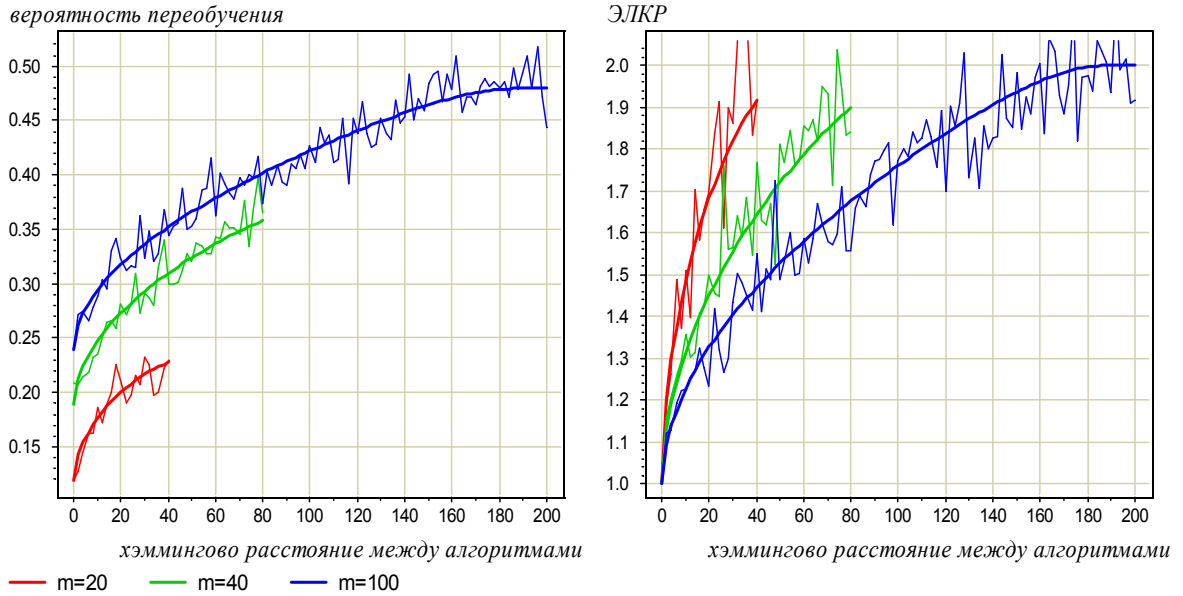
Графики позволяют сделать следующие выводы.

1. VC-оценка ЭЛКР  $\hat{\Delta} = 2$  достигается лишь в том случае, когда нет расслоения (алгоритмы допускают на  $\mathbb{X}$  одинаковое число ошибок,  $m_{01} = m_{10}$ ) и нет схождения (алгоритмы максимально различны,  $m_{00} = m_{11} = 0$ ).

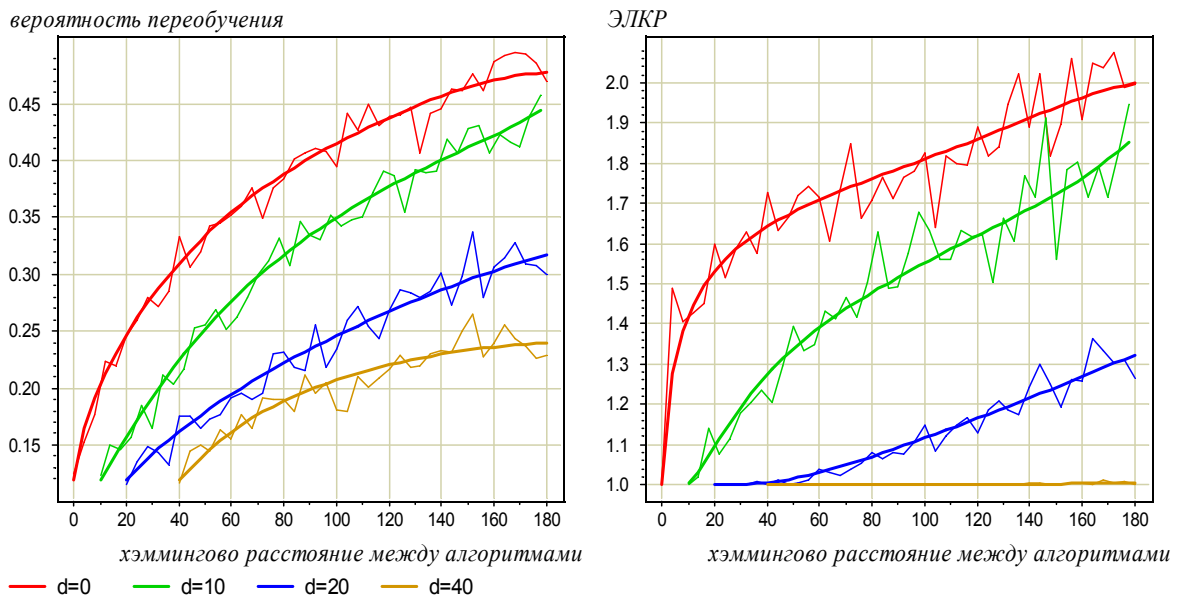
2. Чем сильнее расслоение ( $d = m_{01} - m_{10} > 0$ ), тем меньше вероятность переобучения, и тем ближе ЭЛКР к 1. В данном эксперименте при  $d = 40$  худший из двух алгоритмов уже практически никогда не выбирается.

3. Чем сильнее схождение ( $m_{01}, m_{10} \rightarrow 0$ ), тем меньше вероятность переобучения и тем ближе ЭЛКР к 1. С точки зрения переобучения два схожих алгоритма ведут себя практически как один алгоритм.

**Основной вывод.** Даже в простейшем случае, когда алгоритмов только два, уже возникает явление переобучения и проявляются эффекты расслоения и схождения, снижающие вероятность переобучения.



**Рис. 15.3.** Зависимость вероятности переобучения  $Q_\varepsilon$  и оценки ЭЛКР от различности алгоритмов, когда они допускают одинаковое число ошибок,  $m_{01} = m_{10}$ . Три графика соответствуют трём значениям числа ошибок на полной выборке  $m = m_{01} + m_{11} \in \{20, 40, 100\}$ . Тонкими линиями показаны эмпирические оценки методом Монте-Карло по 1000 случайных разбиений.



**Рис. 15.4.** Зависимость вероятности переобучения и оценки ЭЛКР от различности алгоритмов, когда  $m_{11} = 20$  и второй алгоритм допускает на  $d$  ошибок больше,  $m_{01} = m_{10} + d$ . Четыре графика соответствуют четырём разным значениям  $d \in \{0, 10, 20, 40\}$ . Тонкими линиями показаны эмпирические оценки методом Монте-Карло по 1000 случайных разбиений.

## Резюме

Точные оценки вероятности переобучения к данному моменту известны для ряда модельных семейств алгоритмов: монотонных и унимодальных цепей и сетей, интервалов булева куба и их нижних слоёв, и некоторых других. Эти оценки полезны для понимания того, как свойства расслоения и связности влияют на вероятность переобучения. Модельные семейства служат испытательным полигоном при создании комбинаторной техники оценивания переобучения. Однако для непосредственного практического применения они, как правило, не годятся.

**В следующей лекции** мы перейдём к рассмотрению функционала полного скользящего контроля, для которого в некоторых случаях удаётся получать очень точные оценки, находящие полезные практические применения.

## Упражнения

**Задача 15.1 (5\*).** Получить оценку вероятности переобучения для  $t$  нижних слоёв интервала булева куба через порождающие и запрещающие множества.

**Задача 15.2 (2).** Получить оценку вероятности переобучения для  $t$  нижних слоёв интервала булева куба в случае рандомизированного метода МЭР.

**Задача 15.3 (1).** Верно ли Следствие 15.2.1 для семейства  $A_t$ ,  $t < m$ ?

**Задача 15.4 (1).** Получить оценку вероятности переобучения для пары алгоритмов, пользуясь принципом порождающих и запрещающих множеств.

## 16 Метод ближайшего соседа

Получение точных оценок обобщающей способности невозможно без учёта априорной информации о выборке и методе обучения. В этой лекции будет рассмотрен метод ближайшего соседа совместно с априорной информацией о компактности выборки, а в следующей — метод монотонной классификации совместно с априорной информацией об отношении порядка на выборке.

Рассмотрим задачу классификации с конечным множеством классов  $\mathbb{Y}$ . Индикатор ошибки имеет вид  $I(a, x) = [y(x) \neq a(x)]$ , где  $y: \mathbb{X} \rightarrow \mathbb{Y}$  — функция правильной классификации (целевая зависимость),  $a: \mathbb{X} \rightarrow \mathbb{Y}$  — алгоритм классификации.

В задачах классификации, как правило, справедливо эмпирическое предположение, что классы образуют локализованные «компактные» подмножества объектов, поэтому схожие объекты гораздо чаще оказываются в одном классе, чем в разных. Это предположение называют *гипотезой компактности*. Простейшим методом обучения, построенным на его основе, является метод ближайших соседей.

### §16.1 Профиль компактности выборки

Пусть на множестве  $\mathbb{X} = \{x_1, \dots, x_L\}$  определена функция расстояния  $\rho(x, x')$ .

Относительно каждого объекта  $x_i \in \mathbb{X}$  расположим все остальные  $L - 1$  объектов в порядке возрастания расстояния до  $x_i$ , пронумеровав их двойными индексами:  $x_i = x_{i0}, x_{i1}, x_{i2}, \dots, x_{i,L-1}$ . Таким образом,

$$0 = \rho(x_i, x_{i0}) \leq \rho(x_i, x_{i1}) \leq \dots \leq \rho(x_i, x_{i,L-1}). \quad (16.1)$$

*Метод ближайшего соседа* (nearest neighbor, NN) — это метод обучения  $\mu$ , который запоминает обучающую выборку  $X \subset \mathbb{X}$  и строит алгоритм  $a = \mu X$ , относящий произвольный объект  $x \in \mathbb{X}$  к классу его ближайшего обучающего объекта:

$$a(x) = y(\arg \min_{x' \in X} \rho(x, x')).$$

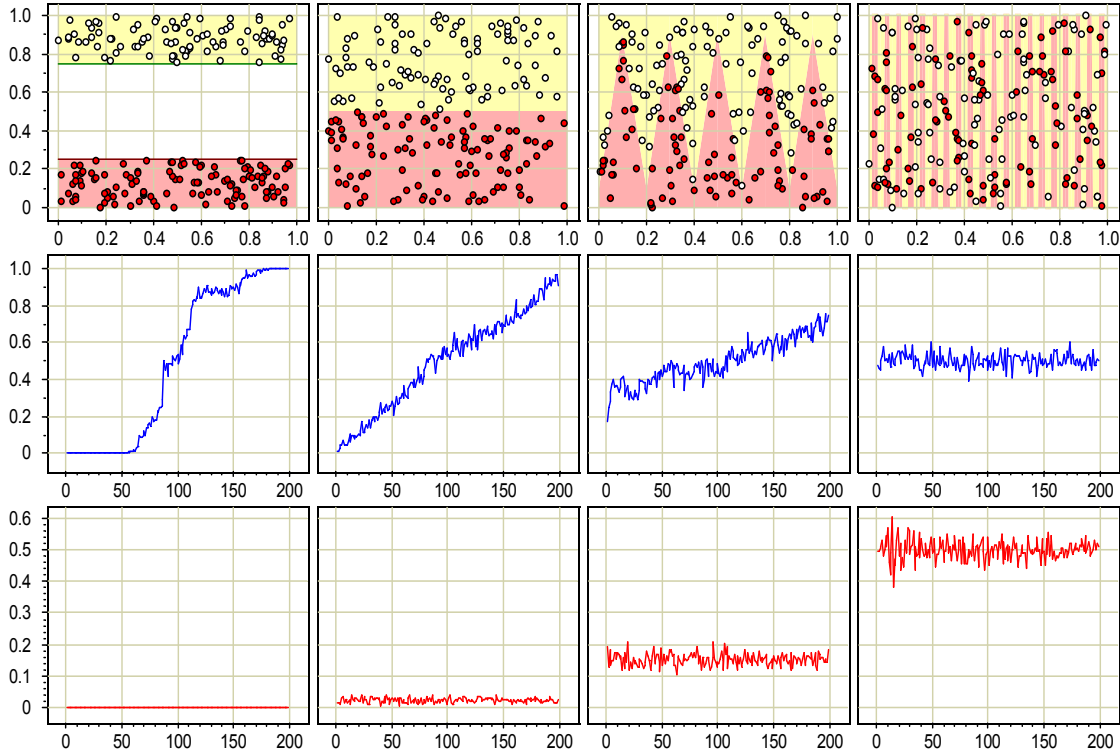
Обозначим через  $I_m(x_i)$  ошибку, возникающую, если правильный ответ  $y(x_i)$  на объекте  $x_i$  заменить ответом на его  $m$ -ом соседе:

$$I_m(x_i) = [y(x_i) \neq y(x_{im})]; \quad i = 1, \dots, L; \quad m = 1, \dots, L - 1.$$

**Определение 16.1.** *Профилем компактности выборки  $\mathbb{X}$  называется функция  $K(m)$ , выражающая долю объектов выборки  $\mathbb{X}$ , для которых правильный ответ не совпадает с правильным ответом на  $m$ -ом соседе:*

$$K(m) = \frac{1}{L} \sum_{i=1}^L I_m(x_i); \quad m = 1, \dots, L - 1.$$

Профиль компактности является формальным выражением гипотезы компактности. Чем проще задача, то есть чем чаще близкие объекты оказываются в одном



**Рис. 16.1.** Верхний ряд: 4 модельные задачи классификации, в порядке возрастания трудности,  $L = 200$ . Средний ряд: профили компактности этих задач. Чем ниже проходит начальный участок профиля, тем «проще» задача для алгоритма NN, и тем выше обобщающая способность. Нижний ряд: зависимость CCV от длины контрольной выборки  $k$  при фиксированной длине обучения  $\ell = 200$ .

классе, тем сильнее «прижимается к нулю» начальный участок профиля. И, наоборот, в задачах, трудных для метода ближайшего соседа, где ближайшие объекты практически не несут информации о классе, профиль вырождается в константу, близкую к 0.5, см. рис. 16.1. Профиль компактности показывает, насколько хорошо метод ближайшего соседа подходит для решения данной задачи при выбранной метрике  $\rho$ .

## §16.2 Точная оценка полного скользящего контроля

Интуитивно очевидная связь профиля компактности с качеством классификации подтверждается следующей теоремой. Идея доказательства взята из [88].

**Теорема 16.1.** Для метода ближайшего соседа  $\mu$  справедливо следующее выражение функционала полного скользящего контроля CCV:

$$CCV(\mu, \mathbb{X}) = \sum_{m=1}^k K(m) \frac{C_{L-1-m}^{\ell-1}}{C_{L-1}^{\ell}}. \quad (16.2)$$

**Доказательство.** Запишем в функционале скользящего контроля частоту ошибок через сумму индикаторов ошибки и переставим местами знаки суммирования:

$$\text{CCV} = \frac{1}{C_L^\ell} \sum_{X, \bar{X}} \frac{1}{k} \sum_{x \in \bar{X}} I(\mu X, x) = \frac{1}{k} \sum_{i=1}^L \frac{1}{C_L^\ell} \underbrace{\sum_{X, \bar{X}} [x_i \in \bar{X}] I(\mu X, x_i)}_{N_i}. \quad (16.3)$$

Внутренняя сумма, обозначенная через  $N_i$ , есть число разбиений выборки  $\mathbb{X}$ , при которых объект  $x_i$  оказывается в контрольной подвыборке и алгоритм  $\mu X$  допускает на нём ошибку. Данная ситуация реализуется для таких разбиений, при которых  $m$  первых объектов из последовательности  $x_{i0}, x_{i1}, \dots, x_{i, L-1}$  попадают в контрольную подвыборку,  $m$ -ый сосед  $x_{im}$  находится в обучающей подвыборке и принадлежит другому классу, то есть  $I_m(x_i) = 1$ . Число способов выбрать  $(\ell - 1)$  обучающих объектов из оставшихся  $(L - 1 - m)$  равно  $C_{L-1-m}^{\ell-1}$ . Число  $m$  может принимать значения только от 1 до  $k$ . Таким образом, искомое число разбиений равно

$$N_i = \sum_{m=1}^k I_m(x_i) C_{L-1-m}^{\ell-1}.$$

Подставляя  $N_i$  в (16.3), используя определение профиля компактности и тождество  $kC_L^\ell = LC_{L-1}^\ell$ , получаем требуемую формулу (16.2). ■

### Некоторые свойства точной оценки полного скользящего контроля.

1. Комбинаторный множитель  $\gamma_m = C_{L-1-m}^{\ell-1} / C_{L-1}^\ell$  убывает с ростом  $m$  быстрее геометрической прогрессии:

$$\gamma_{m+1} = q(m)\gamma_m, \quad q(m) = 1 - \frac{\ell - 1}{L - 1 - m} < \frac{k}{L - 1}.$$

Значение  $\text{CCV}$  мало, если профиль  $K(m)$  принимает малые значения при малых  $m$ , то есть близкие объекты часто лежат в одном классе. При больших  $m$  рост  $K(m)$  компенсируется стремлением комбинаторного множителя  $\gamma_m$  к нулю, поэтому классификации далёких друг от друга объектов практически не влияют на значение  $\text{CCV}$ .

2. При  $k = 1$  профиль компактности вырождается в точку и совпадает с самим функционалом  $\text{CCV}$ , который в этом случае называют скользящим контролем с исключением объектов по одному (leave-one-out, LOO).

В общем случае профиль состоит из  $k$  точек, причём относительный вклад  $K(m)$  быстро уменьшается с ростом  $m$ . Например, при  $k = 1, 2, 3$ :

$$\begin{aligned} k = 1 : & \quad \text{CCV} = K(1); \\ k = 2 : & \quad \text{CCV} = K(1) \frac{\ell}{\ell+1} + K(2) \frac{1}{\ell+1}; \\ k = 3 : & \quad \text{CCV} = K(1) \frac{\ell}{\ell+2} + K(2) \frac{2\ell}{(\ell+1)(\ell+2)} + K(3) \frac{2}{(\ell+1)(\ell+2)}. \end{aligned}$$

С ростом длины контроля  $k$  вклад начальных элементов профиля уменьшается.

3. Вычисление профиля компактности требует  $O(\ell^2)$  операций, упорядочивание объектов по близости —  $O(\ell^2 \log \ell)$  операций. После этого вычисление  $\text{CCV}$  производится за  $O(k)$  операций. Это гораздо быстрее, чем производить суммирование по всем  $C_L^\ell$  разбиениям, что становится практически нереально уже при  $k > 2$ .

### §16.3 Отбор эталонных объектов

В методе ближайшего соседа имеет смысл запоминать не всю обучающую выборку, а только типичные объекты, называемые *эталонами* (prototype). Отбор эталонов преследует одновременно несколько целей: сокращение объёма хранимых данных, повышение скорости классификации, повышение качества классификации благодаря устранению нетипичных объектов. Кроме того, эталоны могут предъявляться пользователю для объяснения результатов классификации.

*Нетипичными* или *шумовыми* называют объекты, находящиеся в окружении объектов чужого класса. Метод ближайшего соседа гарантированно допускает ошибку не только на таких объектах, но и на тех объектах чужого класса, для которых нетипичный объект окажется ближайшим соседом.

Известны эвристические методы отбора эталонов Stolp,  $\lambda$ -Stolp [25], FRiS-Stolp [5], основанные на оценках отношения локальных плотностей классов в каждом объекте. Эти методы неплохо зарекомендовали себя на практике, однако вопросы теоретического обоснования остаются для них открытыми: какой функционал они минимизируют, почему они обладают хорошей обобщающей способностью, не существует ли эвристик, которые могли бы работать ещё лучше.

Рассмотрим задачу выбора подмножества эталонов, минимизирующего функционал CCV для алгоритма ближайшего соседа [13, 73]. Данная задача является NP-полной [114], поэтому для её решения будут применены эвристические алгоритмы последовательного жадного удаления или добавления эталонов.

**Жадные алгоритмы отбора эталонов.** Обозначим через  $\Omega \subseteq \mathbb{X}$  искомое множество эталонов, через  $I_m(x_i, \Omega)$  — ошибку, возникающую, если правильный ответ  $y(x_i)$  на объекте  $x_i$  заменить ответом на его  $m$ -м соседе из множества  $\Omega$ . Определим *профиль компактности относительно множества эталонов*  $\Omega$ :

$$K(m, \Omega) = \frac{1}{L} \sum_{i=1}^L I_m(x_i, \Omega); \quad m = 1, \dots, L-1.$$

Для отбора эталонов удобнее определить величину *вклада*  $T(x_i, \Omega)$  объекта  $x_i$  и представлять CCV в виде суммы вкладов всех объектов.

**Теорема 16.2.** Для метода ближайшего соседа  $\mu_\Omega$ , использующего в качестве эталонов только объекты множества  $\Omega \subseteq \mathbb{X}$ , справедливы два представления CCV:

$$\text{CCV}(\mu_\Omega, \mathbb{X}) = \sum_{m=1}^k K(m, \Omega) \frac{C_{L-1-m}^{\ell-1}}{C_{L-1}^{\ell}} = \sum_{i=1}^L \underbrace{\sum_{m=1}^k I_m(x_i, \Omega) \frac{C_{L-1-m}^{\ell-1}}{LC_{L-1}^{\ell}}}_{T(x_i, \Omega)}. \quad (16.4)$$

Доказательство аналогично Теореме 16.1 и вынесено в упражнения.

Если некоторый объект  $x$  добавляется в  $\Omega$  или удаляется из  $\Omega$ , то для обновления CCV достаточно пересчитать вклады только тех объектов, для которых объект  $x$  является не далее, чем  $k$ -м соседом.



---

**Алгоритм 16.1.** Жадное удаление шумовых и неинформативных объектов.

---

**Вход:** выборка  $\mathbb{X}$ , параметр  $\delta > 0$ ;

**Выход:** множество эталонов  $\Omega \subseteq \mathbb{X}$ ;

---

- 1:  $\Omega := \mathbb{X}$ ;  $Q := \sum_{i=1}^L T(x_i, \Omega)$ ;  $Q_{\min} := Q$ ;
  - 2: **повторять**
  - 3:   count := 0; — счётчик числа удалений
  - 4:   **для всех**  $x \in \Omega$
  - 5:      $\Delta Q := \sum_{x_i \in V_k(x, \Omega)} \left( T(x_i, \Omega \setminus \{x\}) - T(x_i, \Omega) \right)$ ;
  - 6:     **если**  $Q + \Delta Q < Q_{\min} + \delta$  **то**
  - 7:        $Q := Q + \Delta Q$ ;  $\Omega := \Omega \setminus \{x\}$ ;
  - 8:     **если**  $Q < Q_{\min}$  **то**  $Q_{\min} := Q$ ;
  - 9:     count := count + 1;
  - 10: **пока** count > 0;
- 

**Алгоритм 16.2.** Жадное добавление эталонных объектов.

---

**Вход:** выборка  $\mathbb{X}$ , параметр  $\delta > 0$ ;

**Выход:** множество эталонов  $\Omega \subseteq \mathbb{X}$ ;

---

- 1:  $\Omega := \{\text{по одному случайному объекту от каждого класса}\}$ ;
  - $Q := \sum_{i=1}^L T(x_i, \Omega)$ ;  $Q_{\min} := Q$ ;
  - 2: **повторять**
  - 3:   count := 0; — счётчик числа добавлений
  - 4:   **для всех**  $x \in \mathbb{X} \setminus \Omega$
  - 5:      $\Delta Q := \sum_{x_i \in V_k(x, \Omega \cup \{x\})} \left( T(x_i, \Omega \cup \{x\}) - T(x_i, \Omega) \right)$ ;
  - 6:     **если**  $Q + \Delta Q < Q_{\min} - \delta$  **то**
  - 7:        $Q := Q + \Delta Q$ ;  $\Omega := \Omega \cup \{x\}$ ;
  - 8:     **если**  $Q < Q_{\min}$  **то**  $Q_{\min} := Q$ ;
  - 9:     count := count + 1;
  - 10: **пока** count > 0;
- 

Обозначим через  $r_i(x, \Omega)$  ранг объекта  $x \in \Omega$  во множестве  $\Omega$ , упорядоченном по возрастанию расстояний до  $x_i \in \mathbb{X}$ .

**Определение 16.2.** Обратной окрестностью  $m$ -го порядка объекта  $x \in \Omega$  называется множество объектов, для которых объект  $x$  является не далее, чем  $m$ -м соседом:

$$V_m(x, \Omega) = \{x_i \in \mathbb{X} \mid r_i(x, \Omega) \leq m\}, \quad m = 1, \dots, L - 1.$$

В [13] предложен жадный алгоритм удаления, который исключает из  $\Omega$  неэталонные объекты по одному, начиная с  $\Omega = \mathbb{X}$ . На каждом шаге находится тот объект  $x \in \Omega$ , удаление которого минимизирует  $CCV(\mu_{\Omega \setminus \{x\}}, \mathbb{X})$ . Позже был предложен Алгоритм 16.1, который работает намного быстрее [73]. Он удаляет все объекты, удаление которых увеличивает  $CCV$  не более, чем на малую величину  $\delta$ . Параметр  $\delta$

сначала полагают равным нулю, чтобы удалить все шумовые выбросы; при этом  $CCV$  убывает. Когда выбросы будут удалены, параметр  $\delta$  можно немного увеличить, чтобы удалить неинформативные «внутренние» объекты, плотно окружённые объектами своего класса. Когда и они окажутся исчерпаны, параметр  $\delta$  можно снова немного увеличить. Когда  $CCV$  начинает заметно возрастать, процесс удаления объектов прекращается, и все оставшиеся в  $\Omega$  объекты считаются эталонными.

Недостаток жадного удаления в том, что оно работает медленно. Алгоритмы типа *Stolp* обычно используют стратегию жадного добавления, более эффективную с точки зрения объёма вычислений. В алгоритме 16.2 стратегия добавлений используется для оптимизации  $CCV$ . Добавления делаются до тех пор, пока  $CCV$  уменьшается хотя бы на малую величину  $\delta$ . Недостаток этой стратегии в том, что множество отобранных эталонов может существенно зависеть от начального приближения — самых первых объектов, которые выбираются практически случайно.

Компромиссное решение заключается в том, чтобы взять в качестве начального приближения небольшое случайное подмножество объектов в каждом классе, и затем чередовать добавления и удаления эталонов. Мы не приводим здесь этот алгоритм, так он допускает большое число различных эвристик и вариантов реализации.

**О метрических деревьях.** Для эффективного пересчета  $CCV$  необходимо быстро находить объекты окрестности и обратной окрестности. Для этого необходима структура данных, которая позволяла бы эффективно строить, находить и обновлять прямые и обратные окрестности  $k$ -го порядка для любого объекта  $x_i \in X$ . Наиболее подходят для этой цели *метрические деревья*, предназначенные для индексации множества объектов, заданных попарными расстояниями [60, 61]. В среднем каждый запрос на поиск, добавление или удаление объекта обрабатывается метрическим деревом за время порядка  $O(\log L)$ . Использование метрических деревьев позволяет существенно ускорить процесс отбора эталонов.

## §16.4 Эксперименты и выводы

Тестирование алгоритмов отбора эталонов проводилось на двумерной модельной выборке длины  $L = 1000$ , состоящей из двух гауссовских классов с дисперсиями 1 и 2 и расстоянием между центрами 5. Дополнительно в обучающую выборку  $X^L$  вводилось по 5 шумовых объектов в каждом классе. На рис. 16.2 изображена разделяющая поверхность, построенная алгоритмом ближайшего соседа на полной выборке  $X^L$ .

На рис. 16.3 и рис. 16.4 показана разделяющая поверхность для классификатора одного ближайшего соседа по множеству эталонов  $\Omega$ . Рис. 16.3 соответствует стратегии удаления, рис. 16.4 — стратегии добавления эталонов. Большими точками обозначены эталонные объекты, маленькими — периферийные объекты. Для стратегии удаления шумовые объекты показаны треугольниками. Оба алгоритма сглаживают разделяющую поверхность, удаляют все шумовые объекты и оставляют приблизительно одинаковое число эталонов, которые выстраиваются на некотором удалении от границы классов.

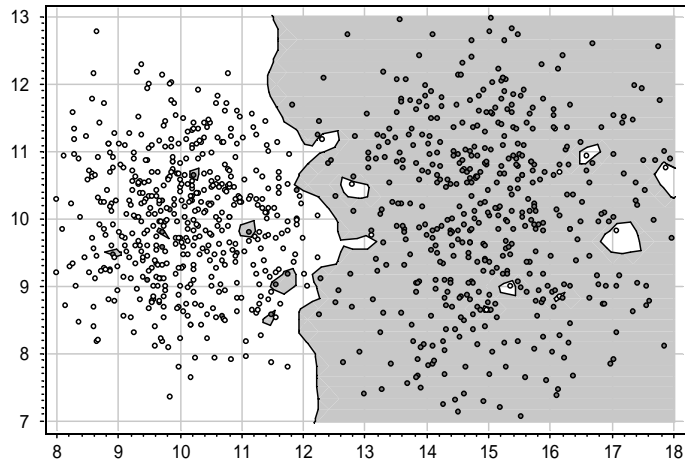


Рис. 16.2. Модельная задача классификации: 1000 объектов, метод 1NN.

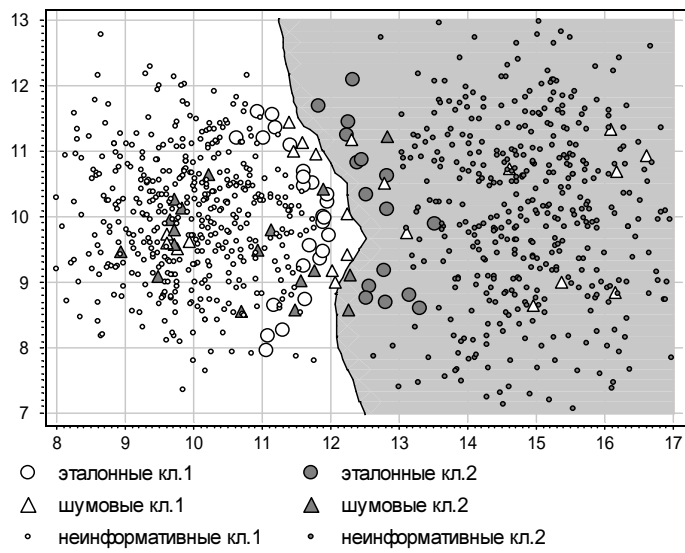


Рис. 16.3. Результат отбора эталонов Алгоритмом 16.1 (стратегия удаления).

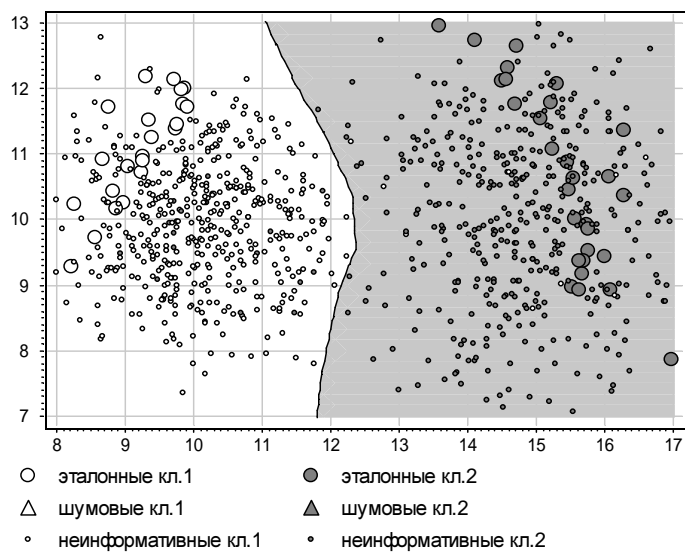
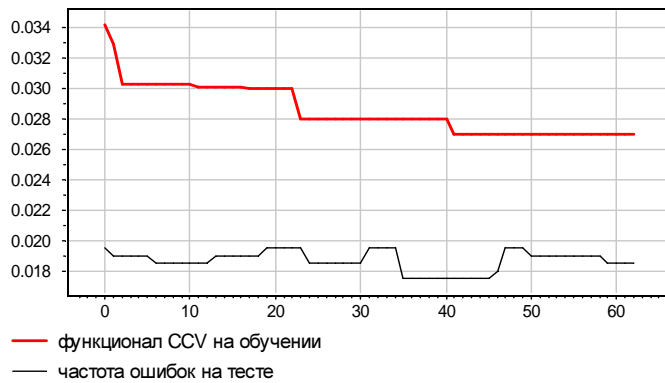


Рис. 16.4. Результат отбора эталонов Алгоритмом 16.2 (стратегия добавления).



**Рис. 16.5.** Зависимость функционала  $Q(\Omega)$  от количества удалённых объектов  $L - |\Omega|$  для Алгоритма 16.1.



**Рис. 16.6.** Зависимость функционала  $Q(\Omega)$  от количества добавленных объектов  $|\Omega|$  для Алгоритма 16.2.

Чтобы проверить обобщающую способность алгоритмов отбора эталонов, строилась тестовая выборка длины 2000 из того же распределения. Эти объекты не использовались при отборе эталонов из  $X^L$ . Частота ошибок на тестовой выборке вычислялась на каждой итерации и выводилась на графиках. В случае жадного удаления (рис. 16.5) она изменяется синхронно со значением CCV на обучающей выборке. Это означает, что данная стратегия практически не подвержена переобучению. В случае жадного добавления (рис. 16.6) столь чёткой синхронности не наблюдалось.

На рис. 16.5 левый участок [0–60] соответствует удалению шумов (около 40 объектов). Правый участок [980–1000] показывает, что число критически важных эталонов в данной выборке не велико (равно 6), но увеличение числа эталонов (до 16) позволяет уменьшить частоту ошибок на тестовой выборке (с 3,6% до 2,0%). На рис. 16.3 выделено 42 эталона, которые обеспечивают минимум  $Q(\Omega)$  и одновременно минимальную частоту ошибок на тестовой выборке 1,8%. Длинный средний участок [60–980], не показанный на рис. 16.5, соответствует этапу удаления периферийных объектов, на котором значение  $Q(\Omega)$  практически постоянно.

## Резюме

Функционал полного скользящего контроля (CCV) определяется как средняя по всем разбиениям частота ошибок на контрольной выборке. Для метода ближайшего соседа точная оценка CCV выражается через профиль компактности выборки, который показывает, насколько адекватна принятая модель сходства объектов в данной задаче. Отбор эталонных объектов по критерию минимума CCV позволяет сократить объём хранимых данных, повысить скорость и качество классификации. При этом объекты разделяются на три типа — шумовые, неинформативные и эталонные, что является полезной дополнительной возможностью во многих приложениях.

**В следующей лекции** будут получены оценки CCV для множества монотонных классификаторов. Интересно отметить, что, хотя парадигмы метрической и монотонной классификации существенно различны, для этих двух случаев получаются удивительно похожие по своей структуре оценки.

## Упражнения

**Задача 16.1 (1).** Доказать теорему 16.2.

**Задача 16.2 (1).** Доказать, что  $\sum_{m=1}^{L-1} K(m) = \frac{1}{L} \sum_{y \in \mathbb{Y}} L_y(L - L_y)$ , где  $L_y$  — число объектов класса  $y$  в выборке  $\mathbb{X}$ .

**Задача 16.3 (1).** Доказать, что  $\sum_{m=1}^{L-1} \gamma_m = 1$ .

**Задача 16.4 (2).** Обобщить теорему 16.1 на случай, когда в вариационном ряду расстояний (16.1) есть одинаковые значения. Сделать пессимистичное предположение, что если для классифицируемого объекта  $x_i$  существует хотя бы один обучающий объект  $x$  на минимальном расстоянии  $\min_{x \in X} \rho(x_i, x)$ , лежащий в другом классе, то алгоритм обязательно допустит ошибку на объекте  $x_i$ .

**Задача 16.5 (5).** Обобщить теорему 16.2 для метода классификации по  $K$  ближайшим соседям.

**Задача 16.6 (5\*).** Получить оценки вероятности переобучения  $Q_\varepsilon$  и вероятности большой частоты ошибок на контроле  $R_\varepsilon$  для метода ближайшего соседа.

## 17 Монотонные классификаторы

Рассмотрим задачу классификации, в которой множество объектов  $\mathbb{X}$  частично упорядочено,  $\mathbb{Y} = \{0, 1\}$ ; множество алгоритмов  $A$  есть множество всех монотонных функций  $a: \mathbb{X} \rightarrow \mathbb{Y}$  (то есть из  $x \leq x'$  следует  $a(x) \leq a(x')$  для всех  $x, x' \in \mathbb{X}$ ); индикатор ошибки имеет вид  $I(a, x) = [y(x) \neq a(x)]$ , где  $y$  — неизвестная целевая зависимость. Предполагается, что функция  $y$  монотонна или «почти монотонна».

На практике ограничения монотонности могут возникать в различных ситуациях. Упомянем три наиболее распространённых.

**1. Априорные ограничения монотонности.** Прикладные эксперты могут давать подсказки (hints), ограничивающие множество допустимых алгоритмов [46]. Примером является утверждение «чем больше значение признака  $f(x)$ , тем выше уверенность, что объект  $x$  принадлежит классу 1». Если таких признаков несколько, утверждение принимает вид «если  $f(x) \leq f(x')$  для всех признаков  $f$  из заданного множества  $G$ , то  $a(x) \leq a(x')$ ». Но это и означает, что функция  $a$  монотонна. Примеры таких утверждений можно найти в задачах медицинского прогнозирования (чем старше пациент и чем выше скорость кровотока в вене, тем выше риск рестеноза шунта), в задачах кредитного скоринга (чем выше доходы заёмщика и чем дольше он проживает на одном месте, тем ниже риск дефолта) и в других предметных областях [101]. Как правило, они носят приблизительный характер, и зависимость  $y$  от признаков на самом деле не в точности монотонна, а «почти монотонна».

**2. Пороговые решающие правила.** Пусть задана функция  $f: \mathbb{X} \rightarrow \mathbb{R}$ . Функция  $a(x) = [f(x) \geq \theta]$  называется *пороговым решающим правилом* с порогом  $\theta$ . Если отношение порядка на  $\mathbb{X}$  задано как  $(x \leq x') \leftrightarrow (f(x) \leq f(x'))$ , то множество всех монотонных классификаторов  $A$  совпадает со множеством всех пороговых решающих правил. На практике пороговые правила широко используются в качестве «строительных блоков» алгоритмов классификации. Это элементарные предикаты в конъюнктивных закономерностях (стр. 100), условия ветвления в решающих деревьях [19, 90, 37] и решающих списках [48, 102], базовые классификаторы в композициях типа бустинга и бэггинга [96, 59], правила принятия окончательного решения во многих алгоритмах классификации на два класса.

**3. Композиции алгоритмов классификации.** Пусть имеется  $T$  алгоритмов классификации вида  $a_t(x) = [b_t(x) \geq 0]$ , где  $b_t: \mathbb{X} \rightarrow \mathbb{R}$ ,  $t = 1, \dots, T$ . Их *композицией* называется алгоритм  $a(x) = F(b_1(x), \dots, b_T(x))$ , где функция  $F: \mathbb{R}^T \rightarrow \mathbb{Y}$  называется *корректирующей операцией*, алгоритмы  $a_t$  называются *базовыми*. Значения  $b_t(x)$  и  $b(x)$  интерпретируются как степень уверенности алгоритмов  $a_t(x)$  и  $a(x)$ , соответственно, в том, что объект  $x$  принадлежит классу 1. Поэтому от функции  $F$  требуется, чтобы она монотонно не убывала по всем  $T$  своим аргументам [41, 11]. Поскольку базовые алгоритмы могут допускать ошибки, зависимость  $y(x_i)$  от  $T$ -мерных векторов  $(b_1(x_i), \dots, b_T(x_i))$  может оказаться не монотонной, но «почти монотонной».

Наиболее распространённым типом композиций является *взвешенное голосование* — линейная комбинация базовых алгоритмов [23, 98, 59, 80]:

$$F(b_1(x), \dots, b_T(x)) = \sum_{t=1}^T \alpha_t b_t(x).$$

При ограничении  $\alpha_t \geq 0$  линейная корректирующая операция является монотонной. Однако не всякая монотонная корректирующая операция является линейной.

Далее нас будет интересовать вопрос, в каких случаях монотонные классификаторы обладают наилучшей обобщающей способностью в смысле функционала полного скользящего контроля ССV. Вообще говоря, обобщающая способность должна быть тем выше, чем лучше выборка согласуется с используемой моделью алгоритмов. В данном случае модель — это множество всех монотонных функций. Поэтому в качестве ответа на поставленный вопрос ожидается получить некоторую характеристику монотонности выборки и явный вид зависимости ССV от этой характеристики.

## §17.1 Профиль монотонности выборки

Допустим, что метод обучения  $\mu$  выбирает алгоритмы из множества  $A$  всех монотонных функций  $a: \mathbb{X} \rightarrow \mathbb{Y}$ .

**Определение 17.1.** *Верхним и нижним клином объекта  $x_i \in \mathbb{X}$  называются, соответственно, множества*

$$W_0(x_i) = \{x \in \mathbb{X}: x_i < x \text{ и } y(x) = 0\};$$

$$W_1(x_i) = \{x \in \mathbb{X}: x < x_i \text{ и } y(x) = 1\}.$$

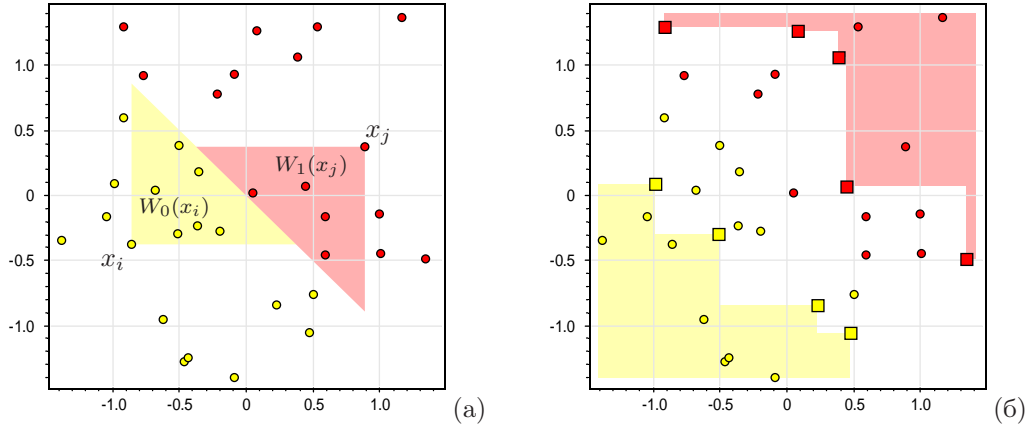
Примеры клиньев показаны на рис. 17.1 (а).

Множество  $W_i = W_{y(x_i)}(x_i)$  будем называть просто *клином* объекта  $x_i$ . Это все объекты того же класса, что и  $x_i$ , лежащие между  $x_i$  и границей классов. Мощность клина  $w_i = |W_i|$  характеризует глубину погружения объекта  $x_i$  в свой класс. Чем меньше  $w_i$ , тем ближе объект к границе класса. Объекты, не имеющие своего клина ( $w_i = 0$ ) будем называть *граничными*.

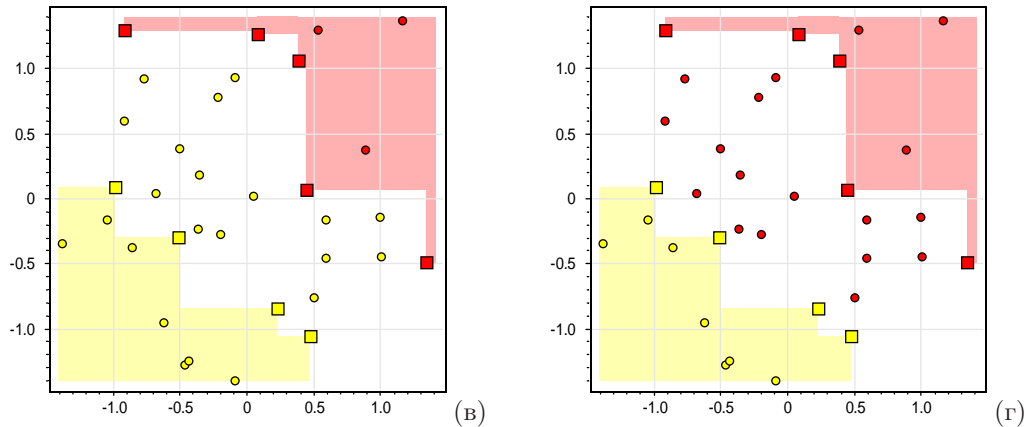
Если монотонный алгоритм допускает ошибку на объекте  $x_i$ , то он допускает ошибку и на всех объектах из клина  $W_i$ . Это основное свойство клина, которое сыграет решающую роль при получении оценок скользящего контроля.

**Определение 17.2.** *Профилем монотонности выборки  $\mathbb{X}$  называется функция  $M(m)$ , равная доле объектов с клином мощности  $m$ :*

$$M(m) = \frac{1}{L} \sum_{i=1}^L [w_i = m]; \quad m = 0, \dots, L - 1.$$



**Рис. 17.1.** Двумерная задача классификации с естественным отношением порядка на множестве  $\mathbb{X} \subset \mathbb{R}^2$ . Рис. (а): верхний клин объекта  $x_i$ , нижний клин объекта  $x_j$ . Рис. (б): одна из допустимых монотонных классификаций генеральной выборки  $\mathbb{X}$ . Объекты обучающей выборки выделены символами  $\square, \blacksquare$ . Фоном показано объединение верхних областей обучающих объектов класса 1 и объединение нижних областей обучающих объектов класса 0.



**Рис. 17.2.** Та же двумерная задача классификации, что на рис. 17.1. Два крайних случая монотонной классификации: все объекты в зазоре между нижними и верхними областями обучающих объектов относятся к классу 0 (а), к классу 1 (б).

Профиль монотонности нормирован:  $\sum_{m=0}^{L-1} M(m) = 1$ .

Профиль монотонности существенно зависит от выборки, причём возможны два крайних случая:

- 1) если в каждом из двух классов объекты попарно несравнимы, то все клинья пусты и профиль сосредоточен в нуле:  $M(m) = [m = 0]$ ;
- 2) если на  $\mathbb{X}$  задан линейный порядок, то  $M(m) = \frac{1}{L}([m < L_0] + [m < L_1])$ , где  $L_y$  — число объектов класса  $y$  в генеральной выборке; если  $L_0 = L_1$ , то профиль монотонности распределён равномерно:  $M(m) = \frac{2}{L} [m < \frac{L}{2}]$ .



## §17.2 Верхняя оценка полного скользящего контроля

**Определение 17.3.** *Степенью немонотонности выборки  $\mathbb{X}$  называется наименьшая частота ошибок, допускаемых на ней монотонными алгоритмами:*

$$\theta(\mathbb{X}) = \min_{a \in A} \nu(a, \mathbb{X}).$$

Выборка  $\mathbb{X}$  называется монотонной, если из  $x_i \leq x_j$  следует  $y(x_i) \leq y(x_j)$  для всех  $x_i, x_j \in \mathbb{X}$ . Выборка монотонна тогда и только тогда, когда  $\theta(\mathbb{X}) = 0$ .

Если метод  $\mu$  минимизирует эмпирический риск, то есть строит алгоритмы с минимальным числом ошибок на обучающей выборке в классе всех монотонных функций  $A$ , то метод  $\mu$  будет корректным на любой монотонной выборке [11].

**Теорема 17.1.** *Если метод  $\mu$  минимизирует эмпирический риск в классе всех монотонных функций и степень немонотонности выборки  $\mathbb{X}$  равна  $\theta$ , то*

$$\text{CCV}(\mu, \mathbb{X}) \leq \sum_{m=0}^{\theta L + k - 1} M(m) H_{L-1}^{\ell, m}(\theta L). \quad (17.1)$$

**Доказательство.** Запишем в функционале скользящего контроля частоту ошибок через сумму индикаторов ошибки и поменяем местами знаки суммирования:

$$\text{CCV} = \frac{1}{C_L^\ell} \sum_{X, \bar{X}} \frac{1}{k} \sum_{x \in \bar{X}} I(\mu X, x) = \frac{1}{k} \sum_{i=1}^L \underbrace{\frac{1}{C_L^\ell} \sum_{X, \bar{X}} [x_i \in \bar{X}] I(\mu X, x_i)}_{N_i}. \quad (17.2)$$

Внутренняя сумма, обозначенная через  $N_i$ , выражает число разбиений выборки  $\mathbb{X}$ , при которых объект  $x_i$  оказывается в контрольной подвыборке, и построенный по обучающей подвыборке алгоритм допускает на нём ошибку.

Оценим  $N_i$ , воспользовавшись тем, что если алгоритм  $a$  монотонный и допускает ошибку на объекте  $x_i$ , то он допускает ошибку и на всех объектах из клина  $W_i$ .

В зависимости от соотношения мощности клина  $w_i$  и степени немонотонности выборки возможны два случая.

Если  $w_i \geq \theta L + k$ , то ни при каком разбиении монотонный алгоритм не будет ошибаться на  $x_i$ , поскольку  $\theta L + k$  есть максимальное число ошибок, которое может допустить монотонная функция на всей выборке  $\mathbb{X}$ . Это вытекает из допущения, что метод  $\mu$  строит алгоритм с минимальным числом ошибок на обучающей выборке в классе всех монотонных функций. Минимальное число ошибок на любой подвыборке  $X$  не превосходит минимального числа ошибок на всей выборке  $\mathbb{X}$ . Следовательно число ошибок на обучении не превышает  $\theta L$ . Таким образом, в этом случае  $N_i = 0$ .

Рассмотрим второй случай, когда  $w_i < \theta L + k$ . Пусть  $s$  — число объектов из  $W_i$ , находящихся в обучающей подвыборке,

$$\max\{0, w_i - k + 1\} \leq s \leq \min\{\theta L, \ell, w_i\}.$$

Имеется  $C_{w_i}^s$  способов выбрать  $s$  обучающих объектов из клина  $W_i$ . Для каждого из этих способов имеется  $C_{L-1-w_i}^{\ell-s}$  вариантов выбрать  $\ell - s$  обучающих объектов из множества  $\mathbb{X} \setminus (W_i \cup \{x_i\})$ . В итоге получаем оценку числа разбиений:

$$N_i \leq \sum_{s=\max\{0, w_i-k+1\}}^{\min\{\theta L, \ell, w_i\}} C_{w_i}^s C_{L-1-w_i}^{\ell-s}. \quad (17.3)$$

Подставим оценку (17.3) в (17.2), воспользуемся тождеством  $kC_L^\ell = LC_{L-1}^\ell$  и учтём, что  $N_i = 0$  при  $w_i \geq \theta L + k$ :

$$\text{CCV} \leq \frac{1}{k} \sum_{\substack{i=1 \\ w_i < \theta L + k}}^L \frac{k}{L} \sum_{s=\max\{0, w_i-k+1\}}^{\min\{\theta L, \ell, w_i\}} \frac{C_{w_i}^s C_{L-1-w_i}^{\ell-s}}{C_{L-1}^\ell}.$$

Применяя определение профиля монотонности, получаем оценку

$$\text{CCV} \leq \sum_{m=0}^{\theta L + k - 1} M(m) \sum_{s=\max\{0, m-k+1\}}^{\min\{\theta L, \ell, m\}} \frac{C_m^s C_{L-1-m}^{\ell-s}}{C_{L-1}^\ell}.$$

Заметив, что сумма по  $s$  совпадает с функцией гипергеометрического распределения  $H_{L-1}^{\ell, m}(\theta L)$ , получаем оценку (17.1). Что и требовалось доказать. ■

**Следствие 17.1.1.** Оценка (17.1) монотонно не убывает по  $\theta$ , достигая наименьшего значения при  $\theta = 0$ , когда выборка монотонна и метод  $\mu$  является корректным на генеральной выборке  $\mathbb{X}$ , то есть когда  $\nu(\mu X, X) = 0$  для всех  $X \in [\mathbb{X}]^\ell$ :

$$\text{CCV}(\mu, \mathbb{X}) \leq \sum_{m=0}^{k-1} M(m) \frac{C_{L-1-m}^\ell}{C_{L-1}^\ell}. \quad (17.4)$$

Наибольшая оценка  $\text{CCV} \leq 1$  достигается, когда  $w_i = 0$  для всех  $i = 1, \dots, L$ . Это означает, что каждый класс представляет собой множество попарно несравнимых объектов, и вся выборка распадается на две антицепи. В этом случае требование монотонности не накладывает никаких ограничений на семейство алгоритмов  $A$ , и оно индуцирует все  $2^L$  способов поделить такую выборку на два класса. Вполне естественно, что переобучение в данном случае максимально.

Наименьшая оценка  $\text{CCV} \leq 2/\ell$  достигается, когда выборка монотонна и линейно упорядочена. В этом случае число клиньев мощности  $w$  не превышает 2 для всех  $w = 1, \dots, k$ . Семейство монотонных алгоритмов  $A$  индуцирует унимодальную цепь. Переобучение в данном случае минимально.

Гипергеометрический множитель в (17.1) убывает с ростом  $m$  быстрее геометрической прогрессии. Чтобы обеспечить малое значение функционала  $\text{CCV}$ , достаточно потребовать, чтобы функция  $M(m)$  принимала малые значения при малых  $m$ . При больших  $m$  её рост компенсируется комбинаторным множителем. Таким образом, качество монотонного классификатора тем выше, чем меньше объектов имеют клинья небольшой мощности. Для этого отношение порядка на множестве объектов  $X$

должно быть близко к линейному вблизи границы классов. Другими словами, множество бинарных векторов ошибок должно быть близко к унимодальной цепи вблизи границы классов.

Таким образом, форма профиля монотонности может рассматриваться как формальное выражение априорной информации о плотности отношения порядка [42, 43] вблизи границы классов.

Заметим, что ёмкость класса монотонных классификаторов бесконечна, поскольку на выборке длины  $L$ , состоящей из попарно несравнимых элементов, реализуется ровно  $2^L$  дихотомий. Таким образом, классическая теория Вапника-Червоненкиса не даёт содержательных оценок качества для данного случая. Известно также, что зависящая от выборки эффективная ёмкость класса монотонных функций не превосходит длины максимальной антицепи в выборке  $X$  [99]. Оценка (17.1) существенно точнее сложностных оценок и никогда не превышает 1.

Интересно отметить большое структурное сходство оценок (16.2) и (17.4), полученных для таких различных, на первый взгляд, априорных ограничений, как компактность и монотонность. Чуть позже мы попробуем разобраться в его причинах.

### §17.3 Точные оценки полного скользящего контроля

Попробуем разобраться, почему оценка (17.1) не является точной. При оценивании числа разбиений  $N_i$  мы не знали, ошибается ли алгоритм  $a = \mu X$  на объекте  $x_i$ , и сделали избыточно сильное пессимистичное предположение, что он всегда на нём ошибается. Если же учесть, как именно алгоритм  $a$  определяется по выборке  $X$ , то возможно получить точные оценки.

Введём следующие определения.

*Нижняя область* объекта  $x_i$  — это множество всех объектов  $x$  таких, что  $x \leq x_i$ .

*Верхняя область* объекта  $x_i$  — это множество всех объектов  $x$  таких, что  $x_i \leq x$ .

Допустим, что выборка  $X$  монотонна. Тогда любой монотонный алгоритм  $a$  правильно классифицирует все объекты, попадающие в нижние области обучающих объектов класса 0 и в верхние области обучающих объектов класса 1, см. рис. 17.1 (б). Про остальные объекты будем говорить, что они *лежат в зазоре* между нижними и верхними областями обучающих объектов. Классификация этих объектов зависит от конструкции алгоритма  $a$ . На рис. 17.1 (б) показана одна из возможных реализаций алгоритма  $a$ . На рис. 17.2 показаны два крайних случая — когда все объекты, лежащие в зазоре, относятся либо к классу 0, либо к классу 1. Начнём с этих крайних случаев, как наиболее простых.

**Монотонные классификаторы по границам зазора.** Назовём *профилем монотонности* класса  $y \in Y$  функцию  $M_y(m)$ , равную доле объектов класса  $y$  с клином мощности  $m$ :

$$M_y(m) = \frac{1}{L} \sum_{i=1}^L [y(x_i) = y] [w_i = m]; \quad m = 0, \dots, L - 1.$$

Очевидно, сумма профилей по классам совпадает с общим профилем:

$$M(m) = M_0(m) + M_1(m), \quad m = 0, \dots, L-1. \quad (17.5)$$

**Теорема 17.2.** *Если выборка  $\mathbb{X}$  монотонна и алгоритм  $a = \mu X$  относит все объекты, лежащие в зазоре, к классу  $y$ , то справедлива точная оценка*

$$\text{CCV}(\mu, \mathbb{X}) = \sum_{m=0}^{k-1} M_{1-y}(m) \frac{C_{L-1-m}^\ell}{C_{L-1}^\ell}. \quad (17.6)$$

**Доказательство.** Допустим, что алгоритм  $a = \mu X$  относит все объекты из зазора к классу  $y = 0$ . Тогда он допускает ошибки только на тех объектах  $x_i$  из зазора, которые принадлежат классу 1. Для этих объектов число разбиений  $N_i$ , при которых  $x_i \in \bar{X}$  и  $I(\mu X, x_i) = 1$ , вычисляется по формуле (17.3). Для остальных объектов из зазора  $N_i = 0$ . Таким образом,

$$N_i = C_{L-1-w_i}^\ell [y(x_i) = 1].$$

Подставив  $N_i$  в (17.2) и воспользовавшись тождеством  $kC_L^\ell = LC_{L-1}^\ell$ , получим (17.6) для случая  $y = 0$ . Аналогичным образом проверяется случай  $y = 1$ . ■

Если предположить, что профиль монотонности делится согласно (17.5) примерно поровну между классами, то из доказанной теоремы следует, что оценка (17.4) для произвольного метода обучения завышена приблизительно вдвое. На самом деле она завышена сильнее, так как относить все объекты зазора к одному классу — очевидно, не самая лучшая стратегия с точки зрения обобщающей способности. Интуиция подсказывает, что оптимальное разделение должно проходить посередине зазора. Чтобы проверить эту гипотезу, необходимо сравнить методы обучения, которые проводят разделяющую поверхность по границам зазора, по середине зазора или (для большей общности) по заданному отношению расстояний до нижней и верхней границ. Точная оценка  $\text{CCV}$  для случая разделения по середине зазора была впервые получена Галиной Махиной в [36].

**Монотонные классификаторы по расстояниям до границ зазора.** Определим для произвольной пары объектов  $x, u \in \mathbb{X}$  две неотрицательные функции: расстояние  $r_0(x, u)$  от  $x$  до нижней области объекта  $u$  и расстояние  $r_1(x, u)$  от  $x$  до верхней области объекта  $u$ . Потребуем, чтобы они удовлетворяли следующим свойствам:

- 1)  $r_0(x, u) = 0$  тогда и только тогда, когда  $x \leq u$ ;
- 2)  $r_1(x, u) = 0$  тогда и только тогда, когда  $x \geq u$ ;
- 3) функция  $r_0(x, u)$  не убывает по  $x$  и не возрастает по  $u$ ;
- 4) функция  $r_1(x, u)$  не возрастает по  $x$  и не убывает по  $u$ .

Определим функцию расстояния от классифицируемого объекта  $x$  до объекта обучающей выборки  $x' \in X$  как расстояние до его нижней области, если  $y(x') = 0$ , или до верхней области, если  $y(x') = 1$ :

$$\rho(x, x') = r_{y(x')}(x, x').$$

Функция  $\rho$  не симметрична и зависит от класса объекта  $x'$ . Впрочем, это обстоятельство не мешает воспользоваться методом ближайшего соседа  $\mu$ , который по обучающей выборке  $X$  строит алгоритм классификации  $a = \mu X$  вида

$$a(x) = y(\arg \min_{x' \in X} \rho(x, x')). \quad (17.7)$$

Данный алгоритм определяет разделяющую поверхность, проходящую посередине зазора. Можно обобщить функцию  $\rho$  так, чтобы зазор разделялся не по середине, а в отношении  $\lambda : (1 - \lambda)$  при заданном значении параметра  $\lambda \in (0, 1)$ :

$$\rho(x, x') = \begin{cases} \lambda r_0(x, x'), & y(x') = 0; \\ (1 - \lambda)r_1(x, x'), & y(x') = 1. \end{cases}$$

При  $\lambda \rightarrow 0$  все объекты из зазора относятся к классу 0, и разделяющая поверхность проходит по верхней границе зазора.

При  $\lambda \rightarrow 1$  все объекты из зазора относятся к классу 1, и разделяющая поверхность проходит по нижней границе зазора.

При  $\lambda = \frac{1}{2}$  разделяющая поверхность проходит посередине зазора.

При крайних значениях  $\lambda \in \{0, 1\}$  функция  $\rho$  может принимать минимальное нулевое значение одновременно на обучающих объектах разных классов, что будет приводить к неоднозначности классификации. Для устранения неоднозначности можно брать крайние значения  $\{\lambda_0, 1 - \lambda_0\}$  при некотором  $\lambda_0$ , очень близком к нулю.

Следующая теорема утверждает, что формула (17.7) действительно определяет монотонную функцию  $a(x)$ .

**Теорема 17.3.** Пусть выборка  $X$  монотонна, функции расстояния  $r_0$  и  $r_1$  удовлетворяют свойствам 1)–4). Тогда функция  $a(x)$  монотонно не убывает по  $x$  и  $a(x) = y(x)$  для всех  $x \in X$ .

Точная оценка  $CCV$  для алгоритма (17.7) даётся той же теоремой 16.1, что и для обычного метода ближайшего соседа. Различий, собственно, нет никаких: точно так же каждый объект  $x_i$  порождает упорядочение всех объектов генеральной выборки по неубыванию расстояний до него, и точно так же определяется профиль компактности.

**Взаимосвязь между профилями компактности и монотонности.** Покажем, что верхняя оценка  $CCV$  по профилю монотонности (17.4) может быть получена в результате ослабления точной оценки  $CCV$  по профилю компактности (16.1).

Запишем точную оценку:

$$CCV = \sum_{m=1}^k K(m) \frac{C_{L-1-m}^{\ell-1}}{C_{L-1}^{\ell}} = \sum_{m=1}^k \frac{1}{L} \sum_{i=1}^L I_m(x_i) \frac{C_{L-1-m}^{\ell-1}}{C_{L-1}^{\ell}},$$

где  $I_m(x_i)$  — индикатор ошибки на объекте  $x_i$ , если классифицировать его по  $m$ -му соседу. Все объекты  $x$  из клина  $W_i$  являются первыми  $w_i = |W_i|$  ближайшими соседями объекта  $x_i$ , поскольку для них и только для них расстояния  $\rho(x_i, x)$  равны

нулю. Объекты из клина принадлежат тому же классу, что и сам  $x_i$ , следовательно,  $I_1(x_i) = \dots = I_m(x_i) = 0$ . Однако, начиная с  $(m + 1)$ -го соседа, могут возникать ошибки. Таким образом, справедлива верхняя оценка для  $I_m$ :

$$I_m(x_i) \leq [w_i < m] = \sum_{s=0}^{m-1} [w_i = s], \quad m = 1, \dots, L - 1.$$

Подставим эту оценку в формулу для CCV и сделаем ряд преобразований:

$$\begin{aligned} \text{CCV} &\leq \sum_{m=1}^k \frac{1}{L} \sum_{i=1}^L \sum_{s=0}^{m-1} [w_i = s] \frac{C_{L-1-m}^{\ell-1}}{C_{L-1}^{\ell}} = \\ &= \sum_{m=1}^k \sum_{s=0}^{m-1} \frac{1}{L} \sum_{i=1}^L [w_i = s] \frac{C_{L-1-m}^{\ell-1}}{C_{L-1}^{\ell}} = \\ &= \sum_{s=0}^{k-1} \sum_{m=s+1}^k M(s) \frac{C_{L-1-m}^{\ell-1}}{C_{L-1}^{\ell}} = \\ &= \sum_{s=0}^{k-1} M(s) \frac{1}{C_{L-1}^{\ell}} (C_{L-1}^{\ell-1} + \dots + C_{L-2-s}^{\ell-1}) = \sum_{m=0}^{k-1} M(m) \frac{C_{L-1-m}^{\ell}}{C_{L-1}^{\ell}}. \end{aligned}$$

Последнее выражение совпадает с верхней оценкой (17.4).

**Расстояния до верхних и нижних областей.** Рассмотрим один естественный способ определить функции  $r_0$ ,  $r_1$ , исходно предложенный в [11] для построения монотонных корректирующих операций. Допустим, что объекты задаются  $n$ -мерными числовыми векторами,  $X \subset \mathbb{R}^n$ ,  $x = (x^1, \dots, x^n)$ ,  $u = (u^1, \dots, u^n)$  и положим

$$\begin{aligned} r_1(x, u) &= \varphi((u^1 - x^1)_+, \dots, (u^p - x^n)_+), \\ r_0(x, u) &= r_1(u, x), \end{aligned}$$

где индекс «+» обозначает операцию срезки:  $z_+ = z \cdot [z \geq 0]$ ; функция  $\varphi(z^1, \dots, z^n)$  не убывает на всей области определения  $[0, +\infty)^n$  и принимает нулевое значение  $\varphi(z^1, \dots, z^n) = 0$  тогда и только тогда, когда  $z^1 = \dots = z^n = 0$ .

В качестве функции  $\varphi$  можно взять максимум  $\max(z^1, \dots, z^n)$ , сумму  $z^1 + \dots + z^n$ , сумму степеней  $(z^1)^p + \dots + (z^n)^p$ , число ненулевых аргументов  $\#\{i: z^i > 0\}$ . В то же время, произведение  $z^1 \dots z^n$  и минимум  $\min(z^1, \dots, z^n)$  не подходят, так как они принимают нулевые значения не только в точке  $(0, \dots, 0)$ .

**ToDo<sup>4</sup> Численный эксперимент.** Целью эксперимента является обоснование того, что проведение разделяющей поверхности посередине зазора является оптимальным с точки зрения минимизации CCV.

## Резюме

Для монотонных алгоритмов классификации верхняя оценка CCV выражается через профиль монотонности выборки, характеризующий плотность отношения порядка вблизи границы классов или степень согласованности ограничения монотонности с наблюдаемой выборкой. Точные оценки также могут быть получены, но для

этого необходимо учитывать конструкцию монотонного классификатора. Для монотонного метода ближайшего соседа удаётся получить точные оценки и обосновать проведение разделяющей поверхности посередине зазора между классами.

## Упражнения

**Задача 17.1 (2).** Обобщить теорему 17.2 на случай, когда выборка не обязательно монотонна.

**Задача 17.2 (1).** Доказать теорему 17.3.

**Задача 17.3 (2).** Для случая, когда выборка не обязательно монотонна, обобщить монотонный классификатор ближайшего соседа (17.7) и теорему 17.3.

**Задача 17.4 (3\*).** Для случая, когда выборка не обязательно монотонна, вывести верхнюю оценку  $CCV$  по профилю монотонности через точную оценку  $CCV$  по профилю компактности.

## 18 Приложение.

### О комбинаторной теории переобучения

Эти заметки адресованы, главным образом, студентам и аспирантам, выбравшим комбинаторную теорию переобучения в качестве темы своей научно-исследовательской работы. Наверное, они будут интересны и тем, кто начинает интересоваться методологией научного исследования. Здесь рассказывается о том, как появлялись постановки задач и результаты, обсуждавшиеся в данном курсе лекций.

Во всяком теоретическом исследовании предшествующий положительный и отрицательный опыт логично подсказывает следующие шаги. Если не научиться замечать эти подсказки, то можно потерять годы, имея результат буквально под носом. Однако без конкретных примеров трудно понять, что значит «опыт подсказывает» и как «замечать подсказки». Попробуем проследить эволюцию некоторых идей комбинаторной теории переобучения в хронологическом порядке, сосредоточившись на поворотных моментах и не умалчивая об ошибках.

Хотя опыт такого рода всегда индивидуален, его всё же можно резюмировать в виде списка старых, как мир, рекомендаций: (1) постоянно читать современные статьи; (2) исследовать простые или вырожденные частные случаи, даже если на первый взгляд это кажется бесполезным; (3) чередовать теоретические исследования с экспериментами; (4) чередовать упорные попытки решить поставленную трудную проблему с попытками изменить саму постановку; (5) чаще собираться на семинары, устраивать мозговые штурмы, думать над проблемой то порознь, то вместе; (6) тщательно продумывать терминологию, упрощать обозначения, искать наиболее рациональные пути доказательств.

Практически для каждой рекомендации можно вспомнить случаи, когда именно она помогла вывести исследование из затянувшегося тупика. Или, наоборот, когда оно заходило в тупик из-за того, что данная рекомендация игнорировалась, будь то по глупости или лени, или просто потому, что столь полезное руководство к действию не было вовремя сформулировано и вывешено над столом.

#### §18.1 Постановка задачи

**1991 год.** Исходная задача, поставленная моим научным руководителем Константином Владимировичем Рудаковым, когда я был студентом третьего курса МФТИ, состояла в следующем. Необходимо было изучить работы В. Л. Матросова [33, 34, 35], в которых доказывалось, что корректные алгебраические замыкания над алгоритмами вычисления оценок имеют конечную ёмкость, следовательно, обучаемы согласно теории В. Н. Вапника и А. Я. Червоненкиса (VC-теории). Затем надо было изучить работы самого К. В. Рудакова [40], в которых показывалось, что можно строить алгебраические замыкания гораздо меньшей степени, о чём В. Л. Матросов в начале 80-х знать никак не мог. Поэтому результаты В. Л. Матросова можно улучшить, что мне и предлагалось сделать.



Разобравшись, я не понял главного — зачем нужны оценки, которые потом нигде не используются. Дело оказалось в том, что VC-оценки, на которые опирался В. Л. Матросов, настолько завышены, что их можно интерпретировать лишь качественно — ёмкость конечна, значит обучаемость есть. Малая вероятность ошибки теорией гарантируется, но, согласно VC-оценкам, достигается на выборках длины порядка миллиардов. В то же время известно, что практические задачи вполне успешно решаются при выборках из сотен а иногда даже десятков объектов.

Частная задача улучшения заведомо завышенной оценки потеряла в тот момент всякий смысл. Проблема завышенности скрывалась где-то внутри VC-теории, и её решение представлялось гораздо более важным.

## §18.2 Долгое топтание на месте

**1991–1995.** По выражению Пойя, в любой теории имеется некая «скрытая движущая пружина», главный момент доказательства, «в который всё и происходит». Внимательное изучение работ Вапника и Червоненкиса показало, что такой момент действительно есть и находится в единственной теореме (стр. 46), где делаются наиболее грубые оценки сверху. Остальная теория никоим образом не связана с проблемой завышенности. Интересно, что этот момент доказательства имеет чисто комбинаторную природу. Вероятностная мера используется лишь для того, чтобы перевести комбинаторный по сути результат на вероятностный язык. Если очистить доказательство от второстепенных деталей, то станет видно, что на самом деле оценивается комбинаторная величина, определяемая следующим образом. Пусть задана конечная генеральная выборка объектов и рассматриваются все её разбиения на две подвыборки, обучающую и контрольную. Задано семейство алгоритмов, в котором мы надеемся найти алгоритм, допускающий мало ошибок. Назовём разбиение плохим, если в семействе существует алгоритм, у которого разность частоты ошибок на контроле и на обучении превышает заданный порог. Доля плохих разбиений — это и есть тот основной функционал качества, вокруг которого строится VC-теория, и для которого получена верхняя VC-оценка.

Константин Владимирович не раз говорил, что на практике решение задач распознавания происходит не так, и никому не придёт в голову перебирать алгоритмы и искать среди них плохие — с малым числом ошибок на обучении и большим на контроле. На практике применяют некоторый метод обучения (настройки параметров) алгоритма, чтобы по заданной обучающей выборке получить один конкретный алгоритм. Поэтому более адекватным функционалом является доля разбиений выборки, при которых именно этот алгоритм переобучен, то есть разность частоты его ошибок на контроле и на обучении превышает заданный порог. Такой функционал впоследствии был назван *вероятностью переобучения* (стр. 12). Очевидно, что он оценивается сверху функционалом равномерной сходимости (5.1), значит, для него также справедлива VC-оценка. Однако вероятность переобучения зависит от выборки и от метода обучения. Потенциально это даёт возможность как-то учитывать или

даже оптимизировать метод обучения. Но как воспользоваться этой возможностью, до сих пор до конца не понятно.

На том этапе никакого нового результата по существу не было. Ключевая теорема VC-теории была переписана на другом языке, более лаконичном и естественном (стр. 46). Это можно было интерпретировать и как ослабление аксиоматики с сохранением результатов: по классике предполагалось, что объекты выбираются случайно и независимо из некоторого существующего, но неизвестного распределения в пространстве объектов; теперь же достаточно предполагать, что выборка произвольна и неслучайна, зато все её разбиения случайны и равновероятны. Фактически, из набора классических вероятностных допущений оставалось только предположение о независимости объектов выборки. Эти соображения впервые были доложены на конференции ММРО-7 в 1995 году.

**1995-2005.** Доказательство основной теоремы со временем удалось ужать буквально до трёх строк. Однако попытки направить его в другое русло оставались бесплодными. Возникавшие идеи через некоторое время поисков в интернете находились в статьях 5–10-летней давности, все до единой, и, как правило, они уже были достаточно глубоко проработаны! Очевидно, я был в этой проблеме не один, и безнадежно отставал. Многие уже было сделано, оставалось только узнавать, как это называется по-английски. Учёт конкретной обучающей выборки — *data dependent bounds*. Метод обучения — *learning algorithm*. Оценки по двум конечным выборкам — *transductive bounds*. Улучшение оценок в процессе обучения за счёт постепенного сужения семейства — *self-bounding algorithms* [66]. Учёт только части алгоритмов с невысоким уровнем ошибок — *algorithmic luckiness* [70], *shell bounds* [82, 81]. Точная оценка полного скользящего контроля для метода ближайшего соседа — *complete cross validation* [88]. И так далее. . .

Научное сообщество уже осознало причины завышенности VC-оценок и активно искало новые пути. Правда, все чужие результаты также оставляли чувство глубокой неудовлетворённости. В какой-то момент произошёл отказ от комбинаторных методов в пользу более сложного математического аппарата функционального анализа и теории концентрации вероятностной меры [83, 57]. Оценки улучшались, но всё же оставались завышенными на порядки. Доказательства усложнялись, и становилось всё труднее проследить, в каких местах возникают «утечки точности». Задача получения точных оценок даже не ставилась, по всей видимости, как полностью безнадежная. Обычно речь шла лишь о зажатых, то есть не сильно завышенных, оценках (*tight bounds*) [81, 94, 47].

### §18.3 Эксперименты с переобучением

**2005 год.** Если проблема не решается теоретически, то с ней надо попробовать разобраться экспериментально. VC-оценка (5.2) представляет собой произведение двух сомножителей. Первый — *коэффициент разнообразия* (*shattering coefficient*) семейства, равный максимальному числу различных бинарных векторов ошибок, порождаемых алгоритмами семейства на генеральной выборке (*i*-я координата вектора

ошибок равна 1, если алгоритм ошибается на  $i$ -м объекте, иначе 0). Второй сомножитель, равный вероятности переобучения фиксированного алгоритма, довольно точно описывается функцией гипергеометрического распределения. Завышен, главным образом, первый сомножитель. Возникает вопрос, каким должно было бы быть значение коэффициента разнообразия, чтобы оценка была не завышенной, а точной? Это значение было названо *эффективным локальным коэффициентом разнообразия* (ЭЛКР).

Чтобы измерить ЭЛКР, был проведён следующий эксперимент (стр. 53). Левая часть неравенства (вероятность переобучения) оценивалась методом Монте-Карло по случайному подмножеству разбиений и делилась на гипергеометрический сомножитель из правой части неравенства. Для экспериментов использовалась библиотека логических алгоритмов классификации, разработанная Денисом Кочедыковым и Андреем Ивахненко. Результаты экспериментов на четырёх реальных задачах классификации были шокирующими: ни на одной из задач ЭЛКР не превысил 10, тогда как обычный коэффициент разнообразия на тех же задачах имел порядки  $10^8$ – $10^{11}$ . Это означало, что VC-оценки завышены на много порядков. Если бы мы захотели определить понятие эффективной локальной ёмкости, то она была бы вырождена — никогда не превышала бы единицы. Отсюда возникает сомнение в состоятельности и полезности понятия ёмкости для оценок переобучения.

Эти результаты не были приняты на ведущие международные конференции COLT и ALT, с объяснением, что трансдуктивная постановка задачи и факт завышенности VC-оценок хорошо известны; только экспериментальный результат был новым и интересным, но одного его не достаточно для публикации. Критика VC-теории была неконструктивна, ничего нового и лучшего взамен не предлагалось. Спасибо рецензентам, указавшим на ряд статей, в том числе на работы Кольчинского [77], Лэнгфорда [81], Хербрича [70], которые действительно были упущены.

**2006 год.** В ходе работы над ошибками описанный эксперимент был переделан тщательнее, и методика оценивания ЭЛКР была усложнена. Во-первых, небольшая завышенность гипергеометрического множителя приводила к занижению ЭЛКР. После исправления этой ошибки значения ЭЛКР на некоторых задачах подросли, но никогда не превышали длины выборки (на стр. 53 показаны уже исправленные результаты). Таким образом, вывод о том, что эффективная локальная ёмкость вырождена и не превышает единицы, сохранялся. Во-вторых, были измерены факторы завышенности, то есть потеря точности на каждом знаке  $\leq$  в цепочке неравенств. Стало понятно, что основных причин завышенности две — это пренебрежение свойствами расслоения и связности, которыми применяемые на практике семейства алгоритмов, как правило, обладают.

Свойство *расслоения* связано с тем, что в конкретной задаче лишь немногие алгоритмы из семейства имеют низкую частоту ошибок, но именно они чаще всего выбираются в результате обучения. Эффективно используется лишь малая часть семейства, в каждой задаче своя.

Свойство *связности* появляется из-за того, что применяемые на практике алгоритмы, как правило, непрерывны по параметрам. Поэтому для каждого алгоритма в семействе найдётся некоторое количество других алгоритмов, у которых векторы ошибок отличаются от данного только на одном объекте. С точки зрения переобучения подмножество столь схожих алгоритмов ведёт себя «почти как один алгоритм».

Эксперименты показали, что свойства расслоения и связности примерно одинаково важны для снижения переобучения. Невозможно получить точные оценки переобучения, учитывая только один из этих двух эффектов. Эксперименты проводились на той же библиотеке логических алгоритмов классификации, что и предыдущие, совместно с Андреем Ивахненко.

Эти результаты были доложены на открытом германско-русском семинаре OGRW в Эттлингене в августе 2007 года. И на этот раз ничего конструктивного не предлагалось. Понятно, что надо учитывать расслоение и связность. Понятно, что VC-оценки не учитывают оба эффекта. Понятно, что в теории статистического обучения пока нет подходов, позволяющих учитывать оба эффекта одновременно и достаточно аккуратно. Не понятно только, что делать дальше.

**Апрель 2007 года.** Если проблема не решается в общем случае, то надо начинать с самых простых частных случаев. Задача о точной комбинаторной оценке вероятности переобучения пары алгоритмов (стр. 137) возникла спонтанно и тут же была предложена студентам второго курса ВМК МГУ в качестве тестовой при поступлении на кафедру ММП (математических методов прогнозирования). Неожиданно несколько человек её быстро решили, но все решения оказались разными. После исправления ошибок стало понятно, что задача не сложная, и можно выводить оценки вероятности переобучения для частных случаев, постепенно двигаясь в сторону обобщений. Однако ещё около года найти другие легко решаемые частные случаи никому не удавалось. Простое увеличение числа алгоритмов казалось поначалу наиболее естественным путём. Случай трёх алгоритмов был несколько сложнее технически, но ничего не добавлял концептуально. Дальнейшее увеличение числа алгоритмов приводило к экспоненциальному росту объёма вычислений и числа ненаблюдаемых параметров в оценке. Для произвольного числа алгоритмов была выписана *блочная оценка* вероятности переобучения (стр. 135). Это точная оценка, довольно изящная по форме, но громоздкая и бесполезная по сути. Стало ясно, что этот путь тупиковый, и ясно, почему — блочная оценка не учитывала в явном виде ключевые свойства расслоения и связности.

Простейшим множеством алгоритмов, которое уже обладает свойствами расслоения и связности, является, по всей видимости, монотонная цепь алгоритмов (стр. 71). Это последовательность алгоритмов, в которой каждый следующий алгоритм делает ошибки на тех же объектах, что и предыдущий, плюс ещё на одном объекте. Выписать оценку вероятности переобучения монотонной цепи сходу не удалось. Тогда возникла идея провести численный эксперимент, посчитав вероятность переобучения методом Монте-Карло. Для наглядности строился график зависимости вероятности переобучения  $Q$  от числа алгоритмов, если брать только  $D$  пер-

вых, самых лучших, алгоритмов цепи (стр. 72). Оказалось, что кривая  $Q(D)$  быстро выходит на насыщение. При длине выборки  $L = 200$  уже при  $D = 6$  вероятность переобучения  $Q$ , достигнув значения 0.1, перестаёт расти, что очень хорошо. Какому эффекту мы обязаны больше — расслоению или связности? Чтобы ответить на этот вопрос, были сгенерированы ещё три модельных семейства: цепь без расслоения и два несвязных семейства, с расслоением и без расслоения. В каждом семействе алгоритмы были упорядочены по возрастанию числа ошибок на генеральной выборке, а наилучший алгоритм во всех семействах был одним и тем же. Эксперимент показал, что связность уменьшает темп роста кривой  $Q(D)$ , а расслоение понижает уровень горизонтальной асимптоты. Приемлемые значения вероятности переобучения даёт только их сочетание. Отсутствие хотя бы одного из двух эффектов приводит к переобучению с вероятностью  $Q = 0.5$  уже при нескольких десятках алгоритмов. Этот простой экспериментальный результат в корне меняет взгляд на существующие теории переобучения. Он означает, что применяемые на практике семейства, содержащие огромное число алгоритмов, с необходимостью должны быть расслоенными и связными, иначе сильное переобучение неизбежно.

Монте-карловский эксперимент был проделан и с парой алгоритмов (стр. 139). Оказалось, что при увеличении различности алгоритмов вероятность переобучения растёт, достигая VC-оценки в единственном наихудшем случае — когда два алгоритма максимально плохи (имеют частоту ошибок 0.5) и максимально различны (их векторы ошибок не совпадают ни на одном объекте). Стоит одному алгоритму оказаться хотя бы немного лучше другого, и вероятность переобучения резко падает — это эффект расслоения. Со сходством оказалось похуже: алгоритмы должны быть очень похожи и различаться буквально на нескольких объектах; далее с ростом их различности вероятность переобучения быстро растёт. Основной вывод заключался в том, что эффекты переобучения, расслоения и связности проявляются даже в самом простом случае, когда обучение сводится к выбору лучшего из двух алгоритмов.

Результаты экспериментов с цепями и парами алгоритмов были доложены на конференции ИОИ в Алуште в июне 2008 года.

## §18.4 «Игрушечные» частные случаи

**Конец 2008 года.** Первые точные оценки вероятности переобучения были найдены для монотонной цепи, унимодальной цепи и единичной окрестности (стр. 74–81). Эти задачи обсуждались на спецкурсе «Теория надёжности обучения по прецедентам» и давались студентам в качестве заданий. В тот момент у меня самого не было решения, но студенты об этом не знали. Были лишь наводящие соображения, которые убеждали, что задачи, наверное, несложные. Решения довольно быстро нашли Жанна Кожахметова и Алексей Куренной, и одновременно студент МФТИ Александр Фрей. обстоятельный разбор и сопоставление этих решений привёл к их обобщению — принципу порождающих и запрещающих множеств, ПЗМ (стр. 61). Он основан на гипотезе, что для каждого алгоритма  $a$  можно указать два подмножества объектов — порождающее  $X_a$  и запрещающее  $X'_a$ , такие, что  $a$  является результа-

том обучения тогда и только тогда, когда  $X_a$  целиком лежит в обучающей выборке, а  $X'_a$  — целиком в контрольной. Этой гипотезы было достаточно, чтобы выписать точную оценку вероятности переобучения и попутно для каждого алгоритма точную оценку вероятности того, что данный алгоритм будет выбран в результате обучения. Для унимодальной цепи алгоритмов эта гипотеза не выполнялась, но была верна более общая гипотеза — что с каждым алгоритмом связано несколько пар порождающих и запрещающих множеств (стр. 64, 77).

Неожиданным открытием стал тот факт, что обобщённая гипотеза верна всегда, то есть множество пар ПЗМ может быть указано для любого метода обучения, любого семейства алгоритмов и любой выборки (стр. 64). Доказательство этого факта элементарно, конструктивно, но неэффективно — точная оценка вероятности переобучения требует экспоненциального по длине выборки объёма вычислений. Спасает лишь то, что выписать систему ПЗМ можно разными способами, среди которых, как правило, удаётся найти эффективные.

С помощью принципа ПЗМ была решена задача о рекуррентном вычислении вероятности переобучения при добавлении алгоритмов по одному (стр. 113). Рекуррентная процедура содержала один простой и красивый момент, который обнаружился лишь после нескольких недель размышлений и горы черновиков, изрисованных бинарными матрицами. Вычисление вероятности переобучения сводилось в итоге к суммированию знакопеременного ряда, причём, обрывая ряд, можно было обменивать точность оценки на время вычислений.

**Начало 2009 года.** Накопившиеся к этому моменты результаты были снова поданы на конференции COLT и ALT, но не приняты, поскольку (1) их теоретическая значимость мала, (2) доказательства перегружены, (3) гипотеза, лежащая в основе метода порождающих и запрещающих множеств, является неестественным и чрезмерно сильным ограничением, (4) все случаи, для которых найдены точные оценки — искусственные и требуют знания скрытых параметров генеральной выборки.

Пожалуй, только с пунктом (3) никак нельзя согласиться, в остальном рецензенты совершенно правы.

Затем были получены оценки ещё для нескольких модельных семейств алгоритмов — слоёв и интервалов булева куба (стр. 132). Это красивые частные случаи, но, к сожалению, за ними так и не обнаружилось никакого общего способа вывода оценок.

## §18.5 Поиск обобщений и альтернативных подходов

Общая оценка, в явном виде учитывающая свойства расслоения и связности, была получена с помощью «умозрительного выполнения» рекуррентной процедуры (стр. 118). Доказательство было громоздким, и в нём никак не удавалось избавиться от сильного дополнительного требования, чтобы в семействе содержался корректный (не допускающий ошибок на генеральной выборке) алгоритм. Это ограничение казалось избыточным, но как его снять, было не ясно.

Примерно в это время нетривиальные результаты стали появляться у всех, кто занимался комбинаторной теорией переобучения.

Денис Кочедыков получил верхние оценки для функционала Вапника-Черво-ненкиса (вероятность большого равномерного отклонения частот ошибок в двух вы-борках), справедливые для произвольного семейства алгоритмов и учитывающие эффект связности. Эффект расслоения для данного функционала не может быть учтён в принципе. Для доказательства использовалась техника цепных разложений или принцип включения-исключения. Проверялась также возможность применения неравенств типа Бонферрони-Галамбоса — это перспективный путь, но пока он не дал каких-либо преимуществ. Используя методы комбинаторной геометрии, Денис доказал, что для линейных классификаторов значения верхней связности (верхняя связность алгоритма — это число алгоритмов семейства, допускающих ошибку на тех же объектах, что и данный, плюс ещё на одном объекте) концентрируются во-круг значения размерности пространства. Позже он получил оценку и для дисперсии связности.

Павел Ботов обобщил монотонную цепь алгоритмов на случай произвольной размерности и решил задачу о переобучении многомерной монотонной сети алго-ритмов (стр. 93). Это обобщение далось на удивление легко. Затем он получил оценки для пучка монотонных цепей и для многомерной унимодальной сети, но в этих случа-ях доказательства были сложнее. Удалось даже показать, что вероятность переобу-чения унимодальной сети приблизительно равна вероятности переобучения монотон-ной сети удвоенной размерности. Доказательства опирались на модифицированный принцип ПЗМ, в котором основная гипотеза формулировалась не для отдельного алгоритма, а целиком для каждого слоя алгоритмов. Из всех изученных к этому моменту модельных семейств многомерная монотонная сеть наиболее приближена к реальности. Она обладает ключевыми свойствами расслоения, связности и размер-ности, которыми обладают и реальные семейства. Возникла идея: нельзя ли реальные семейства аппроксимировать монотонными сетями? Был поставлен эксперимент на реальных задачах классификации и реальных семействах — подопытными оказались нейронная сеть, наивный байесовский классификатор и метод ближайших соседей. Во всех случаях вероятность переобучения действительно неплохо приближалась ве-роятностью переобучения монотонной сети, если подобрать два её параметра: часто-ту ошибок лучшего алгоритма и размерность сети (стр. 98). Но как их подбирать и какую пользу можно из этого извлечь? Пока это открытые вопросы.

Александр Фрей предложил оригинальный подход к получению точных оценок вероятности переобучения для модельных семейств алгоритмов, обладающих свой-ством симметрии. Оценки выводились для метода рандомизированной минимизации эмпирического риска на основе разработанного им теоретико-группового подхода. Развив этот подход, Илья Толстихин получил оценки для хэммингова шара в бу-левым кубе, его нижних слоёв и отдельно взятого центрального слоя шара. А Илья Решетняк заметил, что оценка для центрального слоя шара может быть легко по-лучена и для обычной минимизации эмпирического риска, без привлечения теории групп.

Максим Иванов получил точную комбинаторную оценку функционала полного скользящего контроля для метода ближайших соседей, выразив её через вклады отдельных объектов. Это позволило ему построить достаточно эффективный алгоритм формирования множества эталонов (prototype selection) путём последовательного добавления или удаления объектов (стр. 144).

В сентябре 2009 года все эти результаты были доложены авторами на конференции ММРО-14.

Модельные семейства типа хэмминговых шаров и интервалов булева куба интересны тем, что они в некотором смысле «максимально плотные». Расчёты показали, что они могут содержать десятки и сотни тысяч алгоритмов при относительно небольшой вероятности переобучения. А можно ли аппроксимировать плотные семейства их разреженными подмножествами, и сколько алгоритмов для этого придётся взять? Вспоминая эксперимент по оцениванию ЭЛКР, можно предположить, что нескольких десятков должно хватить. Илья Толстихин показал, что достаточно взять несколько десятков алгоритмов на внешней окружности слоя хэммингова шара, чтобы приблизить вероятность переобучения всего слоя. Таким образом, аппроксимация плотных семейств их разреженными подмножествами действительно возможна. А для небольшого разреженного подмножества вероятность переобучения можно найти разными способами. Например, можно «натянуть» на него другое модельное семейство, для которого известна точная оценка. Эта идея пока ещё никем не проработана. Кроме того, пока не понятно, как должно расслаиваться аппроксимирующее множество.

**22 апреля 2010 года** многолетние усилия завершились защитой докторской диссертации «Комбинаторная теория надёжности обучения по прецедентам». По честному, это были лишь намётки теории. Хорошая теория должна (1) правильно объяснять наблюдаемые явления, (2) быть понятной и одновременно нетривиальной и (3) приносить практическую пользу. Работа была проделана большая, но с понятностью были проблемы, а практические применения и вовсе отложены на потом. Для разработки хорошей теории одной диссертации не достаточно.

Почему же потребовалось более 15 лет, чтобы добраться до первых внятных результатов? Виной тому не столько основная работа, которая отвлекала от теоретических занятий, сколько ошибки в методологии исследования. Алгоритм выхода из творческих тупиков вроде бы известен, но не все им пользуются, либо пользуются сокращённой версией — это и есть главная ошибка. Алгоритм такой:

в цикле повторять

- (1) попытки самостоятельного теоретического решения задачи;
- (2) погружение в современную англоязычную научную литературу;
- (3) проведение экспериментов;
- (4) поиск упрощённых постановок задач, которые удаётся решать;
- (5) обсуждение всех этих задач с коллегами.

Причём все шаги важны, ни один из них нельзя упускать!



Сейчас уже очевидно, что катализатором данного исследования послужил эксперимент с монотонной цепью алгоритмов, который программируется за один вечер, и ничто, в самом деле, не мешало сделать его 15 лет назад.

Ещё одной удачной идеей стало представление семейства алгоритмов в виде графа расслоения–связности (стр. 84). Илья Решетняк первым получил оценку расслоения–связности, справедливую для произвольного семейства алгоритмов, и намного лучшую, чем все предыдущие. Этот результат вошёл в его дипломную работу. Графики вероятности переобучения действительно впечатляли: для линейных классификаторов оценки были завышены лишь в несколько раз, максимум — на порядок. Однако в тот момент оценка показалась неудобной в использовании и вычислительно трудоёмкой. С этой оценкой связан следующий казус.

**Октябрь 2010 года.** На конференцию ИОИ-8 я взял с собой текст Евгения Соколова, студента третьего курса ВМК, который только начинал заниматься комбинаторными оценками. В числе прочего он буквально в пять строк выводил оценку вероятности переобучения, учитывающую верхнюю связность каждого алгоритма, и не требующую существования корректного алгоритма. Вчитавшись в доказательство, я осознал, что это прямое следствие принципа ПЗМ, а верхняя связность — не что иное, как мощность порождающего множества объектов. На радостях я добавил эту теорему в презентацию моего доклада, со ссылкой на Соколова, а своим аспирантам таинственно заявил, что их ждёт приятный сюрприз. После доклада выяснилось, что некоторое время назад они уже рассказывали мне нечто подобное, но настолько невразумительно, что я не смог понять идею и решил, что она совсем сырая. В тот же вечер Андрей Ивахненко показал, что можно улучшить оценку, если учесть ещё и нижнюю связность, и это не что иное, как мощность запрещающего множества объектов. Более того, в запрещающее множество надо включить всякий объект, на котором данный алгоритм допускает ошибку, при условии, что существует монотонно лучший алгоритм, не допускающий на нём ошибку. Это делает оценку намного более точной, так как теперь она существенно учитывает не только связность, но и расслоение. Всё это верно для произвольного множества алгоритмов, а это означает, что предположение о существовании корректного алгоритма наконец снято. И всё это — прямое следствие принципа ПЗМ, причём доказательство занимает несколько строк (стр. 86). Через пару дней я осознал, что этот результат в точности совпадает с оценкой Ильи Решетняка, просто он не использовал ПЗМ и доказывал всё с помощью комбинаторных рассуждений, в которых было легко запутаться. Из всей этой истории следует поучительный вывод: математика — это в первую очередь язык. Правильно подобранные слова и красота формул имеют огромное значение для восприятия результата. Наш совместный результат был должен на конференции РОАИ в декабре 2010 года.

Чуть позже Илья Толстихин, разбираясь с радемахеровской сложностью, заметил, что функционал Вапника-Червоненкиса эквивалентен функционалу вероятности переобучения, если в качестве метода обучения взять не минимизацию частоты ошибок на обучении, а максимизацию разности частот ошибок на контроле и на обу-

чении (*discrepancy maximization*). Вроде бы этот факт был давно известен. Но Илья заметил, что к этому методу обучения также можно применить принцип ПЗМ. Снова короткое доказательство в две строки, из которого следует, что для каждого алгоритма верхняя связность равна мощности порождающего множества объектов, а нижняя связность — мощности запрещающего (стр. 126). Отсюда немедленно следует верхняя оценка вероятности переобучения. Причём этот лёгко доставшийся результат гораздо лучше оценки Кочедыкова, стоившей нескольких страниц выкладок (стр. 128).

Теория в целом стала приобретать изящный вид. Основные содержательные теоремы доказываются в несколько строк, благодаря принципу ПЗМ, претендующему теперь на роль основного универсального инструмента получения оценок.

**24 декабря 2010 года** Андрей Ивахненко защитил кандидатскую диссертацию. В этой работе комбинаторная оценка вероятности переобучения на основе графа расслоения–связности была впервые применена к практическому случаю — семейству конъюнктивных логических закономерностей. Теория была доведена до практической реализации и даже привела к некоторому улучшению качества классификации. Оценка опиралась, опять-таки, на принцип ПЗМ, но теперь основная гипотеза записывалась сразу в виде неравенства, и для каждого алгоритма предполагалось строить только одну пару ПЗМ, что заведомо упрощало получение оценки (стр. 86). Первым идею с неравенством высказал вслух Саша Фрей, хотя она была очевидна. Неочевидным было то, что она приведёт к общей оценке расслоения–связности. Вычисление этой оценки является ресурсоёмким, но всё же вполне реализуемым благодаря двум ухищрениям. Во-первых, перебираются только алгоритмы из нижних слоёв семейства; в хороших случаях их там немного, а в плохих случаях оценка заведомо не нужна. Во-вторых, строятся эффективные процедуры перехода от произвольного вектора ошибок ко всем его соседним в следующем слое. Для семейства конъюнкций эта задача оказалась не вполне тривиальной и потребовала аккуратного описания структуры классов эквивалентности, чтобы не перебирать лишней раз алгоритмы с одинаковыми векторами ошибок (стр. 103). Представляется, что аналогичным образом можно расправляться и с другими семействами.

На данном этапе уже не достаточно понимания взаимосвязи расслоения и связности с переобучением. Модельные семейства интересны лишь как учебные задачи. На первый план выходят задачи получения комбинаторных оценок для реальных семейств, разработка с их помощью новых методов обучения, расширение области применимости комбинаторных оценок обобщающей способности.

## Список литературы

- [1] Айвазян С. А., Бухштабер В. М., Енюков И. С., Мешалкин Л. Д. Прикладная статистика: классификация и снижение размерности. — М.: Финансы и статистика, 1989.
- [2] Алминов Ю. И. Альтернатива методу математической статистики. — Знание, 1980.
- [3] Беляев Ю. К. Вероятностные методы выборочного контроля. — М.: Наука, 1975.
- [4] Большев Л. Н., Смирнов Н. В. Таблицы математической статистики. — М.: Наука, 1983.
- [5] Борисова И. А., Дюбанов В. В., Загоруйко Н. Г., Кутненко О. А. Сходство и компактность // Всеросс. конф. Математические методы распознавания образов-14. — М.: МАКС Пресс, 2009. — С. 89–92.
- [6] Ботов П. В. Точные оценки вероятности переобучения для монотонных и унимодальных семейств алгоритмов // Всеросс. конф. Математические методы распознавания образов-14. — М.: МАКС Пресс, 2009. — С. 7–10.
- [7] Вапник В. Н. Восстановление зависимостей по эмпирическим данным. — М.: Наука, 1979.
- [8] Вапник В. Н., Червоненкис А. Я. О равномерной сходимости частот появления событий к их вероятностям // ДАН СССР. — 1968. — Т. 181, № 4. — С. 781–784.
- [9] Вапник В. Н., Червоненкис А. Я. О равномерной сходимости частот появления событий к их вероятностям // Теория вероятностей и ее применения. — 1971. — Т. 16, № 2. — С. 264–280.
- [10] Вапник В. Н., Червоненкис А. Я. Теория распознавания образов. — М.: Наука, 1974.
- [11] Воронцов К. В. Оптимизационные методы линейной и монотонной коррекции в алгебраическом подходе к проблеме распознавания // ЖВМ и МФ. — 2000. — Т. 40, № 1. — С. 166–176.
- [12] Воронцов К. В. Обзор современных исследований по проблеме качества обучения алгоритмов // Таврический вестник информатики и математики. — 2004. — № 1. — С. 5–24.
- [13] Воронцов К. В., Колосков А. О. Профили компактности и выделение опорных объектов в метрических алгоритмах классификации // Искусственный Интеллект. — 2006. — С. 30–33.
- [14] Воронцов К. В., Решетняк И. М. Точные комбинаторные оценки обобщающей способности онлайн-обучения // Межд. конф. Интеллектуализация обработки информации ИОИ-8. — М.: МАКС Пресс, 2010. — С. 24–27.
- [15] Гаек Я., Шидак З. Теория ранговых критериев. — М.: Наука, 1971.
- [16] Гопла В. Д. Введение в алгебраическую теорию информации. — М.: Наука, 1995.
- [17] Грэхем Р., Кнут Д., Паташник О. Конкретная математика. — М.: Мир, 1998.
- [18] Гуров С. И. Точечная оценка вероятности 0-события // Всеросс. конф. Математические методы распознавания образов-14. — М.: МАКС Пресс, 2009. — С. 22–25.
- [19] Донской В. И. Алгоритмы обучения, основанные на построении решающих деревьев // ЖВМ и МФ. — 1982. — Т. 22, № 4. — С. 963–974.
- [20] Донской В. И. Колмогоровская сложность классов общерекурсивных функций с ограниченной ёмкостью // Таврический вестник информатики и математики. — 2005. —

№ 1. — С. 25–34.

- [21] Дэвид Г. Порядковые статистики. — М.: Наука, 1979. — 336 с.
- [22] Дюличева Ю. Ю. Оценка VCD  $r$ -редуцированного эмпирического леса // *Таврический вестник информатики и математики*. — 2003. — № 1. — С. 31–42.
- [23] Журавлёв Ю. И. Об алгебраическом подходе к решению задач распознавания или классификации // *Проблемы кибернетики*. — 1978. — Т. 33. — С. 5–68.
- [24] Журавлёв Ю. И., Рязанов В. В., Сенько О. В. «Распознавание». Математические методы. Программная система. Практические применения. — М.: Фазис, 2006.
- [25] Загоруйко Н. Г. Прикладные методы анализа данных и знаний. — Новосибирск: ИМ СО РАН, 1999.
- [26] Загоруйко Н. Г., Ёлкина В. Н., Лбов Г. С. Алгоритмы обнаружения эмпирических закономерностей. — Новосибирск: Наука, 1985.
- [27] Кобзарь А. И. Прикладная математическая статистика. — М.: Физматлит, 2006.
- [28] Колмогоров А. Н. Теория информации и теория алгоритмов / Под ред. Ю. В. Прохоров. — М.: Наука, 1987. — 304 с.
- [29] Кочедыков Д. А. Структуры сходства в семействах алгоритмов классификации и оценки обобщающей способности // Всеросс. конф. Математические методы распознавания образов-14. — М.: МАКС Пресс, 2009. — С. 45–48.
- [30] Лбов Г. С. Методы обработки разнотипных экспериментальных данных. — Новосибирск: Наука, 1981.
- [31] Лбов Г. С., Котюков В. И., Машаров Ю. П. Метод обнаружения логических закономерностей на эмпирических таблицах // *Вычислительные системы*. — 1976. — Т. 67. — С. 29–42.
- [32] Матросов В. Л. Корректные алгебры ограниченной ёмкости над множествами некорректных алгоритмов // *ДАН СССР*. — 1980. — Т. 253, № 1. — С. 25–30.
- [33] Матросов В. Л. Ёмкость алгебраических расширений модели алгоритмов вычисления оценок // *ЖВМиМФ*. — 1984. — Т. 24, № 11. — С. 1719–1730.
- [34] Матросов В. Л. Нижние границы ёмкости многомерных алгебр алгоритмов вычисления оценок // *ЖВМиМФ*. — 1984. — Т. 24, № 12. — С. 1881–1892.
- [35] Матросов В. Л. Ёмкость алгоритмических многочленов над множеством алгоритмов вычисления оценок // *ЖВМиМФ*. — 1985. — Т. 25, № 1. — С. 122–133.
- [36] Махина Г. А. Оценка обобщающей способности для монотонных алгоритмов классификации // 16-я международная конференция «Проблемы теоретической кибернетики». Нижний Новгород, 20–25 июня 2011. — 2011.
- [37] Норушис А. Построение логических (древообразных) классификаторов методами нисходящего поиска (обзор) // *Статистические проблемы управления*. Вып. 93 / Под ред. Ш. Раудис. — Вильнюс, 1990. — С. 131–158.
- [38] Орлов А. И. Эконометрика: Учебник для вузов. — М.: Экзамен, 2003. — 576 с.
- [39] Райгородский А. М. Экстремальные задачи теории графов и анализ данных. — М.: НИЦ «Регулярная и хаотическая динамика», Институт компьютерных исследований, 2008. — 118 с.
- [40] Рудаков К. В. Алгебраическая теория универсальных и локальных ограничений для алгоритмов распознавания. — Диссертация на соискание учёной степени д.ф.-м.н.,

- М.: ВЦ РАН. — 1992.
- [41] Рудаков К. В., Воронцов К. В. О методах оптимизации и монотонной коррекции в алгебраическом подходе к проблеме распознавания // *Докл. РАН.* — 1999. — Т. 367, № 3. — С. 314–317.
- [42] Сёмочкин А. Н. Линейные достроения частичного порядка на конечных множествах // *Деп. в ВИНТИ.* — 1998. — № 2964–В98. — С. 19.
- [43] Сёмочкин А. Н. Оценки функционала качества для класса алгоритмов с универсальными ограничениями монотонности // *Деп. в ВИНТИ.* — 1998. — № 2965–В98. — С. 20.
- [44] Смирнов Н. В. Оценка расхождения между эмпирическими кривыми распределения в двух независимых выборках // *Бюлл. Московского ун-та, серия А.* — 1939. — № 2. — С. 3–14.
- [45] Фрей А. И. Точные оценки вероятности переобучения для симметричных семейств алгоритмов // *Всеросс. конф. Математические методы распознавания образов-14.* — М.: МАКС Пресс, 2009. — С. 66–69.
- [46] Abu-Mostafa Y. S. Hints // *Neural Computation.* — 1995. — Vol. 7, no. 4. — Pp. 639–671.
- [47] Ambroladze A., Parrado-Hernández E., Shawe-Taylor J. Tighter PAC-Bayes bounds // *Advances in Neural Information Processing Systems 19* / Ed. by B. Schölkopf, J. Platt, T. Hoffman. — Cambridge, MA: MIT Press, 2007. — Pp. 9–16.
- [48] Anthony M. Generalization error bounds for threshold decision lists // *Journal of Machine Learning Research.* — 2004. — Vol. 5. — Pp. 189–217.
- [49] Anthony M., Shawe-Taylor J. A result of Vapnik with applications // *Discrete Applied Mathematics.* — 1993. — Vol. 47, no. 2. — Pp. 207–217.
- [50] Bartlett P. Lower bounds on the Vapnik-Chervonenkis dimension of multi-layer threshold networks // *Proceedings of the Sixth Annual ACM Conference on Computational Learning Theory.* — ACM Press, New York, NY, 1993. — Pp. 144–150.
- [51] Bartlett P. The sample complexity of pattern classification with neural networks: the size of the weights is more important than the size of the network // *IEEE Transactions on Information Theory.* — 1998. — Vol. 44, no. 2. — Pp. 525–536.
- [52] Bartlett P., Bousquet O., Mendelson S. Localized rademacher complexities // *COLT: 15th Annual Conference on Computational Learning Theory.* — Springer, Berlin, 2002. — Pp. 44–58.
- [53] Bartlett P., Bousquet O., Mendelson S. Local rademacher complexities. — Vol. 33. — Institute of Mathematical Statistics, 2005. — P. 1497–1537.
- [54] Bartlett P. L., Mendelson S., Phillips P. Local complexities for empirical risk minimization // *COLT: 17th Annual Conference on Learning Theory* / Ed. by J. Shawe-Taylor, Y. Singer. — Springer-Verlag, 2004. — Pp. 270–284.
- [55] Bauer M., Godreche C., Luck J. M. Statistics of persistent events in the binomial random walk: Will the drunken sailor hit the sober man? // *J.STAT.PHYS.* — 1999. — Vol. 96. — P. 963.
- [56] Bax E. T. Similar classifiers and VC error bounds: Tech. Rep. CalTech-CS-TR97-14: 1997.
- [57] Boucheron S., Bousquet O., Lugosi G. Theory of classification: A survey of some recent advances // *ESAIM: Probability and Statistics.* — 2005. — no. 9. — Pp. 323–375.

- [58] *Breiman L.* Technical note: Some properties of splitting criteria // *Machine Learning*. — 1996. — Vol. 24. — Pp. 41–47.
- [59] *Breiman L.* Arcing classifiers // *The Annals of Statistics*. — 1998. — Vol. 26, no. 3. — Pp. 801–849.
- [60] *Brin S.* Near neighbor search in large metric spaces // 21th International Conference on Very Large Data Bases (VLDB 1995). — 1995. — Pp. 574–584.
- [61] *Chavez E., Navarro G., Baeza-yates R., Marroquin J. L.* Searching in metric spaces // *ACM Computing Surveys*. — 1999. — Vol. 33. — Pp. 273–321.
- [62] *Chvátal V.* The tail of the hypergeometric distribution // *Discrete Mathematics*. — 1979. — Vol. 25, no. 3. — Pp. 285–287.
- [63] *Cohen W. W.* Fast effective rule induction // Proc. of the 12th International Conference on Machine Learning, Tahoe City, CA. — Morgan Kaufmann, 1995. — Pp. 115–123.
- [64] *Cohen W. W., Singer Y.* A simple, fast and effective rule learner // Proc. of the 16 National Conference on Artificial Intelligence. — 1999. — Pp. 335–342.
- [65] *Frei A. I.* Accurate estimates of the generalization ability for symmetric set of predictors and randomized learning algorithms // *Pattern Recognition and Image Analysis*. — 2010. — Vol. 20, no. 3. — P. 241–250.
- [66] *Freund Y.* Self bounding learning algorithms // COLT: Proceedings of the Workshop on Computational Learning Theory, Morgan Kaufmann Publishers. — 1998.
- [67] *Freund Y., Schapire R. E.* Experiments with a new boosting algorithm // International Conference on Machine Learning. — 1996. — Pp. 148–156.
- [68] *Fürnkranz J., Flach P. A.* Roc ‘n’ rule learning-towards a better understanding of covering algorithms // *Machine Learning*. — 2005. — Vol. 58, no. 1. — Pp. 39–77.
- [69] *Hastie T., Tibshirani R., Friedman J.* The Elements of Statistical Learning, 2nd ed. — Springer, 2009.
- [70] *Herbrich R., C. Williamson R.* Algorithmic luckiness // *Journal of Machine Learning Research*. — 2002. — no. 3. — Pp. 175–212.
- [71] *Herbrich R., Williamson R. C.* Learning and generalization: theoretical bounds. — Cambridge, MA, USA: MIT Press, 2002. — Pp. 619–623.
- [72] *Hosmer D. W., Lemeshow S.* Applied Logistic Regression, second ed. — New York: Wiley, 2000.
- [73] *Ivanov M. N.* Prototype sample selection based on minimization of the complete cross validation functional // *Pattern Recognition and Image Analysis*. — 2010. — Vol. 20, no. 4. — Pp. 427–437.
- [74] *Karpinski M., Macintyre A.* Polynomial bounds for VC dimension of sigmoidal neural networks // 27th ACM Symposium on Theory of Computing, Las Vegas, Nevada, US. — 1995. — Pp. 200–208.
- [75] *Kearns M. J., Mansour Y., Ng A. Y., Ron D.* An experimental and theoretical comparison of model selection methods // 8th Conf. on Computational Learning Theory, Santa Cruz, California, US. — 1995. — Pp. 21–30.
- [76] *Koltchinskii V.* Rademacher penalties and structural risk minimization // *IEEE Transactions on Information Theory*. — 2001. — Vol. 47, no. 5. — Pp. 1902–1914.
- [77] *Koltchinskii V., Panchenko D.* Rademacher processes and bounding the risk of function

- learning // High Dimensional Probability, II / Ed. by D. E. Gine, J. Wellner. — Birkhauser, 1999. — Pp. 443–457.
- [78] *Koltchinskii V., Panchenko D.* Empirical margin distributions and bounding the generalization error of combined classifiers // *The Annals of Statistics*. — 2002. — Vol. 30, no. 1. — Pp. 1–50.
- [79] *Koltchinskii V., Panchenko D., Lozano F.* Further explanation of the effectiveness of voting methods: The game between margins and weights // 14th Annual Conference on Computational Learning Theory, COLT 2001 and 5th European Conference on Computational Learning Theory, EuroCOLT 2001, Amsterdam, The Netherlands, July 2001, Proceedings. — Vol. 2111. — Springer, Berlin, 2001. — Pp. 241–255.
- [80] *Kuncheva L.* Combining pattern classifiers. — John Wiley & Sons, Inc., 2004.
- [81] *Langford J.* Quantitatively Tight Sample Complexity Bounds: Ph.D. thesis / Carnegie Mellon Thesis. — 2002.
- [82] *Langford J., McAllester D.* Computable shell decomposition bounds // Proc. 13th Annu. Conference on Comput. Learning Theory. — Morgan Kaufmann, San Francisco, 2000. — Pp. 25–34.
- [83] *Lugosi G.* On concentration-of-measure inequalities. — Machine Learning Summer School, Australian National University, Canberra. — 2003.
- [84] *Mann H. B., Whitney D. R.* On a test of whether one of two random variables is stochastically larger than the other // *The Annals of Mathematical Statistics*. — 1947. — Vol. 18, no. 1. — Pp. 50–60.
- [85] *Martin J. K.* An exact probability metric for decision tree splitting and stopping // *Machine Learning*. — 1997. — Vol. 28, no. 2-3. — Pp. 257–291.
- [86] *Mazurov V., Khachai M., Rybin A.* Committee constructions for solving problems of selection, diagnostics and prediction // *Proceedings of the Steklov Institute of mathematics*. — 2002. — Vol. 1. — Pp. 67–101.
- [87] *Mertens S., Engel A.* Vapnik-Chervonenkis dimension of neural networks with binary weights // *Phys. Rev. E*. — 1997. — Vol. 55, no. 4. — Pp. 4478–4488.
- [88] *Mullin M., Sukthankar R.* Complete cross-validation for nearest neighbor classifiers // Proceedings of International Conference on Machine Learning. — 2000. — Pp. 639–646.
- [89] *Quinlan J.* Induction of decision trees // *Machine Learning*. — 1986. — Vol. 1, no. 1. — Pp. 81–106.
- [90] *Quinlan J. R.* C4.5: Programs for machine learning. — Morgan Kaufmann, San Francisco, CA, 1993.
- [91] *Quinlan J. R.* Bagging, boosting, and C4.5 // AAI/IAAI, Vol. 1. — 1996. — Pp. 725–730.
- [92] *Rissanen J.* Modeling by shortest data description // *Automatica*. — 1978. — Vol. 14. — Pp. 465–471.
- [93] *Rivest R. L.* Learning decision lists // *Machine Learning*. — 1987. — Vol. 2, no. 3. — Pp. 229–246.
- [94] *Rückert U., Kramer S.* Towards tight bounds for rule learning // Proc. 21th International Conference on Machine Learning, Banff, Canada. — 2004. — P. 90.
- [95] *Scala M.* Hypergeometric tail inequalities: ending the insanity // *Proofs*. — 2009. — Pp. 1–5.

- [96] *Schapire R.* The boosting approach to machine learning: An overview // MSRI Workshop on Nonlinear Estimation and Classification, Berkeley, CA. — 2001.
- [97] *Schapire R. E., Freund Y., Lee W. S., Bartlett P.* Boosting the margin: a new explanation for the effectiveness of voting methods // *Annals of Statistics*. — 1998. — Vol. 26, no. 5. — Pp. 1651–1686.
- [98] *Schapire R. E., Singer Y.* Improved boosting using confidence-rated predictions // *Machine Learning*. — 1999. — Vol. 37, no. 3. — Pp. 297–336.
- [99] *Sill J.* The capacity of monotonic functions // *Discrete Applied Mathematics (special issue on VC dimension)*. — 1998. — Vol. 86. — Pp. 95–107.
- [100] *Sill J.* Monotonicity and connectedness in learning systems: Ph.D. thesis / California Institute of Technology. — 1998.
- [101] *Sill J., Abu-Mostafa Y. S.* Monotonicity hints // *Advances in Neural Information Processing Systems* / Ed. by M. C. Mozer, M. I. Jordan, T. Petsche. — Vol. 9. — The MIT Press, 1997. — Pp. 634–640.
- [102] *Sokolova M., Marchand M., Japkowicz N., Shawe-Taylor J.* The decision list machine // *Advances in Neural Information Processing Systems* 15. — MIT-Press, Cambridge, MA, USA, 2003. — Pp. 921–928.
- [103] *Valiant L. G.* A theory of the learnable // *Communications of the ACM*. — 1984. — Vol. 27. — Pp. 1134–1142.
- [104] *Vapnik V.* Estimation of Dependencies Based on Empirical Data. — Springer-Verlag, New York, 1982.
- [105] *Vapnik V.* The nature of statistical learning theory. — Springer-Verlag, New York, 1995.
- [106] *Vapnik V.* Statistical Learning Theory. — Wiley, New York, 1998.
- [107] *Vayatis N., Azencott R.* Distribution-dependent Vapnik-Chervonenkis bounds // *Lecture Notes in Computer Science*. — 1999. — Vol. 1572. — Pp. 230–240.
- [108] *Vorontsov K. V.* Combinatorial probability and the tightness of generalization bounds // *Pattern Recognition and Image Analysis*. — 2008. — Vol. 18, no. 2. — Pp. 243–259.
- [109] *Vorontsov K. V.* On the influence of similarity of classifiers on the probability of overfitting // *Pattern Recognition and Image Analysis: new information technologies (PRIA-9)*. — Vol. 2. — Nizhni Novgorod, Russian Federation, 2008. — Pp. 303–306.
- [110] *Vorontsov K. V.* Splitting and similarity phenomena in the sets of classifiers and their effect on the probability of overfitting // *Pattern Recognition and Image Analysis*. — 2009. — Vol. 19, no. 3. — Pp. 412–420.
- [111] *Vorontsov K. V.* Exact combinatorial bounds on the probability of overfitting for empirical risk minimization // *Pattern Recognition and Image Analysis*. — 2010. — Vol. 20, no. 3. — Pp. 269–285.
- [112] *Vorontsov K. V., Ivahnenko A. A., Reshetnyak I. M.* Generalization bound based on the splitting and connectivity graph of the set of classifiers // *Pattern Recognition and Image Analysis: new information technologies (PRIA-10)*, St. Petersburg, Russian Federation, December 5–12, 2010. — 2010.
- [113] *Wilcoxon F.* Individual comparisons by ranking methods // *Biometrics Bulletin*. — 1945. — Vol. 1, no. 6. — Pp. 80–83.
- [114] *Zukhba A. V.* NP-completeness of the problem of prototype selection in the nearest



neighbor method // *Pattern Recognition and Image Analysis.* — 2010. — Vol. 20.