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

К. В. Воронцов
http://www.MachineLearning.ru - Участник:Vokov
voron@forecsys.ru
16 февраля 2012 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, либо высказанные в обсуждении страницы «Теория надёжности обучения по прецедентам (курс лекций, К.В.Воронцов)» вики-ресурса www.MachineLearning.ru.



## Содержание

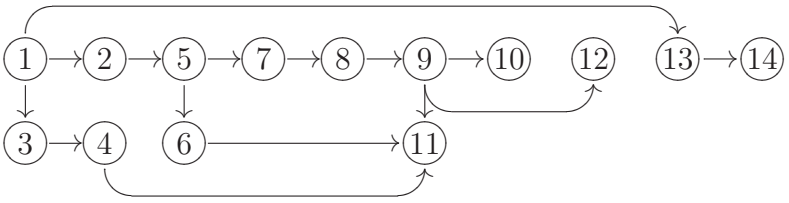
Спецкурс «Теория надёжности обучения по прецедентам» читается студентам кафедры «Математические методы прогнозирования» ВМК МГУ с 2007 года в дополнение к обязательному кафедральному курсу «Математические методы распознавания образов». Спецкурс знакомит студентов с современным состоянием теории статистического 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 первая часть курса, основанная на результатах автора, его учеников и коллег: Д. Кочедыкова, А. Ивахненко, И. Решетняка, А. Фрея, И. Толстихина, П. Ботова, М. Иванова, И. Гуза, Г. Махиной. Вторая 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, но может содержать чуть больше материала, чем можно успеть изложить за полтора часа. 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`.

Автор будет рад любым замечаниям и предложениям, направленным на email `voron@forecsys.ru`.

*Константин Воронцов,*
*февраль 2012 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.

Существует простой алгоритм, который выдаёт правильные решения для объектов обучающей выборки. Он сравнивает новый объект с каждым из обучающих объектов, и в случае полного совпадения выдаёт правильное решение из таблицы исходных данных, а во всех остальных случаях выбирает случайное решение. Такой алгоритм способен обучиться разве что игре в «крестики-нолики». В более содержательных задачах для успешного обучения важно не только запоминать, но и обобщать. Способность алгоритмов правильно находить общие закономерности по эмпирическим данным называют *обобщающей способностью* (generalization ability).

Основная задача *теории статистического обучения*[1] (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.

---

[1] 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 $\mathbb{X} = \{x_1, \ldots, x_L\}$, называемое *генеральной выборкой*; множество $A$, элементы которого называются *алгоритмами*, и бинарная функция $I\colon A \times \mathbb{X} \to \{0, 1\}$, 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*[2] называется отображение $\mu\colon 2^{\mathbb{X}} \to A$, 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:

$$A(X) = \operatorname{Arg}\min_{a \in A} n(a, X) = \big\{a \in A\colon n(a, X) \leqslant n(a', X),\ \forall a' \in A\big\}. \tag{1.1}$$

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 $a = \mu X$:

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

---

[2] В англоязычной литературе метод обучения принято называть *алгоритмом обучения* (learning algorithm) [72], а алгоритм $a\colon \mathbb{X} \to \mathbb{Y}$ — классификатором (classifier), гипотезой (hypothesis), решающей функцией (decision function), либо просто функцией (function). Термин «алгоритм» в смысле отображения $\mathbb{X} \to \mathbb{Y}$ употребляется в работах научной школы академика Ю. И. Журавлёва [26]. Термины «метод» и «алгоритм», обозначающие процедуру построения функции $a$ по выборке данных употребляются в отечественной литературе вперемежку [11, 1, 28, 27].

| | $a_1$ | $a_2$ | $a_3$ | $a_4$ | $a_5$ | $a_6$ | $a_7$ | $a_8$ | |
|---|---|---|---|---|---|---|---|---|---|
| $x_1$ | 1 | 1 | 1 | **0** | 0 | 1 | 1 | 1 | |
| $x_2$ | 0 | 0 | 0 | **0** | 1 | 1 | 0 | 0 | |
| $x_3$ | 0 | 1 | 1 | **0** | 0 | 0 | 0 | 0 | $X$ — наблюдаемая обучающая выборка |
| $x_4$ | 1 | 0 | 1 | **0** | 1 | 0 | 1 | 0 | |
| $x_5$ | 0 | 0 | 1 | **0** | 1 | 1 | 0 | 0 | |
| $x_6$ | 0 | 0 | 0 | **1** | 1 | 1 | 0 | 0 | |
| $x_7$ | 1 | 0 | 0 | **1** | 1 | 1 | 0 | 0 | |
| $x_8$ | 0 | 0 | 0 | **1** | 0 | 0 | 0 | 1 | $\bar{X}$ — скрытая контрольная выборка |
| $x_9$ | 0 | 1 | 1 | **1** | 1 | 1 | 0 | 0 | |
| $x_{10}$ | 0 | 1 | 1 | **1** | 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 $\delta_\mu(X) \geqslant \varepsilon$ при некотором достаточно малом $\varepsilon \in (0, 1)$, то говорят, что метод $\mu$ *переобучается* на выборке $X$.

**Матрица ошибок.** Бинарный вектор-столбец $\vec{a} = \big(I(a, x_i)\big)_{i=1}^L$ 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 $y\colon \mathbb{X} \to \mathbb{Y}$ называется *целевой зависимостью* (target function).

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

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

В качестве *индикатора ошибки* возьмём бинарную функцию $I\colon A \times \mathbb{X} \to \{0,1\}$, 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[3]

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

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

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

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 $\langle \xi, \theta \rangle = \xi_1\theta_1 + \cdots + \xi_n\theta_n$ скалярное произведение векторов в $\mathbb{R}^n$. *Линейные семейства алгоритмов* определяются следующим образом:

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

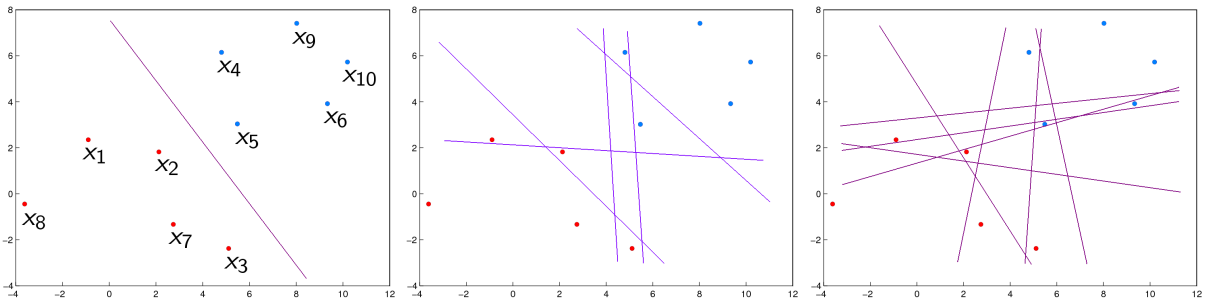
Линейный алгоритм классификации представляет собой гиперплоскость с направляющим вектором $\theta$, разделяющую пространство на две области — классы $-1$ и $+1$.

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.

---

[3] Квадратные скобки переводят логическое значение в числовое: [ложь] = 0, [истина] = 1. Это очень практичное обозначение, называемое *нотацией Айверсона* [19].




| | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| $x_1$ | 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 вверху показаны все алгоритмы с попарно различными векторами ошибок, с числом ошибок, соответственно, 0, 1, 2.

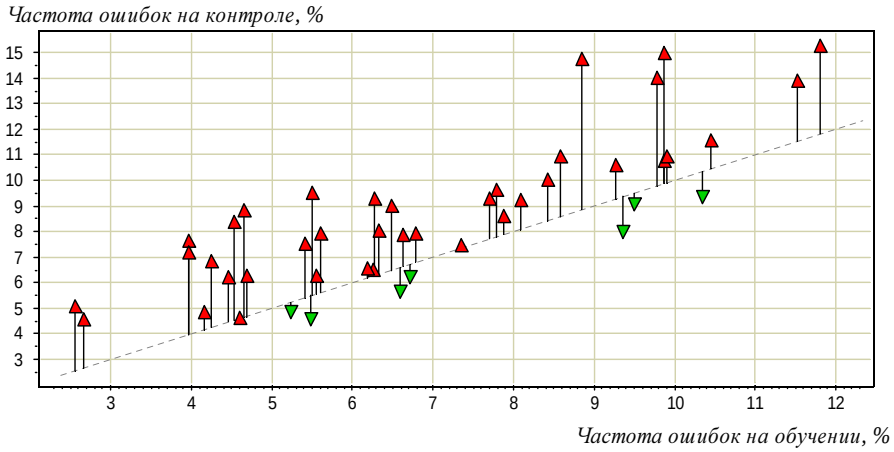



**Рис. 1.3.** Зависимость $\nu(\mu X, \bar{X})$ от $\nu(\mu X, X)$ для задачи прогнозирования отдалённого результата хирургического лечения атеросклероза.

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 биссектирисы. 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 контрольный 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 $\psi\colon \mathbb{X}^L \to \{0,1\}$. 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:*

$$\mathsf{P}_\tau\, \psi(\tau\mathbb{X}) = \frac{1}{L!} \sum_{\tau \in S_L} \psi(\tau\mathbb{X}). \tag{1.2}$$

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

$$F_\xi(z) = \mathsf{P}_\tau\big[\xi(\tau\mathbb{X}) \leqslant z\big]. \tag{1.3}$$

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

$$\mathsf{E}_\tau \xi(\tau\mathbb{X}) = \frac{1}{L!} \sum_{\tau \in S_L} \xi(\tau\mathbb{X}). \tag{1.4}$$

Заметим, что знаки $\mathsf{P}_\tau$ и $\mathsf{E}_\tau$ можно понимать как операцию среднего арифметического по всем перестановкам: $\mathsf{P}_\tau \equiv \mathsf{E}_\tau \equiv \frac{1}{L!}\sum_{\tau\in S_L}$, 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:

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

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

**Эмпирические оценки вероятности.** Непосредственное вычисление величины $\mathsf{P}\,\varphi(X)$ как доли разбиений возможно только при небольших значениях $\ell$ или $k$. В типичных случаях число разбиений $C_L^\ell$ огромно. Приближённой оценкой для $\mathsf{P}\,\varphi(X)$ 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:

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

Далее символами $\hat{\mathsf{P}}$ и $\hat{\mathsf{E}}$ 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 $\varepsilon \in (0,1)$:

$$Q_\varepsilon \equiv Q_\varepsilon(\mu,\mathbb{X}) = \mathsf{P}\big[\delta_\mu(X) \geqslant \varepsilon\big] = \mathsf{P}\big[\nu(\mu X,\bar{X}) - \nu(\mu X, X) \geqslant \varepsilon\big]. \tag{1.5}$$

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

$$R_\varepsilon \equiv R_\varepsilon(\mu,\mathbb{X}) = \mathsf{P}\big[\nu(\mu X,\bar{X}) \geqslant \varepsilon\big], \tag{1.6}$$

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

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

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

Функционалы $Q_\varepsilon$, $R_\varepsilon$ и $C$ характеризуют обобщающую способность метода $\mu$ на выборке $\mathbb{X}$. Их оценивание будет основной задачей на протяжении всего курса.

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

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

Параметр $\varepsilon$ называют *точностью*, а $\eta$ — *надёжностью* оценки [8].

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

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

Итак, обращение верхних оценок $Q_\varepsilon$ или $R_\varepsilon$ даёт верхние оценки частоты ошибок на контроле. Теперь вместо минимизации эмпирического риска $\nu(a, X)$ мы можем минимизировать оценку (1.8), которая отличается от эмпирического риска наличием дополнительного слагаемого $\varepsilon(\eta)$. Тем самым мы фактически переходим к новому методу обучения $\mu'$, который пытается оптимизировать обобщающую способность. Этот приём очень важен в SLT, так как он указывает путь практического применения теоретических оценок вероятности переобучения.

Чтобы применение этого приёма было корректным, оценка (1.8) должна выполняться не для какого-то одного метода обучения $\mu$, а для достаточно широкого класса методов, в который должен попасть и новый метод $\mu'$. В лекции 5 мы рассмотрим оценки Вапника-Червоненкиса, справедливые *для любого* метода обучения. К сожалению, за упрощение постановки задачи приходится платить завышенностью оценок. Разобравшись в причинах завышенности, мы построим более аккуратную комбинаторную теорию, в которой оценки вида (1.8) выводятся для специального класса *монотонных методов обучения*. Это будет сделано в лекции 9.

Может возникнуть вопрос — зачем использовать обращение вместо того, чтобы поступить проще — минимизировать непосредственно ожидаемую частоту ошибок на контроле (1.7). Теоретически, преимущество оценки (1.8) в том, что она является доверительной оценкой. При $\eta$, достаточно близких к нулю, она учитывает не только среднее значение частоты ошибок на контроле, но и величину её разброса. При $\eta = \frac{1}{2}$ она переходит в *медианную оценку*, которая, как правило, мало отличается от ожидаемой частоты ошибок на контроле. Практическое преимущество (1.8) перед (1.7) не очевидно и должно проверяться в экспериментах, в то же время, недостатком может быть вычислительная неэффективность операции обращения.

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

**О предрассудках и переобучении.** Переобучение — это чрезмерно точная подгонка алгоритма $a$ под конкретную обучающую выборку $X$ в ущерб его *обобщающей способности*. Ожидается, что метод $\mu$ обнаружит некоторые общие закономерности генеральной выборки $\mathbb{X}$, но он находит ложные закономерности — *предрассудки*, случайно проявившиеся на относительно небольшой подвыборке $X \subset \mathbb{X}$. Понятие «предрассудка» было введено М. М. Бонгардом в знаменитой книге «Проблемы узнавания» [5], до сих пор не теряющей своей актуальности. Бонгард не даёт строгого определения предрассудка, ограничиваясь следующим примером.

«Пусть человеку, не знающему ни итальянского, ни испанского языков, показали стопку итальянских и стопку испанских книг и сказали, что это объекты, соответственно, I и II классов. Во время поиска достаточных признаков этот человек (ничего не знающий об истинном принципе деления на классы) будет проверять формат книг, число страниц, размеры шрифта, чёткость печати, твёрдость переплёта, цвет обложки, и т. п. И если ему дали небольшое число книг каждого класса, то весьма вероятно, что он отберёт ложные признаки (у него появятся предрассудки). Если, например, случилось, что все итальянские книги были переплетены, а среди испанских нашлись две без переплёта, то у человека может создаться предрассудок, что отсутствие переплёта свидетельствует о принадлежности книги к классу II.

Пусть человеку показали при обучении большое число книг каждого класса, и он решил отбирать только признаки, характеризующие много книг. В этом случае вероятность того, что, например, свойство переплёта отберётся в качестве полезного признака, будет мала. То же самое можно сказать и о любом другом предрассудке (толщине книги, формате и т. п.). Посмотрим, что произойдёт, если человек перейдёт от проверки таких простых признаков к сложным, составным признакам. Он может проверить разность числа слов на чётных и нечётных страницах. Или ту же разность на двадцать второй и двенадцатой страницах. Или произведение средней длины абзаца на на число страниц, начинающихся с абзаца. Или частное от деления числа точек на высоту шрифта и т. д. Вероятность того, что каждый такой предрассудок выдержит проверку, мала. Но сложных признаков существует чрезвычайно много. Поэтому может случиться, что, несмотря на малую вероятность отбора каждого из них, математическое ожидание числа отобранных предрассудков окажется большим. А это означает, что память начнёт забиваться предрассудками. Таким образом, "богатый выбор" не только увеличивает возможности обучаемой машины, но и чреват появлением предрассудков.»

Таким образом, уже в середине 60-х годов из опыта решения практических задач, да и просто из общих соображений, стали ясны две основные закономерности переобучения: оно уменьшается с ростом длины выборки и увеличивается с ростом сложности семейства алгоритмов. В конце 60-х годов В. Н. Вапник и А. Я. Червоненкис впервые предложили количественное описание этих закономерностей [9].

Более точную 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 од-

ному 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 $\delta = \frac{m-s}{k} - \frac{s}{\ell} = \frac{m}{k} - s\frac{L}{\ell k}$ увеличивается с ростом $D$.

Эти рассуждения остаются в силе и в общем случае, когда алгоритмы не являются независимыми и допускают различное число ошибок. При этом проявляются две дополнительные, более тонкие, закономерности переобучения.

Первая закономерность относится к таким семействам алгоритмов, которые разбиваются на *слои* по числу ошибок $m$ и содержат относительно небольшое число хороших алгоритмов в нижних слоях с малым $m$ и огромное количество плохих алгоритмов в верхних слоях с большим $m$. Именно эта ситуация чаще всего наблюдается на практике. Хорошие алгоритмы имеют больше шансов получить минимальное число ошибок на обучающей выборке, в результате метод обучения выбирает их гораздо чаще. Плохие алгоритмы, наоборот, почти никогда не выбираются, и в первом приближении можно полагать, что их в семействе вообще нет. Таким образом, *расслоение* семейства алгоритмов способствует уменьшению его «эффективной сложности» и снижению переобучения.

Вторая закономерность относится к таким семействам алгоритмов, которые содержат много похожих алгоритмов. Эта ситуация также очень распространена на практике. Допустим, семейство алгоритмов классификации задаётся функцией $\varphi(x, \theta)$, непрерывной по параметру $\theta$. Каждому значению $\theta$ соответствует некоторый вектор ошибок. При непрерывном изменении $\theta$ вектор ошибок сначала будет оставаться тем же, потом изменится, скорее всего, только на одном объекте. Например, двигая разделяющую гиперплоскость линейного классификатора, мы последовательно пройдём через все точки выборки, пересекая их по одной, за исключением, быть может, редких особых положений гиперплоскости, при которых она проходит сразу через две или более точек, см. рис. 1.2. Алгоритмы, векторы ошибок которых отличаются только на одном объекте, будем называть *связными*. Добавление в семейство алгоритма, связного с данным, почти не увеличивает сложность семейства, так как это «почти тот же самый алгоритм». Таким образом, *связность* семейства также способствует уменьшению его «эффективной сложности» и снижению переобучения.

В лекциях 8–11 будет показано, что эффекты расслоения и связности снижают переобучение, а их совместный учёт радикально улучшает точность оценок.

## §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 $\langle \mathscr{X}, \Omega, \mathsf{P} \rangle$, где $\mathscr{X}$ — множество допустимых объектов, $\Omega$ — аддитивная $\sigma$-алгебра событий на $\mathscr{X}$, $\mathsf{P}$ — вероятностная мера, определённая на событиях из $\Omega$. В задачах статистического анализа данных обычно предполагается, что множество объектов $\mathbb{X} = \{x_1, \ldots, x_L\}$ является *простой выборкой*, то есть объекты

выбираются из множества $\mathscr{X}$ случайно и независимо согласно вероятностной мере $\mathsf{P}$. Независимость означает, что вероятностная мера на множестве выборок $\mathscr{X}^L$ инвариантна относительно перестановок элементов выборки. В приложениях множество $\mathscr{X}$, как правило, бесконечно, а мера $\mathsf{P}$ неизвестна.

В слабой аксиоматике множество $\mathscr{X}$ 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*

$$\mathsf{P}_\tau\, \psi(\tau\mathbb{X}) = f(\mathbb{X}). \tag{1.9}$$

*Тогда в сильной аксиоматике, если выборка $\mathbb{X}$ простая, то*

$$\mathsf{P}_{\mathbb{X}}\, \psi(\mathbb{X}) = \mathsf{E}_{\mathbb{X}} f(\mathbb{X}). \tag{1.10}$$

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

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

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:

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

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 $\big\{X\colon |\nu(S,X)-P(S)|>\varepsilon\big\}$, 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 $|X'|\to\infty$. 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:

$$Q_\varepsilon^S=\mathsf{P}_{X,\bar X}\big\{|\nu(S,X)-\nu(S,X')|>\varepsilon\big\}. \tag{1.12}$$

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 $\mathsf{P}$ на бесконечном множестве $\mathscr{X}$. 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 $\big\{X,X'\colon |\nu(S,X)-\nu(S,X')|>\varepsilon\big\}$ 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 $\mathsf{P}$ на бесконечном множестве $\mathscr{X}$, либо осуществить предельный переход $\nu(S,X')\to P(S)$ при $|X'|\to\infty$, 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 [33, 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 [18]. 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» [33, 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

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

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

В слабой аксиоматике запись $|X| \to \infty$ 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 $\mathsf{P}_\tau\, \psi(\tau\mathbb{X}) = f(L)$, зависящую от длины выборки, мы можем исследовать асимптотическое поведение числовой функции $f(L)$ при $L \to \infty$. 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 $\mathsf{P}$, $\mathsf{E}$ 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 вычислимых 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: $\mathbb{X} = \left\{-\frac{L}{2}, \ldots, -\frac{3}{2}, -\frac{1}{2}, \frac{1}{2}, \frac{3}{2}, \ldots, \frac{L}{2}\right\}$. Для семейства алгоритмов классификации $a_\theta(x) = \operatorname{sign}(x - \theta)$ с параметром $\theta \in \mathbb{R}$ построить матрицу ошибок и график зависимости $n(a_\theta, \mathbb{X})$ от $\theta$, если целевая зависимость имеет вид:

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

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

**Задача 1.2** (2). В задаче классификации с двумя классами $\mathbb{Y} = \{-1, 1\}$ выборка задана точками на окружности: $\mathbb{X} = \left\{(\sin \varphi_i, \cos \varphi_i) \colon \varphi_i = \left(i - \frac{1}{2}\right)\frac{2\pi}{L},\ i = 1, \ldots, L\right\}$. Целевая зависимость $y(x) = \operatorname{sign} \xi_2$, $x = (\xi_1, \xi_2) \in \mathbb{R}^2$. Для семейства линейных классификаторов $a_\theta(x) = \operatorname{sign}(\theta_1\xi_1 + \theta_2\xi_2 + \theta_0)$, 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 ($m = 0, \ldots, L$) на выборке $\mathbb{X}$?

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

# 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: $S = \{x_i \colon I(a, x_i) = 1\}$, $I(a, x_i) = [x_i \in S]$.

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

$$\mathsf{P}\big[|\nu(\bar{X}) - \nu(X)| \geqslant \varepsilon\big] \leqslant \eta(\varepsilon); \tag{2.1}$$

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

$$\mathsf{P}\big[\nu(\bar{X}) - \nu(X) \geqslant \varepsilon\big] \leqslant \eta(\varepsilon). \tag{2.2}$$

**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:*

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

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

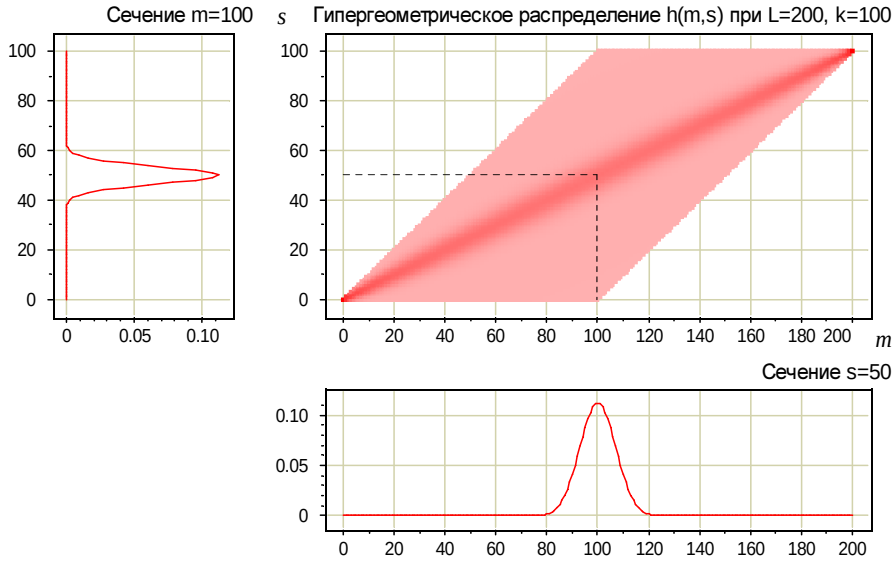



**Рис. 2.1.** Область определения гипергеометрической функции $h(m,s) = h_L^{\ell,\,m}(s)$ при $L = 200$, $\ell = k = 100$, $m = 30$.

## §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 $(m,s)$: $0 \leqslant m \leqslant L$, $\max\{0, m-k\} = s_0 \leqslant s \leqslant s_1 = \min\{\ell, m\}$. 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:

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

Справедлива формула полной вероятности:

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

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 $\bar{H}$:

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

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 $m = 0, \ldots, L$, 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 бóльших значениях параметров $L$, $\ell$, $m$ предельными распределениями для $h(s) = h_L^{\ell,\,m}(s)$ могут быть только распределения одного из четырёх типов:

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

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:*

$$\mathsf{P}\big[\nu(X) \leqslant \varepsilon\big] = \mathcal{H}_L^{\ell,m}(\varepsilon\ell); \tag{2.6}$$

$$\mathsf{P}\big[\nu(\bar{X}) \geqslant \varepsilon\big] = \mathcal{H}_L^{\ell,m}(m - \varepsilon k); \tag{2.7}$$

$$\mathsf{P}\big[\nu(\bar{X}) - \nu(X) \geqslant \varepsilon\big] = \mathcal{H}_L^{\ell,m}(s_m(\varepsilon)), \qquad s_m(\varepsilon) = \tfrac{\ell}{L}(m - \varepsilon k); \tag{2.8}$$

$$\mathsf{P}\big[|\nu(\bar{X}) - \nu(X)| \geqslant \varepsilon\big] = \mathcal{H}_L^{\ell,m}(s_m(\varepsilon)) + \bar{\mathcal{H}}_L^{\ell,m}(\bar{s}_m(\varepsilon)), \quad \bar{s}_m(\varepsilon) = \tfrac{\ell}{L}(m + \varepsilon k). \tag{2.9}$$

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

$$\mathsf{P}\big[\nu(X) \leqslant \varepsilon\big] = \sum_{s=0}^{\ell} \big[\tfrac{s}{\ell} \leqslant \varepsilon\big]\, \mathsf{P}\big[n(X) = s\big] = \sum_{s=s_0}^{\lfloor \varepsilon\ell \rfloor} h_L^{\ell,m}(s) = \mathcal{H}_L^{\ell,m}(\varepsilon\ell);$$

$$\mathsf{P}\big[\nu(\bar{X}) \geqslant \varepsilon\big] = \sum_{s=0}^{\ell} \big[\tfrac{m-s}{k} \geqslant \varepsilon\big]\, \mathsf{P}\big[n(\bar{X}) = m - s\big] = \sum_{s=s_0}^{\lfloor m - \varepsilon k \rfloor} h_L^{\ell,m}(s) = \mathcal{H}_L^{\ell,m}(m - \varepsilon k).$$

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

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

$$\begin{aligned}\mathsf{P}\big[|\nu(\bar{X}) - \nu(X)| \geqslant \varepsilon\big] &= \mathsf{P}\big[\nu(\bar{X}) - \nu(X) \geqslant \varepsilon\big] + \mathsf{P}\big[\nu(X) - \nu(\bar{X}) \geqslant \varepsilon\big] = \\ &= \mathcal{H}_L^{\ell,m}(s_m(\varepsilon)) + \bar{\mathcal{H}}_L^{\ell,m}(\bar{s}_m(\varepsilon)).\end{aligned}$$

■

**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 $L, \ell, m \to \infty$ 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 [8]. По сути, эти две оценки отличаются не принципиально.

Классические неравенства Чебышёва, Чернова, Бернштейна, Хёффдинга [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.

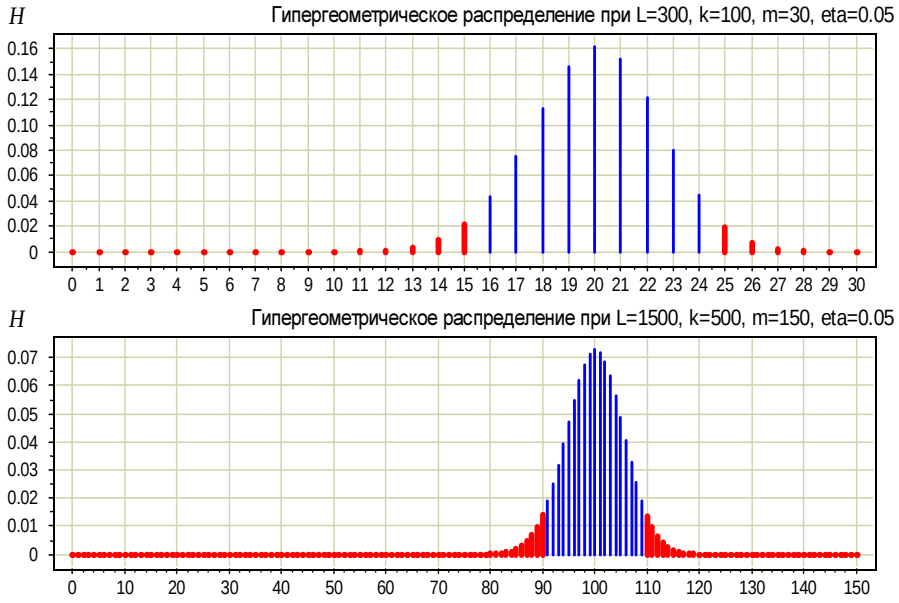



**Рис. 2.2.** Гипергеометрическая функция $h_L^{\ell,m}(s)$. Верхний график получен при $L = 300$, $\ell = 200$, $m = 30$. Выделены крайние левые, $[s_0, s_m(\varepsilon)] = [0, 15]$, и крайние правые $[s_m(\varepsilon), s_1] = [25, 30]$, члены распределения, соответствующие значению надёжности $\eta = 0.05$. 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 $\{0, \ldots, L\} \times \{0, \ldots, \ell\}$ 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 $(b_1, \ldots, b_L)$, 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 $s = s_0, \ldots, s_m(\varepsilon)$.

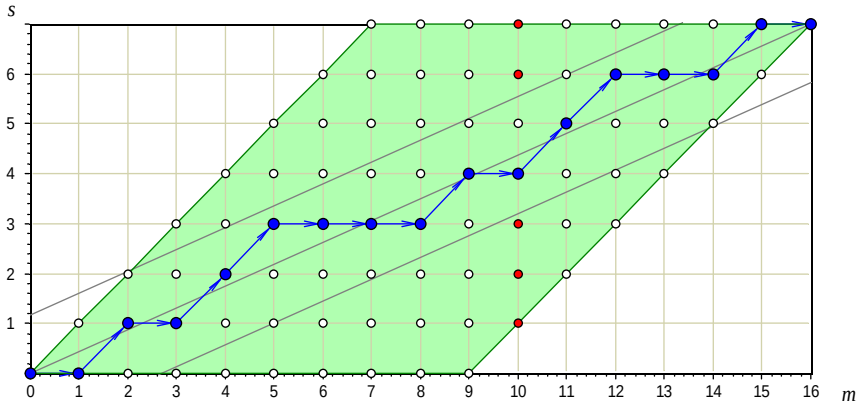



**Рис. 2.3.** Траектория последовательности $\{b_i\}_{i=0}^{L} = 0101100010110010$, при $L = 16$, $\ell = 7$, $\varepsilon = 0.3$. Проведены линии $s = \frac{\ell}{L}m$ и $s = \frac{\ell}{L}(m \pm \varepsilon k)$. При $m = 10$ выделены точки выше верхней линии, $s \geqslant \bar{s}_m(\varepsilon)$, и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

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 $s = s_0, \ldots, s_m(\varepsilon)$, затем по всем $s = \bar{s}_m(\varepsilon), \ldots, s_1$.

**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 возвращений 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 $\mathsf{P}[\nu(\bar{X}) \geqslant \varepsilon] < \eta$, 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 вычисли 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

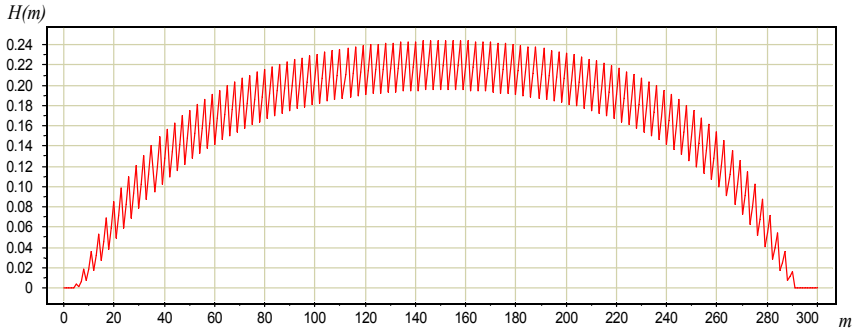



**Рис. 2.4.** Зависимость $H(m) = \mathcal{H}_L^{\ell,m}(s_m(\varepsilon))$ от $m$ при $L = 300,\ \ell = 200,\ \varepsilon = 0.05$.

верхнюю оценку:

$$\mathsf{P}\big[\nu(\bar{X}) - \nu(X) \geqslant \varepsilon\big] \leqslant \max_{m=0,\ldots,L} \mathcal{H}_L^{\ell,m}(s_m(\varepsilon)) \equiv \Gamma_L^\ell(\varepsilon). \tag{2.10}$$

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 [64, 95], с помощью которой можно оценить сверху правую часть (2.10): для любого $\varepsilon > 0$

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

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 $\Gamma_L^\ell(\varepsilon) \to 0$ 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:*

$$n(\mathbb{X}) \leqslant \max\big\{m = m_0, \ldots, L \;\big|\; \mathcal{H}_L^{\ell,m}(s) \geqslant \eta\big\}, \quad \text{где } \ m_0 = \big\lceil s\tfrac{L+2}{\ell+1} - 1 \big\rceil. \tag{2.11}$$

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

$$\mathsf{P}\big[\nu(\bar{X}) \geqslant \varepsilon\big] = \mathcal{H}_L^{\ell,m}(m - \varepsilon k) = H(\varepsilon, m). \tag{2.12}$$

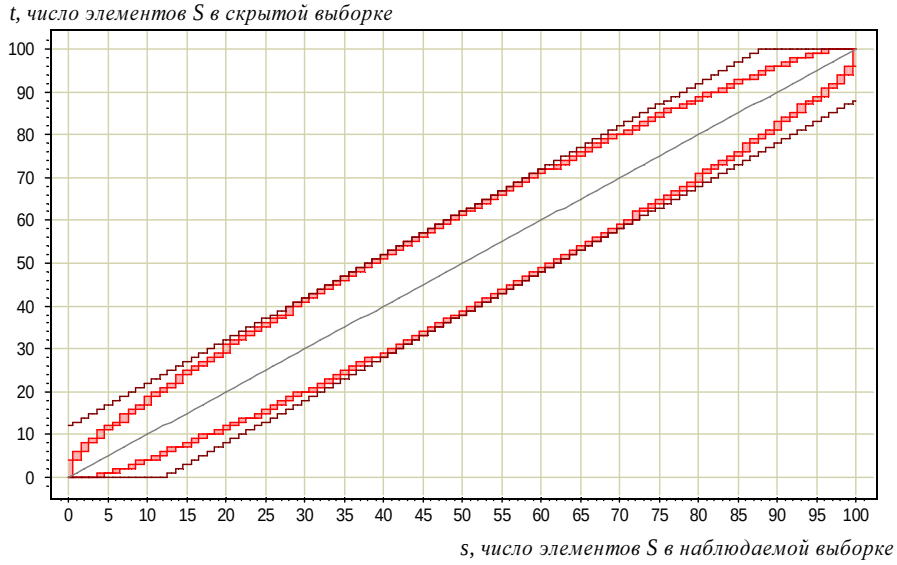



**Рис. 2.5.** Точные верхние и нижние оценки числа $t = n(\bar{X})$ элементов события $S$ в скрытой выборке в зависимости от их числа $s = n(X)$ в наблюдаемой выборке. Условия эксперимента: $L = 200,\ \ell = k = 100,\ \eta = 0.05$.

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 $H(\nu(\bar{X}), m) \geqslant \eta$. 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 $\mathcal{H}_L^{\ell,\,m}(s) \geqslant \eta$. Чтобы разрешить данное неравенство относительно $m$ при фиксированном $s$, найдём максимальное значение $m$, при котором оно выполнено. При максимальном значении $m$ значение $s$ должно находится 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*

$$n(\bar{X}) \leqslant \max\bigl\{t = t_0, \ldots, k \;\bigm|\; \mathcal{H}_L^{\ell,\,s+t}(s) \geqslant \eta\bigr\}, \quad t_0 = \bigl\lceil s\tfrac{k+1}{\ell+1} - 1\bigr\rceil. \tag{2.13}$$

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

$$n(\mathbb{X}) \geqslant \min\bigl\{m = 0, \ldots, m_0 \;\bigm|\; \bar{\mathcal{H}}_L^{\ell,\,m}(s) \geqslant \eta\bigr\};$$
$$n(\bar{X}) \geqslant \min\bigl\{t = 0, \ldots, t_0 \;\bigm|\; \bar{\mathcal{H}}_L^{\ell,\,s+t}(s) \geqslant \eta\bigr\},$$

Вычисление полученных верхних и нижних оценок с использованием рекуррентных соотношений (2.5) требует порядка $O\bigl(n(X)n(\bar{X})\bigr)$ 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 $\mathcal{H}_L^{\ell,\,s+t}(s) = \eta$. 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* [21] — 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:*

$$Q_\varepsilon = \mathsf{P}\big[\nu(a, \bar{X}) - \nu(a, X) \geqslant \varepsilon\big] = \mathcal{H}_L^{\ell,\,m}\left(\tfrac{\ell}{L}(m - \varepsilon k)\right); \tag{2.14}$$

$$R_\varepsilon = \mathsf{P}\big[\nu(a, \bar{X}) \geqslant \varepsilon\big] = \mathcal{H}_L^{\ell,\,m}\big(m - \varepsilon k\big); \tag{2.15}$$

$$C = \mathsf{E}\nu(a, \bar{X}) = \tfrac{m}{L}. \tag{2.16}$$

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

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

■

## 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 слабой аксиоматике возможно получать точные решения для ряда классических задач математической статистики, таких, как построение доверительных интервалов 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 точные оценки $Q_\varepsilon$, $R_\varepsilon$ и $C$ для двухэлементного семейства $A = \{a_1, a_2\}$, если заданы четыре параметра

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

Построить графики зависимости $Q_\varepsilon$, $R_\varepsilon$ и $C$ 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:

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

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 точные оценки $Q_\varepsilon$, $R_\varepsilon$ и $C$ для семейства из трёх алгоритмов $A = \{a_1, a_2, a_3\}$, если заданы восемь параметров

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

Построить графики зависимости $Q_\varepsilon$, $R_\varepsilon$ и $C$ 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 задачи 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 многие *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 $\xi\colon \mathbb{X} \to \mathbb{R}$, значения которой попарно различны на элементах выборки $\mathbb{X}$. Требуется построить по наблюдаемой выборке $X$ семейство вложенных *доверительных интервалов* $\Omega_\varepsilon(X) = \big[\xi_\varepsilon^-(X), \xi_\varepsilon^+(X)\big]$ 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 $U = \{u_1, \ldots, u_t\} \subseteq \mathbb{X}$ называется последовательность значений $\xi(u_1), \ldots, \xi(u_t)$, упорядоченная по возрастанию. Обозначим $s$-е значение вариационного ряда $\xi$ на $U$ через $\xi_U^{(s)}$, тогда $\xi_U^{(1)} < \cdots < \xi_U^{(t)}$.

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

$$\mathsf{P}\big[\xi(\bar{x}) \notin \Omega_\varepsilon(X)\big] = 2\big(1 - \tfrac{\varepsilon}{L}\big), \quad \varepsilon = \lceil \ell/2 \rceil, \ldots, \ell. \tag{3.1}$$

**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:

$$\mathsf{P}\big[\xi(\bar{x}) > \xi_X^{(\varepsilon)}\big] = 1 - \tfrac{\varepsilon}{L}, \quad \varepsilon = \lceil \ell/2 \rceil, \ldots, \ell. \tag{3.2}$$

Полагая в (3.1) $\varepsilon = \ell$, заключаем, что скрытое значение $\xi(\bar{x})$ выходит за пределы диапазона наблюдаемых значений $\big[\xi_X^{(1)}, \xi_X^{(\ell)}\big]$ 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** [24]. 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 *$\alpha$-квантúлью распределения* или квантилью порядка $\alpha$. В частности, $\xi_{\frac{1}{2}}$ есть *медиана распределения*.

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

$$\mathsf{P}\big[\xi_X^{(r)} \leqslant \xi_\alpha \leqslant \xi_X^{(s)}\big] = \sum_{t=r}^{s} C_\ell^t \alpha^t (1-\alpha)^{\ell-t}. \tag{3.3}$$

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

**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 $\mathbb{X} = \{\xi_1, \ldots, \xi_L\}$, состоит из попарно различных значений. Построим её вариационный ряд $\xi_{\mathbb{X}}^{(1)} < \cdots < \xi_{\mathbb{X}}^{(L)}$.

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

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

$$\mathsf{P}\big[\xi_X^{(r)} \leqslant M \leqslant \xi_X^{(s)}\big] = \sum_{t=r}^{s} h_L^{\ell,\,m}(t)\,, \tag{3.4}$$

*где вероятность* $\mathsf{P}$ *понимается в соответствии со слабой аксиоматикой, как доля разбиений генеральной выборки.*

**Доказательство.**

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

Рассмотрим событие $S_m = \big\{\xi_{\mathbb{X}}^{(1)}, \ldots, \xi_{\mathbb{X}}^{(m)}\big\} \subseteq \mathbb{X}$.

Условие $\mathsf{P}\big[\xi_X^{(r)} \leqslant M\big]$ 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

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

Аналогично,

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

Таким образом,

$$\mathsf{P}\big[\xi_X^{(r)} \leqslant M \leqslant \xi_X^{(s)}\big] = \bar{\mathcal{H}}_L^{\ell,\,m}(r) - \bar{\mathcal{H}}_L^{\ell,\,m}(s+1) = \\ = \sum_{t=r}^{s_1} h_L^{\ell,\,m}(t) - \sum_{t=s+1}^{s_1} h_L^{\ell,\,m}(t) = \sum_{t=r}^{s} h_L^{\ell,\,m}(t)\,.$$

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

$$h_L^{\ell,\,m}(t) = C_\ell^t \frac{C_k^{m-t}}{C_L^m} = C_\ell^t \frac{m!}{(m-t)!}\frac{k!}{L!}\frac{(L-m)!}{(k-m+t)!} \to C_\ell^t \frac{m^t (k-m)^{\ell-t}}{L^\ell} = \\ = C_\ell^t \Big(\frac{m}{L}\Big)^t \Big(\frac{k}{L} - \frac{m}{L}\Big)^{\ell-t} \to\ C_\ell^t \alpha^t (1-\alpha)^{\ell-t}.$$

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 связных 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 $T(X) = \sum\limits_{i=1}^{\ell} b_i$. При условии, что гипотеза $H_0$ верна, статистика $T$ подчиняется *биномиальному распределению*:

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

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

$$\frac{1}{2^\ell} \sum_{t=0}^{T(X)} C_\ell^t \;\notin\; \big[\tfrac{\alpha}{2},\, 1 - \tfrac{\alpha}{2}\big]. \tag{3.5}$$

**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 $S = \big\{b_i \in \mathbb{X} \colon b_i = 1\big\}$, приходим к выводу, что статистика $T(X)$ подчиняется *гипергеометрическому распределению*:

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

где вероятность $\mathsf{P}$ 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*:

$$\mathcal{H}_L^{\ell,\,m}\big(T(X)\big) \;\notin\; \big[\tfrac{\alpha}{2},\, 1 - \tfrac{\alpha}{2}\big],$$

**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):

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

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 $X = (x_1, \ldots, x_\ell)$ и $\bar{X} = (x'_1, \ldots, x'_k)$ 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 $H_0\colon \mathsf{P}(x < x') = \frac{1}{2}$. Поэтому он считается не достаточно мощным для проверки гипотезы однородности.

Для простоты в качестве альтернативы рассмотрим $H_1\colon \mathsf{P}(x < x') > \frac{1}{2}$. 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

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

Обычно используется более экономный способ вычисления $U$. Строится вариационный ряд $x^{(1)} < \cdots < x^{(L)}$ объединённой выборки $\mathbb{X} = X \sqcup \bar{X}$. Для простоты будем предполагать, что вариационный ряд не содержит связок. Далее находятся ранги $r(x_i) \in \{1, \ldots, L\}$ 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 $\ell, k > 8$ [43, 4, 32]:

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

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 $\mathsf{P}$ в условии $\mathsf{P}[x < x'] = \frac{1}{2}$ 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 $\mathsf{P}[x < x']$ действительно равна $\frac{1}{2}$:

$$\mathsf{P}[x < x'] = \mathsf{E}\frac{1}{\ell}\sum_{i=1}^{L}\big[x_i \in X\big]\frac{1}{k}\sum_{j=1}^{L}\big[x_j \in \bar{X}\big]\big[x_i < x_j\big] =$$

$$= \frac{1}{\ell k}\sum_{i=1}^{L}\sum_{j=1}^{L}\big[x_i < x_j\big]\underbrace{\mathsf{P}\big[x_i \in X\big]\big[x_j \in \bar{X}\big]}_{C_{L-2}^{\ell-1}/C_L^{\ell}} = \frac{1}{\ell k}\cdot\frac{L(L-1)}{2}\cdot\frac{C_{L-2}^{\ell-1}}{C_L^{\ell}} = \frac{1}{2}.$$

Отсюда следует, что $\mathsf{E}U(X,\bar{X}) = \frac{1}{2}\ell k$. 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 $U(X,\bar{X})$,*

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

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

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

*при начальных условиях*

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

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

Обозначим $b_i = b_i(X) = [x^{(i)} \in X]$. Бинарная последовательность $b_1, \ldots, b_L$ 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 $\sum\limits_{i=1}^{L-1} b_i = \ell$.

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:

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

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 $u \in \{0, \ldots, \ell k\}$.

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 $L \to \infty$ при фиксированной длине наблюдаемой выборки $\ell$ приводит к классическим результатам.

**В следующей лекции** мы рассмотрим ещё одну классическую задачу математической статистики — оценивание 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 $\xi\colon \mathbb{X} \to \mathbb{R}$ и произвольной конечной выборки $U \subseteq \mathbb{X}$ *эмпирическую функцию распределения* $F_\xi\colon \mathbb{R} \to [0,1]$ как долю объектов $x$ выборки $U$, для которых значение $\xi(x)$ не превосходит $z$:

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

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

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

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

$$\mathsf{P}\big[D(X) > \varepsilon\big] \leqslant \eta(\varepsilon), \tag{4.1}$$

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

В классической вероятностной аксиоматике имеет место следующий факт [49].

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

$$\lim_{\ell,k\to\infty} \mathsf{P}\big\{D^{\pm}(X) \geqslant \varepsilon\big\} = \exp\left(-2\varepsilon^2 \tfrac{\ell k}{\ell+k}\right); \tag{4.2}$$

$$\lim_{\ell,k\to\infty} \mathsf{P}\big\{D(X) \geqslant \varepsilon\big\} = 2\sum_{i=1}^{\infty}(-1)^{i+1}\exp\left(-2\varepsilon^2 \tfrac{\ell k}{\ell+k} i^2\right); \tag{4.3}$$

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*:

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

Известны и неасимптотические точные оценки, но они имеют достаточно громоздкий вид [17]. Мы покажем, что точные оценки могут быть выражены более элегантно через усечённый треугольник Паскаля [57], причём доказательство имеет прозрачный геометрический смысл.

## §4.2 Усечённый треугольник Паскаля

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

**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*

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

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 $C_m^s = G_m^s[0, m]$.

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

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

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

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° против часовой стрелки.

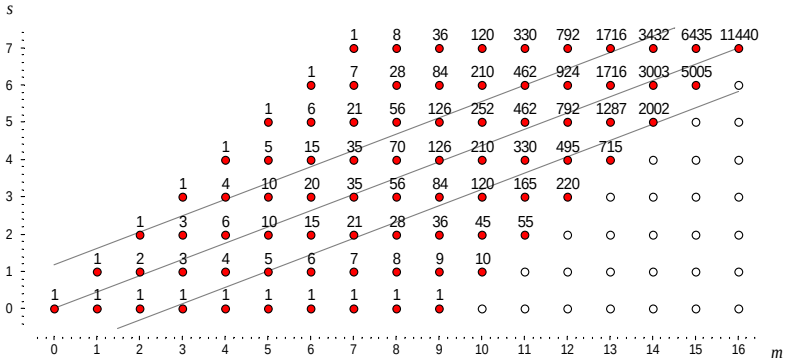



**Рис. 4.1.** Классический треугольник Паскаля $C_m^s = G_m^s[0, m]$ при $L = 16$, $\ell = 7$, $\varepsilon = 0.3$.

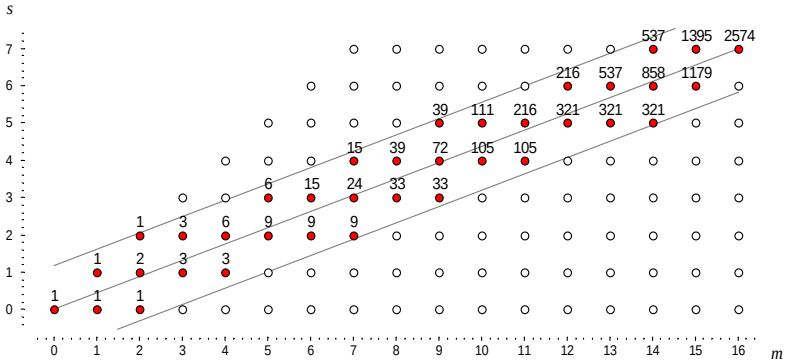



**Рис. 4.2.** Усечённый треугольник Паскаля $G_m^s\big[g_m^-(\varepsilon), g_m^+(\varepsilon)\big]$ при $L = 16$, $\ell = 7$, $\varepsilon = 0.3$.

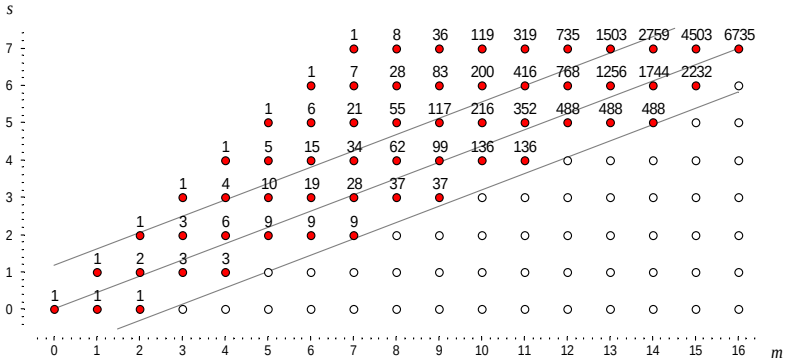



**Рис. 4.3.** Усечённый слева треугольник Паскаля $G_m^s\big[g_m^-(\varepsilon), m\big]$, $L = 16$, $\ell = 7$, $\varepsilon = 0.3$.

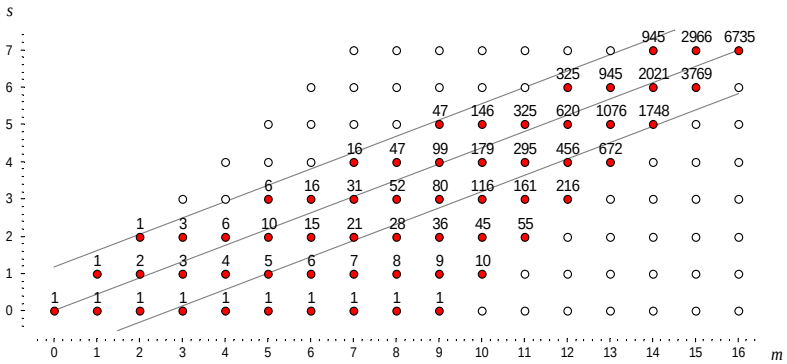



**Рис. 4.4.** Усечённый справа треугольник Паскаля $G_m^s\big[0, g_m^+(\varepsilon)\big]$, $L = 16$, $\ell = 7$, $\varepsilon = 0.3$.

## §4.3 Теорема Смирнова

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

$$\mathsf{P}\big[D^+(X) \leqslant \varepsilon\big] = G_L^\ell\big[0, g_L^+(\varepsilon)\big]/C_L^\ell; \tag{4.5}$$

$$\mathsf{P}\big[D^-(X) \leqslant \varepsilon\big] = G_L^\ell\big[g_L^-(\varepsilon), L\big]/C_L^\ell; \tag{4.6}$$

$$\mathsf{P}\big[D(X) \leqslant \varepsilon\big] = G_L^\ell\big[g_L^-(\varepsilon), g_L^+(\varepsilon)\big]/C_L^\ell. \tag{4.7}$$

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

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

Воспользуемся определением функции распределения:

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

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 $\big[\xi(x^{(i)}) < z\big]$, заменив верхний предел суммирования на $m = \max\big\{i\colon \xi(x^{(i)}) < z\big\}$, 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 $B_m = B_m(X) = b_1 + \cdots + b_m$.

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

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

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

Последнее равенство следует из тождества $\big[\max\limits_m A_m \leqslant 0\big] = \prod\limits_m \big[A_m \leqslant 0\big]$.

2. Рассмотрим подвыборку $X^m = \{x^{(1)}, \ldots, x^{(m)}\}$, 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 $N_0 = \{(X, \bar{X}) \in N \colon b_m(X) = 0\}$ и $N_1 = \{(X, \bar{X}) \in N \colon b_m(X) = 1\}$. Очевидно, $|N_0| = C_{m-1}^s$, $|N_1| = C_{m-1}^{s-1}$.

Нас будет интересовать выражение $H_m^s = \sum\limits_{(X,\bar{X})} \prod\limits_{r=1}^{m} \big[g_r^- \leqslant B_r(X) \leqslant g_r^+\big]$, 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 $(X, \bar{X}) \in N$:

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

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, $H_1^s = \big[g_m^- \leqslant s \leqslant g_m^+\big]$, что совпадает с (4.4).

При $s = 0$ и произвольном $m = 1, \ldots, k$ 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 $r = 1, \ldots, m$. Но тогда $H_m^0 = \prod_{r=1}^m \big[g_r^- \leqslant s \leqslant g_r^+\big]$, что, опять-таки, совпадает с (4.4).

Заметим также, что при $s = 0$ запись $G^{s-1}_{m-1} = 0$ по определению корректна, в то же время $H^{s-1}_{m-1} = 0$, поскольку $N_1 = \varnothing$. Аналогично, при $s = m$ имеем $N_0 = \varnothing$, следовательно, $H^s_{m-1} = 0 = G^s_{m-1}$.

3. Односторонние оценки (4.5) и (4.6) выводится аналогично. Различие в том, что для них выражение под знаком произведения в (4.9) принимает вид, соответственно, либо $\big[0 \leqslant B_m \leqslant g^+_m(\varepsilon)\big]$, либо $\big[g^-_m(\varepsilon) \leqslant B_m \leqslant m\big]$. 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 $b = (b_1, \ldots, b_L)$, 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 $(m, B_m)^L_{m=0}$. Выполнение совокупности условий $\big[g^-_m \leqslant B_m(X) \leqslant g^+_m\big]$ при всех $m = 1, \ldots, L$ 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 $\mathsf{P}\big[D(X) \leqslant \varepsilon\big]$ в точности равен доле таких траекторий. Будем называть их *допустимыми*. Обозначим через $H^s_m$ число допустимых траекторий, проходящих из точки $(0, 0)$ в точку $(m, s)$. Допустимая траектория может придти в $(m, s)$ либо из $(m-1, s-1)$, либо из $(m-1, s)$. Отсюда следует рекуррентная формула для числа допустимых траекторий: $H^s_m = H^s_{m-1} + H^{s-1}_{m-1}$. Однако, если $s \notin \big[g^-_m, g^+_m\big]$, то все такие траектории уже не будут допустимыми, поэтому окончательная формула принимает вид $H^s_m = \big(H^s_{m-1} + H^{s-1}_{m-1}\big)\big[g^-_m \leqslant s \leqslant g^+_m\big]$, что совпадает с определением усечённого треугольника Паскаля: $H^s_m \equiv G^s_m$.

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

$$\varphi^s_m = \frac{m-s}{m}\varphi^s_{m-1} + \frac{s}{m}\varphi^{s-1}_{m-1}.$$

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 [57].

## §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 $\xi(x^{(1)}) < \xi(x^{(2)}) < \cdots < \xi(x^{(L)})$ 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 изменяются только границы усечения $[g_m^-, g_m^+]$. Следующая теорема обобщает критерий Смирнова на случай дискретных распределений.

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

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

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

$$\tilde g_m^+(\varepsilon) = \min\bigl\{g_{i_{h-1}}^+(\varepsilon) + m - i_{h-1},\ g_{i_h}^+(\varepsilon)\bigr\};$$
$$\tilde g_m^-(\varepsilon) = \max\bigl\{g_{i_{h-1}}^-(\varepsilon),\ g_{i_h}^-(\varepsilon) + m - i_h\bigr\};$$

*для всех $m = i_{h-1}{+}1, \ldots, i_h$, 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 $\xi(x_i)$:

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

Максимум достаточно брать не по всем $z \in \mathbb{R}$, а лишь по конечному множеству значений, которые функция $\xi$ принимает на выборке, $z \in \bigl\{\xi(x^{(1)}), \ldots, \xi(x^{(H)})\bigr\}$. Уберём сомножитель $\bigl[\xi(x^{(i)}) < z\bigr]$, заменив верхний предел суммирования $L$ на $m = \max\bigl\{i\colon \xi(x^{(i)}) < z\bigr\}$. 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 $I_H = \{i_1, \ldots, i_H\}$:

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

Аналогично (4.9), получаем:

$$\mathsf{P}\bigl[D(X) \leqslant \varepsilon\bigr] = \mathsf{P} \prod_{m\in I_H} \bigl[g_m^-(\varepsilon) \leqslant B_m \leqslant g_m^+(\varepsilon)\bigr].$$

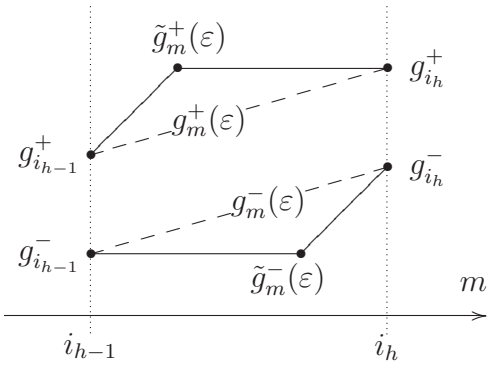



**Рис. 4.5.** Верхние $\tilde g^+_m(\varepsilon)$ и нижние $\tilde g^+_m(\varepsilon)$ границы усечённого треугольника Паскаля в сравнении с линейными границами $g^-_m(\varepsilon)$ и $g^+_m(\varepsilon)$ на отрезке $m = \{i_{h-1},\dots,i_h\}$, соответствующем $h$-й связке.

Единственное отличие от (4.9) заключается в том, что прохождение допустимых траекторий $\big(m, B_m\big)_{m=0}^{L}$ ограничено сверху $g^+_m(\varepsilon)$ и снизу $g^-_m(\varepsilon)$ 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 $\big(i_{h-1}, \lfloor g^+_{i_{h-1}} \rfloor\big)$ и $\big(i_h, \lfloor g^+_{i_h} \rfloor\big)$ допустимая траектория может идти произвольным образом, следовательно, её путь ограничен сверху горизонтальной прямой $B_m \leqslant g^+_{i_h}$ и наклонной прямой $B_m \leqslant g^+_{i_{h-1}} + (m - i_{h-1})$.

Между точками нижней границы $\big(i_{h-1}, \lceil g^-_{i_{h-1}} \rceil\big)$ и $\big(i_h, \lceil g^-_{i_h} \rceil\big)$ допустимая траектория может идти произвольным образом, следовательно, её путь ограничен снизу горизонтальной прямой $B_m \geqslant g^-_{i_{h-1}}$ и наклонной прямой $B_m \geqslant g^-_{i_h} + (m - i_h)$.

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 $\tilde g^+_m(\varepsilon) = g^+_m(\varepsilon)$, $\tilde g^-_m(\varepsilon) = g^-_m(\varepsilon)$, 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 11 мы снова вспомним про критерий Смирнова и покажем, что для одного частного семейства алгоритмов вероятность большого равномерного отклонения также 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 [9, 10, 11, 8]. В середине 80-х она получила широкую мировую известность [105, 106, 107] и вместе с работами Валианта [104] 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 $\mathbb{X} = \{x_1, \ldots, x_L\}$ — конечное множество объектов, $A$ — множество алгоритмов, $I(a,x)$ — индикатор ошибки. Каждый алгоритм $a \in A$ порождает на $\mathbb{X}$ вектор ошибок $\vec{a} = \big(I(a,x_i)\big)_{i=1}^{L}$. Введём ещё несколько понятий и обозначений.

$\vec{A} = \big\{\vec{a} \colon a \in A\big\}$ — 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)[4] 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 $A$.

$\Delta^A(L) = \max\limits_{\mathbb{X}} \Delta^A(\mathbb{X})$ — *функция роста* (growth function) множества алгоритмов $A$ [11, 107]. Максимум берётся по всевозможным выборкам $\mathbb{X} \subset \mathscr{X}$ длины $L$ из некоторого (как правило, бесконечного) множества допустимых объектов $\mathscr{X}$. 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 $\Delta^A(L) \leqslant 2^L$.

**Принцип равномерной сходимости** вводится в VC-теории и многих последующих работах (см. обзоры [108, 59, 13]), 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 $\widetilde{Q}_\varepsilon$ — вероятностью большого равномерного отклонения

[4] В работах В. Н. Вапника и А. Я. Червоненкиса [10, 11, 8] коэффициент разнообразия назывался *индексом системы событий*. Алгоритм $a$ индуцирует событие $S_a = \big\{x \in \mathbb{X} \;\big|\; I(a,x) = 1\big\}$. Семейство $A$ индуцирует систему событий $S = \{S_a | \, a \in A\}$. 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» [44].

частот в двух подвыборках:

$$Q_\varepsilon \leqslant \widetilde{Q}_\varepsilon = \mathsf{P}\Big[\max_{\vec{a}\in\vec{A}} \delta(a,X) \geqslant \varepsilon\Big] = \mathsf{P}\max_{\vec{a}\in\vec{A}}\big[\delta(a,X) \geqslant \varepsilon\big], \tag{5.1}$$

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 $\widetilde{Q}_\varepsilon \to 0$ при $\ell, k \to \infty$ то переобученность также сходится к нулю: $\delta(a,X) \to 0$. Термин «*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 [10, 11, 8].

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

$$Q_\varepsilon \leqslant \Delta^A(\mathbb{X}) \max_{m=1,\ldots,L} \mathcal{H}_L^{\ell,m}\left(s_m(\varepsilon)\right), \quad s_m(\varepsilon) = \tfrac{\ell}{L}(m - \varepsilon k). \tag{5.2}$$

**Доказательство.** Воспользуемся принципом равномерной сходимости (5.1) и оценим сверху максимум бинарных величин их суммой[5]:

$$Q_\varepsilon \leqslant \widetilde{Q}_\varepsilon \leqslant \mathsf{P}\sum_{\vec{a}\in\vec{A}}\big[\delta(a,X) \geqslant \varepsilon\big].$$

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

$$Q_\varepsilon \leqslant \widetilde{Q}_\varepsilon \leqslant \widetilde{\widetilde{Q}}_\varepsilon = \sum_{\vec{a}\in\vec{A}} \mathsf{P}\big[\delta(a,X) \geqslant \varepsilon\big] = \sum_{\vec{a}\in\vec{A}} \mathcal{H}_L^{\ell,m}\left(s_m(\varepsilon)\right), \quad m = n(a,\mathbb{X}). \tag{5.3}$$

Оценим $\mathcal{H}_L^{\ell,m}\left(s_m(\varepsilon)\right)$ 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*

$$R_\varepsilon \leqslant \sum_{\vec{a}\in\vec{A}} \mathcal{H}_L^{\ell,m}\left(m - \varepsilon k\right) \;\leqslant\; \Delta^A(\mathbb{X}) \max_{m=1,\ldots,L} \mathcal{H}_L^{\ell,m}\left(m - \varepsilon k\right).$$

[5] Это можно трактовать и как оценку вероятности объединения событий $\big[\delta(a,X) \geqslant \varepsilon\big]$ 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 $\mathcal{H}_L^{\ell,m}(s_m(\varepsilon))$ 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 $\ell = k$ [8]:

$$\max_m \mathcal{H}_L^{\ell,m}(s_m(\varepsilon)) \leqslant \tfrac{3}{2}e^{-\varepsilon^2\ell}, \quad \ell = k. \tag{5.4}$$

В итоге получается наиболее известная из VC-оценок [107]:

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

$$Q_\varepsilon \leqslant \Delta^A(2\ell) \cdot \tfrac{3}{2}e^{-\varepsilon^2\ell}. \tag{5.5}$$

**Корректные методы обучения.** В VC-теории [8] отдельно рассматривается так называемая *детерминистская постановка задачи*[6], 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 $\mathbb{X}$, $\varepsilon \in [0,1]$*

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

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

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

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

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

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

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

[6] 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 нереализуемым 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 $\ell = k$*

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

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 $\Delta^A(L) \leqslant 2^L$ справедлива более точная полиномиальная оценка

$$\Delta^A(L) \leqslant C_L^0 + C_L^1 + \cdots + C_L^h \leqslant \frac{3}{2}\frac{L^h}{h!}. \tag{5.7}$$

В этом случае вероятность переобучения стремится к нулю при $\ell \to \infty$, 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 $A = \{a(x) = \operatorname{sign}\langle w, x\rangle \mid w \in \mathbb{R}^n\}$ 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 $\Delta^A(L) \leqslant |A|$. 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 [23].

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*

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

**Доказательство.** Подставим в (5.5) верхнюю оценку функции роста (6.1), оценив $h!$ снизу по формуле Стирлинга. Полученная оценка имеет вид $Q_\varepsilon \leqslant \eta(\varepsilon, \ell, h)$. 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 $A_1 \subset A_2 \subset \cdots \subset A_h = A$. 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 $\mathbb{X}$ 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 [59].

## §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 сомножителя (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 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[7].

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.

---

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

| | $\eta = 0.01$ | | | | $\eta = 1$ | | | |
|---|---|---|---|---|---|---|---|---|
| $h$ | $\varepsilon = 0.01$ | $\varepsilon = 0.05$ | $\varepsilon = 0.1$ | $\varepsilon = 0.2$ | $\varepsilon = 0.01$ | $\varepsilon = 0.05$ | $\varepsilon = 0.1$ | $\varepsilon = 0.2$ |
| 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 $Q_\varepsilon \leqslant \eta = \frac{3}{2}\frac{L^h}{h!} \cdot \frac{3}{2}\exp(-\varepsilon^2\ell)$ из Теоремы 5.1. Это наименее точная оценка, использующая аппроксимацию функции роста (6.1) и аппроксимацию гипергеометрического сомножителя (5.4).

| | $\eta = 0.01$ | | | | $\eta = 1$ | | | |
|---|---|---|---|---|---|---|---|---|
| $h$ | $\varepsilon = 0.01$ | $\varepsilon = 0.05$ | $\varepsilon = 0.1$ | $\varepsilon = 0.2$ | $\varepsilon = 0.01$ | $\varepsilon = 0.05$ | $\varepsilon = 0.1$ | $\varepsilon = 0.2$ |
| 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 $Q_\varepsilon \leqslant \eta = (C_L^0 + \cdots + C_L^h)\Gamma_L^\ell(\varepsilon, 1)$. Это также оценка Теоремы 5.1, но функция роста и гипергеометрический сомножитель вычисляются по точным формулам, без применения аппроксимаций.

| | $\eta = 0.01$ | | | | $\eta = 1$ | | | |
|---|---|---|---|---|---|---|---|---|
| $h$ | $\varepsilon = 0.01$ | $\varepsilon = 0.05$ | $\varepsilon = 0.1$ | $\varepsilon = 0.2$ | $\varepsilon = 0.01$ | $\varepsilon = 0.05$ | $\varepsilon = 0.1$ | $\varepsilon = 0.2$ |
| 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 $Q_\varepsilon \leqslant \eta = (C_L^0 + \cdots + C_L^h)C_{L-\lceil\varepsilon k\rceil}^\ell / C_L^\ell$ 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 $\leqslant$, был проведён эксперимент [109] 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 $\leqslant$ в цепочке неравенств. Произведение факторов завышенности даёт общую завышенность VC-оценки:

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

**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 $A_L^\ell = \{\mu X \colon X \subset \mathbb{X},\ |X| = \ell\}$, 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 $P(a) = \mathsf{P}[\mu X = a]$. 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 наихудшей 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

$$\mathsf{P} \max_a [\delta(a, X) \geqslant \varepsilon] \leqslant \sum_a \mathsf{P}[\delta(a, X) \geqslant \varepsilon]$$

| Задача | класс | объектов | $r_1$ | $r_2$ | $r_3$ | $\hat{\Delta}$ | $[\hat{\Delta}_1; \hat{\Delta}_2]$ |
|---|---|---|---|---|---|---|---|
| crx | 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 $[\hat{\Delta}_1; \hat{\Delta}_2]$.

обращается в равенство только когда события $[\delta(a, X) \geqslant \varepsilon]$ 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 [101]. 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 $10^0 \ldots 10^2$, тогда как функция роста находится в пределах $10^5 \ldots 10^{10}$, 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 существенен.

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 [76]. 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* [68] 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 $\hat{\mathsf{E}}\nu(a, \bar{X})$. 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: $Q_\varepsilon = \widetilde{Q}_\varepsilon$.*

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

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

■

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: $\sigma(\mu, \mathbb{X}) = \max\limits_{X \in [\mathbb{X}]^\ell} \nu(\mu X, X)$. Доказать, что для любых $\mu$, $\mathbb{X}$ с ограниченной некорректностью $\sigma(\mu, \mathbb{X}) \leqslant \sigma$ и любого $\varepsilon \in [0,1]$ справедлива оценка, обобщающая (5.2) и (5.6):

$$Q_\varepsilon \leqslant \Delta^A(\mathbb{X}) \max_{m \in M(\varepsilon,\sigma)} \mathcal{H}_L^{\ell,m}\left(s_m(\varepsilon,\sigma)\right), \tag{5.9}$$

где $M(\varepsilon,\sigma) = \{m\colon \varepsilon k \leqslant m \leqslant k + \sigma\ell\}$, $s_m(\varepsilon,\sigma) = \min\bigl\{s_m(\varepsilon), \sigma\ell\bigr\}$.

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

**Задача 5.3** (2). Следуя [58], показать, что если множество векторов ошибок $\vec{A}$ кластеризуется по расстоянию Хэмминга на $S(r)$ кластеров радиуса $r$ каждый, то

$$\mathsf{P}\bigl[\delta_\mu(X) \geqslant \varepsilon + \tfrac{r}{\ell}\bigr] \leqslant S(r) \cdot \max_{m=1,\ldots,L} \mathcal{H}_L^{\ell,m}\left(s_m(\varepsilon)\right).$$

# 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 $L \to \infty$, следовательно, обучение асимптотически состоятельно. Разберём этот фундаментальный результат более подробно.

## §6.1 Связь ёмкости с функцией роста

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

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

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 [8]).** *Если для любой подвыборки $X^{h+1}$ из $\mathbb{X}$ выполняется $\Delta^A(X^{h+1}) < 2^{h+1}$, то $\Delta^A(\mathbb{X}) \leqslant \Phi_L^h$.*

**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 $\Delta^A(X^{h+1}) \leqslant 2^{h+1} - 1 = \Phi_{h+1}^h$, 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 $U' = \{x \in U \colon a(x) = 1\}$. 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

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

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 $\Delta^{A'}(\mathbb{X}) \leqslant \Phi_L^{h-1}$. Таким образом

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

Утверждение индукции доказано для $L+1$. ■

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

Доказательство является несложным техническим упражнением [8].

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

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

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

Теорема доказана. ■

## §6.2 Функция роста множества конъюнкций

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

$$A = \Big\{ a(x) = \bigwedge_{j \in J} [f_j(x) = d_j] \;\Big|\; J \subseteq \{1, \ldots, n\},\ |J| \leqslant K,\ d_j \in D_j \Big\}.$$

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 $D_j$ равномощны, $|D_j| = d$, то $H_k(J) = C^k_{|J|} d^k$. В общем случае величина $H_k(J)$ легко вычисляется по рекуррентным соотношениям:

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

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

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

## §6.3 Ёмкость семейства линейных классификаторов

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

$$A = \big\{a(x) = [\langle w, x\rangle \geqslant 0] \;\big|\; w \in \mathbb{R}^n\big\},$$

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 $a(x_i) = [w_i \geqslant 0] = z_i$. 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

$$\forall X^{n+1} \quad \exists (z_1, \ldots, z_{n+1}) \in Y^{n+1} \quad \forall a \in A \quad \exists i = 1, \ldots, n+1 \quad a(x_i) \neq z_i.$$

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

$$x_{n+1} = b_1 x_1 + \cdots + b_n x_n, \tag{6.2}$$

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

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

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

Левая часть этого равенства строго меньше нуля, поскольку

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

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

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

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 [106].

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

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

Возьмём конкретную выборку объектов $x_i = 10^{-i}$, $i = 1, \ldots, \ell$. 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

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

Из определения величины $\gamma_0$ следует, что $\pi 10^{-\ell} \leqslant \gamma_0 \leqslant 0.3\pi$, 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 [55, 54, 75, 87], решающих деревьев [25], корректных полиномов над алгоритмами вычисления оценок [38], 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 [37, 38, 39, 40].

## 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 мы покажем, что они сильно завышены даже в самом благоприятном случае, когда ёмкость семейства равна единице.

**В следующей лекции** мы рассмотрим базовую технику комбинаторной теории переобучения и убедимся, что для получения точных оценок мало знать коэффициент разнообразия, то есть чисто попарно различных векторов ошибок в семействе. Необходимо также учитывать степень их различности.

# 7 Порождающие и запрещающие множества

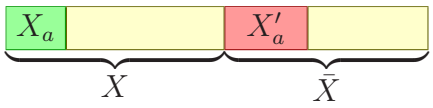
*Принцип порождающих и запрещающих множеств* (ПЗМ) позволяет получать точные (не завышенные, не асимптотические) оценки обобщающей способности [111]. Он основан на предположении, что для каждого алгоритма можно выписать необходимые и достаточные условия того, что он является результатом обучения. Если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

## §7.1 Простая гипотеза ПЗМ

**Гипотеза 7.1.** *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*

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

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:

$\mu X = a$: 


Все остальные объекты $\mathbb{X}\backslash X_a \backslash X'_a$ 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*

$$\sum_{a \in A} \big[\mu X{=}a\big] = 1. \tag{7.2}$$

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*

$$P_a = \mathsf{P}\big[\mu X{=}a\big] = \frac{C_{L_a}^{\ell_a}}{C_L^\ell}. \tag{7.3}$$

**Доказательство.** Согласно гипотезе 7.1

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

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 $X\backslash X_a$, 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 $\mathsf{P}\big(\delta(a,X) \geqslant \varepsilon \mid a\big)$ при условии, что получен алгоритм $a$:

$$Q_\varepsilon = \sum_{a\in A} P_a\, \mathsf{P}\big(\delta(a,X) \geqslant \varepsilon \;\big|\; a\big).$$

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:

$$\mathsf{P}\big(\delta(a,X) \geqslant \varepsilon \;\big|\; a\big) = \mathcal{H}_{L_a}^{\ell_a,\, m_a}\left(s_a(\varepsilon)\right),$$

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, $\delta(a,X) \geqslant \varepsilon$:

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

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*

$$Q_\varepsilon(\mu,\mathbb{X}) = \sum_{a\in A} P_a \mathcal{H}_{L_a}^{\ell_a,\, m_a}\left(s_a(\varepsilon)\right). \tag{7.4}$$

**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 $X\backslash X_a$. Очевидно, значение $Q_\varepsilon$ от этого не изменится:

$$Q_\varepsilon = \mathsf{P}\big[\delta_\mu(X) \geqslant \varepsilon\big] = \mathsf{P}\sum_{a\in A}\big[\mu X{=}a\big]\sum_{s=0}^{\ell_a}\big[n(a,X\backslash X_a) = s\big]\big[\delta(a,X) \geqslant \varepsilon\big]. \tag{7.5}$$

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,

$$\big[\delta(a, X) \geqslant \varepsilon\big] = \big[s \leqslant \tfrac{\ell}{L}\big(n(a, \mathbb{X}) - \varepsilon k\big) - n(a, X_a)\big] = \big[s \leqslant s_a(\varepsilon)\big].$$

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

$$Q_\varepsilon = \sum_{a \in A} \sum_{s=0}^{\ell_a} \underbrace{\mathsf{P}\big[X_a \subseteq X\big]\big[X'_a \subseteq \bar{X}\big]\big[n(a, X \backslash X_a) = s\big]}_{N(a)} \big[s \leqslant s_a(\varepsilon)\big]. \tag{7.6}$$

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 $X \backslash X_a$ длины $\ell_a$ попадает ровно $s$ объектов, на которых алгоритм $a$ допускает ошибку.

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

$$\vec{a} = \Big( \underbrace{X_a; \overbrace{1, \ldots, 1}^{s}; 0, \ldots, 0}_{X \backslash X_a}; \underbrace{X'_a; \overbrace{1, \ldots, 1}^{m_a - s}; 0, \ldots, 0}_{\bar{X} \backslash X'_a} \Big).$$

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

$$Q_\varepsilon = \sum_{a \in A} \frac{C^{\ell_a}_{L_a}}{C^\ell_L} \sum_{s=s_0}^{\ell_a} \big[s \leqslant s_a(\varepsilon)\big] \frac{C^s_{m_a} C^{\ell_a - s}_{L_a - m_a}}{C^{\ell_a}_{L_a}} = \sum_{a \in A} P_a \mathcal{H}^{\ell_a,\, m_a}_{L_a}\big(s_a(\varepsilon)\big).$$

Теорема доказана. ■

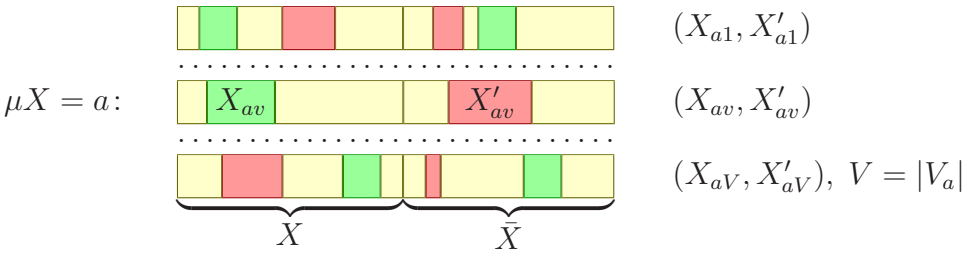
## §7.2 Обобщённая гипотеза ПЗМ

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.** *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*

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

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 $\big[X_{av} \subseteq X\big]\big[X'_{av} \subseteq \bar{X}\big]$ 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):

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

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:

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

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*

$$P_a = \mathsf{P}\big[\mu X{=}a\big] = \sum_{v \in V_a} c_{av} P_{av}; \tag{7.8}$$

$$P_{av} = \mathsf{P}\big[X_{av} \subseteq X\big]\big[X'_{av} \subseteq \bar{X}\big] = \frac{C_{L_{av}}^{\ell_{av}}}{C_L^{\ell}}. \tag{7.9}$$

**Доказательство.** Достаточно применить операцию $\mathsf{P}$ 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*

$$Q_\varepsilon(\mu, \mathbb{X}) = \sum_{a \in A} \sum_{v \in V_a} c_{av} P_{av} \mathcal{H}_{L_{av}}^{\ell_{av},\, m_{av}}\left(s_{av}(\varepsilon)\right). \tag{7.10}$$

**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 $_a$:

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

В остальном доказательство аналогично доказательству Теоремы 7.3. ■

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

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

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:* $\mathcal{H}_L^{\ell,\,0}(s) = [s \geqslant 0]$.

**Доказательство.** При $s \geqslant 0$ сумма $\mathcal{H}_L^{\ell,\,0}(s)$ 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:*

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

**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,

$$\mathcal{H}_{L_{av}}^{\ell_{av},\,0}(s_{av}(\varepsilon)) = \big[s_{av}(\varepsilon) \geqslant 0\big] = \big[\tfrac{\ell}{L}\big(n(a, \mathbb{X}) - \varepsilon k\big) - n(a, X_{av}) \geqslant 0\big] = \big[n(a, \mathbb{X}) \geqslant \varepsilon k\big].$$

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 [16].

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

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

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

$$C(\mu,\mathbb{X}) = \frac{1}{C_L^\ell} \sum_{X\in[\mathbb{X}]^\ell} \sum_{a\in A} \big[\mu X{=}a\big] \frac{1}{k} \sum_{x_i\in\bar{X}} I(a,x_i) =$$

$$= \frac{1}{k} \sum_{a\in A} \sum_{v\in V_a} c_{av} \sum_{i=1}^{L} I(a,x_i) \underbrace{\mathsf{P}\big[X_{av}\subseteq X\big]\big[X'_{av}\subseteq \bar{X}\big]\big[x_i\in\bar{X}\big]}_{p(a,v,x_i)}.$$

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

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

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

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

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

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

Подставляя это выражение в сумму по $i$, получаем

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

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 простых, но уже нетривиальных 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 $X$:

$$A(X) = \operatorname{Arg}\min_{a\in A} n(a, X). \tag{8.1}$$

**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 $\mu X = \arg\min\limits_{a\in A(X)} n(a, \bar{X})$.*

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

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:

**Теорема 8.1.** *Пусть $\mu_{\min}$ — оптимистичный МЭР, $\mu_{\max}$ — пессимистичный МЭР. Тогда для любой выборки $\mathbb{X}$ и любого метода минимизации эмпирического риска $\mu$*

$$Q_\varepsilon(\mu_{\min}, \mathbb{X}) \leqslant Q_\varepsilon(\mu, \mathbb{X}) \leqslant Q_\varepsilon(\mu_{\max}, \mathbb{X}).$$

Наиболее 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:

$$Q_\varepsilon = \mathsf{E}\frac{1}{|A(X)|}\sum_{a\in A(X)}\big[\nu(a,\bar{X}) - \nu(a,X) \geqslant \varepsilon\big]. \tag{8.2}$$

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 $\mathbb{X} = \{x_1, \ldots, x_L\}$ и множеством алгоритмов $A = \{a_1, \ldots, a_D\}$ 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. Тем не менее, изучение модельных семейств представляет интерес по следующим причинам.

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, и могут использоваться как замена реальных семейств при оценивании переобучения [7].

**Эмпирическое оценивание вероятности переобучения по матрице ошибок.** Идея заключается в том, чтобы оценить вероятность переобучения по некоторому подмножеству разбиений генеральной выборки $\mathbb{X} = X_n \sqcup \bar{X}_n$, $n = 1, \ldots, N$:

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

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

**Вход:** матрица ошибок $A = \{a_1, \ldots, a_D\}$; число разбиений $N$; порог $\varepsilon$;
**Выход:** верхняя оценка $\bar{Q}_\varepsilon$, нижняя оценка $\underline{Q}_\varepsilon$.

$\bar{Q}_\varepsilon := 0$; $\underline{Q}_\varepsilon := 0$;
**для всех** разбиений $X_n \sqcup \bar{X}_n = \mathbb{X}$, $n = 1, \ldots, N$
  $A(X_n) := \operatorname{Arg}\min_{a \in A} \nu(a, X_n)$;
  $\bar{a} := \arg\max_{a \in A(X_n)} \nu(a, \bar{X}_n)$; $\bar{Q}_\varepsilon := \bar{Q}_\varepsilon + \frac{1}{N}\big[\delta(\bar{a}, X_n) \geqslant \varepsilon\big]$;
  $\underline{a} := \arg\min_{a \in A(X_n)} \nu(a, \bar{X}_n)$; $\underline{Q}_\varepsilon := \underline{Q}_\varepsilon + \frac{1}{N}\big[\delta(\underline{a}, X_n) \geqslant \varepsilon\big]$;
**вернуть** $\bar{Q}_\varepsilon$, $\underline{Q}_\varepsilon$.

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

**Вход:** матрица ошибок $A = \{a_1, \ldots, a_D\}$; число разбиений $N$; порог $\varepsilon$;
**Выход:** верхняя оценка $\bar{Q}_\varepsilon(d)$ и нижняя оценка $\underline{Q}_\varepsilon(d)$ для подмножества $\{a_1, \ldots, a_d\}$.

сгенерировать и зафиксировать $N$ разбиений:
  $X_n \sqcup \bar{X}_n = \mathbb{X}$, $n = 1, \ldots, N$;
алгоритм, выбранный при $n$-м разбиении пессимистичной и оптимистичной МЭР:
  $\bar{a}_n$, $\underline{a}_n := a_1$, $n = 1, \ldots, N$;
$\bar{Q}_\varepsilon(1)$, $\underline{Q}_\varepsilon(1) := \frac{1}{N}\sum_{n=1}^{N}\big[\delta(a_1, X_n) \geqslant \varepsilon\big]$;
**для всех** $d := 2, \ldots, D$
  **для всех** $n := 1, \ldots, N$
    **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**
      $\bar{Q}_\varepsilon(d) := \bar{Q}_\varepsilon(d-1) - \frac{1}{N}\big[\delta(\bar{a}_n, X_n) \geqslant \varepsilon\big] + \frac{1}{N}\big[\delta(a_d, X_n) \geqslant \varepsilon\big]$;
      $\bar{a}_n := a_d$;
    **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**
      $\underline{Q}_\varepsilon(d) := \underline{Q}_\varepsilon(d-1) - \frac{1}{N}\big[\delta(\underline{a}_n, X_n) \geqslant \varepsilon\big] + \frac{1}{N}\big[\delta(a_d, X_n) \geqslant \varepsilon\big]$;
      $\underline{a}_n := a_d$;

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 вычисляет две оценки вероятности переобучения: верхняя $\bar{Q}_\varepsilon$ соответствует пессимистичной МЭР, нижняя $\underline{Q}_\varepsilon$ — оптимистичной. Алгоритм состоит из двух вложенных циклов: внешний перебирает разбиения, внутренний выбирает из $A$ алгоритм, минимизирующий эмпирический риск при данном разбиении, точнее, два алгоритма — 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 $\{a_1, \ldots, a_d\}$. Предполагается, что алгоритмы упорядочены по числу ошибок на генеральной выборке, $n(a_1, \mathbb{X}) \leqslant \ldots \leqslant n(a_D, \mathbb{X})$.

## §8.3 Связные семейства алгоритмов

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

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

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

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

**Определение 8.7.** *Цепь $A = \{a_0, \ldots, a_D\}$ 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: $D \leqslant L$. Любая подпоследовательность прямой цепи $a_u, \ldots, a_v$, $0 \leqslant u < v \leqslant D$, также является прямой цепью.

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

В монотонной цепи $I(a_{d-1}, x_i) \leqslant I(a_d, x_i)$ для всех $x_i \in \mathbb{X}$, $d = 1, \ldots, D$.
Монотонная цепь является прямой. Отсюда следует ограничение $D \leqslant L - m$.
В случае монотонной цепи объекты выборки можно перенумеровать так, чтобы

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

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

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

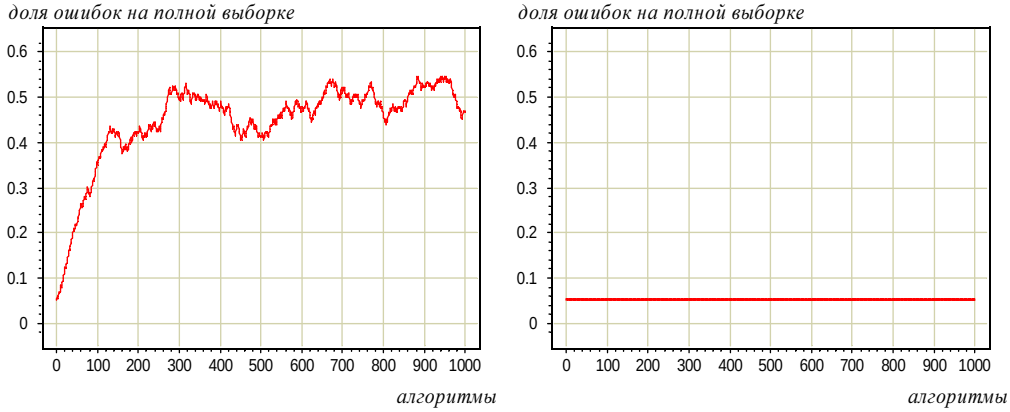



**Рис. 8.1.** Зависимость $\nu(a_d, \mathbb{X})$ от номера алгоритма $d$ в цепи случайных инверсий с расслоением (слева) и без расслоения (справа), при $\ell = k = 100$, $m = 10$.

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 $I(a, x) = \big[a(x, w) \neq y(x)\big]$, где $y(x)$ — истинная классификация объекта $x$, 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

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

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 разделимости.

**Эксперимент с четырьмя модельными семействами.** Цель эксперимента — выяснить, как расслоение и связность влияют на переобучение, какой из двух эффектов важнее или же важно их сочетание. Для этого строятся четыре модельных семейства с одинаковым лучшим алгоритмом $a_0$, который допускает $m$ ошибок на полной выборке. Каждое из четырёх семейств задаётся непосредственно своей матрицей ошибок размера $L \times D$.

1. *Цепь случайных инверсий с расслоением*. Каждый следующий вектор ошибок $a_d$ получается из предыдущего $a_{d-1}$ путём инверсии одной случайно выбранной координаты. Если цепь достаточно длинная, $D \gg L$, то большинство алгоритмов будут допускать около $L/2$ ошибок, рис. 8.1.

2. *Цепь случайных инверсий без расслоения*. Каждый вектор ошибок $a_d$ получается из $a_{d-1}$ также путём одной случайной инверсии, но при нечётных $d$ производятся только инверсии $0 \to 1$, а при чётных $d$ — только $1 \to 0$. В результате число ошибок алгоритмов на полной выборке поочерёдно принимает значения $m$ и $m + 1$, рис. 8.1.

3. *Не-цепь с расслоением* строится по цепи с расслоением путём случайного перемешивания нулей и единиц в каждом вектор-столбце матрицы ошибок $a_d$. Частоты ошибок алгоритмов $\nu(a_d, \mathbb{X})$ при этом не изменяются, но все алгоритмы становятся существенно различными.

4. *Не-цепь без расслоения* строится по цепи без расслоения аналогично.

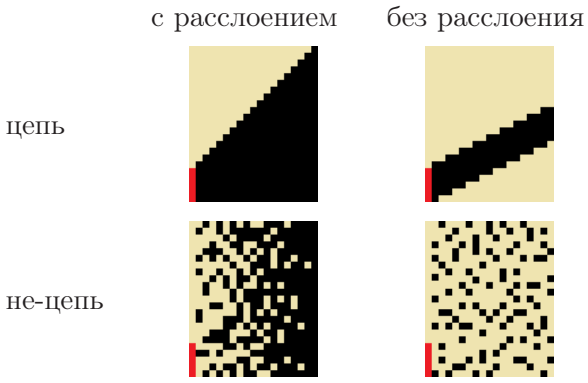



**Рис. 8.2.** Пример матриц ошибок четырёх модельных семейств, когда вместо цепей случайных инверсий берутся прямые цепи.

Вместо цепей случайных инверсий можно было бы взять и прямые цепи, рис. 8.2. Однако их использование накладывает ограничение на число алгоритмов $D \leqslant L$. Кроме того, цепь случайных инверсий представляется более реалистичной моделью однопараметрического семейства алгоритмов.

Сопоставление четырёх модельных семейств позволяет разделить влияние *связности* и *расслоения* на вероятность переобучения.

На рис. 8.3 и рис. 8.4 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, +Р — с расслоением, −Р — 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 семейства $\{a_0, \ldots, a_D\}$ к вероятности переобучения одноэлементного подсемейства $\{a_0\}$.

## Основные выводы.

1. Для семейств без расслоения и связности переобучение может оказаться значительным уже при нескольких десятках алгоритмов в семействе.

2. При больших $D$ только одновременное наличие расслоения и связности позволяет избежать сильного переобучения (нижние кривые на графиках). Таким образом, важно именно сочетание свойств расслоения и связности.

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

4. 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. 8.3 и 8.4.

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

6. Зависимости $Q_\varepsilon(D)$ и $\hat{\Delta}(D)$ имеют горизонтальную асимптоту. В то же время, VC-оценка линейна по $D$, на графиках ЭЛКР ей 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.

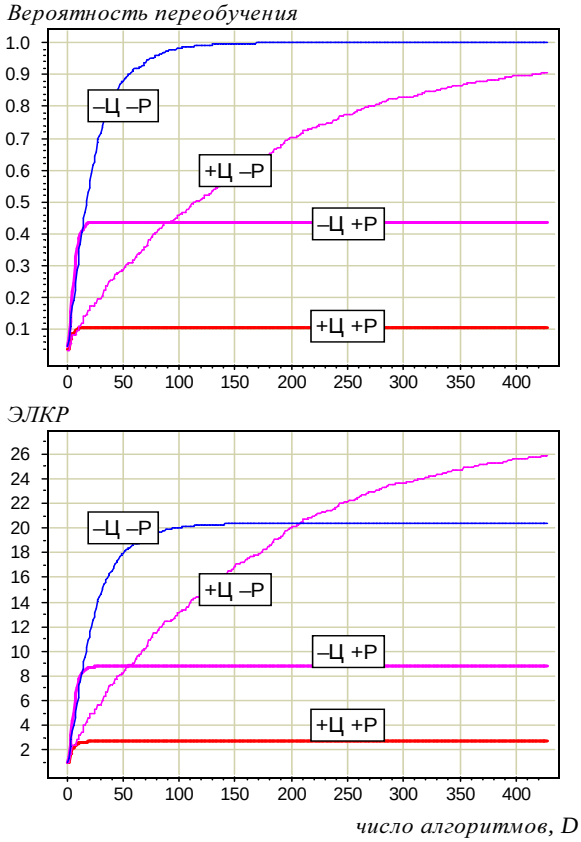



**Рис. 8.3.** Зависимость вероятности переобучения $Q_\varepsilon$ и ЭЛКР $\hat{\Delta}$ от числа алгоритмов $D$. Простая задача: $\nu(a_0, \mathbb{X}) = 0.05$.

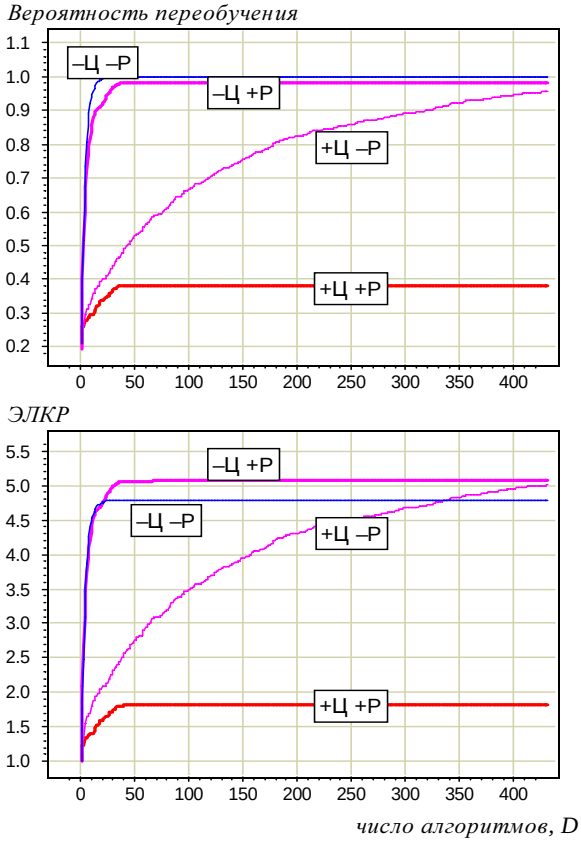



**Рис. 8.4.** Зависимость вероятности переобучения $Q_\varepsilon$ и ЭЛКР $\hat{\Delta}$ от числа алгоритмов $D$. Трудная задача: $\nu(a_0, \mathbb{X}) = 0.25$.

Эксперимент с четырьмя модельным семействами выявил необходимость совместного учёта свойств расслоения и связности и подтолкнул к теоретическому изучению монотонных цепей алгоритмов. Именно с этого эксперимента началось активное развитие комбинаторной теории переобучения.

## §8.4 Точная оценка для монотонной цепи алгоритмов

Точная оценка вероятности переобучения для монотонной цепи алгоритмов может быть получена с помощью метода порождающих и запрещающих множеств.

**Теорема 8.2.** *Пусть $A = \{a_0, \ldots, a_D\}$ — монотонная цепь алгоритмов, $L \geqslant m + D$, $m = n(a_0, \mathbb{X})$, метод обучения $\mu$ является пессимистичной минимизацией эмпирического риска. Тогда вероятность получить алгоритм $a_d$ в результате обучения равна*

$$P_d = C^{\ell-\delta}_{L-d-\delta} / C^\ell_L, \quad d = 0, \ldots, k;$$

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

$$Q_\varepsilon = \sum_{d=0}^{\min\{k, D\}} P_d \mathcal{H}^{\ell-\delta,\, m}_{L-d-\delta} \left( \tfrac{\ell}{L}(m + d - \varepsilon k) \right). \tag{8.3}$$

**Доказательство.** Перенумеруем объекты таким образом, чтобы каждый из алгоритмов $a_d,\ d = 1, \ldots, D$ допускал ошибку на объектах $x_1, \ldots, x_d$. 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:

$$\begin{array}{rccccccccc}
 & x_1 & x_2 & x_3 & & x_D & & \overbrace{\qquad}^{m} & \\
\vec{a}_0 = ( & 0, & 0, & 0, & \ldots & 0, & 0,\ldots,0, & 1,\ldots,1 & ); \\
\vec{a}_1 = ( & 1, & 0, & 0, & \ldots & 0, & 0,\ldots,0, & 1,\ldots,1 & ); \\
\vec{a}_2 = ( & 1, & 1, & 0, & \ldots & 0, & 0,\ldots,0, & 1,\ldots,1 & ); \\
\vec{a}_3 = ( & 1, & 1, & 1, & \ldots & 0, & 0,\ldots,0, & 1,\ldots,1 & ); \\
\ldots & & & & \ldots & & \ldots & \ldots & \\
\vec{a}_D = ( & 1, & 1, & 1, & \ldots & 1, & 0,\ldots,0, & 1,\ldots,1 & );
\end{array}$$

Зафиксируем алгоритм $a_d$ и рассмотрим три случая.

1. Если $d > k$, то число ошибок алгоритма $a_d$ на множестве объектов $\{x_1, \ldots, x_d\}$ 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 $d \leqslant k$ и $d < D$, то метод $\mu$ выберет алгоритм $a_d$, когда объекты $x_1, \ldots, x_d$ будут находиться в контрольной подвыборке $\bar{X}$, а объект $x_{d+1}$ — в обучающей $X$:

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

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

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

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 $\delta = [d < D]$:

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

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 $C^{\ell-\delta}_{L-d-\delta} = 0$, следовательно, $P_d = 0$.

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

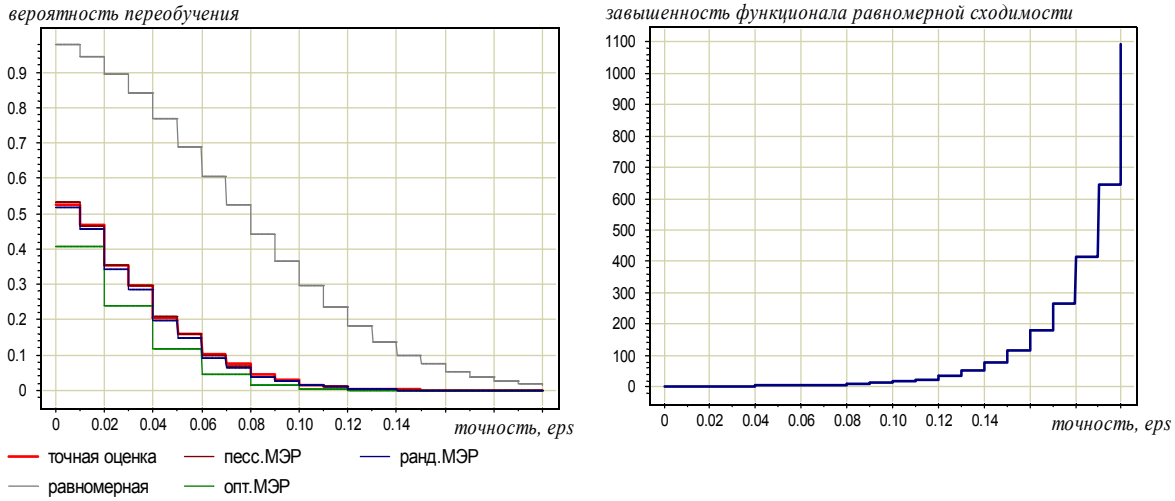



**Рис. 8.5.** Слева: зависимость оценок вероятности переобучения $Q_\varepsilon$ от $\varepsilon$: точная оценка из Теоремы 8.2 и четыре оценки, вычисленные методом Монте-Карло по 1000 случайных разбиений: для пессимистичной, оптимистичной и рандомизированной МЭР. Верхняя кривая соответствует оценке по функционалу равномерной сходимости $\hat{P}_\varepsilon$. Справа: степень завышенности функционала равномерной сходимости $\hat{P}_\varepsilon/Q_\varepsilon$. Все графики построены при $\ell = k = 100,\ m = 20$.

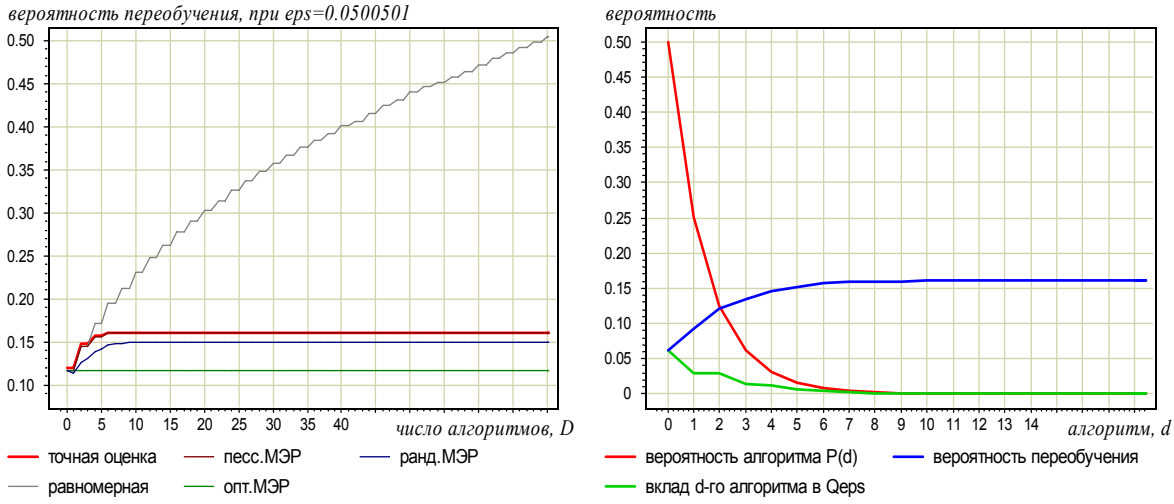



**Рис. 8.6.** Слева: зависимость вероятности переобучения $Q_\varepsilon$ от числа алгоритмов $D$. Справа: вероятность получения каждого из алгоритмов $P_d = \mathsf{P}[\mu X = a_d]$, вклад каждого алгоритма в вероятность переобучения $Q_\varepsilon$, значение $Q_\varepsilon$ для пессимистичной МЭР по подмножеству алгоритмов $\{a_0, \ldots, a_d\}$ как функция от числа алгоритмов $d$. Все графики построены при $\ell = k = 100,\ m = 20,\ \varepsilon = 0.05$.

**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. 8.5 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. 8.5.

В эксперименте вычислялась также эмпирическая оценка функционала равномерной сходимости,

$$\hat{P}_\varepsilon = \hat{\mathsf{P}}\Big[\max_{a\in A}\delta(a,X) \geqslant \varepsilon\Big].$$

Он является завышенной верхней оценкой вероятности переобучения, $Q_\varepsilon \leqslant P_\varepsilon$, см. стр. 48. Правый график на рис. 8.5 показывает, что эта оценка может быть завышенной в сотни раз.

Левый график на рис. 8.6 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. 8.6) 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 слагаемое под знаком суммы $\sum\limits_{a\in A}$ в общей формуле вероятности переобучения (7.10):

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

Рис. 8.6 (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 бóльшую, так и в меньшую сторону, приводит к увеличению числа ошибок.

**Определение 8.9.** *Прямая цепь $A = \{a_D, \ldots, a_1, a_0, a'_1, \ldots, a'_{D'}\}$ называется унимодальной, если её левая ветвь $a_0, a_1, \ldots, a_D$ и правая ветвь $a_0, a'_1, \ldots, a'_{D'}$ 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 $\big\{a(x, w^* + t\delta)\colon t \in \mathbb{R}\big\}$ 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 $a_d$, $d = 1, \ldots, D$ допускал ошибку на объектах $x_1, \ldots, x_d$; а каждый из алгоритмов $a'_d$, $d = 1, \ldots, D$ допускал ошибку на объектах $x'_1, \ldots, x'_d$. Множества объектов $\{x_1, \ldots, x_D\}$ и $\{x'_1, \ldots, x'_D\}$ 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 $\mathbb{X}$ разбитой на четыре блока:

$$
\begin{array}{rccccccccccccc}
 & x_1 & x_2 & x_3 & & x_D & x'_1 & x'_2 & x'_3 & & x'_D & & \overbrace{\qquad}^{m} & \\
\vec{a}_0 = ( & 0, & 0, & 0, & \ldots & 0, & 0, & 0, & 0, & \ldots & 0, & 0,\ldots,0, & 1,\ldots,1 & ); \\
\vec{a}_1 = ( & 1, & 0, & 0, & \ldots & 0, & 0, & 0, & 0, & \ldots & 0, & 0,\ldots,0, & 1,\ldots,1 & ); \\
\vec{a}_2 = ( & 1, & 1, & 0, & \ldots & 0, & 0, & 0, & 0, & \ldots & 0, & 0,\ldots,0, & 1,\ldots,1 & ); \\
\vec{a}_3 = ( & 1, & 1, & 1, & \ldots & 0, & 0, & 0, & 0, & \ldots & 0, & 0,\ldots,0, & 1,\ldots,1 & ); \\
\ldots & & & & \ldots & & & & & \ldots & & \ldots & \ldots & \\
\vec{a}_D = ( & 1, & 1, & 1, & \ldots & 1, & 0, & 0, & 0, & \ldots & 0, & 0,\ldots,0, & 1,\ldots,1 & ); \\
 & & & & & & & & & & & & & \\
\vec{a}'_1 = ( & 0, & 0, & 0, & \ldots & 0, & 1, & 0, & 0, & \ldots & 0, & 0,\ldots,0, & 1,\ldots,1 & ); \\
\vec{a}'_2 = ( & 0, & 0, & 0, & \ldots & 0, & 1, & 1, & 0, & \ldots & 0, & 0,\ldots,0, & 1,\ldots,1 & ); \\
\vec{a}'_3 = ( & 0, & 0, & 0, & \ldots & 0, & 1, & 1, & 1, & \ldots & 0, & 0,\ldots,0, & 1,\ldots,1 & ); \\
\ldots & & & & \ldots & & & & & \ldots & & \ldots & \ldots & \\
\vec{a}'_D = ( & 0, & 0, & 0, & \ldots & 0, & 1, & 1, & 1, & \ldots & 1, & 0,\ldots,0, & 1,\ldots,1 & ).
\end{array}
$$

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

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

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

$$Q_\varepsilon = \frac{C_{L-2}^{\ell-2}}{C_L^\ell}\mathcal{H}_{L-2}^{\ell-2,\,m}(s_0(\varepsilon)) + \sum_{d=1}^{\min\{k,D\}}\Bigg(2\frac{C_{L-d-\delta}^{\ell-\delta}}{C_L^\ell}\mathcal{H}_{L-d-\delta}^{\ell-\delta,\,m}(s_d(\varepsilon)) - \\ - \delta\frac{C_{L-2d-2}^{\ell-1}}{C_L^\ell}\mathcal{H}_{L-2d-2}^{\ell-1,\,m}(s_d(\varepsilon)) - \\ - \frac{C_{L-2d-\delta}^{\ell-\delta}}{C_L^\ell}\mathcal{H}_{L-2d-\delta}^{\ell-\delta,\,m}(s_d(\varepsilon))\Bigg). \tag{8.4}$$

**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: $[\mu X{=}a_d] = [x_{d+1} \in X][x_1, \ldots, x_d \in \bar{X}]$. 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 $a_{d+1}', \ldots, a_D'$ также минимизирует эмпирический риск. Это происходит, когда объекты $x_1', \ldots, x_{d+1}'$ лежат в контрольной выборке. Следовательно,

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

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:

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

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

$$[\mu X{=}a_D] = [x_1, \ldots, x_D \in \bar{X}];$$
$$[\mu X{=}a_D'] = [x_1', \ldots, x_D' \in \bar{X}]\big(1 - [x_1, \ldots, x_D \in \bar{X}]\big).$$

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 | $L_{av}$ | $\ell_{av}$ | $c_{av}$ |
|---|---|---|---|---|---|
| $a_0$ | $\{x_1, x'_1\}$ | $\varnothing$ | $L-2$ | $\ell-2$ | $+1$ |
| $a_d$, $d=1,\ldots,D-1$ | $\{x_{d+1}\}$ | $\{x_1,\ldots,x_d\}$ | $L-d-1$ | $\ell-1$ | $+1$ |
| | $\{x_{d+1}\}$ | $\{x_1,\ldots,x_d,x'_1,\ldots,x'_{d+1}\}$ | $L-2d-2$ | $\ell-1$ | $-1$ |
| $a'_d$, $d=1,\ldots,D-1$ | $\{x'_{d+1}\}$ | $\{x'_1,\ldots,x'_d\}$ | $L-d-1$ | $\ell-1$ | $+1$ |
| | $\{x'_{d+1}\}$ | $\{x'_1,\ldots,x'_d,x_1,\ldots,x_d\}$ | $L-2d-1$ | $\ell-1$ | $-1$ |
| $a_D$ | $\varnothing$ | $\{x_1,\ldots,x_D\}$ | $L-D$ | $\ell$ | $+1$ |
| $a'_D$ | $\varnothing$ | $\{x'_1,\ldots,x'_D\}$ | $L-D$ | $\ell$ | $+1$ |
| | $\varnothing$ | $\{x'_1,\ldots,x'_D,x_1,\ldots,x_D\}$ | $L-2D$ | $\ell$ | $-1$ |

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 $C^{\ell-\delta}_{L-d-\delta}$ 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 переобучается. Этот факт служит косвенным обоснованием для процедур одномерной оптимизации, широко применяемых в машинном обучении.

**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 Теорему 8.1.

**Задача 8.2** (1). Доказать, что цепь алгоритмов $A = \{a_0, \ldots, a_D\}$ 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 8.3** (1). Проверить, что в случае монотонной цепи условия $[\mu X{=}a]$ в Теореме 8.2 определены корректно, доказав тождество $\sum_{a \in A} [\mu X{=}a] = 1$.

Подсказка: воспользоваться тождеством $[x \in X] = 1 - [x \in \bar{X}]$.

**Задача 8.4** (1). Проверить, что в случае унимодальной цепи условия $[\mu X{=}a]$ в Теореме 8.3 определены корректно, доказав тождество $\sum_{a \in A} [\mu X{=}a] = 1$.

Подсказка: сначала решить предыдущую задачу.

**Задача 8.5** (1). Для монотонной цепи, используя выражения для $P_a$ и комбинаторные тождества, доказать равенство $\sum_{a \in A} P_a = 1$.

**Задача 8.6** (1). Для унимодальной цепи, используя выражения для $P_a$ и комбинаторные тождества, доказать равенство $\sum_{a \in A} P_a = 1$.

**Задача 8.7** (1). Выписать оценку полного скользящего контроля $C(\mu, \mathbb{X})$ для монотонной и унимодальной цепи.

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 8.8** (1). $A = \{a_0, a_1, \ldots, a_D\}$ — 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 8.9** (3). $A = \{a_{-D'}, \ldots, a_0, \ldots, a_D\}$ — несимметричная унимодальная цепь алгоритмов с ветвями различной длины, $D \neq D'$.

**Задача 8.10** ($4^*$). Разорванная цепь $A = \{a_0, \ldots, a_D, a_{D+1}, \ldots, a_C\}$, где $a_0, \ldots, a_D$ и $a_{D+1}, \ldots, a_C$ — две монотонные цепи, $n(a_0, \mathbb{X}) = m$, $n(a_{D+1}, \mathbb{X}) = m_1 > m + D + 1$, причём алгоритм $a_{D+1}$ ошибается на всех объектах, на которых ошибается $a_D$, и ещё на $m_1 - m - D$ объектах.

**Задача 8.11** ($5^*$). $A = \{a_0, \ldots, a_D\}$ — прямая цепь без расслоения; число ошибок, чередуясь, принимает два значения: $n(a_d, \mathbb{X}) = m + (d \bmod 2)$, $d = 0, \ldots, D$.

**Задача 8.12** (5*). Матрица ошибок $m$-диагональна, то есть алгоритм $a_d$ допускает ошибки на объектах $x_i$, $i = d+1, \ldots, d+m$, для всех $d = 0, \ldots, D$.

**Задача 8.13** (5*). Матрица ошибок образуется циклическим сдвигом первых $D+1$ координат вектора ошибок с $m$ ошибками, то есть алгоритм $a_d$ допускает ошибки на объектах $x_{1+(i-1) \bmod (D+1)}$, $i = d+1, \ldots, d+m$, для всех $d = 0, \ldots, D$.

**Задача 8.14** (7*). $A = \{a_0, \ldots, a_C, \ldots, a_D\}$ — прямая цепь, составленная из цепи без расслоения $a_0, \ldots, a_C$ и монотонной цепи $a_C, \ldots, a_D$ с общим алгоритмом $a_C$.

**Задача 8.15** (8*). $A = \{a_{-D'}, \ldots, a_{-C'}, \ldots, a_C, \ldots, a_D\}$ — прямая цепь, составленная из монотонной цепи $a_{-C'}, \ldots, a_{-D'}$, цепи без расслоения $a_{-C'}, \ldots, a_C$ и монотонной цепи $a_C, \ldots, a_D$, с общими алгоритмами $a_{-C'}$ и $a_C$ соответственно.

**Задача 8.16** (10*). Произвольная цепь алгоритмов.

**Задача 8.17** (5). Пучок из $h$ монотонных цепей $\{a_0, a_{j1}, \ldots, a_{jH}\}$, $j = 1, \ldots, h$, с общим лучшим алгоритмом $a_0$, $n(a_0, \mathbb{X}) = m$. Предполагается, что множества объектов $X(j) = \{x_i \colon I(a_0, x_i) = 0 \text{ и } I(a_{jH}, x_i) = 1\}$, $j = 1, \ldots, h$, попарно не пересекаются.

## Практикум

В практическом задании предлагается реализовать Алгоритм 8.2 и использовать его для проверки теоретических оценок и исследования новых семейств.

**Задача 8.18** (5). Написать программу, позволяющую:

- генерировать модельные семейства алгоритмов $A = \{a_1, \ldots, a_D\}$ 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 $\widetilde{Q}_\varepsilon$ 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 $\bar{Q}_\varepsilon, \underline{Q}_\varepsilon, \hat{Q}_\varepsilon, \widetilde{Q}_\varepsilon$ для подмножества $A(d) = \{a_1, \ldots, a_d\}$;
  — точные значения $\bar{Q}_\varepsilon, \underline{Q}_\varepsilon, \hat{Q}_\varepsilon, \widetilde{Q}_\varepsilon$ 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 $\bar{Q}_\varepsilon, \underline{Q}_\varepsilon, \hat{Q}_\varepsilon, \widetilde{Q}_\varepsilon$, 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 $\bar{Q}_\varepsilon, \underline{Q}_\varepsilon, \hat{Q}_\varepsilon, \widetilde{Q}_\varepsilon$ от параметров модельного семейства.

**Задача 8.19** (3)**.** Монотонная цепь с параметрами $m$ и $D$.

**Задача 8.20** (3)**.** Унимодальная цепь с параметрами $m$ и $D$.

**Задача 8.21** (3)**.** Единичная окрестность с параметрами $m$ и $D$ (см. Задачу 8.8).

**Задача 8.22** (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 8.23** (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 8.24** (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 $\{a_0, \ldots, a_D\}$ и её подмножества $\{a_0, a_t, a_{2t}, a_{3t}, \ldots\}$ при $t = 1, \ldots, D$.

# 9 Оценки расслоения–связности

Принцип *порождающих и запрещающих множеств* (ПЗМ) универсален, но не очень удобен в использовании. Если же наложить на множество алгоритмов дополнительные ограничения расслоения и связности, то ПЗМ легко выписываются в терминах графа Хассе частично упорядоченного множества векторов ошибок. Накладываемые ограничения не слишком обременительны, и многие практические семейства им удовлетворяют. Получаемые оценки вероятности переобучения зависят от количественных характеристик расслоения и связности множества алгоритмов, являются точными для некоторых нетривиальных семейств, а при игнорировании эффектов расслоения и связности переходят в оценки Вапника-Червоненкиса.

## §9.1 Граф расслоения–связности

Напомним основные обозначения:
$\mathbb{X} = \{x_1, \ldots, x_L\}$ — 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 $b$;
$A_m = \big\{a \in A \colon n(a, \mathbb{X}) = m\big\}$ — *$m$-й слой* множества $A$.

Будем полагать, что все векторы ошибок $\vec{a} = \big(I(a, x_i)\big)_{i=1}^{L}$, 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: $a \leqslant b$ тогда и только тогда, когда $I(a, x) \leqslant I(b, x)$ для всех $x \in \mathbb{X}$. Определим $a < b$ если $a \leqslant b$ 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 $E = \big\{(a, b) \colon a \prec b\big\}$.*

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.

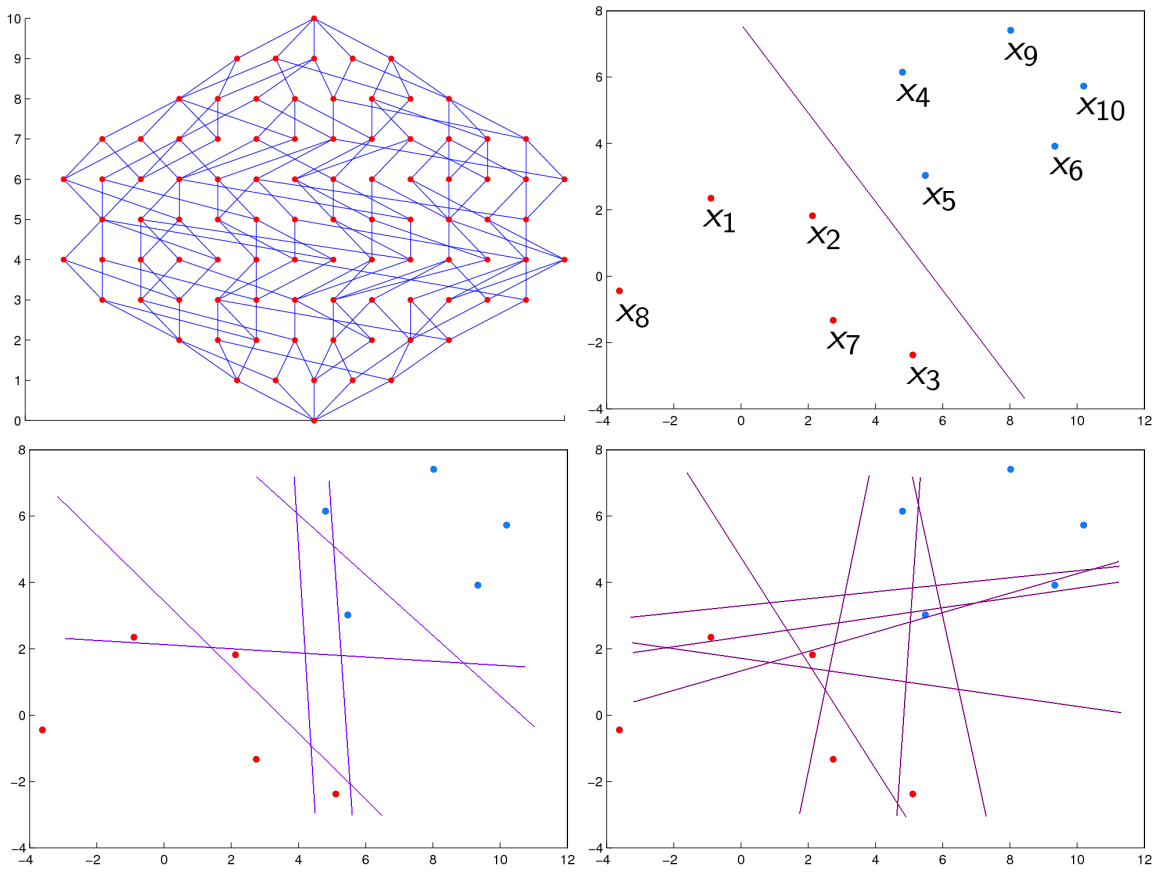



**Рис. 9.1.** Пример графа расслоения–связности (вверху слева; по вертикальной оси отложены номера слоёв), порождаемого семейством линейных алгоритмов классификации на выборке из 10 объектов, по 5 объектов каждого класса (вверху справа). Первый слой образуется 5 алгоритмами с одной ошибкой (внизу слева), второй слой — 8 алгоритмами с двумя ошибками (внизу справа), и т. д.

В типичном случае граф расслоения–связности изоморфен графу транзитивной редукции отношения порядка $\leqslant$, называемому также *диаграммой Хáссе*. Отличие 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.

## §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 $X \in [\mathbb{X}]^\ell$:

$$\big[\mu X{=}a\big] = \sum_{v \in V_a} c_{av} \big[X_{av} \subseteq X\big] \big[X'_{av} \subseteq \bar{X}\big].$$

Чтобы упростить оценку, ослабим условие до необходимого, заменив равенство неравенством, оставим по одной паре ПЗМ для каждого алгоритма и положим $c_{av}=1$.

**Гипотеза 9.1.** *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*

$$\big[\mu X{=}a\big] \leqslant \big[X_a \subseteq X\big]\big[X'_a \subseteq \bar{X}\big], \quad \forall X \in [\mathbb{X}]^\ell. \tag{9.1}$$

**Теорема 9.1.** *Если гипотеза 9.1 справедлива, то вероятность переобучения оценивается сверху по формуле*

$$Q_\varepsilon(\mu,\mathbb{X}) \;\leqslant\; \bar{Q}_\varepsilon(\mu,\mathbb{X}) = \sum_{a\in A} \bar{P}_a \mathcal{H}_{L_a}^{\ell_a,\,m_a}\left(s_a(\varepsilon)\right); \tag{9.2}$$

$$\begin{aligned} P_a = \mathsf{P}[\mu X = a] \;\leqslant\; \bar{P}_a &= C_{L_a}^{\ell_a}/C_L^\ell; \\ L_a &= L - |X_a| - |X'_a|; \\ \ell_a &= \ell - |X_a|; \\ m_a &= n(a, \mathbb{X}\backslash X_a \backslash X'_a); \\ s_a(\varepsilon) &= \tfrac{\ell}{L}\big(n(a,\mathbb{X}) - \varepsilon k\big) - n(a, X_a). \end{aligned} \tag{9.3}$$

Единственное отличие от Теоремы 7.3 в том, что равенство (7.1) в Гипотезе 7.1 заменяется неравенством (9.1) в Гипотезе 9.1. Ход доказательства сохраняется в точности. Знак сравнения $=$ или $\leqslant$ без изменения переходит из (9.1) в (9.3) для каждого алгоритма $a$. Поэтому равенство в оценке (9.2) достигается тогда и только тогда, когда в (9.1) имеет место равенство для всех алгоритмов $a$.

**Следствие 9.1.1.** *Оценка* (9.2) *является достижимой. Равенство достигается тогда и только тогда, когда* $\sum\limits_{a\in A} \bar{P}_a = 1$. *Значение* $\sum\limits_{a\in A} \bar{P}_a$ *ориентировочно показывает степень завышенности верхней оценки вероятности переобучения* (9.2).

**Монотонные методы обучения.** Обозначим проекцию вектора ошибок $a$ на подвыборку $X \subseteq \mathbb{X}$ через $\vec{a}_X = \big(I(a,x_i)\big)_{x_i\in X}$. На проекциях естественным образом вводится отношение частичного порядка: $\vec{a}_X \leqslant \vec{b}_X$ тогда и только тогда, когда $I(a,x) \leqslant I(b,x)$ для всех $x \in X$. Определим $\vec{a}_X < \vec{b}_X$ если $\vec{a}_X \leqslant \vec{b}_X$ и $\vec{a}_X \neq \vec{b}_X$.

**Определение 9.2.** *Метод обучения $\mu$ называется монотонным, если $\mu(X) \in A_M(X)$ для любой обучающей выборки $X \in [\mathbb{X}]^\ell$, где*

$$A_M(X) = \operatorname{Arg}\min_{a\in A} M(a,X),$$

*и критерий $M(a,X)$ является строго монотонной функцией вектора ошибок, то есть для любых $X \subset \mathbb{X}$ и $a,b \in A$, таких, что $\vec{a}_X < \vec{b}_X$, справедливо $M(a,X) < M(b,X)$.*

Примером монотонного метода является минимизация эмпирического риска, а также её обобщение — минимизация *взвешенного эмпирического риска*,

$$M(a,X) = \sum_{x_i\in X} w_i I(a,x_i) \to \min_{a\in A},$$

где $w_i$ — неотрицательный вес объекта $x_i$. Введение весов объектов — довольно распространённый приём. Он, в частности, применяется, когда классы не сбалансированы, и объектам малочисленного класса требуется придать больший вес. Иногда меньший вес назначают тем объектам, информация о которых менее точна. Ещё минимизация взвешенного эмпирического риска применяется для обучения базовых алгоритмов в композициях типа бустинга [97]; в этом случае веса объектов пересчитываются по специальным формулам перед построением каждого базового алгоритма.

Допустим, что про метод обучения $\mu$ известно только, что он минимизирует монотонный критерий $M(a, X)$, но не известно, как именно он выбирает алгоритм из множества $A_M(X)$. Тогда будем предполагать, что реализуется худший случай.

**Определение 9.3.** *Монотонный метод обучения $\mu$ называется пессимистичным, если для любой обучающей выборки $X \in [\mathbb{X}]^\ell$ из множества $A_M(X)$ выбирается наиболее переобученный алгоритм:*

$$\mu X = \arg \max_{a \in A_M(X)} \delta(a, X).$$

Пессимистичный метод не реализуем на практике, так как он использует информацию о скрытой выборке. Он интересен тем, что даёт верхние оценки вероятности переобучения, справедливые для любого монотонного метода обучения.

**Теорема 9.2.** *Для любой выборки $\mathbb{X}$ и любого монотонного метода обучения $\mu$ с критерием $M(a, X)$ справедлива оценка $Q_\varepsilon(\mu, \mathbb{X}) \leqslant Q_\varepsilon(\mu_{\max}, \mathbb{X})$, где $\mu_{\max}$ — пессимистичный монотонный метод обучения с тем же критерием.*

**ПЗМ для монотонных методов обучения.** 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 $a$:

$$X_a = \big\{x \in \mathbb{X} \;\big|\; \exists b \in A\colon a \prec b,\ I(a, x) < I(b, x)\big\}, \tag{9.4}$$

и $X'_a$ — множество объектов $x \in \mathbb{X}$, таких, что $a$ ошибается на $x$ и существует лучший алгоритм $b \in A$, который не ошибается на $x$:

$$X'_a = \big\{x \in \mathbb{X} \;\big|\; \exists b \in A\colon b < a,\ I(b, x) < I(a, x)\big\}. \tag{9.5}$$

**Лемма 9.3.** *Если $\mu$ — пессимистичный монотонный метод обучения, то множество $X_a$ (9.4) является порождающим, а множество $X'_a$ (9.5) — запрещающим для алгоритма $a$ в смысле Гипотезы 9.1.*

**Доказательство.** Пусть $\mu X = a$. Докажем от противного, что тогда выполняются два условия: $X_a \subseteq X$ и $X'_a \subseteq \bar{X}$.

Допустим, что найдётся $x_{ab} \in X_a$, не лежащий в $X$. Векторы ошибок $a$ и $b$ отличаются только на объекте $x_{ab}$. Следовательно,

$$\vec{a}_X = \vec{b}_X, \quad M(a, X) = M(b, X), \quad \nu(a, X) = \nu(b, X).$$

Значит, оба алгоритма, $a$ и $b$, принадлежат $A_M(X)$. В то же время,

$$\vec{a}_{\bar{X}} < \vec{b}_{\bar{X}}, \quad \nu(a,\bar{X}) < \nu(b,\bar{X}), \quad \delta(a,X) < \delta(b,\bar{X}),$$

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 $\vec{b}_X < \vec{a}_X$. Поскольку метод $\mu$ минимизирует монотонный критерий $M(a,X)$, выбор алгоритма $b$ будет предпочтительнее, чем $a$, что противоречит условию $\mu X = a$. Значит, $X'_a \subseteq \bar{X}$.

Лемма доказана. ■

## Верхняя оценка вероятности переобучения.

**Определение 9.4.** *Верхней связностью $u(a)$ (up-connectivity) алгоритма $a \in A$ будем называть число рёбер графа, исходящих из вершины $a$:*

$$u(a) = \#\big\{x_{ab} \in \mathbb{X} \colon a \prec b\big\} = |X_a|.$$

**Определение 9.5.** *Нижней связностью $d(a)$ (down-connectivity) алгоритма $a \in A$ будем называть число рёбер графа, входящих в вершину $a$:*

$$d(a) = \#\big\{x_{ba} \in \mathbb{X} \colon b \prec a\big\}.$$

Связность $u(a)$ (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 9.6.** *Неполноценностью $q(a)$ (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 $x$:*

$$q(a) = |X'_a|.$$

В терминах графа расслоения–связности $q(a)$ есть число различных объектов $x_{bc}$, соответствующих всевозможным рёбрам $(b,c)$ на путях, ведущих к вершине $a$.

Справедливо неравенство $d(a) \leqslant q(a) \leqslant n(a,\mathbb{X})$.

Равенство $q(a) = d(a)$ достигается на всех алгоритмах двух самых нижних слоёв.

Равенство $q(a) = n(a,\mathbb{X})$ достигается в случае, когда существует корректный алгоритм $a_0 \in A$: $n(a_0,\mathbb{X}) = 0$.

**Теорема 9.4 (оценка расслоения–связности).** *Для произвольной выборки $\mathbb{X}$, произвольного монотонного метода обучения $\mu$ и произвольного $\varepsilon \in (0,1)$*

$$Q_\varepsilon(\mu,\mathbb{X}) \leqslant \sum_{a\in A} \frac{C^{\ell-u}_{L-u-q}}{C^\ell_L} \mathcal{H}^{\ell-u,\,m-q}_{L-u-q}\left(\tfrac{\ell}{L}(m-\varepsilon k)\right), \tag{9.6}$$

*где $u = u(a)$ — верхняя связность, $q = q(a)$ — неполноценность, $m = n(a,\mathbb{X})$ — число ошибок алгоритма $a$ на генеральной выборке.*

**Доказательство.** Данная оценка следует непосредственно из общей оценки (9.2), Теоремы 9.2 и Леммы 9.3, если заметить, что произвольный алгоритм $a \in A$ ошибается на всех объектах запрещающего множества $X'_a$ и не ошибается на всех объектах порождающего множества $X_a$: $|X_a| = u(a),\ n(a, X_a) = 0,\ |X'_a| = n(a, X'_a) = q(a)$. ■

**Некоторые свойства оценки расслоения–связности.**

1. Благодаря комбинаторному сомножителю $C^{\ell-u}_{L-u-q}/C^{\ell}_{L}$ вклад каждого алгоритма $a$ в оценку $Q_\varepsilon$ экспоненциально убывает с ростом неполноценности $q$ и связности $u$. 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. Числитель (9.6) обращается в нуль при $\ell - u \leqslant L - u - q$. Поэтому ненулевой вклад в сумму могут давать только алгоритмы нижних слоёв, для которых $q(a) \leqslant k$. Таким образом, при малых $k$ послойным вычислением можно получать не приближённое, а точное значение $\bar{Q}_\varepsilon$. С другой стороны, при совсем малых $k = 1, 2, 3$ игнорируются алгоритмы с $q(a) > k$, которые при бóльших $k$ могли бы вносить существенный вклад в переобучение. Отсюда следует практическая рекомендация — брать число $k$ не меньше, чем число «эффективно работающих» нижних слоёв.

3. Если пренебречь расслоением и связностью, положив $q = u = 0$ для каждого $a \in A$ в формуле (9.6), то получится лучшая из VC-оценок (5.3):

$$Q_\varepsilon(\mu, \mathbb{X}) \leqslant \sum_{a \in A} \mathcal{H}^{\ell,\,m}_{L}\left(\tfrac{\ell}{L}(m - \varepsilon k)\right).$$

4. Оценка расслоения–связности является достижимой. Неравенство $Q_\varepsilon \leqslant \bar{Q}_\varepsilon$ переходит в равенство, когда условие (9.1) является равенством. Это так, в частности, для монотонной цепи алгоритмов (стр. 76) и её многомерного обобщения — монотонной сети алгоритмов, которая будет рассмотрена ниже.

5. 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 9.5.** *Если $\mu$ — монотонный метод обучения и семейство $A$ корректно, то*

$$Q_\varepsilon(\mu, \mathbb{X}) \leqslant \sum_{a \in A} \frac{C^{\ell-u}_{L-u-m}}{C^{\ell}_{L}} \big[m \geqslant \varepsilon k\big], \tag{9.7}$$

*где $u = u(a),\ m = n(a, \mathbb{X})$.*

**Доказательство.** Если в $A$ существует корректный алгоритм $a_0$, то неполноценность любого алгоритма $a \in A$ равна числу его ошибок, $q(a) = n(a, \mathbb{X})$. Поэтому в (9.6) можно подставить $m - q = 0$. Тогда, согласно Лемме 7.7 (стр. 68), гипергеометрическое распределение вырождается: $\mathcal{H}^{\ell-u,\,0}_{L-u-m}\left(\frac{\ell}{L}(m - \varepsilon k)\right) = [m \geqslant \varepsilon k]$. Отсюда вытекает утверждение теоремы. ■

**Снова монотонная цепь алгоритмов.** Ранее уже была получена точная оценка вероятности переобучения (8.3) для монотонной цепи алгоритмов $a_0 \prec a_1 \prec \cdots \prec a_D$, в которой лучший алгоритм допускает $m$ ошибок, $n(a_0, \mathbb{X}) = m$:

$$Q_\varepsilon = \sum_{d=0}^{\min\{k,D\}} \frac{C_{L-d-\delta}^{\ell-\delta}}{C_L^\ell} \mathcal{H}_{L-d-\delta}^{\ell-\delta,\,m}\left(\tfrac{\ell}{L}(m+d-\varepsilon k)\right), \quad \delta = [d < D].$$

Легко убедиться, что она совпадает с общей оценкой расслоения–связности (9.6). Это следует из того, что

1) связность равна 1 для всех алгоритмов, кроме последнего, $u(a_d) = [d < D]$;

2) неполноценность и число ошибок связаны равенством $n(a_d, \mathbb{X}) = q(a_d) + m$;

3) числитель (9.6) обращается в нуль при $\ell - \delta \leqslant L - d - \delta$, то есть $d \leqslant k$, следовательно, ненулевой вклад в сумму дают только первые $k$ алгоритмов.

Таким образом, монотонная цепь алгоритмов — это пример семейства, для которого общая оценка расслоения–связности (9.6) является точной.

## §9.3 Профиль расслоения–связности

В ряде случаев общая оценка (9.6) приводится к более удобному виду.

**Определение 9.7.** *Профилем расслоения–связности множества алгоритмов $A$ называется матрица $(\Delta_{mu})$ размера $(L+1)\times(L+1)$, где $\Delta_{mu}$ — число алгоритмов в $m$-м слое с верхней связностью $u$:*

$$\Delta_{mu} = \sum_{a\in A} \big[n(a,\mathbb{X}) = m\big]\big[u(a) = u\big].$$

**Теорема 9.6.** *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*

$$Q_\varepsilon \leqslant \sum_{m=m_0}^{L} \sum_{u=0}^{L} \Delta_{mu} \frac{C_{L-u-m+m_0}^{\ell-u}}{C_L^\ell} \mathcal{H}_{L-u-m+m_0}^{\ell-u,\,m_0}\left(\tfrac{\ell}{L}(m-\varepsilon k)\right). \tag{9.8}$$

**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 $q(a)$. Значит,

$$Q_\varepsilon \leqslant \sum_{m=m_0}^{L} \sum_{a\in A_m} \frac{C_{L-u(a)-m+m_0}^{\ell-u(a)}}{C_L^\ell} \mathcal{H}_{L-u(a)-m+m_0}^{\ell-u(a),\,m_0}\left(\tfrac{\ell}{L}(m-\varepsilon k)\right).$$

Перегруппировав слагаемые и воспользовавшись определением профиля расслоения–связности, получим утверждение теоремы. ■

**Теорема 9.7.** *Пусть верны предположения предыдущей теоремы и исток является корректным алгоритмом, $m_0 = n(a_0, \mathbb{X}) = 0$. Тогда справедлива верхняя оценка вероятности переобучения*

$$Q_\varepsilon \leqslant \sum_{m=\lceil \varepsilon k \rceil}^{L} \sum_{u=0}^{L} \Delta_{mu} \frac{C_{L-u-m}^{\ell-u}}{C_L^\ell}. \tag{9.9}$$

**Доказательство.** В оценку (9.8) подставим $m_0 = 0$. Гипергеометрическое распределение вырождается согласно Лемме 7.7, откуда и получаем оценку (9.9). ■

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[8] $\Delta_{mu}$ от $m$ и $u$, для множества линейных алгоритмов классификации и линейно разделимых двумерных выборок длины $L = 20,\ 50,\ 100$.

Глядя на графики, можно выдвинуть гипотезу, что профиль расслоения–связности $\Delta_{mu}$ является с высокой точностью *сепарабельным*:

$$\Delta_{mu} \approx \Delta_m \lambda_u,$$

где $\Delta_m = |A_m|$ — число алгоритмов в $m$-м слое, $\lambda_u$ — доля алгоритмов со связностью $u$.

Вектор $(\Delta_m)_{m=0}^L$ предлагается называть *профилем расслоения*, а вектор $(\lambda_u)_{u=0}^L$ — *профилем связности* множества алгоритмов $A$.

Очевидно, профиль связности нормирован: $\lambda_0 + \cdots + \lambda_L = 1$.

Проверка гипотезы сепарабельности пока остаётся открытой проблемой.

**Гипотеза о концентрации профиля связности вокруг значения размерности.** На рис. 9.2 видно, что профиль связности концентрируется вокруг значения $u = 2$, равного размерности пространства. С увеличением длины выборки доминирование данной компоненты профиля усиливается.

Эксперименты Дениса Кочедыкова с линейными классификаторами [34, 77] показывают, что и при бóльших значениях размерности $h$ максимум профиля связности достигается при $u = h$, см. рис. 9.3.

Можно выдвинуть гипотезу, что профиль связности концентрируется вокруг значения размерности не только в случае линейных классификаторов. Более того, можно предположить, что если выборка де-факто располагается в подпространстве меньшей размерности, то максимум профиля связности указывает на значение «эффективной размерности» выборки. Проверка этих гипотез также пока остаётся открытой проблемой.

[8] Вычислительные эксперименты выполнены Ильёй Решетняком.

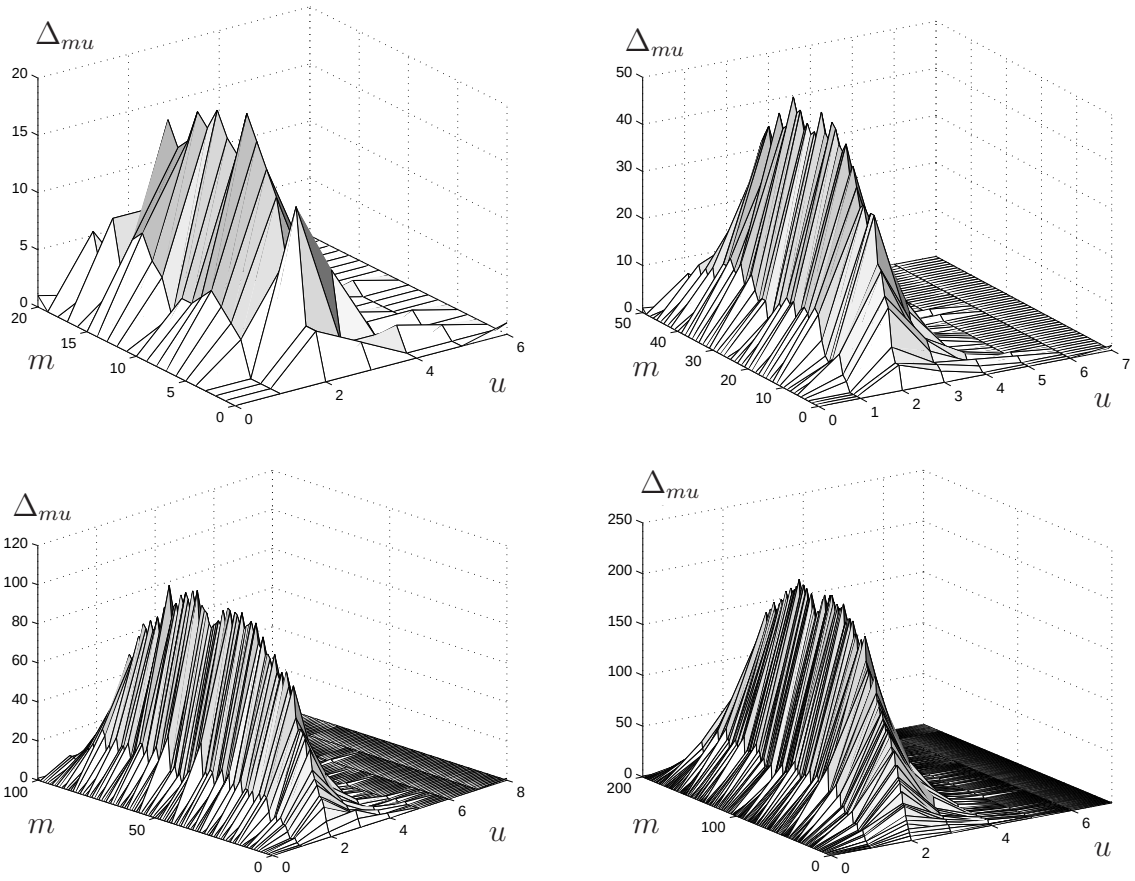



**Рис. 9.2.** Профили расслоения–связности для двумерных выборок длины $L = 20$, 50, 100, 200. Профиль $\Delta_{mu}$ — это количество алгоритмов с числом ошибок $m$ на генеральной выборке и верхней связностью $u$.

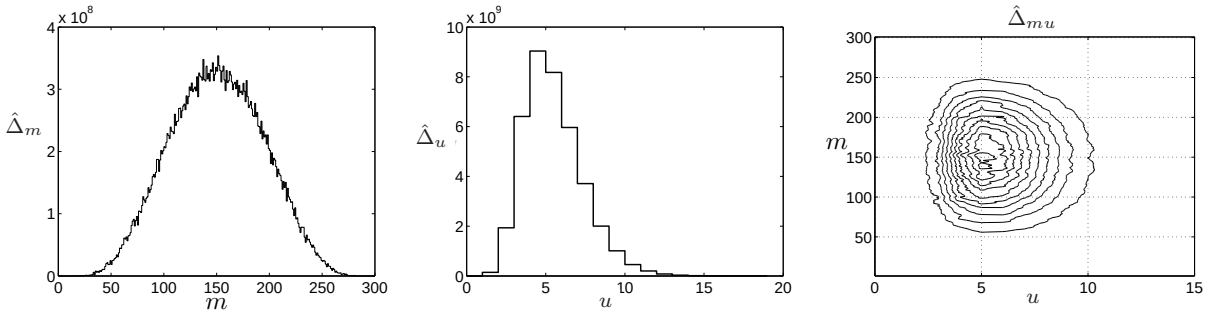



**Рис. 9.3.** Оценки профиля расслоения $\hat{\Delta}_m$, ненормированного профиля связности $\hat{\Delta}_u = \lambda_u|A|$ и изолиний профиля расслоения-связности $\hat{\Delta}_{mu}$ для семейства линейных классификаторов в $\mathbb{R}^5$, при $L = 300$, согласно [77].

Не выяснен также до конца вопрос о ширине пика профиля связности. Заметим, что на рис. 9.2 и рис. 9.3 показаны профили только *верхней* связности. Есть гипотеза, что полусумма верхней и нижней связности гораздо сильнее концентрируется вокруг значения размерности, чем верхняя и нижняя связность по отдельности. Обозначим среднюю верхнюю связность в слое $m$ через $u_m$, среднюю нижнюю связность — че-

рез $d_m$. Число рёбер, соединяющих алгоритмы в двух соседних слоях, можно записать двумя способами: $\Delta_m u_m = \Delta_{m+1} d_{m+1}$. В нижних слоях профиль расслоения $\Delta_m$, как правило, стремительно возрастает, поэтому верхние связности должны систематически превышать нижние. Отсюда следуют два вывода. Во-первых, замена верхних связностей алгоритмов нижних слоёв значением размерности будет, скорее всего, завышать оценку вероятности переобучения. Во-вторых, правая ветвь профиля связности (см. средний график на рис. 9.3) образована, скорее всего, именно алгоритмами нижних слоёв, поэтому её оценивание представляет наибольший интерес.

## 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 эффективного приближённого вычисления вероятности переобучения, возможно, будет достаточно перебрать алгоритмы из нескольких нижних слоёв.

**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 размерностью.

## 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 (9.6) обращалась бы в равенство? Возможно ли сформулировать эти ограничения в терминах графа расслоения–связности?

**Задача 9.2** ($5^*$)**.** Обобщить оценку расслоения–связности (9.6) на случай, когда в графе расслоения–связность имеются «дырки», то есть для некоторых пар алгоритмов $(a, b)$, $a < b$, не существует алгоритма $c \in A$, такого, что $a \prec c < b$.

**Задача 9.3** ($10^*$)**.** Обосновать гипотезу сепарабельности для линейных алгоритмов классификации.

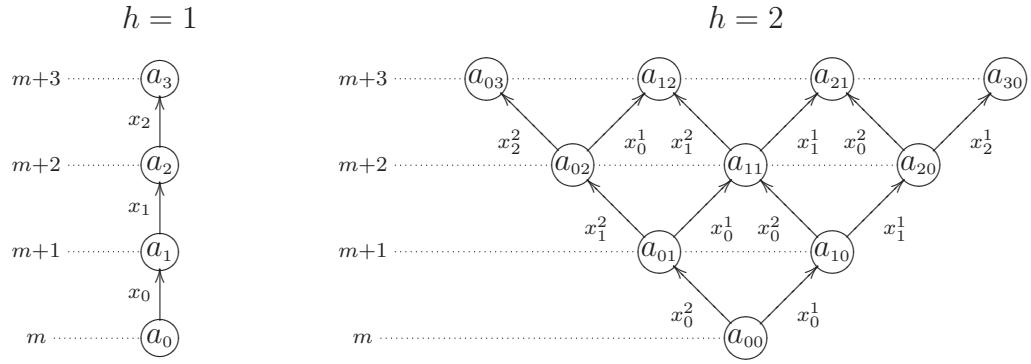
# 10 Многомерные сети алгоритмов

Многомерные сети алгоритмов — это модельные семейства, обладающие не только свойствами расслоения и связности, но и размерностью, что сближает их с реальными семействами. Эти семейства интересны ещё и тем, что для них оценки расслоения–связности являются точными.

## §10.1 Двумерная монотонная сеть

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 некоторому фрагменту прямоугольной сети размерности $h$.

На рисунке показаны четыре нижних слоя графов расслоения–связности монотонной цепи (слева) и двумерной сети (справа). Горизонтальные линии соответствуют слоям, $m$ — число ошибок лучшего алгоритма на генеральной выборке.




В двумерной сети при переходе от произвольного алгоритма $a_{ij}$ к алгоритму $a_{i+1,j}$ возникает ошибка на объекте $x_i^1$, а при переходе от $a_{ij}$ к $a_{i,j+1}$ — на объекте $x_j^2$. Число алгоритмов в слое $m+t$ равно $t+1$. Верхняя связность $u(a_{ij}) = 2$ совпадает с размерностью $h = 2$. Неполноценность $q(a_{ij})$ равна $t$, поскольку $a_{ij}$ отличается от лучшего алгоритма $a_{00}$ ровно на $t$ объектах. Подставив все эти данные в оценку (9.6) и сгруппировав слагаемые по слоям, получим оценку расслоения–связности для $H$ нижних слоёв двумерной сети алгоритмов:

$$Q_\varepsilon \leqslant \sum_{t=0}^{H}(t+1)\frac{C_{L-t-2}^{\ell-2}}{C_L^\ell}\mathcal{H}_{L-t-2}^{\ell-2,\,m}\left(\tfrac{\ell}{L}(m+t-\varepsilon k)\right).$$

Пока это лишь предварительные соображения, показывающие, что оценки расслоения-связности можно получать довольно просто. Далее мы обобщим эту оценку на случай произвольной размерности и более аккуратно учтём отсутствие связности в верхнем слое. Мы также убедимся, что оценка расслоения–связности для $h$-мерной монотонной сети является точной. Первым эту оценку получил Павел Ботов [7].

## §10.2 Монотонная сеть произвольной размерности

Пусть $J = (j_1, \ldots, j_h) \in \{0, \ldots, H\}^h$ — целочисленный вектор индексов.

Введём на векторах индексов отношение частичного порядка: для произвольных $J = (j_1, \ldots, j_h)$ и $K = (k_1, \ldots, k_h)$ положим $J \leqslant K$, если $j_d \leqslant k_d$ для всех $d = 1, \ldots, h$. Положим $J < K$, если $J \leqslant K$ и $J \neq K$. Положим $|J| = j_1 + \cdots + j_h$.

**Определение 10.1.** *Монотонной сетью алгоритмов размерности $h$ и высоты $H$ называется множество алгоритмов*

$$A = \big\{a_J \colon J \in \{0, \ldots, H\}^h,\ |J| \leqslant H\big\}, \tag{10.1}$$

*матрица ошибок которого задаётся разбиением генеральной выборки на три непересекающихся подмножества $\mathbb{X} = X_0 \sqcup X_1 \sqcup \big\{x_t^d \colon t = 0, \ldots, H-1,\ d = 1, \ldots, h\big\}$ и индикатором ошибки, определяемым для всех $a_J \in A,\ x \in \mathbb{X}$ как*

$$I(a_J, x) = \begin{cases} 0, & x \in X_0; \\ 1, & x \in X_1; \\ [t < j_d], & x = x_t^d. \end{cases} \tag{10.2}$$

**Интерпретация.** Монотонная сеть — 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 $x_0^d, \ldots, x_{H-1}^d$, независимо от значений остальных параметров. В реальных задачах это, как правило, не так, хотя известны примеры, когда реальное семейство порождает монотонную сеть на специально подобранной выборке (упражнения 10.2, 10.3).

**Свойства монотонной сети.**

1. Если $J < K$, то $a_J < a_K$.
2. Для всех $a_J \in A$ число ошибок $n(a_J, \mathbb{X}) = m + |J|$, где $m = |X_1|$.
3. Если множество $A$ имеет структуру (10.1) при заданных $h$ и $H$ и обладает свойствами 1 и 2, то его матрица ошибок однозначно определяется формулой (10.2).
4. Все векторы ошибок алгоритмов $a_J \in A$ попарно различны.
5. В $A$ существует единственный *лучший* алгоритм $a_0 = a_{(0,\ldots,0)}$ с минимальным числом ошибок $n(a_0, \mathbb{X}) = m$.
6. Размерность $h$ и высота $H$ удовлетворяют ограничению $m + hH \leqslant L$.
7. $a_J \prec a_K$ тогда и только тогда, когда векторы $J$ и $K$ отличаются только по одной координате, $j_d + 1 = k_d$. Ребру $(a_J, a_K)$ графа расслоения-связности взаимно однозначно соответствует объект $x_{JK} = x_{j_d}^d$, для которого $I(a_J, x_{JK}) < I(a_K, x_{JK})$.
8. Число алгоритмов в слое $m+t$ равно числу векторов $J = (j_1, \ldots, j_h)$ таких, что $|J| = j_1 + \ldots + j_h = t$ и равно числу способов выбрать $t$ предметов из $h$ с повторениями,

$$|A_{m+t}| = \bar{C}_h^t = C_{h+t-1}^{h-1}.$$

9. Просуммировав число алгоритмов по слоям и воспользовавшись известным комбинаторным тождеством, нетрудно найти число алгоритмов во всей сети:

$$|A| = C_{h-1}^{h-1} + \cdots + C_{h+H-1}^{h-1} = C_{H+h}^h.$$

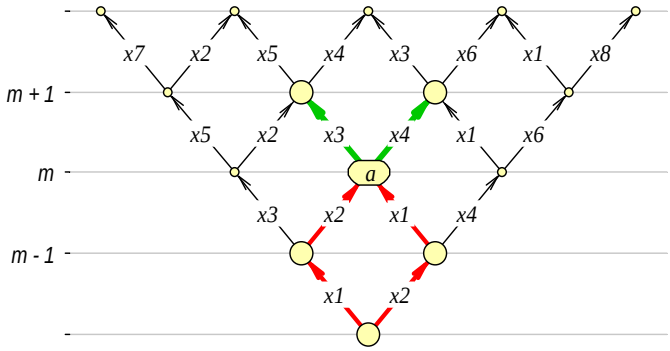



**Рис. 10.1.** Пример двумерной сети алгоритмов. Для алгоритма $a$ верхняя связность $u(a) = \#\{x3, x4\} = 2$, неполноценность $q(a) = \#\{x1, x2\} = 2$.

## §10.3 Точная оценка расслоения–связности

Обозначим через $A'$ множество всех алгоритмов $a_J$ монотонной сети $A$, за исключением алгоритмов последнего слоя, $A' = A \setminus A_{m+H}$. Из формул (9.4), (9.5) и графа расслоения–связности (см. рис. 10.1) следует, что для любого алгоритма $a_J \in A'$

$X_J = \{x_t^d : t = j_d,\ d = 1, \ldots, h\}$ — порождающее множество;
$X'_J = \{x_t^d : t < j_d,\ d = 1, \ldots, h\}$ — запрещающее множество.

Лемма 9.3 даёт лишь необходимое условие того, что алгоритм $a_J$ будет получен как результат обучения: $[\mu X{=}a_J] \leqslant [X_J \subseteq X]\,[X'_J \subseteq \bar{X}]$. Однако в случае многомерной монотонной сети необходимое условие является также и достаточным.

**Лемма 10.1.** *Если $\mu$ — пессимистичный монотонный метод обучения и $|J| < H$, то*

$$[\mu X{=}a_J] = [X_J \subseteq X]\,[X'_J \subseteq \bar{X}] \quad \textit{для каждого } a_J \in A'.$$

**Доказательство.** Докажем, что если $X_J \subseteq X$ и $X'_J \subseteq \bar{X}$, то только алгоритм $a_J$ может быть результатом обучения.

Множество объектов $x$, на которых $I(a_J, x) = 1$, есть $X_1 \sqcup X'_J$. Подмножество $X'_J$ целиком лежит в контроле, а на объектах из $X_1$ ошибаются все алгоритмы сети. Поэтому $(\vec{a}_J)_X \leqslant (\vec{a}_K)_X$ для всех $a_K \in A$, следовательно, $a_J \in A_M(X)$.

Покажем, что $A_M(X) = \{a_K \in A : K \leqslant J\}$. Действительно, если $K \not\leqslant J$, то существует координата $d \in \{1, \ldots, h\}$, такая, что $k_d > j_d$. Объект $x = x_{j_d}^d$ принадлежит $X_J$, следовательно, он принадлежит и $X$, однако $I(a_J, x) < I(a_K, x)$, поэтому $(\vec{a}_J)_X < (\vec{a}_K)_X$, значит, $a_K \notin A_M(X)$.

Покажем теперь, что никакой другой алгоритм $a_K \in A_M(X)$ не может быть результатом обучения $\mu X$. Поскольку $K < J$, существует координата $d \in \{1, \ldots, h\}$, такая, что $k_d < j_d$. Объект $x = x_{k_d}^d$ принадлежит $X'_J$, следовательно, он принадлежит и $\bar{X}$, поэтому $(\vec{a}_K)_{\bar{X}} < (\vec{a}_J)_{\bar{X}}$, и пессимистичный метод $\mu$ выберет алгоритм $a_J$.

Таким образом, только алгоритм $a_J$ может быть результатом обучения. ■

**Теорема 10.2 (О монотонной сети).** *Пусть $A$ — монотонная сеть размерности $h$ и высоты $H$, $\mu$ — пессимистичный монотонный метод обучения и $k < H$. Тогда*

$$Q_\varepsilon(\mu, \mathbb{X}) = \sum_{t=0}^{k} C_{h+t-1}^{t} \frac{C_{L-h-t}^{\ell-h}}{C_L^\ell} \mathcal{H}_{L-h-t}^{\ell-h,\,m}\left(\tfrac{\ell}{L}(m+t-\varepsilon k)\right). \tag{10.3}$$

**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. Случай $|J| \leqslant k$ и $|J| = H$ невозможен в силу условия $k < H$.

3. Если $|J| \leqslant k$ и $|J| < H$, то для алгоритма $a_J$ выполнены условия Леммы 10.1:

$$[\mu X{=}a_J] = [X_J \subseteq X][X'_J \subseteq \bar{X}].$$

В этом случае верхняя связность $u(a_J) = |X_J| = h$, неполноценность $q(a_J) = |J|$, оценка расслоения–связности (9.6) является равенством:

$$Q_\varepsilon = \sum_{J\colon |J|\leqslant k} P_J \mathcal{H}_{L-h-q}^{\ell-h,\,m+|J|-q}\left(\tfrac{\ell}{L}\big(m+|J|-\varepsilon k\big)\right), \quad P_J = \frac{C_{L-h-q}^{\ell-h}}{C_L^\ell}, \tag{10.4}$$

где $P_J = \mathsf{P}[\mu X{=}a_J]$ — вероятность получить алгоритм $a_J$ в результате обучения, $u = u(a_J)$ — верхняя связность, $q = q(a_J) = |J|$ — неполноценность алгоритма $a_J$.

Верхняя связность и неполноценность в каждом слое одинаковы у всех алгоритмов. Заменим сумму по алгоритмам суммой по слоям $t = 0, \ldots, H$, введя под знак суммы число алгоритмов в слое $C_{h+t-1}^t$. Поскольку при $t > k$ алгоритмы вообще не вносят вклад в оценку, из (10.4) следует требуемое равенство (10.3). ■

**Следствие 10.2.1.** *При $k < H$ вероятность того, что методом $\mu$ будет получен какой-либо алгоритм $a_J$ из слоя $m + t$, где $t = |J|$, равна*

$$P_t \equiv \mathsf{P}\big[n(\mu X, \mathbb{X}) = m{+}t\big] = C_{h+t-1}^t P_J = C_{h+t-1}^t \frac{C_{L-h-t}^{\ell-h}}{C_L^\ell}. \tag{10.5}$$

## §10.4 Вычислительные эксперименты

**Эксперимент 1: вычисление оценки расслоения–связности.** На рис. 10.2 слева показан график зависимости вероятности $P_t$ получить алгоритм из слоя $m + t$, вычисленной по формуле (10.5), от числа $t$. Справа показан график зависимости вероятности переобучения $Q_\varepsilon$ от порога $\varepsilon$. На каждом графике изображены 9 кривых, соответствующих различным значениям размерности $h = 1, \ldots, 9$.

Зависимость $P_t$ многомерной сети на отрезке $[0, H-1]$ унимодальна. Вероятность последнего слоя немного выше, так как для него верхняя связность равна

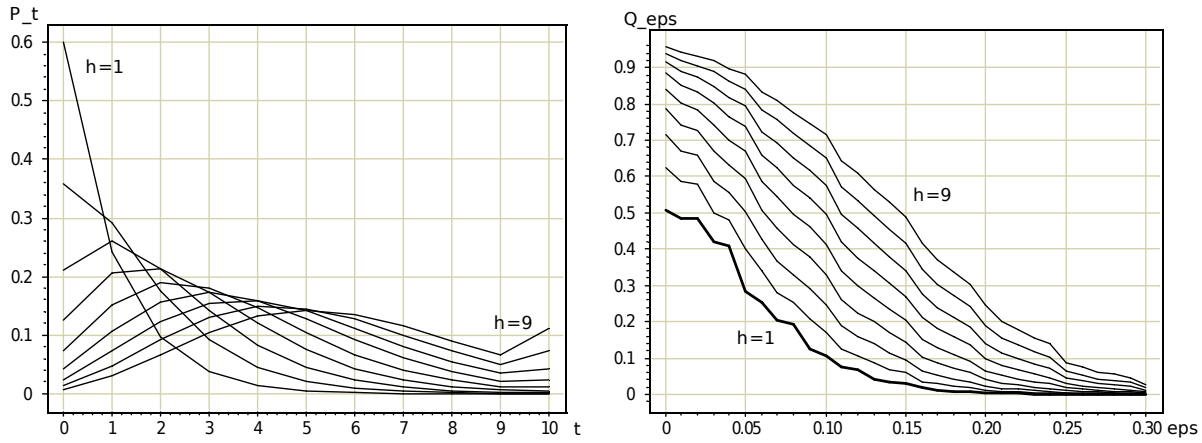



**Рис. 10.2.** Зависимости $P_t$ от $t$ (слева) и $Q_\varepsilon$ от $\varepsilon$ (справа) для монотонных сетей размерностей $h = 1, \ldots, 9$ при $L = 100,\ \ell = 60,\ m = 10,\ H = 10$.

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 $Q_\varepsilon$ от $\varepsilon$ для задач из репозитория UCI (sonar, breast-cancer-wisconsin) на методах классификации NaiveBayes, SVM, DecisionTree, NeuralNetwork [7]. Экспериментальные кривые переобученности $\hat{Q}_\varepsilon$, 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. 10.3, 10.4, 10.5 («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. 10.5. 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 бóльшую размерность, бóльшую переобученность и бóльшую ошибку на контроле.

## 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.

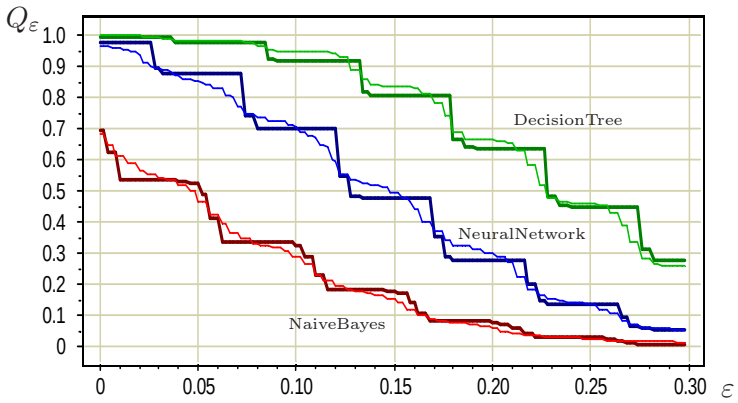



**Рис. 10.3.** Вероятность переобучения $Q_\varepsilon$ алгоритмов NaiveBayes, NeuralNetwork, DecisionTree и монотонных сетей размерностей, соответственно, $h = 8, 28, 46$. Средняя частота ошибок на обучении/контроле, соответственно: 0.28/0.32; 0.05/0.20; 0.02/0.26. Задача sonar, 20 признаков, $L = 208$, $\ell = 187$.

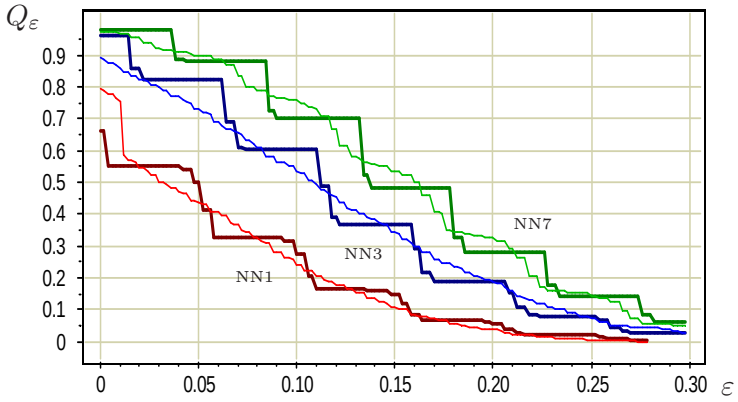



**Рис. 10.4.** Вероятность переобучения $Q_\varepsilon$ алгоритмов NeuralNetwork с числом нейронов в скрытом слое $1, 3, 7$ и монотонных сетей размерностей, соответственно, $h = 8, 22, 30$. Средняя частота ошибок на обучении/контроле, соответственно: 0.29/0.34; 0.12/0.23; 0.03/0.19. Задача sonar, 20 признаков, $L = 208$, $\ell = 187$.

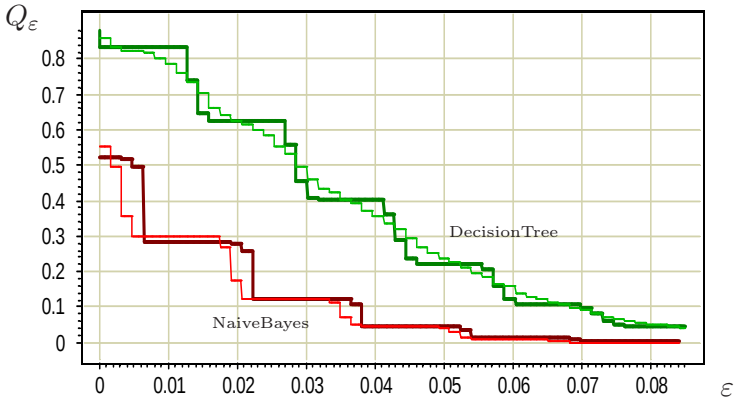



**Рис. 10.5.** Вероятность переобучения $Q_\varepsilon$ алгоритмов NaiveBayes, DecisionTree и монотонных сетей размерности $h = 1, 20$. Средняя частота ошибок на обучении/контроле, соответственно: 0.04/0.04; 0.03/0.06. Задача breast-cancer-wisconsin, 9 признаков, $L = 700$, $\ell = 630$.

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 вернёмся к функционалу равномерного отклонения, который используется в VC-теории, и покажем, что связность для него учесть можно, а расслоение — нельзя.

## Упражнения

**Задача 10.1 (1).** Доказать девять свойств монотонной сети.

**Задача 10.2 (1).** Привести пример выборки в $\mathbb{R}^2$, для которой множество линейных классификаторов (не обязательно всех возможных) порождает монотонную сеть алгоритмов.

**Задача 10.3 ($3^*$).** Привести пример выборки в $\mathbb{R}^h$, для которой множество линейных классификаторов (не обязательно всех возможных) порождает монотонную сеть алгоритмов.

**Задача 10.4 ($5^*$).** Сформулировать требования к семейству алгоритмов, при которых оценка расслоения–связности является точной.

**Задача 10.5 (5).** Получить оценку вероятности переобучения для многомерной монотонной сети, когда по каждой размерности $d$ задана своя высота $H_d$.

**Задача 10.6 (8).** Получить оценку вероятности переобучения для унимодальной $h$-мерной сети.

**Задача 10.7 (5).** Получить оценку вероятности переобучения для пучка $h$ монотонных цепей длины $H$.

# 11 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 [108, 59, 13]). 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.

## §11.1 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

$$Q_\varepsilon(\mu, \mathbb{X}) = \mathsf{P}\big[\delta(\mu X, X) \geqslant \varepsilon\big]$$

оценивается сверху вероятностью большого равномерного (по множеству алгоритмов $A$) отклонения частот в двух подвыборках:

$$Q_\varepsilon(\mu, \mathbb{X}) \leqslant \widetilde{Q}_\varepsilon(A, \mathbb{X}) = \mathsf{P} \max_{a \in A} \big[\delta(a, X) \geqslant \varepsilon\big] = \mathsf{P}\Big[\max_{a \in A} \delta(a, X) \geqslant \varepsilon\Big].$$

**Метод максимизации переобученности.** Принцип порождающих и запрещающих множеств может быть применён к функционалу $\widetilde{Q}_\varepsilon(A, \mathbb{X})$, если специальным образом ввести метод обучения. Эта идея принадлежит Илье Толстихину.

**Определение 11.1.** *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,

$$Q_\varepsilon(\mu, \mathbb{X}) = \widetilde{Q}_\varepsilon(A, \mathbb{X}).$$

**Лемма 11.1.** *Если метод $\mu$ максимизирует переобученность, то для каждого алгоритма $a \in A$ множество $X_a = \{x_{ab} \in \mathbb{X}\colon a \prec b\}$ является порождающим, а множество $X'_a = \{x_{ba} \in \mathbb{X}\colon b \prec a\}$ — запрещающим в смысле Гипотезы 9.1:*

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

**Доказательство.** Из того, что $\mu X = a$ следует

$$\delta(a, X) \geqslant \delta(b, X) \quad \text{для всех } b \in A. \tag{11.1}$$

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

$$\delta(b, X) = \nu(b, \bar{X}) - \nu(b, X) = \nu(a, \bar{X}) + \tfrac{1}{k} - \nu(a, X) > \delta(a, X),$$

что противоречит (11.1), 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

$$\delta(b, X) = \nu(b, \bar{X}) - \nu(b, X) = \nu(a, \bar{X}) - \nu(a, X) + \tfrac{1}{\ell} > \delta(a, X),$$

что противоречит (11.1), значит, $x_{ba} \in \bar{X}$. В силу его произвольности $X'_a \subseteq \bar{X}$. ■

**Оценка связности.** Из Леммы 11.1 и общей оценки расслоения–связности (9.2) следует оценка связности для вероятности равномерного отклонения.

**Теорема 11.2.** *Пусть векторы ошибок всех алгоритмов из $A$ попарно различны. Тогда имеет место верхняя оценка вероятности равномерного отклонения:*

$$\widetilde{Q}_\varepsilon(A, \mathbb{X}) \leqslant \sum_{a \in A} \frac{C^{\ell-u}_{L-u-d}}{C^\ell_L} \mathcal{H}^{\ell-u,\ m-d}_{L-u-d}\left(\tfrac{\ell}{L}(m - \varepsilon k)\right), \tag{11.2}$$

*где $u = u(a) = |X_a|$ — 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 (9.6) для пессимистичной МЭР. Единственное различие заключается в том, что неполноценность $q(a)$ заменяется на нижнюю связность $d(a)$. Неполноценность растёт с номером слоя линейно, соответственно, вклады алгоритмов в $Q_\varepsilon$ убывают экспоненциально. Нижняя связность $d(a)$ не превышает $q(a)$ и, как показывают эксперименты [34], концентрируется вокруг одного и того же значения во всех слоях, поэтому алгоритмы всех слоёв вносят примерно равный вклад в оценку (11.2). Таким образом, эта оценка учитывает связность, но не учитывает расслоение.

## §11.2 Техника цепных разложений

Метод оценивания вероятности равномерного отклонения, основанный на *цепных разложениях* (chain expansion), предложен Денисом Кочедыковым в [34].

**Теорема 11.3.** *Для любых* $\mathbb{X}$, $A$ *и любого* $\varepsilon \in [0,1]$

$$\widetilde{Q}_\varepsilon(A,\mathbb{X}) \leqslant \sum_{a\in A}\Big[\big\{s_a(\varepsilon)\big\} < \tfrac{\ell}{L}\Big]\frac{C^\ell_{L-d}}{C^\ell_L}h^{\ell,\,m-d}_{L-d}\big(s_a(\varepsilon)\big)\,. \tag{11.3}$$

*где* $d = d(a)$ *— нижняя связность алгоритма* $a$, $m = n(a,\mathbb{X})$ *— число ошибок алгоритма* $a$ *на генеральной выборке,* $s_a(\varepsilon) = \frac{\ell}{L}\big(m - \varepsilon k\big)$,

**Доказательство.** Обозначим через $U_a(X,\varepsilon)$ или просто $U_a$ индикатор переобученности алгоритма $a$ на выборке $X$:

$$U_a = \big[\delta(a,X) \geqslant \varepsilon\big] = \big[n(a,X) \leqslant s_a(\varepsilon)\big].$$

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*:

$$\widetilde{Q}_\varepsilon = \mathsf{P}\max_a U_a = \sum_{a\in A}\mathsf{P}\,U_a\prod_{b\in A\colon\, b<a}\bar{U}_b.$$

Оставим в цепном разложении только такие сомножители $\bar{U}_b$, что $b \prec a$. Остальные сомножители тривиально оценим сверху единицей, $\bar{U}_b \leqslant 1$, $b \not\prec a$. Тогда

$$\widetilde{Q}_\varepsilon \leqslant \sum_{a\in A}\mathsf{P}\prod_{b\in A\colon\, b\prec a}U_a\bar{U}_b.$$

Рассмотрим произведение $U_a\bar{U}_b$ для произвольных $b$ и $a$ таких, что $b \prec a$:

$$\begin{aligned}
U_a\bar{U}_b &= \big[n(a,X) \leqslant s_a(\varepsilon)\big]\big[n(b,X) > s_b(\varepsilon)\big] = \\
&= \big[s_a(\varepsilon) - \tfrac{\ell}{L} = s_b(\varepsilon) < n(b,X) \leqslant n(a,X) \leqslant s_a(\varepsilon)\big] = \\
&= \big[s_a(\varepsilon) - \tfrac{\ell}{L} < n(a,X) \leqslant s_a(\varepsilon)\big]\big[n(b,X) = n(a,X)\big] = \\
&= \big[n(a,X) = \lfloor s_a(\varepsilon)\rfloor\big]\big[\{s_a(\varepsilon)\} < \tfrac{\ell}{L}\big]\big[x_{ba}\in\bar{X}\big],
\end{aligned}$$

где $\lfloor s\rfloor$ — целая часть $s$, $\{s\}$ — дробная часть $s$. Тогда

$$\widetilde{Q}_\varepsilon \leqslant \sum_{a\in A}\mathsf{P}\prod_{b\in A\colon\, b\prec a}U_a\bar{U}_b = \sum_{a\in A}\mathsf{P}\prod_{b\in A\colon\, b\prec a}\big[n(a,X) = \lfloor s_a(\varepsilon)\rfloor\big]\big[\{s_a(\varepsilon)\} < \tfrac{\ell}{L}\big]\big[x_{ba}\in\bar{X}\big].$$

Сомножители, не зависящие от $b$, вынесем за знак произведения. Введём множество $X'_a = \big\{x_{ba}\in\mathbb{X}\colon b\prec a\big\}$ и заметим, что

$$\prod_{b\in A\colon\, b\prec a}\big[x_{ba}\in\bar{X}\big] = \big[X'_a\subseteq\bar{X}\big].$$

Таким образом,

$$\widetilde{Q}_\varepsilon \leqslant \sum_{a\in A} \big[\{s_a(\varepsilon)\} < \tfrac{\ell}{L}\big]\, \mathsf{P}\big[n(a,X) = \lfloor s_a(\varepsilon)\rfloor\big]\,\big[X'_a \subset \bar{X}\big].$$

Учитывая, что $|X'_a| = d(a)$, получим оценку (11.2). Теорема доказана. ■

Оценка (11.3) по своей структуре аналогична оценке (11.2). 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 (11.3) учитывает только нижнюю связность, что в итоге делает её более слабой по сравнению с оценкой (11.2). При этом обе оценки не учитывают расслоение.

## §11.3 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 $b_i = b_i(X) = [x_i \in X]$:

$$\begin{aligned} D(X) &= \max_{a\in A} \delta(a,X) = \max_{a\in A}\big(\nu(a,\bar{X}) - \nu(a,X)\big) = \\ &= \max_{a\in A} \sum_{i=1}^{L} I(a,x_i)\Big(\tfrac{1}{k}\underbrace{[x_i\in\bar{X}]}_{1-b_i} - \tfrac{1}{\ell}\underbrace{[x_i\in X]}_{b_i}\Big) = \\ &= \frac{L}{\ell k}\,\max_{a\in A} \sum_{i=1}^{L} I(a,x_i)\left(\frac{\ell}{L} - b_i\right). \end{aligned} \tag{11.4}$$

**Монотонная цепь алгоритмов** — это модельное семейство $A = \{a_0, a_1, \ldots, a_D\}$, в котором алгоритм $a_0$ допускает $m$ ошибок на генеральной выборке, а каждый последующий алгоритм хуже предыдущего на одном объекте (определение 8.8, стр. 73). Перенумеруем объекты $x_1, \ldots, x_L$ таким образом, чтобы

$$I(a_d, x_i) = \big[i \leqslant m + d\big], \quad i = 1,\ldots,L, \quad d = 0,\ldots,D. \tag{11.5}$$

**Теорема 11.4.** *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 $\varepsilon$:*

$$\mathsf{P}\Big[\max_{a\in A}\delta(a,X) \leqslant \varepsilon\Big] = \frac{G_L^\ell\big[g_L^-(\varepsilon), L\big]}{C_L^\ell},$$

*где $g_t^-(\varepsilon)$ — левая граница усечённого треугольника Паскаля:*

$$g_t^-(\varepsilon) = \tfrac{\ell}{L}(t - \varepsilon k)\big[m \leqslant t \leqslant m + D\big].$$

**Доказательство.** В случае монотонной цепи представление (11.4) упрощается после подстановки в него (11.5):

$$D(X) = \frac{L}{\ell k} \max_{d=0..D} \sum_{i=1}^{m+d} \left( \frac{\ell}{L} - b_i \right) = \frac{L}{\ell k} \max_{d=0..D} \left( \frac{\ell(m+d)}{L} - B_{m+d} \right),$$

где $B_{m+d} = b_1 + \cdots + b_{m+d}$. 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 $b_1, \ldots, b_L$ от «ожидаемого» числа единиц $\frac{\ell}{L}(m+d)$.

Найдём теперь вероятность того, что $D(X)$ не превышает $\varepsilon$:

$$\begin{aligned} \mathsf{P}\big[D(X) \leqslant \varepsilon\big] &= \mathsf{P}\Big[ \max_{d=0..D} \big(\tfrac{\ell}{L}(m+d) - B_{m+d}\big) \leqslant \tfrac{\ell}{L}\varepsilon k \Big] = \\ &= \mathsf{P} \prod_{d=0}^{D} \Big[ \tfrac{\ell}{L}\big(m+d-\varepsilon k\big) \leqslant B_{m+d} \Big] = \mathsf{P} \prod_{t=1}^{L} \big[ g_t^-(\varepsilon) \leqslant B_t \big]. \end{aligned}$$

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. ■

## 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 применим оценку расслоения–связности к семейству пороговых конъюнкций над вещественными признаками. Это первый пример не модельного семейства, широко используемого при решении прикладных задач.

## Упражнения

**Задача 11.1** ($5^*$)**.** Получить точную оценку вероятности равномерного отклонения $\widetilde{Q}_\varepsilon(A, \mathbb{X})$ для унимодальной цепи алгоритмов.

**Задача 11.2** ($10^*$)**.** Получить точную оценку $\widetilde{Q}_\varepsilon(A, \mathbb{X})$ для произвольного множества алгоритмов, заданного графом расслоения–связности.

# 12 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*.

## §12.1 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* $f_j \colon \mathbb{X} \to \mathbb{R}$, $j = 1, \ldots, n$.

Логические методы классификации основаны на построении композиций информативных, хорошо интерпретируемых логических закономерностей.

**Логические правила и требование интерпретируемости.** *Предикатом* будем называть произвольное отображение вида $r \colon \mathbb{X} \to \{0, 1\}$. Если $r(x) = 1$, то будем говорить, что предикат *выделяет* объект $x$.

*Правилом* (rule) будем называть предикат из некоторого фиксированного семейства предикатов $R$, обладающих свойством *интерпретируемости* — они достаточно просты и допускают запись на естественном языке в терминах предметной области. Эти требования формализуются в самой конструкции семейства $R$. Мы рассмотрим наиболее распространённый вид правил — *пороговые конъюнкции*:

$$r(x;\theta) = \prod_{j \in J} \big[ f_j(x) \lessgtr_j \theta^j \big], \tag{12.1}$$

где $J \subseteq \{1, \ldots, n\}$ — подмножество признаков, $\theta^j \in \mathbb{R}$ — *порог* по $j$-му признаку, $\theta = (\theta^j)_{j \in J}$ — *вектор порогов*, $\lessgtr_j$ — один из знаков сравнения $\{\leqslant, \geqslant\}$.

Число $|J|$ называется *рангом конъюнкции*. Обычно оно ограничивается сверху в угоду всё той же интерпретируемости (как утверждают психологи, людям трудно 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.** *Логической закономерностью* класса $y \in \mathbb{Y}$ будем называть правило $r \in R$, выделяющее в 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:

$$p(r, X) = \#\big\{x_i \in X \;\big|\; r(x_i) = 1,\ y_i = y\big\} \to \max;$$
$$n(r, X) = \#\big\{x_i \in X \;\big|\; r(x_i) = 1,\ y_i \neq y\big\} \to \min.$$

Введём также число положительных и отрицательных примеров в выборке $X$:

$$P(X) = \#\{x_i \in X \colon y_i = y\};$$
$$N(X) = \#\{x_i \in X \colon y_i \neq y\}.$$

На практике два критерия $p, n$ сворачивают в один *эвристический критерий информативности* $\mathscr{H}(p, n) \to \max$. Известны десятки критериев, дающих немного разные результаты, но ни один из них нельзя назвать безусловно предпочтительным [85, 69]. Некоторые критерии являются чисто эвристическими, другие имеют более строгие обоснования в теории информации и математической статистике.

**Пример 12.1 (эвристические критерии).** *Критерий точности* (accuracy) — это доля правильных классификаций, то есть отношение общего числа выделенных положительных и невыделенных отрицательных примеров к длине выборки:

$$\mathscr{H}(p, n) = \frac{p + N - n}{P + N}. \tag{12.2}$$

Иногда нормировку убирают. Иногда вводят штраф $\gamma > 1$ за выделение отрицательных объектов, поощряя тем самым поиск «более чистых» закономерностей [29]:

$$\mathscr{H}(p, n) = p - \gamma n.$$

**Пример 12.2 (критерий бустинга).** В теории бустинга [99] возникает критерий разности квадратных корней:

$$\mathscr{H}(p, n) = \sqrt{p} - \sqrt{n}.$$

В задачах классификации с несбалансированными классами (когда отношение $P : N$ существенно отлично от единицы) используется также нормированный критерий

$$\mathscr{H}(p, n) = \sqrt{p/P} - \sqrt{n/N}.$$

**Пример 12.3 (точный тест Фишера).** Если принять в качестве нулевой гипотезы предположение, что предикаты $r(x)$ и $\big[y(x) = y\big]$ являются независимыми случайными величинами, то при фиксированном числе выделяемых объектов $(p + n)$ число выделяемых отрицательных примеров $n$ подчиняется гипергеометрическому распределению $h_{P+N}^{p+n,\,N}(n)$. Достигаемый уровень значимости $\alpha = \mathcal{H}_{P+N}^{p+n,\,N}(n)$ равен вероятности чисто случайной реализации наблюдаемого или ещё меньшего значения $n$. Чем меньше $\alpha$, тем менее правдоподобна гипотеза о независимости. В качестве критерия информативности берут $-\log \alpha$ как меру неслучайности взаимосвязи $r$ и $y$:

$$\mathscr{H}(p, n) = -\log \mathcal{H}_{P+N}^{p+n,\,N}(n).$$

**Пример 12.4 (энтропийный критерий).** Асимптотическим приближением гипергеометрического критерия является *энтропийный критерий информативности* или *выигрыш информации* (information gain, IGain) [89]:

$$\mathscr{H}(p,n) = h\left(\frac{P}{P+N}\right) - \frac{p+n}{P+N}\,h\left(\frac{p}{p+n}\right) - \frac{P+N-p-n}{P+N}\,h\left(\frac{P-p}{P+N-p-n}\right),$$

где $h(q) = -q\log_2 q - (1-q)\log_2(1-q)$ — функция энтропии пары исходов с вероятностями $q$ и $1-q$.

**Пример 12.5 (критерий Джини).** На практике используют также *индекс Джини* (Gini impurity), отличающийся от IGain только функцией $h(q) = 4q(1-q)$, которая неплохо аппроксимирует функцию энтропии пары исходов [60].

**Композиции закономерностей.** Каждая закономерность выделяет лишь часть выборки и относит её к одному из классов. Поэтому классификатор $a\colon \mathbb{X} \to \mathbb{Y}$ строят из большого числа закономерностей, объединяя их в композицию, например, с помощью *взвешенного голосования*:

$$a(x) = \arg\max_{y\in\mathbb{Y}} \sum_{r\in R_y} w_r r(x), \tag{12.3}$$

где $R_y$ — множество закономерностей класса $y$, $w_r \geqslant 0$ — 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 [73] или методом опорных векторов [70]. 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 [66].

## §12.2 Задача предсказания информативности

Итак, имеется три типа параметров, обучаемых по выборке, и, соответственно, три возможных 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 $r$.

Естественно полагать, что хороший алгоритм $a(x)$ можно построить только из непереобученных закономерностей, имеющих высокую информативность как на обучающей, так и на контрольной выборке.

Нашей целью будет построение такого критерия $\mathscr{H}'$, который предсказывал бы значение заданного стандартного критерия $\mathscr{H}$ на скрытой выборке после оптимизации порогов $\theta^j$. *Предсказанную информативность* $\mathscr{H}'$ предполагается затем использовать в качестве критерия выбора подмножеств признаков $J$.

Проблемой переобучения весов $w_r$ мы заниматься не будем, отчасти потому, что она уже неплохо исследована [98, 56, 66, 79], отчасти, чтобы сосредоточиться на применении оценки расслоения–связности к семейству пороговых конъюнкций.

Плюс предлагаемой схемы в том, что она не требует что-либо менять в существующих методах обучения закономерностей, за исключением критерия информативности. Поэтому её легко встраивать в готовые библиотеки алгоритмов.

Минус же в том, что не учитывается переобучение, связанное с оптимизацией подмножеств $J$ и весов $w_r$. Таким образом, мы пока не можем дать окончательного решения проблемы переобучения для взвешенного голосования конъюнкций.

**Монотонность критериев информативности.** Введём для фиксированного класса $y \in \mathbb{Y}$ на множестве правил $R$ индикатор ошибки

$$I(r, x_i) = \big[r(x_i) \neq [y_i{=}y]\big], \quad r \in R, \quad x_i \in \mathbb{X}. \tag{12.4}$$

Множество правил $R$ порождает конечное множество векторов ошибок $\big(I(r, x_i)\big)_{i=1}^{L}$, на котором естественным образом вводятся отношения порядка и предшествования.

Следующая теорема показывает, что большинство известных критериев информативности являются монотонными в смысле Определения 9.2 (стр. 9.2). Отсюда вытекает, что оценка расслоения–связности (9.6) применима независимо от того, какой из критериев информативности используется для обучения правил.

**Теорема 12.1.** *Если функция $\mathscr{H}(p, n)$ строго монотонно возрастает по $p$ и строго монотонно убывает по $n$, то критерий $M(r, X) = -\mathscr{H}\big(p(r, X), n(r, X)\big)$ является монотонным относительно индикатора ошибки* (12.4)*, и, соответственно, метод максимизации информативности является монотонным методом обучения правил.*

**Доказательство.** Обозначим проекцию вектора ошибок $r$ на подвыборку $X \subseteq \mathbb{X}$ через $\vec{r}_X = \big(I(a, x_i)\big)_{x_i \in X}$. Напомним определение монотонного критерия $M$: для любых $X \subset \mathbb{X}$ и $r, v \in R$, таких, что $\vec{r}_X < \vec{v}_X$, справедливо $M(r, X) < M(v, X)$.

Возьмём произвольные $X \subset \mathbb{X}$ и $r, v \in R$, такие, что $\vec{r}_X < \vec{v}_X$. Отсюда и из определения индикатора ошибки немедленно следует $p(r, X) \geqslant p(v, X)$ и $n(r, X) \leqslant n(v, X)$, причём хотя бы одно из этих двух неравенств строгое. Следовательно,

$$\mathscr{H}\big(p(r, X), n(r, X)\big) > \mathscr{H}\big(p(v, X), n(v, X)\big),$$

что равносильно $M(r, X) < M(v, X)$. Таким образом, $M$ удовлетворяет определению монотонного критерия. ■

**Критерий предсказанной точности.** Число ошибок правила $r$ относительно индикатора ошибки (12.4) договоримся обозначать буквой $m$ вместо обычного $n$, чтобы не путать с числом отрицательных примеров:

$$m(r, X) = \sum_{x \in X} I(r, x) \;=\; n(r, X) + P(X) - p(r, X).$$

Тогда частота ошибок $\nu(r,X) = m(r,X)/|X|$ связана с критерием точности (12.2):

$$\mathscr{H}\big(p(r,X), n(r,X)\big) = 1 - \nu(r,X),$$

а минимизация эмпирического риска эквивалентна максимизации точности. Благодаря этому модифицированный критерий $\mathscr{H}'$ строится особенно просто. Запишем оценку расслоения–связности (9.6): для любого $\varepsilon \in (0,1)$

$$\mathsf{P}\big[\nu(r,\bar{X}) - \nu(r,X) \geqslant \varepsilon\big] \leqslant \sum_{r\in R} \frac{C^{\ell-u}_{L-u-q}}{C^{\ell}_{L}} \mathcal{H}^{\ell-u,\,m-q}_{L-u-q}\left(\tfrac{\ell}{L}(m-\varepsilon k)\right), \tag{12.5}$$

где $u = u(r)$ — верхняя связность, $q = q(r)$ — неполноценность, $m = m(r,\mathbb{X})$ — число ошибок правила $r$ на генеральной выборке. Обозначим правую часть неравенства через $\eta_{\ell,k}(\varepsilon)$, обратную к ней функцию — через $\varepsilon_{\ell,k}(\eta)$. Тогда с вероятностью не менее $(1-\eta)$ справедлива оценка

$$\nu(r,\bar{X}) \leqslant \nu(r,X) + \varepsilon_{\ell,k}(\eta),$$

или, после перехода от частот к точностям,

$$\mathscr{H}\big(p(r,\bar{X}), n(r,\bar{X})\big) \geqslant \mathscr{H}\big(p(r,X), n(r,X)\big) - \varepsilon_{\ell,k}(\eta).$$

Взяв $\eta = \frac{1}{2}$, получим в правой части нижнюю оценку медианы точности на скрытой выборке. Её и примем в качестве предсказания информативности:

$$\mathscr{H}'(p,n) = \mathscr{H}(p,n) - \varepsilon_{\ell,k}(\tfrac{1}{2}).$$

В общем случае столь простой взаимосвязи между критерием $\mathscr{H}$ и частотой ошибок $\nu$ уже нет, и нужно раздельно оценивать $p(r,\bar{X})$ снизу и $n(r,\bar{X})$ сверху.

**Ошибки I и II рода.** Если $r$ — закономерность класса $y$, то невыделение объекта класса $y$ называют «пропуском цели» или *ошибкой первого рода*, а выделение объекта чужого класса — «ложной тревогой» или *ошибкой второго рода*. Генеральная выборка разбивается на два подмножества $\mathbb{X}' = \{x_i \in \mathbb{X}\colon y_i = y\}$ и $\mathbb{X}'' = \{x_i \in \mathbb{X}\colon y_i \neq y\}$. Таким образом, возникают ещё два определения индикатора ошибки:

$$\begin{aligned} I'(r,x_i) &= \big[r(x_i)=0\big]\big[y_i = y\big] = I(r,x_i)\big[x_i \in \mathbb{X}'\big]; \\ I''(r,x_i) &= \big[r(x_i)=1\big]\big[y_i \neq y\big] = I(r,x_i)\big[x_i \in \mathbb{X}''\big]; \end{aligned}$$

и, соответственно, два определения числа и частоты ошибок:

$$\begin{aligned} m'(r,X) &= P(X) - p(r,X); & \nu'(r,X) &= m'(r,X)/|X|; \\ m''(r,X) &= n(r,X); & \nu''(r,X) &= m''(r,X)/|X|. \end{aligned}$$

В силу тождества $I'(r,x_i) + I''(r,x_i) = I(r,x_i)$ имеет место разложение общего числа ошибок и общей частоты ошибок на ошибки I и II рода:

$$m'(r,X) + m''(r,X) = m(r,X); \qquad \nu'(r,X) + \nu''(r,X) = \nu(r,X).$$

Выпишем оценки расслоения–связности отдельно для частот $\nu'$ и $\nu''$, предполагая, что метод обучения является монотонным относительно индикатора ошибки $I$. Для обеих оценок множество правил $R$ и метод обучения $\mu$ одинаковы. Следовательно, одинаковы также порождающие и запрещающие множества $X_r$, $X'_r$, граф расслоения–связности и вероятности получения правил $P_r = \mathsf{P}[\mu X = r]$. Леммы 7.2 и 9.3 остаются в силе. Однако Теоремы 7.4 и 9.4 требуют ревизии, поскольку теперь частоты $\nu'$ и $\nu''$ определяются по своим индикаторам ошибки $I'$ и $I''$, отличным от общего индикатора $I$, породившего граф расслоения–связности.

**Теорема 12.2 (оценка расслоения–связности для ошибок I и II рода).** *Для произвольной выборки $\mathbb{X}$, произвольного монотонного метода обучения $\mu$ и произвольного $\varepsilon \in (0,1)$*

$$Q'_\varepsilon = \mathsf{P}\big[\nu'(r,\bar{X}) - \nu'(r,X) \geqslant \varepsilon\big] \;\leqslant\; \sum_{r\in R} \frac{C^{\ell-u}_{L-u-q}}{C^{\ell}_{L}} \mathcal{H}^{\ell-u,\,m'-q'}_{L-u-q}\left(\tfrac{\ell}{L}(m'-\varepsilon k)\right); \qquad (12.6)$$

$$Q''_\varepsilon = \mathsf{P}\big[\nu''(r,\bar{X}) - \nu''(r,X) \geqslant \varepsilon\big] \;\leqslant\; \sum_{r\in R} \frac{C^{\ell-u}_{L-u-q}}{C^{\ell}_{L}} \mathcal{H}^{\ell-u,\,m''-q''}_{L-u-q}\left(\tfrac{\ell}{L}(m''-\varepsilon k)\right); \qquad (12.7)$$

*где $u = |X_r|$ — верхняя связность правила $r$, $q = |X'_r|$, $q' = |X'_r \cap \mathbb{X}'|$, $q'' = |X''_r \cap \mathbb{X}''|$ — неполноценность правила $r$ относительно индикаторов ошибки $I, I', I''$ соответственно, $m' = m'(r,\mathbb{X})$, $m'' = m''(r,\mathbb{X})$ — число ошибок правила $r$ на генеральной выборке относительно индикаторов ошибки $I', I''$ соответственно.*

Обозначим правые части неравенств (12.6) и (12.7) через $\eta'_{\ell,k}(\varepsilon)$ и $\eta''_{\ell,k}(\varepsilon)$ соответственно, а обратные к ним функции — через $\varepsilon'_{\ell,k}(\eta)$ и $\varepsilon''_{\ell,k}(\eta)$. Тогда с вероятностью не менее $(1-\eta)$ справедливы оценки

$$\nu'(r,\bar{X}) \leqslant \nu'(r,X) + \varepsilon'_{\ell,k}(\eta),$$
$$\nu''(r,\bar{X}) \leqslant \nu''(r,X) + \varepsilon''_{\ell,k}(\eta).$$

Возьмём $\eta = \frac{1}{2}$, чтобы получить медианные оценки. Возвращаясь от частот ошибок $\nu', \nu''$ к обозначениям $p, n$, получим, с вероятностью не менее $\frac{1}{2}$,

$$p(r,\bar{X}) \geqslant k\big(p(r,X)/\ell - \varepsilon'_{\ell,k}(\tfrac{1}{2}) - \delta\big),$$
$$n(r,\bar{X}) \leqslant k\big(n(r,X)/\ell + \varepsilon''_{\ell,k}(\tfrac{1}{2})\big),$$

где $\delta = \frac{1}{\ell}P(X) - \frac{1}{k}P(\bar{X})$ — поправка на нестратифицированность классов, которая равна нулю, если доли положительных примеров в обучении и контроле одинаковы. Подставляя правые части этих оценок в заданный стандартный критерий информативности $\mathscr{H}$, получим *критерий предсказанной информативности*:

$$\mathscr{H}'(p,n) = \mathscr{H}\Big(k\big(p/\ell - \varepsilon'_{\ell,k}(\tfrac{1}{2}) - \delta\big),\; k\big(n/\ell + \varepsilon''_{\ell,k}(\tfrac{1}{2})\big)\Big). \qquad (12.8)$$

**Переход от ненаблюдаемой оценки к наблюдаемой.** Функции $\varepsilon'_{\ell,k}$, $\varepsilon''_{\ell,k}$ предсказывают число ошибок I и II рода на контрольной выборке длины $k$ по числу ошибок I и II рода на обучающей выборке длины $\ell$ и графу расслоения–связности. При этом для построения графа используется полная выборка $\mathbb{X}$ с известными классификациями $y_i$ всех объектов $x_i \in \mathbb{X}$. Значит, выборка $\mathbb{X}$ всё же предполагается наблюдаемой. Хотелось бы использовать все имеющиеся данные $\mathbb{X}$ для обучения закономерностей, а информативность предсказывать для некоторой неизвестной выборки $\bar{\mathbb{X}}$.

Допустим, что реализовалось одно из равновероятных разбиений *супер-выборки* $\mathscr{X} = \mathbb{X} \sqcup \bar{\mathbb{X}}$ длины $L+K$ на наблюдаемую $\mathbb{X}$ длины $L$ и скрытую $\bar{\mathbb{X}}$ длины $K$. Если бы мы знали скрытую выборку, то могли бы, повторив в точности все выкладки, построить функции $\varepsilon'_{L,K}$, $\varepsilon''_{L,K}$ для предсказания информативности на скрытой выборке $\bar{\mathbb{X}}$ по наблюдаемым значениям $p = p(r,\mathbb{X})$ и $n = n(r,\mathbb{X})$. По аналогии с (12.8) имеем *ненаблюдаемый (unobservable) критерий предсказанной информативности*:

$$\mathscr{H}'_{\text{un}}(p,n) = \mathscr{H}\Big(K\big(p/L - \varepsilon'_{L,K}(\tfrac{1}{2}) - \delta\big),\ K\big(n/L + \varepsilon''_{L,K}(\tfrac{1}{2})\big)\Big). \tag{12.9}$$

Однако мы не можем вычислить величины $\varepsilon'_{L,K}$, $\varepsilon''_{L,K}$, поскольку выборка $\bar{\mathbb{X}}$ неизвестна. Заменим их наблюдаемыми $\varepsilon'_{\ell,k}$, $\varepsilon''_{\ell,k}$ и пренебрежём поправкой $\delta$. Получим *наблюдаемый (observable) критерий предсказанной информативности*:

$$\mathscr{H}'_{\text{ob}}(p,n) = \mathscr{H}\Big(K\big(p/L - \varepsilon'_{\ell,k}(\tfrac{1}{2})\big),\ K\big(n/L + \varepsilon''_{\ell,k}(\tfrac{1}{2})\big)\Big). \tag{12.10}$$

Проделанная замена основана на гипотезе, что оценки переобученности $\varepsilon'_{\ell,k}$, $\varepsilon''_{\ell,k}$ существенно зависят от характеристик расслоения и связности семейства правил $R$, но слабо зависят от длины выборки.

Для проверки этой гипотезы Мариной Дударенко был проведён следующий эксперимент. В пространстве $\mathbb{R}^2$ синтезировались модельные супер-выборки $\mathscr{X}$ с классификациями $y_i = r(x_i,\theta)$, на которые накладывался шум — ответы $y_i$ на некоторых объектах $x_i$ инвертировались. Уровень шума варьировался от 0% до 50% объектов. Менялось также распределение шума — вблизи границы классов, равномерно, вдали от границы классов. Каждая супер-выборка многократно разбивалась случайным образом на наблюдаемую и скрытую, $\mathscr{X} = \mathbb{X} \sqcup \bar{\mathbb{X}}$. По подвыборке $\mathbb{X}$ вычислялись оценки $\varepsilon'_{\ell,k}$, $\varepsilon''_{\ell,k}$. По супер-выборке $\mathscr{X}$ вычислялись оценки $\varepsilon'_{L,K}$, $\varepsilon''_{L,K}$. Использовался энтропийный критерий информативности $\mathscr{H}$.

Рис. 12.1 показывает, что зависимость ненаблюдаемой информативности от наблюдаемой близка к линейной, даже несмотря на малый объём данных $L = K = 60$.

На рис. 12.2 показана зависимость линейной корреляции $\text{corr}(\mathscr{H}_{\text{un}}, \mathscr{H}_{\text{ob}})$ от длины супер-выборки. Корреляция становится близка к единице на выборках порядка нескольких десятков объектов. При увеличении уровня шума необходимая длина выборки увеличивается, но не критично.

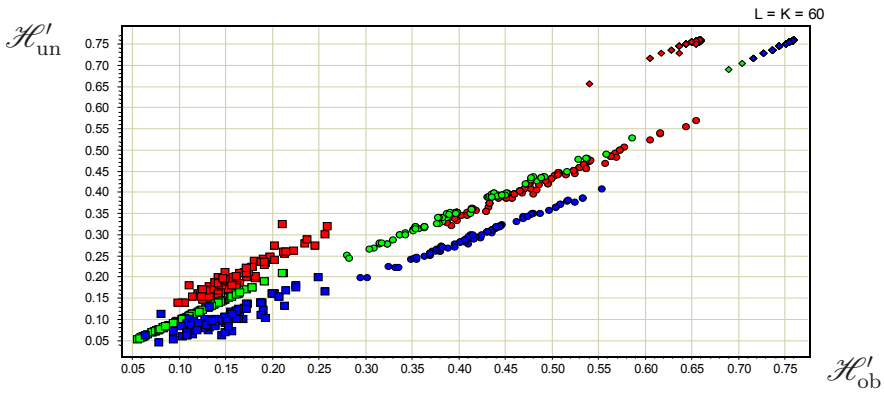



**Рис. 12.1.** Зависимость $\mathscr{H}'_{\text{un}}$ от $\mathscr{H}'_{\text{ob}}$ при $L = K = 60$ для двумерных модельных выборок при различном уровне шума ($\diamond$ — 0%, $\circ$ — 10%, $\square$ — 50%) и различном распределении шумовых объектов (синий — на границе классов, зелёный — равномерно, красный — вдали от границы классов). Точки соответствуют разбиениям супер-выборки на наблюдаемую и скрытую.

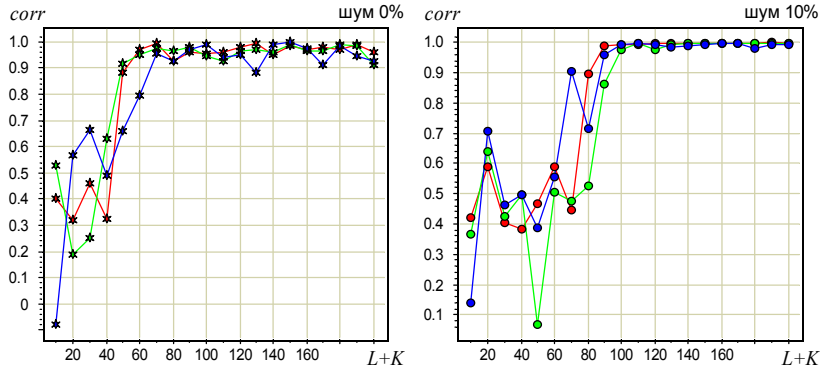



**Рис. 12.2.** Зависимость корреляции $\mathscr{H}_{\text{un}}$ и $\mathscr{H}_{\text{ob}}$ от длины супер-выборки и зашумлённости данных при различном уровне и распределении шума.

## §12.3 Классы эквивалентности пороговых конъюнкций

Нашей ближайшей целью будет поиск конструктивного описания множества всех правил $R$ вида (12.1) с попарно различными векторами ошибок для эффективного вычисления оценок расслоения–связности (12.6) и (12.7).

Будем полагать, что в правилах вида (12.1) множество признаков $J$ фиксировано, и все знаки сравнения $\lesseqgtr_j$ суть $\leqslant$ (возможность выбора знаков сравнения можно реализовать, добавив $n$ признаков $-f_1(x), \ldots, -f_n(x)$).

Определим индикатор ошибки $I(r, x_i) = \big[r(x_i) \neq [y_i{=}y]\big]$, для любого правила $r \in R$ и любого объекта $x_i \in \mathbb{X}$, где $y \in \mathbb{Y}$ — фиксированный класс.

Чтобы вычислить оценку расслоения–связности (9.6), необходимо выполнить суммирование по множеству всех правил с попарно различными векторами ошибок. Для этого введём отношение эквивалентности на правилах и рассмотрим описание классов эквивалентности, предложенное Андреем Ивахненко в [30, 31, 112].

**Классы эквивалентности правил.** 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 $1, \ldots, L$. Значения порогов $\theta^j$ в правилах (12.1) также выбираются из целых значений $0, \ldots, L$. Таким образом, и объекты, и правила описываются элементами одного и того же дискретного множества $\Theta = \{0, \ldots, L\}^{|J|}$. 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: $(u \leqslant v) \;\leftrightarrow\; \forall j \in J\ (u^j \leqslant v^j)$. Положим $(u < v) \;\leftrightarrow\; (u \leqslant v$ и $u \neq v)$.

**Пример 12.6.** На рис. 12.3 показан пример задачи с $n = 2$ признаками, $L = 10$ объектами и семейство правил $r(x; \theta) = [x \leqslant \theta] = [x^1 \leqslant \theta^1]\,[x^2 \leqslant \theta^2]$. Каждому правилу соответствует узел целочисленной прямоугольной сетки $(\theta^1, \theta^2) \in \Theta = \{0, \ldots, L\}^2$.

**Лемма 12.3.** *Пусть $E \subseteq R$ — класс эквивалентности правил. Тогда классу $E$ принадлежит правило $r(x; \theta_E)$, где $\theta_E^j = \min\limits_{\theta \in E} \theta^j$ для всех $j \in J$.*

**Доказательство.** В силу эквивалентности и бинарности правил из $E$, предикат $r_E(x) = \prod\limits_{\theta \in E} r(x; \theta)$ 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 (12.1):

$$r_E(x) = \prod_{\theta \in E} \prod_{j \in J} [x^j \leqslant \theta^j] = \prod_{j \in J} [x^j \leqslant \min_{\theta \in E} \theta^j] = \prod_{j \in J} [x^j \leqslant \theta_E^j] = r(x; \theta_E).$$

Таким образом, правило $r(x; \theta_E)$ также принадлежит $E$. Лемма доказана. ■

Будем называть правило $r_E(x) = r(x; \theta_E)$ *стандартным представителем* класса эквивалентности $E$. На рис. 12.3 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 $\theta_S^j = \max\limits_{x \in S} x^j$, $j \in J$. 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 $\theta_\varnothing^j = 0$, $j \in J$.

**Теорема 12.4.** *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 12.3 уменьшение порога $\theta_E^j$ на единицу приводит к уменьшению значения

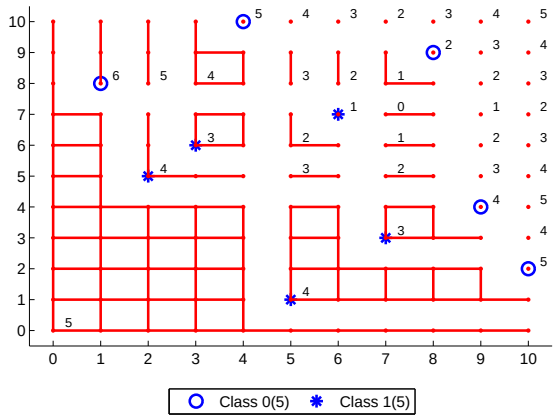



**Рис. 12.3.** Двумерная выборка из $L = 10$ объектов, по 5 объектов в каждом классе. Объекты отмечены крупными точками. Правилам соответствуют мелкие точки в узлах сетки. Эквивалентные правила соединены отрезками. Рядом с каждым классом эквивалентности указано число ошибок на генеральной выборке $m(r, \mathbb{X})$.

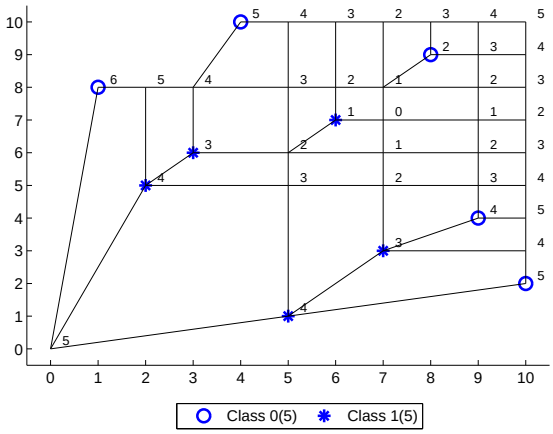



**Рис. 12.4.** Граф связей между стандартными представителями классов эквивалентных правил для выборки, представленной на Рис. 12.3.

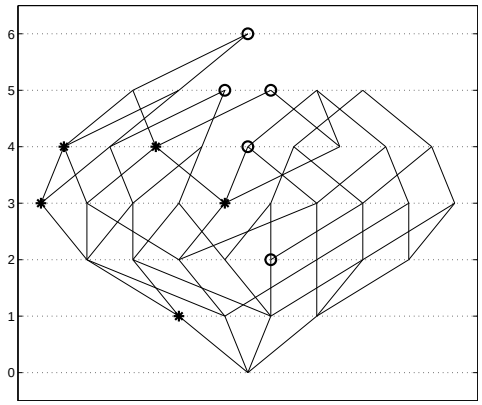


**Рис. 12.5.** Граф расслоения–связности, изоморфный графу связей на Рис. 12.4. По вертикальной оси отложено число ошибок правил $m(r, \mathbb{X})$.

$r(x;\theta_E)$ с 1 до 0 на некотором объекте $x \in \mathbb{X}$, таком, что $x \leqslant \theta_E$ 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 $S = \left\{x \in \mathbb{X} \;\middle|\; x \leqslant \theta_E \text{ и } \exists j \in J\colon x^j = \theta_E^j\right\}$ 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. ■

Из теоремы следует, что $|S| \leqslant |J|$, в силу предположения, что значения каждого признака попарно различны на всей выборке $\mathbb{X}$.

**Свойства граничных подмножеств.** Обозначим через $M_j$ множество всех граничных подмножеств мощности $j$ и перечислим их основные свойства.

1. $M_1$ состоит из всех $L$ одноэлементных подмножеств, $M_1 = \big\{\{x_1\},\ldots,\{x_L\}\big\}$.
2. $M_2$ состоит из всех пар несравнимых объектов из $\mathbb{X}$.
3. Мощность граничных подмножеств не может превышать ранга конъюнкции: $M_j = \varnothing$ при $j > |J|$.
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 $j = 2,\ldots,|J|$ к каждому подмножеству $S' \in M_{j-1}$ 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 $M_j$.

Для вычисления оценок расслоения–связности этот алгоритм не достаточно эффективен, так как он не перебирает правила по слоям снизу вверх и не подсчитывает характеристики связности и неполноценности каждого правила.

**Графические представления классов эквивалентности.** На рис. 12.4 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 $m(r_E, \mathbb{X})$. На Рис. 12.4 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 $m(r_E, \mathbb{X}) = m$, см. Рис. 12.5. Слои располагаются в порядке возрастания числа ошибок снизу вверх.

## §12.4 Послойное вычисление оценки расслоения–связности

Рассмотрим теперь алгоритм перебора правил, позволяющий эффективно вычислять характеристики связности и неполноценности 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, $u(\theta) = |X_\theta|$ — верхняя связность;

$X'_\theta$ — запрещающее множество, $q(\theta) = |X'_\theta|$ — неполноценность;

$q'(\theta), q''(\theta)$ — неполноценности отдельно по ошибкам I и II рода;

$m(\theta) = m(r, \mathbb{X})$ — число ошибок правила на генеральной выборке;

$m'(\theta), m''(\theta)$ — число ошибок I и II рода правила на генеральной выборке.

Будем строить только такие векторы порогов $\theta'$, которые являются граничными точками и, согласно Теореме 12.4, одновременно стандартными представителями классов эквивалентности правил. Построение разбивается на два этапа, см. Алгоритм 12.1.

На первом этапе (шаги 1–5) строятся все соседние правила $r(x;\theta')$, получаемые из $\theta$ путём уменьшения некоторых порогов, $\theta' \leqslant \theta$. Для этого из граничного подмножества $S(\theta) = \{x \in S \mid \exists j \in J\colon x^j = \theta^j\}$ 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, $\theta' \geqslant \theta$. 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

**Алгоритм 12.1.** 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:**

$V_\theta$, $X_\theta$, $X'_\theta$, $m(\theta)$, $m'(\theta)$, $m''(\theta)$, $q(\theta)$, $q'(\theta)$, $q''(\theta)$, $u(\theta)$.

*Этап 1 — построение окрестных правил путём уменьшения порогов:*

$V_\theta := \varnothing$;
**для всех** $x \in \mathbb{X}$ таких, что $r(x;\theta) = 1$ и $\exists j \in J\colon \theta^j = x^j$
  **для всех** $j \in J$ таких, что $x^j = \theta^j$
    $\theta'^j := \max\bigl\{x_i^j \bigm| x_i \in \mathbb{X},\ x_i < \theta\bigr\}$;
  Добавить$(\theta, \theta', x)$;

*Подготовка к этапу 2 — поиск максимальной граничной точки $\bar\theta$:*

**для всех** $j \in J$
  $\bar\theta^j := \min\bigl\{L, x^j \bigm| x \in \mathbb{X},\ x^j > \theta^j,\ x^t \leqslant \theta^t,\ t \neq j\bigr\}$;

*Этап 2 — построение окрестных правил путём увеличения порогов:*

**для всех** $j \in J$
  **для всех** $x$ таких, что $\theta^j < x^j \leqslant \bar\theta^j$
    **если** $x < \bar\theta$ и $x$.проверен = нет **то**
      Проверить$(x)$;

**ПРОЦЕДУРА** Проверить$(x)$
**для всех** $j \in J$ таких, что $\theta^j < x^j$
  **для всех** $\tilde x$ таких, что $\theta^j < \tilde x^j < x^j$
    **если** $\theta < \tilde x < x$ **то**
      $x$.проверен := плохой;
      **если** $\tilde x$.проверен = нет **то** Проверить$(\tilde x)$;
      **выход**;
$x$.проверен := хороший;
$\theta'^j := \max\{\theta^j, x^j\}$, для всех $j \in J$;
Добавить$(\theta, \theta', x)$;

**ПРОЦЕДУРА** Добавить$(\theta, \theta', x_i)$
добавить $\theta'$ в список $V_\theta$;
**если** $r(x_i;\theta) = [y_i = y]$ **то**
  *правило $\theta'$ находится слоем выше, чем $\theta$:*
  $X_\theta := X_\theta \cup \{x_i\}$;  $u(\theta) := |X_\theta|$;
**иначе**
  *правило $\theta'$ находится слоем ниже, чем $\theta$:*
  $m(\theta) := m(\theta') + 1$;  $m'(\theta) := m'(\theta') + [y_i = y]$;  $m''(\theta) := m''(\theta') + [y_i \neq y]$;
  $X'_\theta := X'_\theta \cup X'_{\theta'} \cup \{x_i\}$;  $q(\theta) := |X'_\theta|$;
  $q'(\theta) := \#\bigl\{x_j \in X'_\theta\colon y_j = y\bigr\}$;  $q''(\theta) := \#\bigl\{x_j \in X'_\theta\colon y_j \neq y\bigr\}$;

**Алгоритм 12.2.** Вычисление оценки расслоения–связности для семейства пороговых конъюнкций.

**Вход:** $J$ — набор признаков, $y$ — класс правил, $\mathbb{X}$ — выборка.
**Выход:** $Q'_\varepsilon, Q''_\varepsilon$ — оценки вероятности переобучения (12.6), (12.7).

1: $\Theta := \operatorname{Arg}\min_\theta m(\theta, \mathbb{X})$; $\quad Q'_\varepsilon := 0$; $\quad Q''_\varepsilon := 0$;
2: **повторять**
3: $\quad Q'_{\varepsilon,m} := 0$; $\quad Q''_{\varepsilon,m} := 0$; $\quad \Theta' := \varnothing$;
4: $\quad$ **для всех** $\theta \in \Theta$
5: $\quad\quad$ построить окрестность $V_\theta$ с помощью Алгоритма 12.1;
6: $\quad\quad P_\theta := C^{\ell-u(\theta)}_{L-u(\theta)-q(\theta)} / C^\ell_L$;
7: $\quad\quad Q'_{\varepsilon,m} := Q'_{\varepsilon,m} + P_\theta \mathcal{H}^{\ell-u(\theta),\, m'(\theta)-q'(\theta)}_{L-u(\theta)-q(\theta)} \big(\frac{\ell}{L}(m'(\theta) - \varepsilon k)\big)$;
8: $\quad\quad Q''_{\varepsilon,m} := Q''_{\varepsilon,m} + P_\theta \mathcal{H}^{\ell-u(\theta),\, m''(\theta)-q''(\theta)}_{L-u(\theta)-q(\theta)} \big(\frac{\ell}{L}(m''(\theta) - \varepsilon k)\big)$;
9: $\quad\quad \Theta' := \Theta' \cup \big\{\theta' \in V_\theta \colon m(\theta') = m(\theta) + 1\big\}$;
10: $\quad\quad Q'_\varepsilon := Q'_\varepsilon + Q'_{\varepsilon,m}$; $\quad Q''_\varepsilon := Q''_\varepsilon + Q''_{\varepsilon,m}$; $\quad \Theta := \Theta'$;
11: **пока** вклады слоя $Q'_{\varepsilon,m}$ и $Q''_{\varepsilon,m}$ не станут малы.

позволит в дальнейшем сократить поиск. Каждый объект выборки $x \in \mathbb{X}$ может находиться в одном из трёх статусов: $x.\mathsf{проверен} \in \{\mathsf{нет}, \mathsf{плохой}, \mathsf{хороший}\}$. 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 $\theta < x \leqslant \bar\theta$, и хотя бы по одной координате объект $x$ лежит выше порога: $\theta^j < x^j$. Для каждого такого объекта вызывается процедура $\mathsf{Проверить}(x)$. Она устанавливает статус объекта $x.\mathsf{хороший}$, 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 $x.\mathsf{плохой}$, и процедура $\mathsf{Проверить}(\tilde x)$ 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 12.1 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.

**Послойный перебор классов эквивалентности.** Алгоритм 12.2 вычисляет $Q_\varepsilon$ — оценку вероятности переобучения, перебирая правила по слоям снизу вверх. Сначала

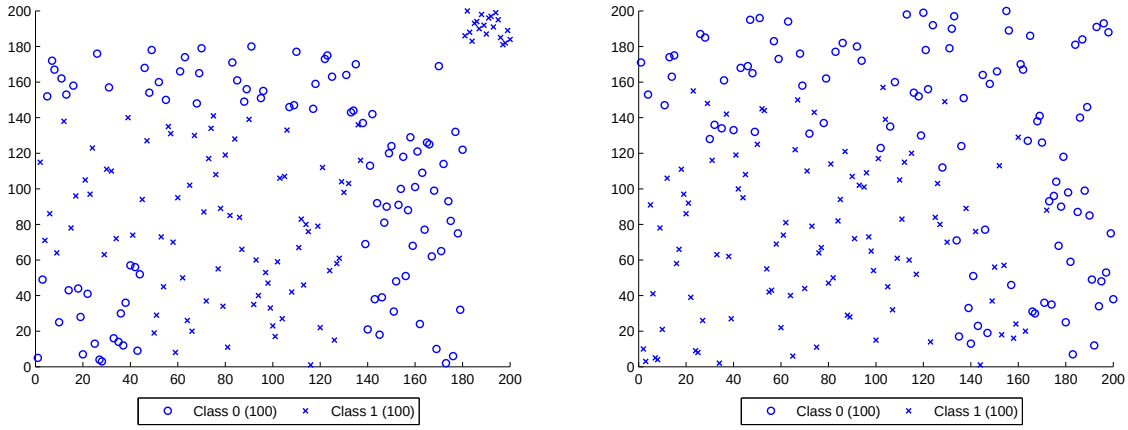



**Рис. 12.6.** Выборка 1 (слева) и выборка 2 (справа) в осях двух признаков.

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.

## §12.5 Эксперимент на модельных данных

Возьмем две двумерных ($n = 2$) двухклассовых ($\mathbb{Y} = \{0, 1\}$) сбалансированных ($P = N = L/2$) модельных выборки длины $L = 200$. Будем разбивать их на обучение и контроль поровну, $\ell = k = 100$. В семействе правил (12.1) фиксируем операцию сравнения $\leqslant$ и будем строить только правила класса 1. Для поиска правил используем гипергеометрический критерий информативности (точный тест Фишера) $\mathscr{H}(p, n) = -\ln\bigl(C_P^p C_N^n / C_{P+N}^{p+n}\bigr)$.

Выборки сгенерируем следующим образом.

В первой выборке число ошибок лучшего правила равно 40. Граница между классами четкая, поскольку все 40 шумовых объектов расположены вдали от границы классов, рис. 12.6 (слева).

Во второй выборке число ошибок лучшего правила 23, т. е. почти вдвое меньше, шумовые объекты расположены вдоль границы классов, рис. 12.6 (справа).

Хотя во второй выборке существует правило, почти вдвое лучшее, чем в первой, найти его по обучающей подвыборке практически невозможно из-за переобучения. Значения критерия информативности на обучающей выборке почти одинаковы (незначимое предпочтение отдаётся второй выборке), однако на контроле найденное правило оказывается значимо лучше для первой выборки:

| | выборка 1 | выборка 2 |
|---|---|---|
| обучение | 40,29 | 40,47 |
| контроль | 39,13 | 34,46 |

| $\eta$ | оценка Монте-Карло | | оценка расслоения-связности | |
|---|---|---|---|---|
| | выборка 1 | выборка 2 | выборка 1 | выборка 2 |
| 0.1 | 24.44 | 16.29 | 23.23 | 13.42 |
| 0.2 | 28.07 | 22.13 | 27.11 | 20.58 |
| 0.3 | 32.28 | 28.28 | 30.45 | 25.74 |
| 0.4 | 34.35 | 32.98 | 33.39 | 27.59 |
| 0.5 | 37.63 | 36.12 | 34.93 | 31.21 |

**Таблица 12.1.** Значения модифицированного критерия $\mathscr{H}'$ на обучении на двух модельных выборках, при пяти различных значениях $\eta$. Вероятность переобучения оценивалась двумя способами: методом Монте-Карло по 200 случайным разбиениям и по формуле расслоения–связности.

Причина в том, что у второй выборки существенно менее выражено расслоение, мощности нижних слоёв очень быстро возрастают, рис. 12.7.

Возникает вопрос — возможно ли с помощью комбинаторных оценок, использующих информацию о расслоении и связности на обучающей выборке, предсказать информативность на скрытой контрольной выборке? Таблица 12.1 даёт утвердительный ответ.

Предсказанная информативность $\mathscr{H}'$ всегда меньше для второй выборки. Таким образом, нам действительно удаётся предсказать, что найти хорошее правило при сильно зашумлённой границе классов невозможно из-за переобучения.

Сравнение с методом Монте-Карло показывает, что предсказанная информативность $\mathscr{H}'$, вычисляемая по оценке расслоения–связности, всегда пессимистично занижена, поскольку завышена оценка $p(r, \bar{X})$ и занижена оценка $n(r, \bar{X})$. Заниженность информативности составляет в данной задаче 1–2 для первой выборки и 3–5 для второй. Это согласуется с теоретическим выводом, что оценка расслоения–связности менее точна для выборок с менее выраженным расслоением. При поиске закономерностей такая смещённость оценки является благоприятной — чем слабее закономерность, тем сильнее занижена предсказанная информативность.

Заметим также, что оценка расслоения-связности вычисляется существенно быстрее, чем эмпирическая оценка по методу Монте-Карло.

Рассмотрим теперь вклад каждого слоя в оценку расслоения–связности. График зависимости накопленного значения $P_r$ от номера слоя представлен на рисунке 12.8. Видно, что основной вклад вносят слои с малым числом ошибок.

Если бы оценки вероятностей $P_r$ были точными, их сумма равнялась бы единице. Накопленное значение $P_r$ в крайней правой точке графика позволяет судить о степени завышенности оценок расслоения–связности. Для выборки 1 с «хорошим расслоением» оценка завышена всего в 2 раза; для выборки 2 с «плохим расслоением» — в 40 раз.

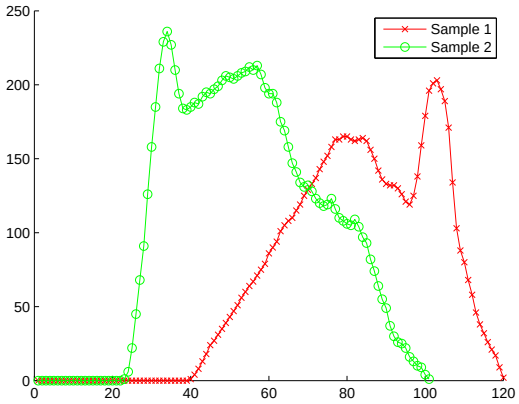



**Рис. 12.7.** Зависимость числа правил в слое от номера слоя $m(r, \mathbb{X})$ для двух модельных выборок.

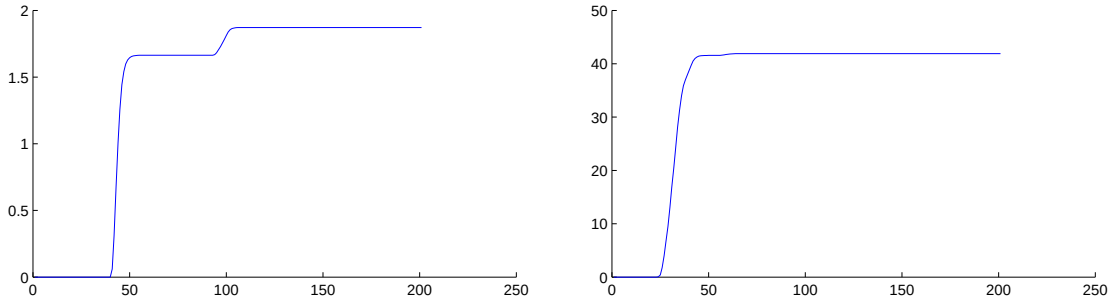



**Рис. 12.8.** Зависимость накопленного значения $P_r$ от номера слоя $m(r, \mathbb{X})$ для выборки 1 (слева) и выборки 2 (справа).

## §12.6 Эксперимент на реальных данных

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 [36, 35]; 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 [73];

*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 [66]. 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 | **2.9** | 19.7 | 20.7 | 18.0 | 32.7 |
| 2 | RIPPER+opt | 15.2 | 5.5 | 20.1 | 23.2 | 18.0 | **31.3** |
| 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 | **14.0** | 4.3 | 21.8 | 20.1 | 18.4 | 31.9 |
| 6 | SLIPPER | 15.7 | 4.3 | **19.4** | **17.4** | **12.3** | 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 | **WV+MC** | **13.9** | **3.0** | **19.5** | **18.3** | **13.2** | **30.7** |
| 11 | **DL+MC** | 14.5 | 3.5 | 19.8 | 18.7 | 13.8 | 32.8 |
| 12 | **WV+SC** | **14.1** | **3.2** | **19.3** | **18.1** | **13.4** | **30.2** |
| 13 | **DL+SC** | 14.4 | 3.6 | 19.5 | 18.6 | 13.6 | 32.3 |

**Таблица 12.2.** 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 $\mathscr{H}'(p,n)$. 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 [65] и SLIPPER [66].

Таблица 12.2 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. Заменив 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 перейдём к рассмотрению функционала полного скользящего контроля, для которого в некоторых случаях удаётся получать очень точные оценки, находящие полезные практические применения. Для начала мы займёмся методом ближайшего соседа — это один из самых простых и часто используемых методов классификации.

## Упражнения

**Задача 12.1** (1). Доказать теорему 12.1.

**Задача 12.2** (2). Доказать теорему 12.2.

**Задача 12.3** (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 12.4** (2). Доказать, что каждый класс эквивалентности правил представляет собой диаграмму Юнга.

**Задача 12.5** (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 связные компоненты графа будут взаимно однозначно соответствовать классам эквивалентности правил.

**Задача 12.6** (2). Описать структуру классов эквивалентности семейства правил $r(x; \theta_1, \theta_2) = \big[\theta_1 \leqslant x \leqslant \theta_2\big]$, где все значения $x_i \in \mathbb{X} \subset \mathbb{R}$ попарно различны. Предложить эффективный алгоритм их послойного перебора. Подсказка: рассмотреть эквивалентное семейство двухпризнаковых конъюнкций $r(x; \theta_1, \theta_2) = \big[-x \leqslant \theta_1\big]\big[x \leqslant \theta_2\big]$.

**Задача 12.7** (3). Описать структуру классов эквивалентности и предложить эффективный алгоритм их послойного перебора для случая, когда матрица ошибок порождается индикатором ошибок I рода $I'(r, x_i)$ или II рода $I''(r, x_i)$.

**Задача 12.8** (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 12.9** ($3^*$)**.** Обобщить Алгоритм 12.2 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.

# 13 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 $I(a,x) = \big[y(x) \neq a(x)\big]$, где $y\colon \mathbb{X} \to \mathbb{Y}$ — функция правильной классификации (целевая зависимость), $a\colon \mathbb{X} \to \mathbb{Y}$ — 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.

## §13.1 Профиль компактности выборки

Пусть на множестве $\mathbb{X} = \{x_1, \ldots, x_L\}$ 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: $x_i = x_{i0},\ x_{i1},\ x_{i2}, \ldots, x_{i,L-1}$. Таким образом,

$$0 = \rho(x_i, x_{i0}) \leqslant \rho(x_i, x_{i1}) \leqslant \cdots \leqslant \rho(x_i, x_{i,L-1}). \tag{13.1}$$

*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:

$$a(x) = y\big(\arg\min_{x' \in X} \rho(x, x')\big).$$

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:

$$I_m(x_i) = \big[y(x_i) \neq y(x_{im})\big]; \quad i = 1, \ldots, L; \quad m = 1, \ldots, L-1.$$

**Определение 13.1.** *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:*

$$K(m) = \frac{1}{L} \sum_{i=1}^{L} I_m(x_i); \quad m = 1, \ldots, L-1.$$

Профиль компактности является формальным выражением гипотезы компактности. Чем проще задача, то есть чем чаще близкие объекты оказываются в одном

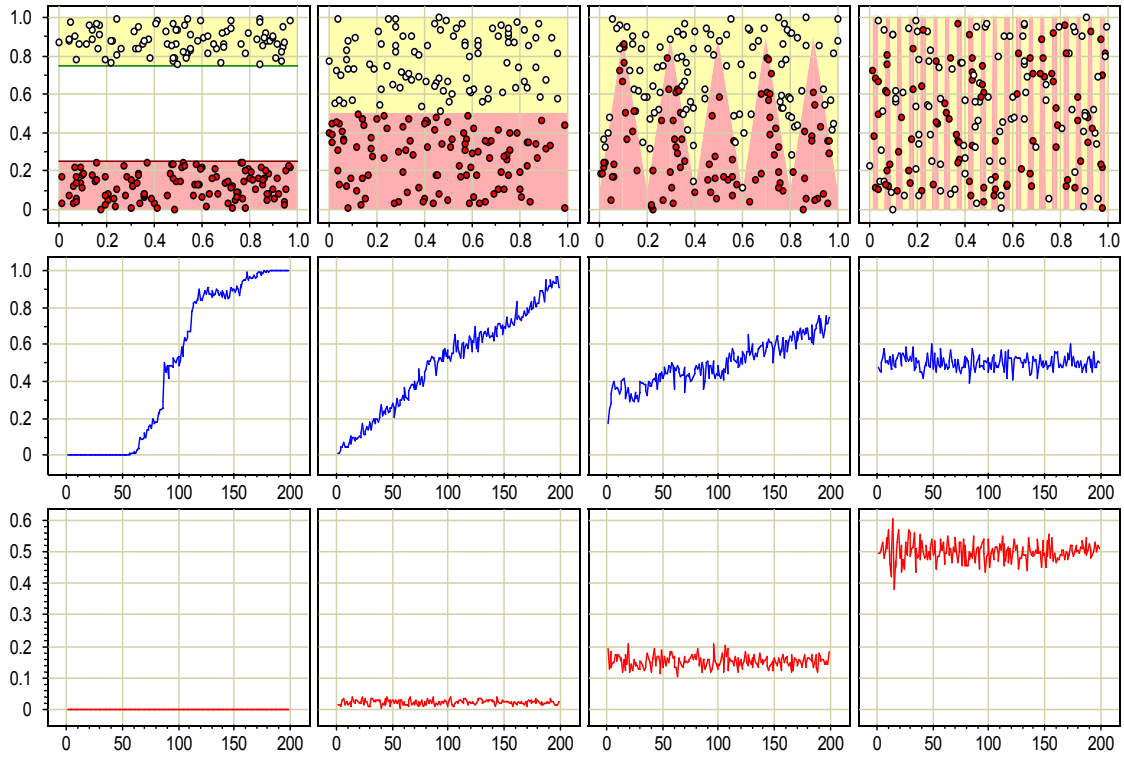


**Рис. 13.1.** Верхний ряд: 4 модельные задачи классификации, в порядке возрастания трудности, $L = 200$. Средний ряд: профили компактности этих задач. Чем ниже проходит начальный участок профиля, тем «проще» задача для алгоритма NN, и тем выше обобщающая способность. Нижний ряд: зависимость CCV от длины 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. 13.1. Профиль компактности показывает, насколько хорошо метод ближайшего соседа подходит для решения данной задачи при выбранной метрике $\rho$.

## §13.2 Точная оценка полного скользящего контроля

Интуитивно очевидная связь профиля компактности с качеством классификации подтверждается следующей теоремой. Идея доказательства взята из [88].

**Теорема 13.1.** *Для метода ближайшего соседа $\mu$ справедливо следующее выражение функционала полного скользящего контроля* CCV*:*

$$\mathrm{CCV}(\mu, \mathbb{X}) = \sum_{m=1}^{k} K(m) \frac{C_{L-1-m}^{\ell-1}}{C_{L-1}^{\ell}}. \tag{13.2}$$

**Доказательство.** Запишем в функционале скользящего контроля частоту ошибок через сумму индикаторов ошибки и переставим местами знаки суммирования:

$$\mathrm{CCV} = \frac{1}{C_L^\ell} \sum_{X,\bar{X}} \frac{1}{k} \sum_{x \in \bar{X}} I(\mu X, x) = \frac{1}{k} \sum_{i=1}^{L} \frac{1}{C_L^\ell} \underbrace{\sum_{X,\bar{X}} [x_i \in \bar{X}]\, I(\mu X, x_i)}_{N_i}. \tag{13.3}$$

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 $x_{i0}, x_{i1}, \ldots, x_{i,L-1}$ 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 (13.3), используя определение профиля компактности и тождество $kC_L^\ell = LC_{L-1}^\ell$, получаем требуемую формулу (13.2). ■

### 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:

$$\gamma_{m+1} = q(m)\gamma_m, \qquad q(m) = 1 - \frac{\ell - 1}{L - 1 - m} < \frac{k}{L-1}.$$

Значение 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 CCV.

2. При $k = 1$ профиль компактности вырождается в точку и совпадает с самим функционалом 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 $k = 1, 2, 3$:

$$\begin{aligned}
k = 1: &\quad \mathrm{CCV} = K(1);\\
k = 2: &\quad \mathrm{CCV} = K(1)\tfrac{\ell}{\ell+1} + K(2)\tfrac{1}{\ell+1};\\
k = 3: &\quad \mathrm{CCV} = K(1)\tfrac{\ell}{\ell+2} + K(2)\tfrac{2\ell}{(\ell+1)(\ell+2)} + K(3)\tfrac{2}{(\ell+1)(\ell+2)}.
\end{aligned}$$

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 CCV производится за $O(k)$ операций. Это гораздо быстрее, чем производить суммирование по всем $C_L^\ell$ разбиениям, что становится практически нереально уже при $k > 2$.

## §13.3 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 [28], FRiS-Stolp [6], 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 [14, 74]. 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* $\Omega$:

$$K(m, \Omega) = \frac{1}{L} \sum_{i=1}^{L} I_m(x_i, \Omega); \quad m = 1, \ldots, L-1.$$

Для отбора эталонов удобнее определить величину *вклада* $T(x_i, \Omega)$ объекта $x_i$ и представлять CCV в виде суммы вкладов всех объектов.

**Теорема 13.2.** *Для метода ближайшего соседа $\mu_\Omega$, использующего в качестве эталонов только объекты множества $\Omega \subseteq \mathbb{X}$, справедливы два представления CCV:*

$$\mathrm{CCV}(\mu_\Omega, \mathbb{X}) = \sum_{m=1}^{k} K(m, \Omega) \frac{C_{L-1-m}^{\ell-1}}{C_{L-1}^{\ell}} = \sum_{i=1}^{L} \underbrace{\sum_{m=1}^{k} I_m(x_i, \Omega) \frac{C_{L-1-m}^{\ell-1}}{L C_{L-1}^{\ell}}}_{T(x_i, \Omega)}. \tag{13.4}$$

Доказательство аналогично Теореме 13.1 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 13.1.** Жадное удаление шумовых и неинформативных объектов.

**Вход:** выборка $\mathbb{X}$, параметр $\delta > 0$;
**Выход:** множество эталонов $\Omega \subseteq \mathbb{X}$;

$\Omega := \mathbb{X}$; $Q := \sum_{i=1}^{L} T(x_i, \Omega)$; $Q_{\min} := Q$;
**повторять**
  count := 0; — *счётчик числа удалений*
  **для всех** $x \in \Omega$
    $\Delta Q := \sum\limits_{x_i \in V_k(x,\Omega)} \Big(T\big(x_i, \Omega\backslash\{x\}\big) - T(x_i, \Omega)\Big)$;
    **если** $Q + \Delta Q < Q_{\min} + \delta$ **то**
      $Q := Q + \Delta Q$; $\Omega := \Omega\backslash\{x\}$;
      **если** $Q < Q_{\min}$ **то** $Q_{\min} := Q$;
      count := count + 1;
**пока** count $> 0$;

**Алгоритм 13.2.** Жадное добавление эталонных объектов.

**Вход:** выборка $\mathbb{X}$, параметр $\delta > 0$;
**Выход:** множество эталонов $\Omega \subseteq \mathbb{X}$;

$\Omega :=$ {по одному случайному объекту от каждого класса};
$Q := \sum_{i=1}^{L} T(x_i, \Omega)$; $Q_{\min} := Q$;
**повторять**
  count := 0; — *счётчик числа добавлений*
  **для всех** $x \in \mathbb{X}\backslash\Omega$
    $\Delta Q := \sum\limits_{x_i \in V_k(x,\Omega\cup\{x\})} \Big(T\big(x_i, \Omega \cup \{x\}\big) - T(x_i, \Omega)\Big)$;
    **если** $Q + \Delta Q < Q_{\min} - \delta$ **то**
      $Q := Q + \Delta Q$; $\Omega := \Omega \cup \{x\}$;
      **если** $Q < Q_{\min}$ **то** $Q_{\min} := Q$;
      count := count + 1;
**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 13.2.** *Обратной окрестностью $m$-го порядка объекта $x \in \Omega$ называется множество объектов, для которых объект $x$ является не далее, чем $m$-м соседом:*

$$V_m(x, \Omega) = \big\{x_i \in \mathbb{X} \;\big|\; r_i(x, \Omega) \leqslant m\big\}, \quad m = 1, \ldots, L-1.$$

В [14] 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 $\mathrm{CCV}(\mu_{\Omega\backslash\{x\}}, \mathbb{X})$. Позже был предложен Алгоритм 13.1, который работает намного быстрее [74]. 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 13.2 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 как 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 $x_i \in \mathbb{X}$. Наиболее подходят для этой цели *метрические деревья*, предназначенные для индексации множества объектов, заданных попарными расстояниями [62, 63]. 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.

## §13.4 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. 13.2 изображена разделяющая поверхность, построенная алгоритмом ближайшего соседа на полной выборке $X^L$.

На рис. 13.3 и рис. 13.4 показана разделяющая поверхность для классификатора одного ближайшего соседа по множеству эталонов $\Omega$. Рис. 13.3 соответствует стратегии удаления, рис. 13.4 — 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.

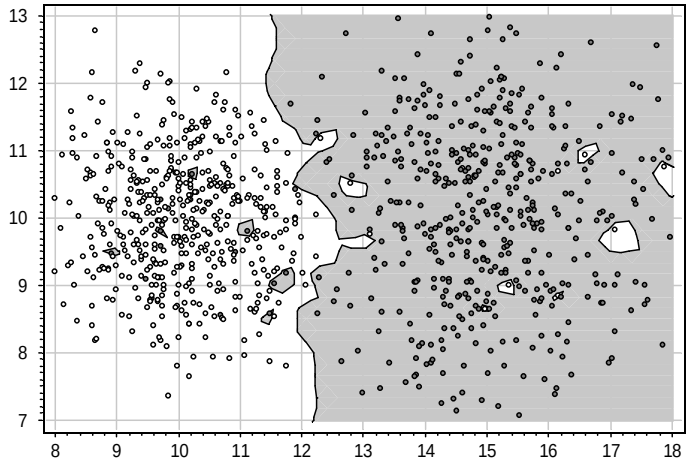



**Рис. 13.2.** Модельная задача классификации: 1000 объектов, метод 1NN.

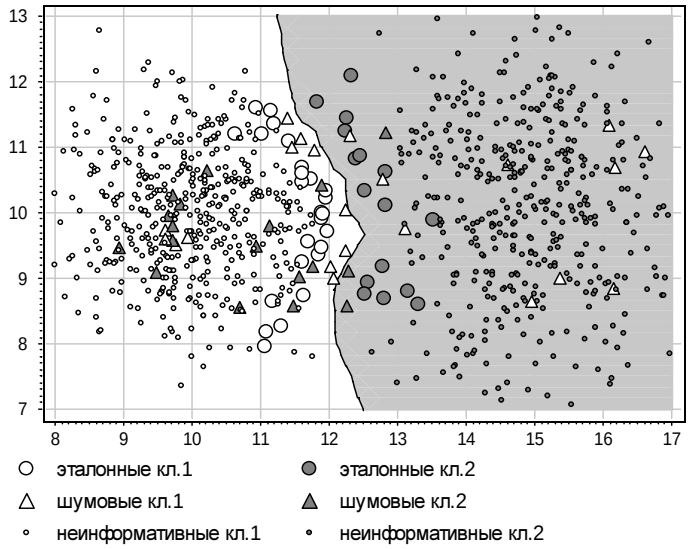



**Рис. 13.3.** Результат отбора эталонов Алгоритмом 13.1 (стратегия удаления).

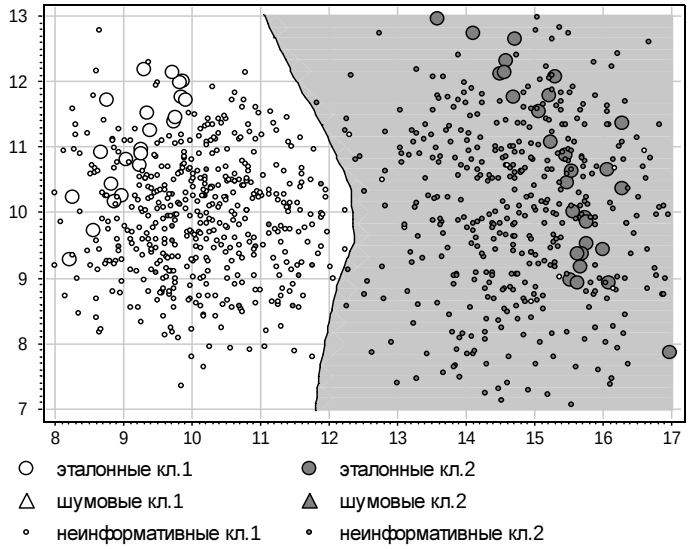



**Рис. 13.4.** Результат отбора эталонов Алгоритмом 13.2 (стратегия добавления).


**Рис. 13.5.** Зависимость функционала $Q(\Omega)$ от количества удалённых объектов $L - |\Omega|$ для Алгоритма 13.1.

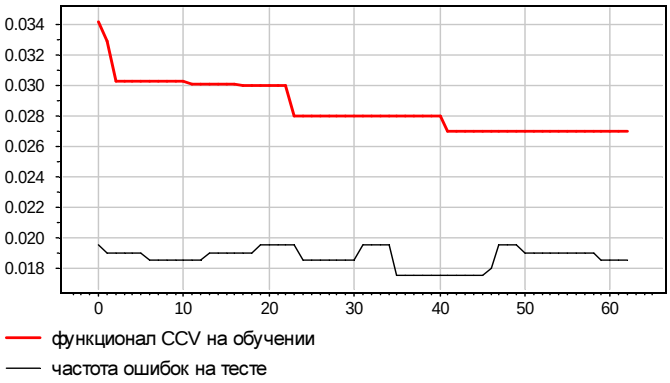



**Рис. 13.6.** Зависимость функционала $Q(\Omega)$ от количества добавленных объектов $|\Omega|$ для Алгоритма 13.2.

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. 13.5) она изменяется синхронно со значением CCV на обучающей выборке. Это означает, что данная стратегия практически не подвержена переобучению. В случае жадного добавления (рис. 13.6) столь чёткой синхронности не наблюдалось.

На рис. 13.5 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. 13.3 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. 13.5, 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 $K(m)$. Он определяется как доля объектов выборки, для которых $m$-й сосед лежит в другом классе. По форме начального участка профиля компактности можно судить, подходит ли выбранная модель сходства объектов для данной задачи. 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 информацией во многих приложениях.

**В следующей лекции** будут получены оценки CCV для семейства монотонных классификаторов. Хотя парадигмы метрической и монотонной классификации на первый взгляд совершенно различны, для этих двух случаев получаются удивительно похожие по своей структуре оценки.

## Упражнения

**Задача 13.1** (1). Доказать теорему 13.2.

**Задача 13.2** (1). Доказать, что $\sum\limits_{m=1}^{L-1} K(m) = \frac{1}{L} \sum\limits_{y \in \mathbb{Y}} L_y(L - L_y)$, где $L_y$ — число объектов класса $y$ в выборке $\mathbb{X}$.

**Задача 13.3** (1). Доказать, что $\sum\limits_{m=1}^{L-1} \gamma_m = 1$.

**Задача 13.4** (2). Обобщить теорему 13.1 на случай, когда в вариационном ряду расстояний (13.1) есть одинаковые значения. Сделать пессимистичное предположение, что если для классифицируемого объекта $x_i$ существует хотя бы один обучающий объект $x$ на минимальном расстоянии $\min\limits_{x \in X} \rho(x_i, x)$, лежащий в другом классе, то алгоритм обязательно допустит ошибку на объекте $x_i$.

**Задача 13.5** (5). Обобщить теорему 13.2 для метода классификации по $K$ ближайшим соседям.

**Задача 13.6** ($5^*$). Получить оценки вероятности переобучения $Q_\varepsilon$ и вероятности большой частоты ошибок на контроле $R_\varepsilon$ для метода ближайшего соседа.

# 14 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 $a\colon \mathbb{X} \to \mathbb{Y}$ (то есть из $x \leqslant x'$ следует $a(x) \leqslant a(x')$ для всех $x, x' \in \mathbb{X}$); индикатор ошибки имеет вид $I(a, x) = \big[y(x) \neq a(x)\big]$, 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 [50]. 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 $f(x) \leqslant f(x')$ для всех признаков $f$ из заданного множества $G$, то $a(x) \leqslant a(x')$». 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 [102]. 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 $f\colon \mathbb{X} \to \mathbb{R}$. Функция $a(x) = \big[f(x) \geqslant \theta\big]$ называется *пороговым решающим правилом* с порогом $\theta$. Если отношение порядка на $\mathbb{X}$ задано как $\big(x \leqslant x'\big) \leftrightarrow \big(f(x) \leqslant f(x')\big)$, 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. 108), условия ветвления в решающих деревьях [22, 90, 42] и решающих списках [53, 103], базовые классификаторы в композициях типа бустинга и бэггинга [96, 61], правила принятия окончательного решения во многих алгоритмах классификации на два класса.

**3. Композиции алгоритмов классификации.** Пусть имеется $T$ алгоритмов классификации вида $a_t(x) = \big[b_t(x) \geqslant 0\big]$, где $b_t\colon \mathbb{X} \to \mathbb{R}$, $t = 1, \ldots, T$. Их *композицией* называется алгоритм $a(x) = F\big(b_1(x), \ldots, b_T(x)\big)$, где функция $F\colon \mathbb{R}^T \to \mathbb{Y}$ 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 [46, 12]. Поскольку базовые алгоритмы могут допускать ошибки, зависимость $y(x_i)$ от $T$-мерных векторов $\big(b_1(x_i), \ldots, b_T(x_i)\big)$ может оказаться не монотонной, но «почти монотонной».

Наиболее распространённым типом композиций является *взвешенное голосование* — линейная комбинация базовых алгоритмов [26, 99, 61, 80]:

$$F\big(b_1(x),\ldots,b_T(x)\big) = \sum_{t=1}^{T} \alpha_t b_t(x).$$

Обычно предполагается, что $\alpha_t \geqslant 0$; в таком случае линейная корректирующая операция является монотонной. Требование линейности для корректирующих операций выглядит менее обоснованным, чем требование монотонности. Линейные функции имеют чисто практические преимущества: их проще строить, и коэффициенты $\alpha_t$ можно интерпретировать как степени доверия базовым алгоритмам. Монотонные корректирующие операции образуют существенно более широкое множество функций, что позволяет надеяться на выигрыш в качестве классификации, но в то же время это может приводить к переобучению. Вопрос о том, какие корректирующие операции лучше, линейные или монотонные, относительно мало исследован. Эксперименты свидетельствуют, что монотонные корректирующие операции могут обладать лучшей обобщающей способностью при значительно меньшем количестве базовых алгоритмов [20], однако пока нет чёткого понимания, почему это происходит.

Далее нас будет интересовать вопрос, в каких случаях монотонные классификаторы обладают наилучшей обобщающей способностью в смысле функционала полного скользящего контроля CCV. 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 CCV от этой характеристики.

## §14.1 Профиль монотонности выборки

Допустим, что метод обучения $\mu$ выбирает алгоритмы из множества $A$ всех монотонных функций $a\colon \mathbb{X} \to \mathbb{Y}$.

**Определение 14.1.** *Верхним и нижним клином объекта $x_i \in \mathbb{X}$ называются, соответственно, множества*

$$W_0(x_i) = \{x \in \mathbb{X}\colon x_i < x \text{ и } y(x) = 0\};$$
$$W_1(x_i) = \{x \in \mathbb{X}\colon x < x_i \text{ и } y(x) = 1\}.$$

Примеры клиньев показаны на рис. 14.1 (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*.

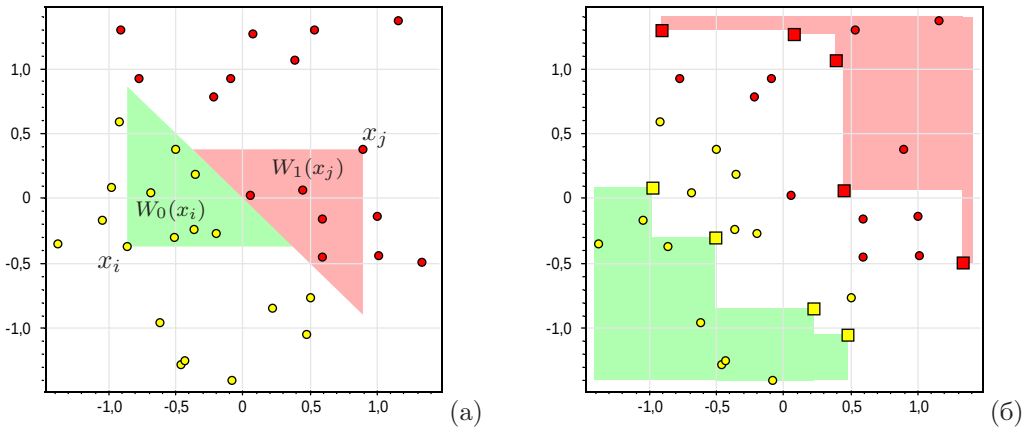



**Рис. 14.1.** Двумерная задача классификации с естественным отношением порядка на множестве $\mathbb{X} \subset \mathbb{R}^2$. Рис. (а): верхний клин объекта $x_i$, нижний клин объекта $x_j$. Рис. (б): одна из допустимых монотонных классификаций генеральной выборки $\mathbb{X}$. Объекты обучающей выборки выделены символами □, ■. Фоном показано объединение верхних областей обучающих объектов класса 1 и объединение нижних областей обучающих объектов класса 0.

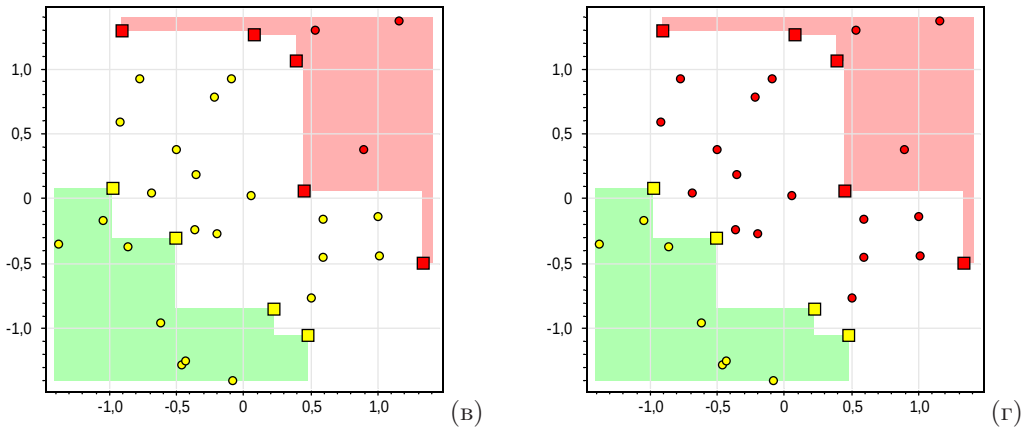



**Рис. 14.2.** Та же двумерная задача классификации, что на рис. 14.1. Два крайних случая монотонной классификации: все объекты в зазоре между нижними и верхними областями обучающих объектов относятся к классу 0 (а), к классу 1 (б).

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 14.2.** *Профилем монотонности выборки* $\mathbb{X}$ *называется функция* $M(m)$, *равная доле объектов с клином мощности* $m$:

$$M(m) = \frac{1}{L}\sum_{i=1}^{L}[w_i = m]; \quad m = 0, \ldots, L-1.$$

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 $M(m) = \frac{1}{L}\big([m < L_0] + [m < L_1]\big)$, где $L_y$ — число объектов класса $y$ в генеральной выборке; если $L_0 = L_1$, то профиль монотонности распределён равномерно: $M(m) = \frac{2}{L}\big[m < \frac{L}{2}\big]$.

## §14.2 Верхняя оценка полного скользящего контроля

**Определение 14.3.** *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 $x_i \leqslant x_j$ следует $y(x_i) \leqslant y(x_j)$ 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 [12].

**Теорема 14.1.** *Если метод $\mu$ минимизирует эмпирический риск в классе всех монотонных функций и степень немонотонности выборки $\mathbb{X}$ равна $\theta$, то*

$$\mathrm{CCV}(\mu, \mathbb{X}) \leqslant \sum_{m=0}^{\theta L + k - 1} M(m)\, \mathcal{H}_{L-1}^{\ell, m}(\theta L)\,. \tag{14.1}$$

**Доказательство.** Запишем в функционале скользящего контроля частоту ошибок через сумму индикаторов ошибки и поменяем местами знаки суммирования:

$$\mathrm{CCV} = \frac{1}{C_L^\ell} \sum_{X, \bar{X}} \frac{1}{k} \sum_{x \in \bar{X}} I(\mu X, x) = \frac{1}{k} \sum_{i=1}^{L} \frac{1}{C_L^\ell} \underbrace{\sum_{X, \bar{X}} [x_i \in \bar{X}]\; I(\mu X, x_i)}_{N_i}\,. \tag{14.2}$$

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 $w_i \geqslant \theta L + k$, 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,

$$\max\{0, w_i - k + 1\} \leqslant s \leqslant \min\{\theta L, \ell, w_i\}.$$

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:

$$N_i \leqslant \sum_{s=\max\{0,w_i-k+1\}}^{\min\{\theta L,\ell,w_i\}} C_{w_i}^s C_{L-1-w_i}^{\ell-s}. \tag{14.3}$$

Подставим оценку (14.3) в (14.2), воспользуемся тождеством $kC_L^\ell = LC_{L-1}^\ell$ и учтём, что $N_i = 0$ при $w_i \geqslant \theta L + k$:

$$\mathrm{CCV} \leqslant \frac{1}{k} \sum_{\substack{i=1 \\ w_i < \theta L + k}}^{L} \frac{k}{L} \sum_{s=\max\{0,w_i-k+1\}}^{\min\{\theta L,\ell,w_i\}} \frac{C_{w_i}^s C_{L-1-w_i}^{\ell-s}}{C_{L-1}^\ell}.$$

Применяя определение профиля монотонности, получаем оценку

$$\mathrm{CCV} \leqslant \sum_{m=0}^{\theta L+k-1} M(m) \sum_{s=\max\{0,m-k+1\}}^{\min\{\theta L,\ell,m\}} \frac{C_m^s C_{L-1-m}^{\ell-s}}{C_{L-1}^\ell}.$$

Заметив, что сумма по $s$ совпадает с функцией гипергеометрического распределения $\mathcal{H}_{L-1}^{\ell,m}(\theta L)$, получаем оценку (14.1). Что и требовалось доказать. ■

**Следствие 14.1.1.** *Оценка (14.1) 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 $X \in [\mathbb{X}]^\ell$:*

$$\mathrm{CCV}(\mu, \mathbb{X}) \leqslant \sum_{m=0}^{k-1} M(m) \frac{C_{L-1-m}^\ell}{C_{L-1}^\ell}. \tag{14.4}$$

Наибольшая оценка $\mathrm{CCV} \leqslant 1$ достигается, когда $w_i = 0$ для всех $i = 1, \ldots, L$. 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 $\mathrm{CCV} \leqslant 2/\ell$ достигается, когда выборка монотонна и линейно упорядочена. В этом случае число клиньев мощности $w$ не превышает 2 для всех

$w = 1, \ldots, k$. Семейство монотонных алгоритмов $A$ индуцирует унимодальную цепь. Переобучение в данном случае минимально.

Гипергеометрический множитель в (14.1) убывает с ростом $m$ быстрее геометрической прогрессии. Чтобы обеспечить малое значение функционала 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, или, 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 [47, 48] 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 $\mathbb{X}$ [100]. Оценка (14.1) существенно точнее сложностных оценок и никогда не превышает 1.

Интересно отметить большое структурное сходство оценок (13.2) и (14.4), полученных для таких различных, на первый взгляд, априорных ограничений, как компактность и монотонность. Чуть позже мы попробуем разобраться в его причинах.

## §14.3 Точные оценки полного скользящего контроля

Попробуем разобраться, почему оценка (14.1) 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 $x \leqslant x_i$.

*Верхняя область* объекта $x_i$ — это множество всех объектов $x$ таких, что $x_i \leqslant x$.

Допустим, что выборка $\mathbb{X}$ 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. 14.1 (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. 14.1 (б) показана одна из возможных реализаций алгоритма $a$. На рис. 14.2 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 $y \in \mathbb{Y}$ функцию $M_y(m)$, равную доле объектов класса $y$ с клином мощности $m$:

$$M_y(m) = \frac{1}{L}\sum_{i=1}^{L}\big[y(x_i) = y\big]\big[w_i = m\big]; \quad m = 0, \ldots, L-1.$$

Очевидно, сумма профилей по классам совпадает с общим профилем:

$$M(m) = M_0(m) + M_1(m), \quad m = 0, \ldots, L-1. \tag{14.5}$$

**Теорема 14.2.** *Если выборка $\mathbb{X}$ монотонна и алгоритм $a = \mu X$ относит все объекты, лежащие в зазоре, к классу $y$, то справедлива точная оценка*

$$\mathrm{CCV}(\mu, \mathbb{X}) = \sum_{m=0}^{k-1} M_{1-y}(m)\frac{C^{\ell}_{L-1-m}}{C^{\ell}_{L-1}}. \tag{14.6}$$

**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 (14.3). Для остальных объектов из зазора $N_i = 0$. Таким образом,

$$N_i = C^{\ell}_{L-1-w_i}\big[y(x_i) = 1\big].$$

Подставив $N_i$ в (14.2) и воспользовавшись тождеством $kC^{\ell}_{L} = LC^{\ell}_{L-1}$, получим (14.6) для случая $y = 0$. Аналогичным образом проверяется случай $y = 1$. ■

Если предположить, что профиль монотонности делится согласно (14.5) примерно поровну между классами, то из доказанной теоремы следует, что оценка (14.4) 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 CCV для случая разделения по середине зазора получена Галиной Махиной в [41].

**Монотонные классификаторы по расстояниям до границ зазора.** Определим для произвольной пары объектов $x, u \in \mathbb{X}$ расстояние $r(x, u)$ между нижней областью объекта $x$ и верхней областью объекта $u$. Потребуем, чтобы функция $r$ обладала следующими свойствами:

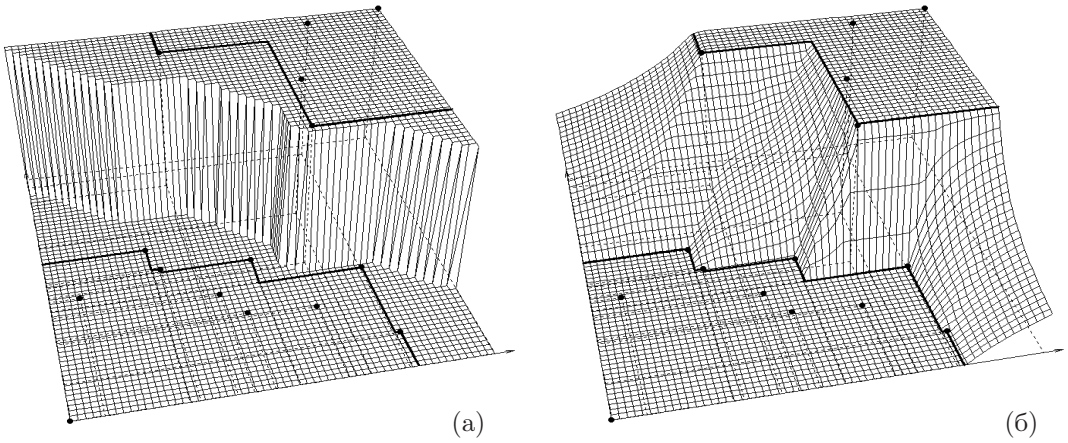



**Рис. 14.3.** Монотонные функции, проходящие через точки монотонной обучающей выборки в $\mathbb{R}^2$: (а) алгоритм классификации $a(x)$, (б) непрерывная функция $b(x)$. Ступенчатыми линиями показаны нижняя и верхняя границы зазора.

1) $r(x,u) = 0$ тогда и только тогда, когда $x \geqslant u$;

2) $r(x,u)$ не возрастает по $x$ не убывает по $u$.

Функцию $r(x,u)$ можно интерпретировать также и как расстояние от объекта $x$ до верхней области $u$, и как расстояние от объекта $u$ до нижней области $x$.

Рассмотрим алгоритм ближайшего соседа $a = \mu X$,

$$a(x) = y\big(\arg\min_{x' \in X} \rho(x,x')\big), \tag{14.7}$$

определив функцию расстояния $\rho(x,x')$ от классифицируемого объекта $x$ до объекта обучающей выборки $x' \in X$ как расстояние до его нижней области, если $y(x') = 0$, и до его верхней области, если $y(x') = 1$:

$$\rho(x,x') = \begin{cases} (1-\lambda) r(x',x), & y(x') = 0, \\ \lambda r(x,x'), & y(x') = 1, \end{cases}$$

где $\lambda \in (0,1)$ определяет положение разделяющей поверхности внутри зазора:

при $\lambda \to 0$ разделяющая поверхность проходит по нижней границе зазора, и все объекты из зазора относятся к классу 1;

при $\lambda \to 1$ разделяющая поверхность проходит по верхней границе зазора, и все объекты из зазора относятся к классу 0;

при $\lambda = \frac{1}{2}$ разделяющая поверхность проходит посередине зазора, рис. 14.3 (а).

При крайних значениях $\lambda \in \{0,1\}$ функция $\rho$ может принимать минимальное значение 0 одновременно на обучающих объектах разных классов, что будет приводить к неоднозначности выбора класса в (14.7). Неоднозначность не возникнет, если брать крайние значения $\{\lambda_0, 1-\lambda_0\}$ при некотором $\lambda_0$, очень близком к нулю.

Алгоритм (14.7) можно определить эквивалентным образом как $a(x) = \big[b(x) > \lambda\big]$, где $b(x)$ — относительное расстояние до ближайшей нижней области, рис. 14.3 (б):

$$b(x) = \frac{d_0(x)}{d_0(x) + d_1(x)}, \qquad d_0(x) = \min_{\substack{x' \in X \\ y(x')=0}} \rho(x,x'), \qquad d_1(x) = \min_{\substack{x' \in X \\ y(x')=1}} \rho(x,x').$$

**Теорема 14.3.** *Пусть выборка $\mathbb{X}$ монотонна, функция $r(x,u)$ удовлетворяет свойствам 1) и 2). Тогда функции $a(x)$ и $b(x)$ монотонно не убывают по $x$ и проходят через точки обучающей выборки: $a(x) = b(x) = y(x)$ для всех $x \in X$, при этом функция $b(x)$ непрерывна.*

Доказательство можно найти в [12].

Точная оценка CCV для алгоритма (14.7) даётся той же теоремой 13.1, что и для обычного метода ближайшего соседа. 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 (14.4) может быть получена в результате ослабления точной оценки CCV по профилю компактности (13.1).

Запишем точную оценку:

$$\text{CCV} = \sum_{m=1}^{k} K(m) \frac{C_{L-1-m}^{\ell-1}}{C_{L-1}^{\ell}} = \sum_{m=1}^{k} \frac{1}{L} \sum_{i=1}^{L} I_m(x_i) \frac{C_{L-1-m}^{\ell-1}}{C_{L-1}^{\ell}},$$

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, $I_1(x_i) = \cdots = I_m(x_i) = 0$. Однако, начиная с $(m+1)$-го соседа, могут возникать ошибки. Таким образом, справедлива верхняя оценка для $I_m$:

$$I_m(x_i) \leqslant [w_i < m] = \sum_{s=0}^{m-1} [w_i = s], \quad m = 1, \ldots, L-1.$$

Подставим эту оценку в формулу для CCV и сделаем ряд преобразований:

$$\begin{aligned}
\text{CCV} &\leqslant \sum_{m=1}^{k} \frac{1}{L} \sum_{i=1}^{L} \sum_{s=0}^{m-1} [w_i = s] \frac{C_{L-1-m}^{\ell-1}}{C_{L-1}^{\ell}} = \\
&= \sum_{m=1}^{k} \sum_{s=0}^{m-1} \frac{1}{L} \sum_{i=1}^{L} [w_i = s] \frac{C_{L-1-m}^{\ell-1}}{C_{L-1}^{\ell}} = \\
&= \sum_{s=0}^{k-1} \sum_{m=s+1}^{k} M(s) \frac{C_{L-1-m}^{\ell-1}}{C_{L-1}^{\ell}} = \\
&= \sum_{s=0}^{k-1} M(s) \frac{1}{C_{L-1}^{\ell}} \left( C_{\ell-1}^{\ell-1} + \cdots + C_{L-2-s}^{\ell-1} \right) = \sum_{m=0}^{k-1} M(m) \frac{C_{L-1-m}^{\ell}}{C_{L-1}^{\ell}}.
\end{aligned}$$

Последнее выражение совпадает с верхней оценкой (14.4).

**Расстояния до верхних и нижних областей.** Рассмотрим один естественный способ определить функцию $r$, предложенный в [12] для построения монотонных корректирующих операций. Пусть объекты задаются $n$-мерными числовыми векторами, $\mathbb{X} \subset \mathbb{R}^n$, $x = (x^1, \ldots, x^n)$, $u = (u^1, \ldots, u^n)$. Положим

$$r(x,u) = \varphi\left((u^1 - x^1)_+, \ldots, (u^n - x^n)_+\right),$$

где индекс «+» обозначает операцию срезки: $z_+ = z \cdot [z \geqslant 0]$; функция $\varphi(z^1, \ldots, z^n)$ не убывает на всей области определения $[0, +\infty)^n$ и принимает нулевое значение $\varphi(z^1, \ldots, z^n) = 0$ тогда и только тогда, когда $z^1 = \cdots = z^n = 0$.

В качестве функции $\varphi$ подходят: максимум $\max(z^1, \ldots, z^n)$, сумма $z^1 + \cdots + z^n$, $p$-норма $\left((z^1)^p + \cdots + (z^n)^p\right)^{\frac{1}{p}}$, число ненулевых аргументов $\#\{i\colon z^i > 0\}$. Произведение $z^1 \cdots z^n$ и минимум $\min(z^1, \ldots, z^n)$ не подходят, так как они принимают нулевые значения не только в точке $(0, \ldots, 0)$.

**Восстановление частично заданных монотонных булевых функций.** Пусть объекты описываются $n$ бинарными признаками, $\mathbb{X} \subseteq \{0,1\}^n$. Тогда задача построения монотонного классификатора совпадает с известной задачей теории булевых функций о восстановлении частично заданной монотонной булевой функции. Стандартные алгоритмы, решающие данную задачу, обычно основаны на дополнительном требовании минимизировать сложность получаемой булевой функции $a(x)$ [51]. Критерий обобщающей способности CCV является новым в данной задаче.

Дополнительное затруднение связано с тем, что естественное в данном случае расстояние Хэмминга $\rho(x, x')$ принимает дискретный набор значений $0, \ldots, n$, что не позволяет однозначно ранжировать соседей и определять $y(x_{im})$ — класс $m$-го соседа объекта $x_i$. В результате монотонные классификаторы вида (14.7) могут отличаться на объектах, равноудалённых от границ классов 0 и 1.

Для получения верхней оценки CCV рассмотрим пессимистичный метод обучения $\mu_p$, который строит классификатор $\mu_p(X)$, ошибающийся на всех таких объектах.

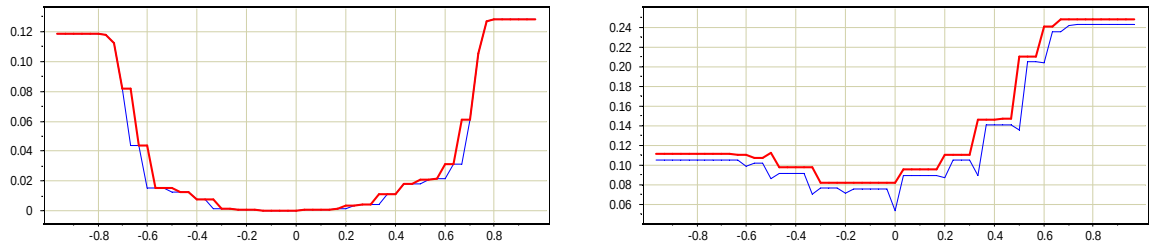
Нижняя оценка получается для оптимистичного метода $\mu_o$, при котором $\mu_o(X)$ правильно классифицирует все такие объекты.

Для каждого $x_i \in \mathbb{X}$ определим функции целочисленного аргумента $v \in \{0, \ldots, n\}$:

$$t_i(v) = \sum_{x_j \in \mathbb{X} \setminus x_i} \big[\rho_\lambda(x_i, x_j) < (1 + \lambda - 2y_i\lambda)v\big];$$

$$s_i(v) = \sum_{x_j \in \mathbb{X} \setminus x_i} \big[\rho_\lambda(x_i, x_j) = (1 + \lambda - 2y_i\lambda)v\big]\big[y_i \neq y_j\big];$$

$$p_i(v) = \sum_{x_j \in \mathbb{X} \setminus x_i} \big[\rho_\lambda(x_i, x_j) = v\big]\big[y_i = y_j\big].$$




(а) (б)

**Рис. 14.4.** Зависимость $C_o$ и $C_p$ от $\lambda$ при $\ell = k = 100$, $n = 15$: (а) выборка с широким зазором и числом объектов в классах $(100, 100)$; (б) выборка с зазором сложной формы и несбалансированными классами $(150, 50)$.

**Теорема 14.4 ([15]).** *Пусть выборка $\mathbb{X}$ монотонна. Тогда $C_o \equiv C(\mu_o, \mathbb{X}) \leqslant C(\mu, \mathbb{X}) \leqslant C(\mu_p, \mathbb{X}) \equiv C_p$, где*

$$C_p = \sum_{i=1}^{L} \sum_{v=0}^{n} \frac{C^{\ell}_{L-t_i(v)-1} - C^{\ell}_{L-t_i(v)-s_i(v)-1}}{kC^{\ell}_{L}}.$$

$$C_o = \sum_{i=1}^{L} \sum_{v=0}^{n} \frac{C^{\ell}_{L-t_i(v)-p_i(v)-1} - C^{\ell}_{L-t_i(v)-s_i(v)-p_i(v)-1}}{kC^{\ell}_{L}}.$$

**Численный эксперимент.** Цель эксперимента — проверить гипотезу, что проведение разделяющей поверхности посередине зазора ($\lambda = 0$) минимизирует CCV.

Эксперимент проводился на бинарных модельных данных, при различных значениях $n$, $L$, $\ell$, при различной сбалансированности классов, при различной форме зазора. Практически во всех экспериментах наблюдался чётко выраженный минимум CCV при $\lambda = 0$, при этом оценки $C_o$ и $C_p$ были очень близки, рис. 14.4. В задачах с малыми выборками или существенно несбалансированными классами положение минимума иногда незначительно отличалось от $\lambda = 0$.

## 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 14.1** (2). Обобщить теорему 14.2 на случай, когда выборка не обязательно монотонна.

**Задача 14.2** (1). Доказать теорему 14.3.

**Задача 14.3** (2)**.** Для случая, когда выборка не обязательно монотонна, обобщить монотонный классификатор ближайшего соседа (14.7) и теорему 14.3.

**Задача 14.4** ($3^*$)**.** Для случая, когда выборка не обязательно монотонна, вывести верхнюю оценку CCV по профилю монотонности через точную оценку CCV по профилю компактности.

**Задача 14.5** (4)**.** Доказать Теорему 14.4.

# 15 Приложение. О комбинаторной теории переобучения

Эти заметки адресованы, главным образом, студентам и аспирантам, выбравшим комбинаторную теорию переобучения в качестве темы своей научно-исследовательской работы. Возможно, они будут полезны и тем, кто интересуется методологией научного исследования. Здесь в хронологическом порядке рассказывается о том, как возникали и решались задачи, рассмотренные в данном курсе лекций.

Во всяком исследовании предшествующие результаты, как положительные, так и отрицательные, подсказывают направление дальнейших поисков. Иногда можно не заметить эти подсказки и потерять годы, имея результат буквально под носом. Мы учимся замечать их на конкретных примерах, собственных и чужих ошибках. Считается, что этот опыт индивидуален и передаётся от учителя к ученику в процессе работы. С другой стороны, все пользуются примерно одним и тем же «Алгоритмом НИР», который состоит в следующем: на каждом шаге выполняется определённая работа; порядок работ выбирается по ситуации, но все они важны, и ни одну нельзя упускать:

1) погружение в современную (в основном англоязычную) научную литературу;
2) решение простых частных задач, даже если они на первый взгляд бесполезны;
3) чередование теоретических исследований с экспериментами;
4) чередование попыток решить задачу с попытками изменить её постановку;
5) чередование самостоятельных размышлений с семинарами и обсуждениями;
6) упрощение обозначений и доказательств, поиск удачной терминологии.

Практически для каждого пункта можно вспомнить случаи, когда именно эта работа помогла вывести исследование из затянувшегося тупика. Или, наоборот, когда тупик возникал из-за того, что какая-то из этих работ подолгу откладывалась.

## §15.1 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 [38, 39, 40], 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 [45], 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 лишь интерпретировать 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 в тот момент потеряла для меня всякий смысл. Проблема завышенности скрывалась где-то внутри VC-теории, и её решение представлялось гораздо более важным.

## §15.2 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. 48), 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.

На том этапе никакого нового результата по существу не было. Ключевая теорема VC-теории была переписана на другом языке, более лаконичном и естественном

(стр. 48). 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 безнадёжно отставал. Многое уже было сделано, оставалось только узнавать, как это называется по-английски. Оценки, использующие информацию из 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 [67]. Учёт только части алгоритмов с невысоким уровнем ошибок — algorithmic luckiness [71], 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, 59]. 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 безнадёжная. Обычно речь шла лишь о зажатых, то есть не сильно завышенных, оценках (tight bounds) [81, 94, 52].

## §15.3 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. 55). 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 [78], Лэнгфорда [81], Хербрича [71], 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. 55 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 $\leqslant$ 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 переобучении пары алгоритмов (стр. **??**) 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. **??**). 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. 73). 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 $D$ первых, самых лучших, алгоритмов цепи (стр. 74). 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. Како-

му 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. **??**). 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.

## §15.4 «Игрушечные» частные случаи

**Конец 2008 года.** Первые точные оценки вероятности переобучения были найдены для монотонной цепи, унимодальной цепи и единичной окрестности (стр. 76–83). 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. 63). 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 результатом 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. 66, 79).

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. 66). 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 иногда удаётся найти эффективные. Скорее даже угадать, чем найти, что превращает эту деятельность в своего рода искусство.

С помощью принципа ПЗМ была решена задача о рекуррентном вычислении вероятности переобучения при добавлении алгоритмов по одному (стр. **??**). 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. **??**). Это красивые частные случаи, но, к сожалению, за ними так и не обнаружилось никакого общего способа вывода оценок.

## §15.5 Поиск обобщений и альтернатив

Общая оценка, в явном виде учитывающая свойства расслоения и связности, была получена с помощью «умозрительного выполнения» рекуррентной процедуры (стр. **??**). 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-Червоненкиса (вероятность большого равномерного отклонения частот ошибок в двух выборках), 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 вокруг значения размерности пространства. Позже он получил оценку и для дисперсии связности.

Павел Ботов обобщил монотонную цепь алгоритмов на случай произвольной размеренности и решил задачу о переобучении многомерной монотонной сети алгоритмов (стр. 97). Это обобщение далось на удивление легко. Затем он получил оценки для пучка монотонных цепей и для многомерной унимодальной сети, но в этих случаях доказательства были сложнее. Удалось даже показать, что вероятность переобучения унимодальной сети приблизительно равна вероятности переобучения монотонной 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 размерности, 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 вероятностью переобучения монотонной сети, если подобрать два её параметра: частоту ошибок лучшего алгоритма и размерность сети (стр. 101). 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 свойством симметрии (стр. **??**). 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 булевом кубе, его нижних слоёв и отдельно взятого центрального слоя шара. А Илья Решетняк заметил, что оценка для центрального слоя шара может быть легко получена 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 от-

дельных объектов. Это позволило ему построить достаточно эффективный алгоритм формирования множества эталонов (prototype selection) путём последовательного добавления или удаления объектов (стр. 131).

В сентябре 2009 года все эти результаты были доложены авторами на конференции ММРО-14.

Модельные семейства типа хэммингових 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 НИР» гарантированно выводит из творческих тупиков, если ни один из шагов не упускать. Теперь 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. 86). 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. 89). 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 рademахеровской 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 обучении (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. 103). Отсюда немедленно следует верхняя оценка вероятности переобучения. Причём этот лёгко доставшийся результат гораздо лучше оценки Кочедыкова, стоившей нескольких страниц выкладок (стр. 105).

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 улучшение 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. 88). 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 лишний раз алгоритмы с одинаковыми векторами ошибок (стр. 115). 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. Основные вопросы — как применять комбинаторные оценки к реальным семействам и как улучшать с их помощью методы обучения.

## §15.6 Десять открытых проблем

В заключение приведём перечень основных проблем комбинаторной теории переобучения, которые пока ждут своего решения.

1. Получение оценок, учитывающих связи между алгоритмами с хеммингловым расстоянием, большим 1. Их игнорирование является, по всей видимости, основной причиной завышенности оценки расслоения–связности.

2. Получение оценок, не зависящих от характеристик полной выборки $\mathbb{X}$, а только от наблюдаемых величин, которые могут быть вычислены по конкретной обучающей подвыборке $X$.

3. Расширение класса методов обучения, для которых комбинаторные оценки позволяют улучшать обобщающую способность.

4. Получение эффективно вычислимых оценок для произвольных цепей алгоритмов. Затем, с их помощью, совершенствование методов обучения, сводящихся к решению последовательности одномерных задач оптимизации.

5. Получение эффективно вычислимых оценок расслоения–связности для семейства линейных классификаторов. Затем, с их помощью, совершенствование методов отбора признаков.

6. Обоснование и, возможно, уточнение принципов регуляризации, аппроксимации эмпирического риска, максимизации зазора.

7. Обобщение комбинаторных оценок на случай небинарных функций потерь.

8. Проверка гипотезы о том, что значения связности концентрируются вокруг значения локальной размерности пространства параметров.

9. Проверка гипотезы о сепарабельности профиля расслоения–связности.

10. Получение оценок обобщающей способности для задач динамического (online) обучения.

# 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] *Алимов 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] *Бонгард М. М.* Проблема узнавания. — М.: Наука, 1967.

[6] *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.

[7] *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.

[8] *Вапник В. Н.* Восстановление зависимостей по эмпирическим данным. — М.: Наука, 1979.

[9] *Вапник В. Н., Червоненкис А. Я.* О равномерной сходимости частот появления событий к их вероятностям // *ДАН СССР*. — 1968. — Т. 181, № 4. — С. 781–784.

[10] *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.

[11] *Вапник В. Н., Червоненкис А. Я.* Теория распознавания образов. — М.: Наука, 1974.

[12] *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.

[13] *Воронцов К. В.* Обзор современных исследований по проблеме качества обучения алгоритмов // *Таврический вестник информатики и математики*. — 2004. — № 1. — С. 5–24.

[14] *Воронцов К. В., Колосков А. О.* Профили компактности и выделение опорных объектов в метрических алгоритмах классификации // *Искусственный Интеллект*. — 2006. — С. 30–33.

[15] *Воронцов К. В., Махина Г. А.* Принцип максимизации зазора для монотонного классификатора ближайшего соседа // 15-я всероссийская конференция «Математические методы распознавания образов», Петрозаводск. — 2011. — С. 64–67.

[16] *Воронцов К. В., Решетняк И. М.* Точные комбинаторные оценки обобщающей способности онлайнового обучения // Межд. конф. Интеллектуализация обработки информации ИОИ-8. — М.: МАКС Пресс, 2010. — С. 24–27.

[17] *Гаек Я., Шидак З.* Теория ранговых критериев. — М.: Наука, 1971.

[18] *Гоппа В. Д.* Введение в алгебраическую теорию информации. — М.: Наука, 1995.

[19] *Грэхем Р., Кнут Д., Паташник О.* Конкретная математика. — М.: Мир, 1998.

[20] *Гуз И. С.* Нелинейные монотонные композиции классификаторов // Математические методы распознавания образов-13. — М.: МАКС Пресс, 2007. — С. 111–114.

[21] *Гуров С. И.* Точечная оценка вероятности 0-события // Всеросс. конф. Математические методы распознавания образов-14. — М.: МАКС Пресс, 2009. — С. 22–25.

[22] *Донской В. И.* Алгоритмы обучения, основанные на построении решающих деревьев // *ЖВМиМФ.* — 1982. — Т. 22, № 4. — С. 963–974.

[23] *Донской В. И.* Колмогоровская сложность классов общерекурсивных функций с ограниченной ёмкостью // *Таврический вестник информатики и математики.* — 2005. — № 1. — С. 25–34.

[24] *Дэйвид Г.* Порядковые статистики. — М.: Наука, 1979. — 336 с.

[25] *Дюличева Ю. Ю.* Оценка VCD $r$-редуцированного эмпирического леса // *Таврический вестник информатики и математики.* — 2003. — № 1. — С. 31–42.

[26] *Журавлёв Ю. И.* Об алгебраическом подходе к решению задач распознавания или классификации // *Проблемы кибернетики.* — 1978. — Т. 33. — С. 5–68.

[27] *Журавлёв Ю. И., Рязанов В. В., Сенько О. В.* «Распознавание». Математические методы. Программная система. Практические применения. — М.: Фазис, 2006.

[28] *Загоруйко Н. Г.* Прикладные методы анализа данных и знаний. — Новосибирск: ИМ СО РАН, 1999.

[29] *Загоруйко Н. Г., Ёлкина В. Н., Лбов Г. С.* Алгоритмы обнаружения эмпирических закономерностей. — Новосибирск: Наука, 1985.

[30] *Ивахненко А. А.* О вероятности переобучения пороговых конъюнкций // Межд. конф. Интеллектуализация обработки информации ИОИ-8. — М.: МАКС Пресс, 2010. — С. 57–60.

[31] *Ивахненко А. А.* Точная верхняя оценка вероятности переобучения для корректных логических правил // *Труды МФТИ.* — 2010. — Т. 2, № 3. — С. 81–87.

[32] *Кобзарь А. И.* Прикладная математическая статистика. — М.: Физматлит, 2006.

[33] *Колмогоров А. Н.* Теория информации и теория алгоритмов / Под ред. Ю. В. Прохоров. — М.: Наука, 1987. — 304 с.

[34] *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.

[35] *Лбов Г. С.* Методы обработки разнотипных экспериментальных данных. — Новосибирск: Наука, 1981.

[36] *Лбов Г. С., Котюков В. И., Машаров Ю. П.* Метод обнаружения логических закономерностей на эмпирических таблицах // *Вычислительные системы.* — 1976. — Т. 67. — С. 29–42.

[37] *Матросов В. Л.* Корректные алгебры ограниченной ёмкости над множествами некорректных алгоритмов // *ДАН СССР.* — 1980. — Т. 253, № 1. — С. 25–30.

[38] *Матросов В. Л.* Ёмкость алгебраических расширений модели алгоритмов вычисления оценок // *ЖВМиМФ.* — 1984. — Т. 24, № 11. — С. 1719–1730.

[39] *Матросов В. Л.* Нижние границы ёмкости многомерных алгебр алгоритмов вычисления оценок // *ЖВМиМФ.* — 1984. — Т. 24, № 12. — С. 1881–1892.

[40] *Матросов В. Л.* Ёмкость алгоритмических многочленов над множеством алгоритмов вычисления оценок // *ЖВМиМФ.* — 1985. — Т. 25, № 1. — С. 122–133.

[41] *Махина Г. А.* Оценка обобщающей способности для монотонных алгоритмов класси-

фикации // 16-я международная конференция «Проблемы теоретической кибернетики». Нижний Новгород, 20–25 июня 2011. — 2011. — С. 307–310.

[42] *Норушис А.* Построение логических (древообразных) классификаторов методами нисходящего поиска (обзор) // Статистические проблемы управления. Вып. 93 / Под ред. Ш. Раудис. — Вильнюс, 1990. — С. 131–158.

[43] *Орлов А. И.* Эконометрика: Учебник для вузов. — М.: Экзамен, 2003. — 576 с.

[44] *Райгородский А. М.* Экстремальные задачи теории графов и анализ данных. — М.: НИЦ «Регулярная и хаотическая динамика», Институт компьютерных исследований, 2008. — 118 с.

[45] *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.

[46] *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.

[47] *Сёмочкин А. Н.* Линейные достроения частичного порядка на конечных множествах // *Деп. в ВИНИТИ.* — 1998. — № 2964–В98. — С. 19.

[48] *Сёмочкин А. Н.* Оценки функционала качества для класса алгоритмов с универсальными ограничениями монотонности // *Деп. в ВИНИТИ.* — 1998. — № 2965–В98. — С. 20.

[49] *Смирнов Н. В.* Оценка расхождения между эмпирическими кривыми распределения в двух независимых выборках // *Бюлл. Московского ун-та, серия А.* — 1939. — № 2. — С. 3–14.

[50] *Abu-Mostafa Y. S.* Hints // *Neural Computation.* — 1995. — Vol. 7, no. 4. — Pp. 639–671.

[51] *Akers S. B.* A truth table method for synthesis of combinational logic // *IRE Trans.* — 1961. — Vol. EC-10, no. 4. — Pp. 604–615.

[52] *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.

[53] *Anthony M.* Generalization error bounds for threshold decision lists // *Journal of Machine Learning Research.* — 2004. — Vol. 5. — Pp. 189–217.

[54] *Anthony M., Shawe-Taylor J.* A result of Vapnik with applications // *Discrete Applied Mathematics.* — 1993. — Vol. 47, no. 2. — Pp. 207–217.

[55] *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.

[56] *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.

[57] *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.

[58] *Bax E. T.* Similar classifiers and VC error bounds: Tech. Rep. CalTech-CS-TR97-14: 1997.

[59] *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.

[60] *Breiman L.* Technical note: Some properties of splitting criteria // *Machine Learning.* — 1996. — Vol. 24. — Pp. 41–47.

[61] *Breiman L.* Arcing classifiers // *The Annals of Statistics.* — 1998. — Vol. 26, no. 3. — Pp. 801–849.

[62] *Brin S.* Near neighbor search in large metric spaces // 21th International Conference on Very Large Data Bases (VLDB 1995). — 1995. — Pp. 574–584.

[63] *Chavez E., Navarro G., Baeza-yates R., Marroquin J. L.* Searching in metric spaces // *ACM Computing Surveys.* — 1999. — Vol. 33. — Pp. 273–321.

[64] *Chvátal V.* The tail of the hypergeometric distribution // *Discrete Mathematics.* — 1979. — Vol. 25, no. 3. — Pp. 285–287.

[65] *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.

[66] *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.

[67] *Freund Y.* Self bounding learning algorithms // COLT: Proceedings of the Workshop on Computational Learning Theory, Morgan Kaufmann Publishers. — 1998.

[68] *Freund Y., Schapire R. E.* Experiments with a new boosting algorithm // International Conference on Machine Learning. — 1996. — Pp. 148–156.

[69] *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.

[70] *Hastie T., Tibshirani R., Friedman J.* The Elements of Statistical Learning, 2nd ed. — Springer, 2009.

[71] *Herbrich R., C.Williamson R.* Algorithmic luckiness // *Journal of Machine Learning Research.* — 2002. — no. 3. — Pp. 175–212.

[72] *Herbrich R., Williamson R. C.* Learning and generalization: theoretical bounds // The handbook of brain theory and neural networks, 2nd Edition. — Cambridge, MA, USA: MIT Press, 2002. — Pp. 619–623.

[73] *Hosmer D. W., Lemeshow S.* Applied Logistic Regression, second ed. — New York: Wiley, 2000.

[74] *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.

[75] *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.

[76] *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.

[77] *Kochedykov D. A.* A combinatorial approach to hypothesis similarity in generalization bounds // *Pattern Recognition and Image Analysis.* — 2011. — Vol. 21, no. 4. — P. (to appear).

[78] *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.

[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 // AAAI/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.* A brief introduction to boosting // International Joint Conference on Artificial Intelligence. — 1999. — Pp. 1401–1406.

[98] *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.

[99] *Schapire R. E., Singer Y.* Improved boosting using confidence-rated predictions // *Machine Learning*. — 1999. — Vol. 37, no. 3. — Pp. 297–336.

[100] *Sill J.* The capacity of monotonic functions // *Discrete Applied Mathematics (special issue on VC dimension)*. — 1998. — Vol. 86. — Pp. 95–107.

[101] *Sill J.* Monotonicity and connectedness in learning systems: Ph.D. thesis / California Institute of Technology. — 1998.

[102] *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.

[103] *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.

[104] *Valiant L. G.* A theory of the learnable // *Communications of the ACM*. — 1984. — Vol. 27. — Pp. 1134–1142.

[105] *Vapnik V.* Estimation of Dependencies Based on Empirical Data. — Springer-Verlag, New York, 1982.

[106] *Vapnik V.* The nature of statistical learning theory. — Springer-Verlag, New York, 1995.

[107] *Vapnik V.* Statistical Learning Theory. — Wiley, New York, 1998.

[108] *Vayatis N., Azencott R.* Distribution-dependent Vapnik-Chervonenkis bounds // *Lecture Notes in Computer Science*. — 1999. — Vol. 1572. — Pp. 230–240.

[109] *Vorontsov K. V.* Combinatorial probability and the tightness of generalization bounds // *Pattern Recognition and Image Analysis*. — 2008. — Vol. 18, no. 2. — Pp. 243–259.

[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.* Tight combinatorial generalization bounds for threshold conjunction rules // 4th International Conference on Pattern Recognition and Machine Intelligence (PReMI'11). June 27 – July 1, 2011. — Lecture Notes in Computer Science. Springer-Verlag, 2011. — Pp. 66–73.

[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. — Pp. 484–494.