Федеральное государственное образовательное учреждение
высшего профессионального образования
«Московский физико–технический институт
(государственный университет)»

---

На правах рукописи

УДК 519.22

Фрей Александр Ильич

# Теоретико-групповой подход в комбинаторной теории переобучения

Специальность 05.13.17 — теоретические основы информатики

Диссертация на соискание ученой степени
кандидата физико-математических наук

Научный руководитель: д. ф.-м. н. Воронцов Константин Вячеславович

Москва
2013

# Оглавление

# Введение

Диссертационная работа посвящена проблеме повышения точности комбинаторных оценок вероятности переобучения.

**Актуальность темы.** При решении задач обучения по прецедентам, восстановления зависимостей по эмпирическим данным, классификации, распознавания образов, прогнозирования часто возникает проблема переобучения. Она состоит в том, что решающая функция (алгоритм), построенная по конечной обучающей выборке, может допускать ошибки на объектах контрольной выборки существенно чаще, чем на объектах обучающей выборки. Для контроля переобучения на этапе построения алгоритма необходимо иметь оценки вероятности переобучения. Такие оценки известны в статистической теории обучения, однако они либо сильно завышены, либо имеют слишком узкую область применимости.

**Степень разработанности темы.** Основы статистической теории обучения были заложены в работах В. Н. Вапника и А. Я. Червоненкиса в конце 60-х годов. Ими была доказана состоятельность обучения по прецедентам и получены количественные оценки, связывающие обобщающую способность метода обучения с длиной обучающей выборки и сложностью семейства алгоритмов. Основной проблемой этих оценок является их завышенность. Для устранения завышенности предлагалось строить оценки, зависящие от выборки (D. Haussler, 1992); учитывать ширину зазора, разделяющего классы (P. Bartlett, 1998); строить оценки на основе локальной рademахеровской сложности семейства алгоритмов (V. Koltchinskii, 1998); учитывать априорные распределения на множестве алгоритмов (L. Valiant, 1982; D. McAllester, 1999; J. Langford, 2005); а также ряд других подходов.

Комбинаторная теория переобучения показала, что для повышения точности оценок и сокращения переобучения необходимо одновременно учитывать эффекты расслоения и сходства в семействах алгоритмов (К. В. Воронцов, 2010). Была получена оценка расслоения-связности, справедливая для широкого класса семейств, представимых в виде связного графа (К. В. Воронцов, А. А. Ивахненко, И. М. Решетняк, 2010). Для некоторых модельных частных случаев было показано, что этого достаточно для получения неулуч-

шаемых (точных) оценок. Таким образом, комбинаторная теория переобучения является новым перспективным подходом. Данная работа направлена на ее дальнейшее развитие: расширение границ применимости, разработку новых методов вывода оценок обобщающей способности и повышение точности этих оценок.

**Цели и задачи работы:** повышение точности комбинаторных оценок вероятности переобучения; переход от требования связности к более слабому требованию сходства алгоритмов; разработка новых методов получения оценок обобщающей способности, применимых к несвязным семействам алгоритмов высокой мощности.

**Научная новизна.** Впервые получены неулучшаемые оценки вероятности переобучения для рандомизированного метода минимизации эмпирического риска. Для их получения разработан новый теоретико-групповой подход, основанный на учете симметрий множества алгоритмов. С его помощью получены неулучшаемые оценки вероятности переобучения для девяти модельных семейств алгоритмов. Получена комбинаторная оценка вероятности переобучения, основанная на разложении множества алгоритмов на непересекающиеся подмножества (кластеры). Каждый кластер пополняется алгоритмами до объемлющего множества алгоритмов с известной точной оценкой вероятности переобучения. Итоговая оценка учитывает сходство алгоритмов внутри каждого кластера и расслоение алгоритмов по числу ошибок между разными кластерами. Данная оценка применима к широкому классу семейств, в том числе и к семействам, не обладающим свойством связности.

**Теоретическая и практическая значимость.** Данная работа вносит существенный вклад в развитие комбинаторной теории переобучения и расширяет границы ее применимости на несвязные семейства алгоритмов высокой мощности.

**Методы исследования.** Для получения оценок вероятности переобучения использована слабая (перестановочная) вероятностная аксиоматика, комбинаторная теория переобучения, элементы комбинаторки, теории групп, теории вероятностей и теории графов. Для проверки точности комбинаторных оценок проведены вычислительные эксперименты на модельных данных и задачах из репозитория UCI.

**Положения, выносимые на защиту.**

1. Теоретико-групповой метод орбит, позволяющий выводить оценки вероятности переобучения для симметричных семейств алгоритмов и рандомизированного метода минимизации эмпирического риска.

2. Точные оценки вероятности переобучения рандомизированного метода минимизации эмпирического риска для модельных семейств: монотонной и унимодальной сетей, слоя хэммингова шара и ряда других.

3. Общая оценка вероятности переобучения, основанная на разложении и покрытии множества алгоритмов.

4. Экспериментальное подтверждение того, что новая оценка в некоторых случаях менее завышена по сравнению с другими комбинаторными оценками вероятности переобучения.

**Степень достоверности и апробация работы.** Достоверность результатов подтверждена математическими доказательствами, экспериментальной проверкой полученных оценок переобучения на реальных задачах классификации; публикациями результатов исследования в рецензируемых научных изданиях, в том числе рекомендованных ВАК РФ. Результаты работы докладывались, обсуждались и получили одобрение специалистов на следующих научных конференциях и семинарах:

- Всероссийская конференция «Математические методы распознавания образов» ММРО-14, 2009 г. [31];
- 52-я научная конференция МФТИ, 2009 г. [32];
- Международная конференция «Интеллектуализация обработки информации» ИОИ-8, 2010 г. [33];
- Всероссийская конференция «Математические методы распознавания образов» ММРО-15, 2011 г. [34];
- Международная конференция «25th European Conference on Operational Research», 2012 г.

- Международная конференция «Интеллектуализация обработки информации» ИОИ-9, 2012 г. [35];
- Научные семинары отдела Интеллектуальных систем Вычислительного центра РАН и кафедры «Интеллектуальные системы» МФТИ, 2010 – 2013 г.г.

**Публикации по теме диссертации** в изданиях из списка ВАК РФ — одна [54]. Другие публикации по теме диссертации: [31, 32, 33, 34, 35, 72, 36].

**Структура и объем работы.** Работа состоит из введения, пяти глав, заключения, списка использованных источников, включающего 78 наименований. Общий объем работы составляет 102 страницы.

**Краткое содержание работы по главам.** В главе 1 дается краткий обзор современного состояния теории статистического обучения (statistical learning theory). Обсуждается проблема завышенности классических оценок переобучения, приводятся мотивации перехода от классических теоретико-вероятностных постановок задач к комбинаторным.

В главе 2 вводится комбинаторная постановка задачи. Описываются эффекты расслоения и связности, а также их влияние на переобучение. Приводится комбинаторная оценка расслоения-связности. Демонстрируются результаты двух экспериментов, указывающих на слабое место этой оценки. Анализ и устранение причин завышенности оценки расслоения-связности является основной целью данной диссертационной работы.

В главе 3 вводятся понятия рандомизированного метода обучения и группы симметрии множества алгоритмов. Предлагается теоретико-групповой метод орбит для вывода оценок вероятности переобучения симметричных семейств алгоритмов. Предлагается общая оценка вероятности переобучения, основанная на разложении и покрытии множества алгоритмов. Метод порождающих и запрещающих множеств, предложенный К. В. Воронцовым, обобщается по двум направлениям: во-первых, на случай разложения множества алгоритмов на кластеры, во-вторых, на случай рандомизированного метода обучения.

В главе 4 получены точные оценки вероятности переобучения рандомизированного метода минимизации эмпирического риска (РМЭР) для девяти модельных семейств алгоритмов. При выводе оценок используется математический аппарат, разработанный в главе 4: разложение вероятности переобучения по орбитам действия группы симметрий

на множестве алгоритмов или на множестве разбиений выборки, а также теорема о порождающих и запрещающих множеств для РМЭР.

В главе 5 описываются вычислительные эксперименты на реальных данных из репозитория UCI. В экспериментах сравнивается завышенность уже известных комбинаторных оценок вероятности переобучения и новой оценки, основанной на разложении и покрытии множества алгоритмов.

## Благодарности

Автор выражает глубокую признательность научному руководителю д.ф.-м.н. Константину Вячеславовичу Воронцову за постановку задачи и постоянную поддержку в ходе исследований, заведующему кафедрой «Интеллектуальные системы» чл.-корр. РАН Константину Владимировичу Рудакову за ценные замечания, а также Андрею Ивахненко, Илье Толстихину, Евгению Соколову и другим коллегам по учебе в аспирантуре МФТИ за плодотворные обсуждения работы.

# Глава 1. Теория статистического обучения

В данной главе приводится обзор общепринятого подхода к оценке обобщающей способности, который основан на классическом определении вероятности [49, 52]. Настоящий обзор не претендует на полноту изложения. Основное внимание будет уделено оценкам, использующим понятие VC-размерности, локальным оценкам избыточного риска и PAC-Bayes оценкам переобучения, применимым к линейным решающим правилам.

## 1.1 Основные определения SLT

В данном разделе мы будем придерживаться обозначений, введенных в [59]. Пусть $(\Omega, \Sigma, \mathcal{P})$ — вероятностное пространство, а $X, X_1, \ldots, X_n$ — одинаково распределенные случайные величины, принимающие значения в измеримом пространстве $(S, \mathscr{A})$ с общим распределением $P$. Через $P_n$ обозначим эмпирическое распределение, построенное по $n$ наблюдениям:

$$P_n = \frac{1}{n} \sum_{j=1}^{n} \delta_{X_j},$$

где $\delta_x$, $x \in S$ — дельта-функция Дирака. Пусть $\mathscr{F} = \{f \colon S \to [0,1]\}$ — класс измеримых функций. В дальнейшем мы будем интерпретировать значения функций $f \in \mathscr{F}$ как «потери», а математическое ожидание $f(X)$,

$$\mathsf{E} f(X) = \int_S f dP = Pf,$$

как риск, связанный с определенным «решающим правилом».

**Пример 1.1.** Рассмотрим задачу классификации. Пусть $\mathbb{X}$ — множество объектов, а $\mathbb{Y}$ — множество вещественных (или целочисленных) ответов. Положим $S = \mathbb{X} \times \mathbb{Y}$. Для произвольного алгоритма классификации $g \colon \mathbb{X} \to \mathbb{Y}$ и прецедента $(x, y) \in S$ определим величину потерь как $f(x, y) = (g(x) - y)^2$. После нормировки эти потери можно рассматривать как функцию $f \colon S \to [0, 1]$. Тогда класс функций потерь $\mathscr{F}$ получится, если перебрать все классификаторы $g$ из определенного семейства (например, из семейства всех линейных классификаторов в исходном признаковом пространстве задачи). В дальнейшем для краткости обозначений нам будет удобнее забыть об этой структуре множеств $S$ и $\mathscr{F}$.

Нас будет интересовать *задача минимизации риска*

$$Pf \to \min, f \in \mathscr{F}, \tag{1.1}$$

в условиях неизвестного распределения $P$. Поэтому вместе с задачей (1.1) будет также рассматриваться *задача минимизации эмпирического риска*

$$\frac{1}{n}\sum_{j=1}^{n} f(X_j) = \int_S f dP_n = P_n f \to \min, f \in \mathscr{F}. \tag{1.2}$$

**Определение 1.1.** *Избыточным риском функции $f \in \mathscr{F}$ назовем величину*

$$\mathcal{E}(f) = Pf - \inf_{g\in\mathscr{F}} Pg.$$

Обозначим через $\hat{f} = \hat{f}_n \in \operatorname{Arg}\min_{f\in\mathscr{F}} P_n f$ решение задачи минимизации эмпирического риска (1.2). Функция $\hat{f}_n$ используется как приближенное решение задачи минимизации истинного риска (1.1), а значит, избыточный риск $\mathcal{E}(\hat{f}_n)$ является естественной мерой ошибки приближенного решения. Кроме избыточного риска полезно рассматривать уклонение $Pf - P_n f$, показывающее, как точно эмпирический риск приближает истинный риск.

Семейство случайных величин $\{(P - P_n)f\}_{f\in\mathscr{F}}$ называют *эмпирическим процессом, индексированным классом $\mathscr{F}$*. Нормой эмпирического процесса называют величину $||P - P_n||_{\mathscr{F}} \equiv \sup_{f\in\mathscr{F}} |(P - P_n)f|$.

В дальнейшем нас будут интересовать оценки следующего вида:

$$\begin{aligned} &\mathsf{P}^n\left\{\mathcal{E}(\hat{f}_n) \geqslant \varepsilon\right\} \leqslant B_1(\varepsilon, n, \mathscr{F}),\\ &\mathsf{P}^n\left\{P\hat{f}_n - P_n\hat{f}_n \geqslant \varepsilon\right\} \leqslant B_2(\varepsilon, n, \mathscr{F}), \end{aligned} \tag{1.3}$$

а также доверительные интервалы при заданном уровне надежности $\eta$:

$$\begin{aligned} &\mathsf{P}^n\left\{\mathcal{E}(\hat{f}_n) \geqslant \varepsilon_1(\eta, n, \mathscr{F})\right\} \leqslant \eta,\\ &\mathsf{P}^n\left\{P\hat{f}_n - P_n\hat{f}_n \geqslant \varepsilon_2(\eta, n, \mathscr{F})\right\} \leqslant \eta. \end{aligned} \tag{1.4}$$

Во всех оценках (1.3) и (1.4) вероятность $\mathsf{P}^n \equiv P^{\otimes n}$ соответствует реализациям выборки $(X_1, \ldots, X_n) \subset S$.

## 1.2 Неравенства концентрации меры

Неравенства концентрации меры [48, 51, 65] играют большую роль при выводе оценок вида (1.3) и (1.4). В данном параграфе мы рассмотрим простейший случай (класс $\mathscr{F}$,

состоящий лишь из одной функции $f$) и покажем, как в этой ситуации применить неравенство Хевдинга. Случай $|\mathscr{F}| = 1$ означает, что всякое обучение отсутствует, а различия между истинным риском $Pf$ и эмпирическим риском $P_n f$ возникают лишь из-за нашей вероятностной модели данных.

Заметим, что $Pf - P_n f = \mathsf{E} f(X) - \frac{1}{n}\sum\limits_{j=1}^{n} f(X_j)$. Следовательно, по закону больших чисел эмпирический риск хорошо аппроксимирует истинный риск при больших значениях $n$:

$$\mathsf{P}^n\Big\{\lim_{n\to\infty}\frac{1}{n}\sum_{j=1}^{n} f(X_j) - \mathsf{E} f(X) = 0\Big\} = 1.$$

Неравенство Хевдинга дает количественную оценку на скорость этой сходимости.

**Теорема 1.1 (Хевдинг).** *Пусть $X_1, \ldots, X_n$ — независимые одинаково распределенные случайные величины, принимающие значения на отрезке $[a, b]$. Тогда для всех $\varepsilon > 0$ выполнено*

$$\mathsf{P}^n\Big\{\mathsf{E} f(X) - \frac{1}{n}\sum_{j=1}^{n} f(X_j) \geqslant \varepsilon\Big\} \leqslant \exp\Big(-\frac{2n\varepsilon^2}{(b-a)^2}\Big). \tag{1.5}$$

Обозначим правую часть (1.5) через $\delta$ и выразим $\varepsilon$ через $\delta$. Получим, что с вероятностью не менее $1 - \delta$ выполнено следующее:

$$Pf - P_n f \leqslant (b-a)\sqrt{\frac{\log\frac{1}{\delta}}{2n}}. \tag{1.6}$$

Более точную оценку можно получить из неравенства Бернштейна. Оно уточняет неравенство Хевдинга благодаря учету дисперсии случайных величин $X_1, \ldots, X_n$.

**Теорема 1.2 (Бернштейн).** *Пусть $X_1, \ldots, X_n$ — независимые (но не обязательно одинаково распределенные) случайные величины с нулевым математическим ожиданием, принимающие значения на отрезке $[-c, c]$. Пусть*

$$\sigma^2 = \frac{1}{n}\sum_{j=1}^{n} \mathsf{D} X_j.$$

*Тогда для любого $\varepsilon > 0$ выполнено*

$$\mathsf{P}^n\Big\{\frac{1}{n}\sum_{j=1}^{n} X_j > \varepsilon\Big\} \leqslant \exp\Big(-\frac{n\varepsilon^2}{2\sigma^2 + 2c\varepsilon/3}\Big).$$

При использовании неравенства Бернштейна получим следующую оценку, выполненную с вероятностью не менее $1 - \delta$:

$$Pf - P_n f \leqslant \sqrt{\frac{2\sigma^2 \log\frac{1}{\delta}}{n}} + \frac{2\log\frac{1}{\delta}}{3n}, \tag{1.7}$$

где $\sigma^2 = \mathsf{D}f(X)$.

К сожалению, неравенства (1.6) и (1.7) оказываются верными лишь для фиксированной функции $f$, и оказываются неверными, если функция выбирается из класса $\mathscr{F}$ по данным $(X_1, \ldots, X_n)$.

## 1.3 Теория Вапника-Червоненкиса

Чтобы справиться с ситуацией $|\mathscr{F}| > 1$, величину $P\hat{f}_n - P_n\hat{f}_n$ для функции $\hat{f} \in \mathscr{F}$ ограничивают сверху супремумом по всему классу функций [4, 5]:

$$P\hat{f}_n - P_n\hat{f}_n \leqslant \sup_{f\in\mathscr{F}} Pf - P_nf.$$

Для конечного класса функций $|\mathscr{F}| < \infty$ эту величину легко оценить с помощью неравенства Буля. Действительно, пусть $\mathscr{F} = \{f_1, \ldots, f_N\}$. Тогда для каждой функции $f_i$ рассмотрим множество

$$C_i = \{(X_1, \ldots, X_n) \in S^n \colon Pf_i(X) - P_nf_i(X) > \varepsilon\}$$

— множество тех выборок, для которых $f_i$ оказалась преобученной. Тогда по неравенству Буля выполнено

$$\mathsf{P}^n\{C_1 \cup \cdots \cup C_N\} \leqslant \sum_{i=1}^{N} \mathsf{P}^n\{C_i\}.$$

Применяя для каждой $f_i$ неравенство Хевдинга, получим

$$\mathsf{P}^n\left\{\sup_{f\in\mathscr{F}} Pf - P_nf \geqslant \varepsilon\right\} \leqslant N\varepsilon(-2n\varepsilon^2).$$

Это значит, с вероятностью не менее $1-\delta$ выполнено

$$P\hat{f}_n - P_n\hat{f}_n \leqslant \sqrt{\frac{\log N + \log\frac{1}{\delta}}{2n}}.$$

Теория Вапника-Червоненкиса [6, 71] позволяет справиться даже с бесконечным классом функций $\mathscr{F}$ в важном частном случае бинарной функции потерь ($f\colon S \to \{0,1\}$). Для этого класс функций $\mathscr{F}$ «проецируется» на конечную выборку. Более строго, рассмотрим выборку $(X_1, \ldots, X_n)$, и пусть

$$\mathscr{F}_{X_1,\ldots,X_n} = \{(f(X_1), \ldots, f(X_n)\colon f \in \mathscr{F}\}.$$

Мощность этого множества называют *коэффициентом разнообразия*. Коэффициент разнообразия зависит и от семейства функций $\mathscr{F}$, и от выборки $(X_1, \ldots, X_n)$.

**Определение 1.2.** *Функцией роста* $S_{\mathscr{F}}(n)$ *называют максимальное число способов, которым* $n$ *объектов могут быть классифицированы функциями из* $\mathscr{F}$*:*

$$S_{\mathscr{F}}(n) = \sup_{(X_1,\ldots,X_n)} |\mathscr{F}_{X_1,\ldots,X_n}|.$$

**Теорема 1.3 (Вапник, Червоненкис, [6]).** *Для всех* $\delta > 0$ *с вероятностью не менее* $1-\delta$ *выполнено*

$$P\hat{f}_n - P_n\hat{f}_n \leqslant 2\sqrt{2\frac{\log S_{\mathscr{F}}(2n) + \log\frac{2}{\delta}}{n}}.$$

Очевидно, что для бинарной функции потерь функция роста не превосходит $2^n$.

**Определение 1.3.** *Пусть* $h$ *— минимальное число, такое что* $S_{\mathscr{F}}(h) < 2^h$*. Тогда* $h$ *называют размерностью Вапника-Червоненкиса (или* $VC$*-размерностью) для семейства* $\mathscr{F}$*.*

Оказывается, что для многих реальных семейств алгоритмов $h < \infty$, а функция роста при $n \geqslant h$ растет полиномиально:

$$S_{\mathscr{F}}(h) \leqslant \sum_{i=0}^{h} C_n^i \leqslant \left(\frac{en}{h}\right)^h.$$

Следовательно, выполнена оценка

$$P\hat{f}_n - P_n\hat{f}_n \leqslant 2\sqrt{2\frac{h\log\frac{2en}{h} + \log\frac{2}{\delta}}{n}}. \tag{1.8}$$

Таким образом, в случае конечной $VC$-размерности эмпирический риск сходится к истинному риску равномерно по классу функций $\mathscr{F}$. Это доказывает состоятельность методов машинного обучения по конечным выборкам данных. Вместе с тем, оценка (1.8) не применима на практике из-за ее высокой завышенности, возникающей при использовании супремума по всему классу функций.

**Композиции алгоритмов.** Дальнейшее увеличение точности VC-оценок шло по пути учета структуры семейства алгоритмов и специфических свойств метода обучения. Отметим, что большинство применяемых на практике методов обучения (в частности, метод $k$ ближайших соседей, метод парзеновского окна, метод потенциальных функций [1], линейные классификаторы, и другие) фактически являются композицией более простых алгоритмов. В работах [11, 12, 13, 14] изучается широкий класс корректных линейных и алгебраических композиций алгоритмов вычисления оценок. Для таких композиций были получены нетривиальные оценки вероятности ошибки [21, 22, 23], а также разработана

теория универсальных и локальных ограничений [26, 27, 28], которая позволила унифицировать как методику построения алгебраических композиций, так и приёмы доказательства их ключевых свойств (таких как регулярность и полнота). Другим широким направлением в области алгоритмических композиций является комитетный метод формирования алгоритмов распознавания [19, 20], для которого также удалось получить оценки емкости класса решающих правил и найти достаточное условие равномерной сходимости частот к вероятностям по классу комитетных событий [37, 38, 25].

## 1.4 Радемахеровский процесс

Радемахеровский процесс [62, 60] позволяет получать оценки, вычислимые по наблюдаемой выборке и не зависящие от неизвестных вероятностных распределений. Кроме этого, радемахеровский процесс является крайне полезным математическим инструментом, удобным при доказательстве некоторых утверждений (например, теоремы 1.3).

**Определение 1.4.** *Пусть* $\varepsilon_1, \ldots, \varepsilon_n$ *— независимые случайные величины, такие что* $P(\varepsilon_i = +1) = P(\varepsilon_i = -1) = \frac{1}{2}$. *Тогда радемахеровским процессом называют следующий эмпирический процесс:*

$$R_n(f) = \frac{1}{n}\sum_{j=1}^{n} \varepsilon_j f(X_j), \quad f \in \mathscr{F}.$$

*Среднюю норму этого процесса* $\mathsf{E}_\varepsilon ||R_n||_{\mathscr{F}} = \mathsf{E}_\varepsilon \sup\limits_{f\in\mathscr{F}} |R_n(f)|$ *называют радемахеровской сложностью семейства* $\mathscr{F}$, *где* $\mathsf{E}_\varepsilon$ *обозначает усреднение по радемахеровским случайным величинам* $\varepsilon_1, \ldots, \varepsilon_n$.

Большая радемахеровская сложность семейства $\mathscr{F}$ означает, что для каждой реализации вектора шума $(\varepsilon_i)_{i=1}^n$ в семействе найдется функция, хорошо коррелирующая с этим вектором.

Радемахеровский процесс обладает рядом полезных математических свойств. В частности, он ограничивает величину $\mathsf{E}^n||P - P_n||_{\mathscr{F}}$, где $\mathsf{E}^n$ означает усреднение по всем реализациям выборки $(X_1, \ldots, X_n)$. Этот результат часто называют «симметризацией» благодаря интересному приему доказательства, основанному на искусственном введении дополнительной выборки $(X'_1, \ldots, X'_n)$.

**Лемма 1.4.** *(Симметризация)*

$$\frac{1}{2}\mathsf{E}^n\mathsf{E}_\varepsilon||R_n||_{\mathscr{F}_c} \leqslant \mathsf{E}^n||P - P_n||_{\mathscr{F}} \leqslant 2\mathsf{E}^n\mathsf{E}_\varepsilon||R_n||_{\mathscr{F}},$$

*где* $\mathscr{F}_c = \{f - Pf, f \in \mathscr{F}\}$.

Следующее неравенство является еще одним часто используемым приемом при работе с рademахеровскими процессами:

**Лемма 1.5.** *(Неравенство сжатия) Пусть функции класса $\mathscr{F}$ принимают значения из $[-1,1]$. Пусть функция $\varphi\colon [-1,1]\to\mathbb{R}$ — липшецева с константой $L$, и $\varphi(0)=0$. Тогда для класса функций $\varphi\circ\mathscr{F}=\{\varphi\circ f\colon f\in\mathscr{F}\}$ справедливо неравенство*

$$\mathsf{E}_\varepsilon||R_n||_{\varphi\circ\mathscr{F}}\leqslant 2L\mathsf{E}_\varepsilon||R_n||_{\mathscr{F}}.$$

*В частности, при $\varphi(t)=t^2$ выполнено следующее:*

$$\mathsf{E}_\varepsilon\sup_{f\in\mathscr{F}}\Big|\sum_{j=1}^n\varepsilon_j f^2(X_j)\Big|\leqslant 4\mathsf{E}_\varepsilon\sup_{f\in\mathscr{F}}\Big|\sum_{j=1}^n\varepsilon_j f(X_j)\Big| \tag{1.9}$$

Рассмотрим еще одно неравенство концентрации, обобщающее неравенство Хевдинга.

**Теорема 1.6 (МакДиармид).** *Пусть функция $F\colon S^n\to\mathbb{R}$ для всех $i=1,\dots,n$ и фиксированного $c>0$ удовлетворяет условию ограниченной вариации:*

$$\sup_{z_1,\dots,z_n,z_i'}|F(z_1,\dots,z_i,\dots,z_n)-F(z_1,\dots,z_i',\dots,z_n)|\leqslant c,$$

*и пусть $X_1,\dots,X_n$ — независимые одинаково распределенные случайные величины. Тогда для произвольного $\varepsilon>0$ выполнено*

$$\mathsf{P}^n\Big\{|F(X_1,\dots,X_n)-\mathsf{E}^nF(X_1,\dots,X_n)|>\varepsilon\Big\}\leqslant 2\exp\Big(-\frac{2\varepsilon^2}{nc^2}\Big). \tag{1.10}$$

Заметим, что оба выражения $||P-P_n||_{\mathscr{F}}$ и $||R_n||_{\mathscr{F}}$ являются функциями от $(X_1,\dots,X_n)$, удовлетворяющими условию ограниченной вариации с константой $c=\frac{1}{n}$. Это позволяет применить оценку (1.10) к $||P-P_n||_{\mathscr{F}}$, затем воспользоваться леммой симметризации 1.4, после чего вновь применить (1.10), но для $||R_n||_{\mathscr{F}}$. Итоговое неравенство дает верхнюю оценку на избыточный риск, в которой правая часть зависит лишь от наблюдаемой выборки и не содержит неизвестных вероятностных распределений.

Обратив оценку (1.10), получим, что для любого $\delta>0$ с вероятностью не менее $1-\delta$ выполнено

$$P\hat f-P_n\hat f\leqslant 2\mathsf{E}_\varepsilon\sup_{f\in\mathscr{F}}|R_nf|+\sqrt{\frac{2\log\frac{2}{\delta}}{n}}. \tag{1.11}$$

Данная оценка, как и неравенство Хевдинга (1.5), не учитывает дисперсию класса функций. Соответствующее обобщение неравенства Бернштейна, учитывающего дисперсии, оказалось намного более трудной задачей, решенной с помощью неравенства Талаграна.

## 1.5 Неравенство Талаграна

В статистической теории обучения используется следующая формулировка неравенства Талаграна:

**Теорема 1.7 (Талагран, [70]).** *Пусть $(X_1, \dots, X_n)$ — независимые случайные величины, пусть $\mathscr{F}$ — класс функций, равномерно ограниченных константой $U > 0$. Тогда для любого $\varepsilon > 0$*

$$\mathsf{P}^n \left\{ \Big| ||P_n f||_{\mathscr{F}} - \mathsf{E}^n ||P_n f||_{\mathscr{F}} \Big| > \varepsilon \right\} \leqslant K \exp\Big\{ - \frac{1}{K} \frac{\varepsilon}{nU} \log\Big(1 + \frac{\varepsilon U}{nV}\Big) \Big\},$$

*где $P_n f = \frac{1}{n} \sum\limits_{j=1}^{n} f(X_j)$, $K$ — некоторая константа, $V$ — любое число, удовлетворяющее условию*

$$V \geqslant \mathsf{E}^n \sup_{f \in \mathscr{F}} \sum_{j=1}^{n} f^2(X_j).$$

В этой теореме константу $V$ следует интерпретировать как дисперсию класса функций $\mathscr{F}$. Более конструктивное выражение для $V$ можно получить, применив лемму 1.4 о симметризации, а затем неравенство сжатия (1.9):

$$\mathsf{E}^n \sup_{f \in \mathscr{F}} \sum_{j=1}^{n} f^2(X_j) \leqslant n \sup_{f \in \mathscr{F}} Pf^2 + 8nU \mathsf{E}_\varepsilon ||R_n||_{\mathscr{F}} \equiv V.$$

Аналогичные неравенства, но с более строгими константами, были доказаны в [50] и [58].

$$\mathsf{P}^n \left\{ ||P_n - P||_{\mathscr{F}} \geqslant \mathsf{E}^n ||P_n - P||_{\mathscr{F}} + \sqrt{2 \frac{t}{n} \big( \sigma_P^2(\mathscr{F}) + 2\mathsf{E}^n ||P_n - P||_{\mathscr{F}} \big)} + \frac{t}{3n} \right\} \leqslant e^{-t};$$

$$\mathsf{P}^n \left\{ ||P_n - P||_{\mathscr{F}} \leqslant \mathsf{E}^n ||P_n - P||_{\mathscr{F}} - \sqrt{2 \frac{t}{n} \big( \sigma_P^2(\mathscr{F}) + 2\mathsf{E}^n ||P_n - P||_{\mathscr{F}} \big)} - \frac{t}{n} \right\} \leqslant e^{-t};$$

где

$$\sigma_P^2(\mathscr{F}) \equiv \sup_{f \in \mathscr{F}} \Big( Pf^2 - (Pf)^2 \Big).$$

Благодаря учету дисперсии функций класса $\mathscr{F}$, неравенство Талаграна позволяет существенно повысить точность оценок на избыточный риск.

## 1.6 Локальные оценки избыточного риска

Все оценки избыточного риска, приведенные в прошлых параграфах, основывались на следующем неравенстве:

$$P\hat{f}_n - P_n \hat{f}_n \leqslant \sup_{f \in \mathscr{F}} Pf - P_n f,$$

где супремум берется по всему классу функций $\mathscr{F}$. Это приводит к завышенности оценок, поскольку на практике методы обучения перебирают не все функции из $\mathscr{F}$, а лишь малую часть алгоритмов классификации, расположенных в окрестности лучшего решения.

В работах [42, 44, 43, 61] эта проблема решена следующим образом. Для произвольного $\delta > 0$ из семейства $\mathscr{F}$ выделяется *$\delta$-минимальное множество*, у функций которого истинный риск не превосходит $\delta$:

$$\mathscr{F}(\delta) = \mathscr{F}_P(\delta) = \{f \in \mathscr{F} \colon \mathcal{E}_P(f) \leqslant \delta\}.$$

Пусть функция $\bar{f} \in \operatorname{Arg}\min\limits_{f\in\mathscr{F}} Pf$ минимизирует истинный риск, а функция $\hat{f} \in \operatorname{Arg}\min\limits_{f\in\mathscr{F}} P_n f$ минимизирует эмпирический риск. Обозначим $\hat{\delta} = \mathcal{E}_P(\hat{f})$ и допустим, что $\bar{f} \in \mathscr{F}$. Тогда $\hat{f}, \bar{f} \in \mathscr{F}(\hat{\delta})$ и $P_n\hat{f} \leqslant P_n\bar{f}$. Следовательно,

$$\hat{\delta} = \mathcal{E}_P(\hat{f}) = P(\hat{f} - \bar{f}) = P_n(\hat{f} - \bar{f}) + (P - P_n)(\hat{f} - \bar{f}),$$

а значит,

$$\hat{\delta} \leqslant \sup_{f,g\in\mathscr{F}(\delta)} |(P_n - P)(f - g)|.$$

Допустим, что существует неслучайная верхняя оценка вида

$$U_n(\delta) \geqslant \sup_{f,g\in\mathscr{F}(\delta)} |(P_n - P)(f - g)|, \tag{1.12}$$

верная с высокой вероятностью равномерно по $\delta$. Тогда избыточный риск $\varepsilon_P(\hat{f})$ будет с той же вероятностью ограничен сверху наибольшим решением неравенства $\delta \leqslant U_n(\delta)$.

Оказывается, что оценку вида (1.12) можно построить с помощью неравенства Талаграна. Пусть $D(\mathscr{F})$ обозначает $L_2(P)$-диаметр $\delta$-минимального множества:

$$D(\delta) = \sup_{f,g\in\mathscr{F}(\delta)} \Big(P(f - g)^2 - (P(f - g))^2\Big).$$

Определим функцию $\varphi_n(\delta)$ следующим образом:

$$\varphi_n(\delta) = \mathsf{E}^n \sup_{f,g\in\mathscr{F}(\delta)} |(P_n - P)(f - g)|;$$

**Теорема 1.8 (Колчинский, [59]).** *Пусть $\{\delta_j\}_{j\geqslant 0}$ — убывающая последовательность положительных чисел, такая что $\delta_0 = 1$. Рассмотрим кусочно-постоянную функцию $U_n(\delta)$, определенную на промежутках $\delta \in (\delta_{j+1}, \delta_j]$ следующей формулой:*

$$U_n(\delta, t) = \varphi_n(\delta_j) + \sqrt{2\frac{t}{n}(D^2(\delta_j) + 2\varphi_n(\delta_j))} + \frac{t}{2n}, \text{ где } t > 0.$$

*Положим*

$$\delta_n(t) = \sup\{\delta \in (0,1] \colon \delta \leqslant U_n(\delta)\}.$$

*Тогда для всех* $\delta \geqslant \delta_n(t)$ *выполнено*

$$\mathsf{P}^n \left\{ \mathcal{E}(\hat{f}_n) > \delta \right\} \leqslant C(\delta) e^{-t}, \tag{1.13}$$

*где* $C(\delta)$ *— количество членов последовательности* $\{\delta_j\}_{j \geqslant 0}$*, превышающих* $\delta$*.*

Зафиксируем число $q > 1$ и положим $\delta_j = q^{-j}$. Тогда в условиях прошлой теоремы получим оценку

$$\mathsf{P}^n \left\{ \mathcal{E}(\hat{f}) \geqslant \delta \right\} \leqslant \left( \log_q \frac{q}{\delta} \right) e^{-t},$$

справедливую при $\delta \geqslant \delta_n(t)$.

На оценки, приведенные в данном параграфе, можно смотреть как на итерационный процесс. На первой итерации неравенство концентрации меры применяется ко всему семейству функций $\mathscr{F}$, и получается верхняя оценка на избыточный риск $\mathcal{E}(\hat{f}) \leqslant \delta_0$. На следующем шаге оценка применяется уже к $\delta_0$-минимальному множеству $\mathscr{F}(\delta_0)$, и получается новая оценка $\mathcal{E}(\hat{f}) \leqslant \delta_1$. При этом сужение рассматриваемого класса функций $\mathscr{F} \supset \mathscr{F}(\delta_0) \supset \mathscr{F}(\delta_1) \supset \ldots$ приводит к уменьшению дисперсии и, как следствие, к повышению точности оценки на каждой следующей итерации. Именно для этого важно использовать неравенства типа Бернштейна, а не Хевдинга. Отметим, что на каждой итерации неравенство верно лишь с определенной вероятностью. Таким образом, вероятность ошибки накапливается, но весь процесс удалось организовать так, что он сходится к итоговой оценке (1.13).

## 1.7 PAC-Bayes оценки

В данном параграфе будут рассмотрены оценки обобщающей способности, полученные для семейства линейных решающих правил. Пусть $\mathbb{X} = \mathbb{R}^d$ — множество объектов, описанных $d$ вещественными признаками, а $\mathbb{Y} = \{+1, -1\}$ — метки целевых классов. Каждый вектор весов $\mathbf{w} \in R^d$ определяет линейный классификатор $c_{\mathbf{w}}$, действующий на объекты $\mathbf{x} \in R^d$ по правилу $c_{\mathbf{w}}(\mathbf{x}) = \operatorname{sign}(\mathbf{w}^T \mathbf{x})$.

Как и в прошлых главах, для каждого классификатора $c_{\mathbf{w}}$ нас интересует его истинный

риск $P(c_{\mathbf{w}})$ и эмпирический риск $P_n(c_{\mathbf{w}})$ на выборке $(\mathbf{x}_i, y_i)_{i=1}^n$:

$$\begin{aligned} P(c_{\mathbf{w}}) &= \int\limits_{(x,y)\in S} I(\text{sign}(\mathbf{w}^T\mathbf{x}), y)dP, \\ P_n(c_{\mathbf{w}}) &= \frac{1}{n}\sum_{i=1}^{n} I(\text{sign}(\mathbf{w}^T\mathbf{x}_i), y_i), \end{aligned} \tag{1.14}$$

где $I(y_1, y_2) = [y_1 \neq y_2]$ — функция потерь, [истина] $= 1$, [ложь] $= 0$.

Основная идея PAC-Bayes подхода [63, 64, 67, 69] заключается в рассмотрении *стохастических классификаторов*. Пусть $C$ — множество классификаторов, а $Q$ задает распределение вероятности на этом множестве. Истинный и эмпирический риски стохастического классификатора определены следующим образом:

$$\begin{aligned} P(Q) &= \mathsf{E}_{\mathbf{w}\sim Q} P(c_{\mathbf{w}}) = \int\limits_{w\in C}\int\limits_{(x,y)\in S} I(\text{sign}(\mathbf{w}^T\mathbf{x}), y)dPdQ, \\ P_n(Q) &= \mathsf{E}_{\mathbf{w}\sim Q} P_n(c_{\mathbf{w}}) = \int\limits_{w\in C} \frac{1}{n}\sum_{i=1}^{n} I(\text{sign}(\mathbf{w}^T\mathbf{x}_i), y_i)dQ, \end{aligned} \tag{1.15}$$

Следующая теорема дает оценку обобщающей способности для стохастического классификатора.

**Теорема 1.9 (Langford, [63]).** *Пусть $Q_0$ — произвольное априорное распределение на множестве классификаторов. Тогда для любого $\delta \in (0,1)$ выполнено*

$$\mathsf{P}^n\left\{\forall Q, kl\big(P_n(Q), P(Q)\big) \leqslant \frac{KL(Q||Q_0) + \ln\frac{n+1}{\delta}}{n}\right\} \geqslant 1-\delta,$$

*где $KL(Q||Q_0)$ обозначает дивергенцию Кульбака - Лейблера между распределениями $Q$ и $Q_0$, $kl(q||p) = q\log\frac{q}{p} + (1-q)\log\frac{1-q}{1-p}$ определена при $q, p \in [0,1]$ и обозначает KL-дивергенцию между двумя случайными величинами Бернулли.*

С точки зрения задачи обучения по прецедентам, распределение $Q_0$ в теореме 1.9 следует воспринимать как априорное распределение, выбранное до реализации обучающей выборки. Распределение $Q$ можно выбрать уже после того, как стала известна обучающая выборка $(\mathbf{x}_i, y_i)_{i=1}^n$.

Теорему 1.9 можно применить к семейству линейных классификаторов. Пусть результатом обучения является классификатор $\hat{\mathbf{w}} \in \mathbb{R}^n$. В качестве априорного распределения $Q_0$ всегда выбирают многомерное нормальное распределение $\mathcal{N}(\mathbf{0}, I_d)$ с единичной дисперсией $I_d$ и нулевым вектором математического ожидания. Апостериорное распределение $Q$ полагают равным $\mathcal{N}(\mu\hat{\mathbf{w}}, I_d)$, т. е. вновь используют нормальное распределение с единичной

дисперсией, но вектор математического ожидания выбирают в направлении классификатора $\hat{\mathbf{w}}$; длиной вектора управляет константа $\mu$, которая может быть выбрана произвольно.

**Теорема 1.10 (Langford, [63]).** *Пусть $\mathbb{X} = \mathbb{R}^d$, $\mu > 0$ — константа, $\hat{\mathbf{w}} \in \mathbb{R}^d, ||\hat{\mathbf{w}}|| = 1$ — единичный вектор. Пусть $Q(\mu, \hat{\mathbf{w}})$ обозначает распределение на линейных классификаторах $c_{\mathbf{w}}$, где $\mathbf{w} \sim \mathcal{N}(\mu\hat{\mathbf{w}}, I_d)$. Тогда для любого $\delta \in (0, 1)$ с вероятностью не менее $1 - \delta$ относительно реализаций обучающей выборки $(\mathbf{x}_i, y_i)_{i=1}^n$*

$$kl(P_n(Q(\mu, \hat{\mathbf{w}})), P(Q(\mu, \hat{\mathbf{w}}))) \leqslant \frac{\frac{\mu^2}{2} + \ln\frac{n+1}{\delta}}{n}$$

*выполнено для всех $\mu > 0$ и всех $\hat{\mathbf{w}} \in \mathbb{R}^d$ при $||\hat{\mathbf{w}}|| = 1$. Кроме этого, эмпирический риск стохастического классификатора может быть записан в виде*

$$P_n(Q(\mu, \hat{\mathbf{w}})) = \frac{1}{n}\sum_{i=1}^{n}\overline{\Phi}(\mu\gamma(\hat{\mathbf{w}}, \mathbf{x}_i, y_i)),$$

*где $\gamma(\hat{\mathbf{w}}, \mathbf{x}, y)) = y\frac{\hat{\mathbf{w}}^T\mathbf{x}}{||\mathbf{x}_i||}$ — отступ прецедента $(\mathbf{x}, y)$ от разделяющей плоскости $\hat{\mathbf{w}}$;*

$$\overline{\Phi}(t) = 1 - \Phi(t) = \int_t^\infty \frac{1}{\sqrt{2\pi}}e^{\tau^2/2}d\tau$$

*дает вероятность правого хвоста нормального распределения.*

Можно показать, что для любого $\mu > 0$ истинный риск классификатора $c_{\hat{\mathbf{w}}}$ не превосходит удвоенного риска стохастического классификатора $Q(\mu, \hat{\mathbf{w}})$. Это позволяет применить теорему 1.10 к детерминированному линейному классификатору. Алгоритм 1.7.1 показывает явную схему вычислений. Отметим, что теорема 1.10 справедлива для произвольного значения параметра $\mu$, что позволяет минимизировать оценку по этому параметру.

**Алгоритм 1.7.1** PAC-Bayes оценка для линейного классификатора

**Вход:** Классификатор $\hat{\mathbf{w}} \in \mathbb{R}^d$, выборка $(\mathbf{x}_i, y_i)_{i=1}^n$, параметры $\delta$;

**Выход:** Верхняя оценка $E = P(\hat{\mathbf{w}})$ на истинный риск классификатора $\hat{\mathbf{w}}$.

1: Нормировать классификатор: $\hat{\mathbf{w}} := \hat{\mathbf{w}} \Big/ ||\hat{\mathbf{w}}||$;

2: Вычислить отступы: $\gamma_i := y_i \hat{\mathbf{w}}^T \mathbf{x}_i$, при $i = 1, \ldots, n$;

3: Сгенерировать логарифмическую сетку $M := \exp(-7 \colon 0.01 \colon 15)$;

4: **для всех** $\mu \in M$

5: Вычислить эмпирический риск: $E_n := \sum_{i=1}^{n} \overline{\Phi}(\mu\gamma_i)$;

6: Положить $T := \frac{\mu^2}{2n} + \frac{1}{n} \ln \frac{n+1}{\delta}$;

7: Положить $E_\mu := \max \left\{ p \in [0,1] \colon kl(E_n, p) \leqslant T \right\}$;

8: **Вернуть** $P(\hat{\mathbf{w}}) := 2 \cdot \operatorname{Arg} \min_{\mu \in M} E_\mu$.

## 1.8 Основные выводы

Основная задача теории статистического обучения (SLT) — вывод теоретических оценок обобщающей способности для методов машинного обучения. Подобные оценки используются при отборе признаков, выборе моделей при структурной минимизации риска, а также как альтернатива вычислительно трудоемкой процедуре скользящего контроля.

В основе SLT лежит модель, основанная на неизвестном и принципиально ненаблюдаемом вероятностном распределении на множестве всех объектов классификации. Модель SLT гарантирует, что оценки обобщающей способности будут выполнены для новых объектов, недоступных на этапе обучения, с заранее фиксированной вероятностью.

Оценки, полученные в SLT, остаются слишком завышенными для их применения в большинстве прикладных задач. Основная причина в том, что SLT дает оценки худшего случая, справедливые для любого вероятностного распределения на пространстве объектов, а также для практически любого метода обучения. Получить более точные оценки возможно лишь за счет учета специфических свойств конкретных задач и методов обучения.

# Глава 2. Комбинаторный подход

В данной главе вводятся основные определения и понятия комбинаторного подхода к оценке обобщающей способности.

## 2.1 Основные определения

Пусть задана генеральная выборка $\mathbb{X} = \{x_1, \ldots, x_L\}$, состоящая из $L$ объектов. Пусть $\mathscr{A}$ — некоторое семейство алгоритмов (например, семейство всех линейных разделяющих поверхностей в исходном признаковом пространстве задачи). Каждый алгоритм классификации $\alpha \in \mathscr{A}$, примененный к выборке $\mathbb{X}$, порождает бинарный вектор ошибок $a \equiv \big(I(\alpha, x_i)\big)_{i=1}^{L}$, где $I(\alpha, x_i) \in \{0, 1\}$ — индикатор ошибки алгоритма $\alpha$ на объекте $x_i$. Множество попарно-различных векторов ошибок, индуцированных семейством $\mathscr{A}$ и выборкой $\mathbb{X}$, называется *матрицей ошибок семейства* $\mathscr{A}$ и обозначается через $A$. В дальнейшем выборка $\mathbb{X}$ предполагается фиксированной, поэтому алгоритмы $\alpha \in \mathscr{A}$ будут отождествляться с векторами своих ошибок на выборке $\mathbb{X}$, а семейство $\mathscr{A}$ — с матрицей ошибок $A$.

**Замечание 2.1.** Важно помнить, что на объектах выборки $\mathbb{X} = \{x_1, \ldots, x_L\}$ выбран фиксированный порядок (нумерация от 1 до $L$). Это позволяет рассматривать бинарные *векторы* ошибок алгоритмов, в которых порядок следования нулей и единиц определяется выбранным порядком объектов выборки. Вместе с тем на алгоритмах $\alpha \in \mathscr{A}$ и на их векторах ошибок порядок не введен. Говоря «матрица ошибок», мы на самом деле имеем в виду неупорядоченное множество векторов ошибок. Таким образом, матрица ошибок $A$ рассматривается как подмножество $A \subset \mathbb{A} \equiv \{0, 1\}^L$ множества всех возможных векторов ошибок. Кроме этого, вместо термина «матрица ошибок алгоритмов» часто будет употребляться термин «множество алгоритмов».

**Замечание 2.2.** В ряде случаев мы будем искусственно генерировать матрицу ошибок алгоритмов $A$ в явном виде, не указывая, из какого семейства $\mathscr{A}$ она была получена. В таких случаях мы будем говорить о *модельном множестве алгоритмов*.

**Метод обучения.** *Методом обучения* называют отображение вида $\mu\colon 2^{\mathbb{X}} \to A$. Для произвольной обучающей выборки $X \subset \mathbb{X}$ метод обучения возвращает алгоритм $a = \mu X$ из фиксированного множества $A$. Мы будем использовать обозначения $\mu(X, A)$ в тех случаях, когда нужно явно подчеркнуть, из какого множества алгоритмов производится выбор.

Для выборки $X \subset \mathbb{X}$ обозначим через $n(a, X) = \sum\limits_{x \in X} I(a, x)$ *число ошибок*, а через $\nu(a, X) = n(a, X)/|X|$ — *частоту ошибок* алгоритма $a$ на выборке $X$. Подмножество $A_m = \{a \in A\colon n(a, \mathbb{X}) = m\}$ называют *$m$-слоем* алгоритмов.

Частоту ошибок на обучающей выборке называют *эмпирическим риском*. *Минимизация эмпирического риска* (МЭР) — это метод обучения, который из заданного множества $A \subset \mathbb{A}$ выбирает алгоритм $a \in A$, допускающий наименьшее число ошибок на обучающей выборке $X$. Таким образом, для всех $X \subset \mathbb{X}$ выполнено $\mu X \in A(X)$, где множество

$$A(X) = \operatorname{Arg}\min_{a \in A} n(a, X) \tag{2.1}$$

соответствует слою алгоритмов с минимальным числом ошибок на обучающей выборке $X$. Из-за дискретности функции $n(a, X)$ минимальный эмпирический риск может достигаться сразу для нескольких алгоритмов. Чтобы разрешить эту неоднозначность, вводится *пессимистический* МЭР. Он разрешает неоднозначность, выбирая в $A(X)$ алгоритм с наибольшим числом ошибок на полной выборке:

$$\mu X \in \operatorname{Arg}\max_{a \in A(X)} n(a, \mathbb{X}). \tag{2.2}$$

Пессимистический МЭР не может быть реализован на практике, т.к. он подглядывает в скрытую часть выборки. Тем не менее, пессимистический МЭР является удобной теоретической конструкцией, поскольку он позволяет получать верхние оценки на вероятность переобучения любого МЭР.

**Перестановочная вероятность.** Обозначим через $[\mathbb{X}]^\ell$ множество всех разбиений генеральной выборки $\mathbb{X}$ на обучающую выборку $X$ длины $\ell$ и контрольную выборку $\bar{X}$ длины $k = L - \ell$.

Для предиката $\varphi\colon [\mathbb{X}]^\ell \to \{0, 1\}$ и вещественной функции $\psi\colon [\mathbb{X}]^\ell \to \mathbb{R}$ определим оператор вероятности $\mathsf{P}$ и матожидания $\mathsf{E}$:

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

Если *уклонение частот* $\delta(a,X) = \nu(a,\bar{X}) - \nu(a,X)$ превосходит фиксированный порог $\varepsilon > 0$, то говорят, что алгоритм $a = \mu X$ является *переобученным*. Нашей целью является оценка *вероятности переобучения* [7, 73]:

$$Q_\varepsilon(\mu,\mathbb{X}) = \mathsf{P}\big[\delta(\mu,X) \geqslant \varepsilon\big]. \tag{2.3}$$

где для краткости введено обозначение $\delta(\mu,X) = \delta(\mu X,X)$, а квадратные скобки действуют по правилам [истина] $= 1$, [ложь] $= 1$.

*Обращением оценки* $Q_\varepsilon \leqslant \eta(\varepsilon)$ называют неравенство вида

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

выполненное с вероятностью не менее $1-\eta$, где $\varepsilon(\eta)$ является обратной функцией для $\eta(\varepsilon)$. *Медианой оценки* $Q_\varepsilon \leqslant \eta(\varepsilon)$ называют инверсию в точке $\eta = 1/2$.

Средние частоты ошибок на обучении и на контроле определим следующим образом:

$$\nu_\ell(\mu,\mathbb{X}) = \mathsf{E}\nu(\mu X,X); \tag{2.4}$$

$$\bar{\nu}_\ell(\mu,\mathbb{X}) = \mathsf{E}\nu(\mu X,\bar{X}). \tag{2.5}$$

**Пример 2.1.** В следующей таблице показан пример модельного множества алгоритмов, для которого минимизация эмпирического риска приводит к переобучению. Столбцы таблицы соответствуют алгоритмам, строки — объектам обучающей выборки $\{x_1,\ldots,x_\ell\}$ и контрольной выборки $\{x_{\ell+1},\ldots,x_L\}$. Единица в $[i,d]$-й ячейке таблицы означает, что алгоритм $a_d$ допускает ошибку на объекте $x_i$.

| | $a_1$ | $a_2$ | $\cdots$ | $a_d$ | $\cdots$ | $a_D$ |
|---|---|---|---|---|---|---|
| $x_1$ | 0 | 1 | $\cdots$ | 0 | $\cdots$ | 1 |
| $\cdots$ | 1 | 1 | $\cdots$ | 0 | $\cdots$ | 0 |
| $x_\ell$ | 0 | 0 | $\cdots$ | 0 | $\cdots$ | 0 |
| $x_{\ell+1}$ | 1 | 1 | $\cdots$ | 1 | $\cdots$ | 1 |
| $\cdots$ | 1 | 0 | $\cdots$ | 1 | $\cdots$ | 0 |
| $x_L$ | 0 | 0 | $\cdots$ | 1 | $\cdots$ | 0 |

В данном примере переобучение могло быть следствием «неудачного» разбиения генеральной выборки на обучение и контроль. Поэтому функционал вероятности переобучения (2.3) содержит усреднение по всем разбиениям $\mathbb{X} = X \sqcup \bar{X}$ на обучение и контроль фиксированной длины.

Определение (2.3) впервые упоминается в работе [56] для частного случая $k = 1$. Кроме этого, оно встречается в работах [47, 45], на этот раз для произвольного $k$, но с существенными различиями в обозначениях. Это определение не случайно напоминает процедуру полной кросс-валидации, поскольку при решении практических задач кросс-валидация зарекомендовала себя как наиболее точный инструмент для оценки обобщающей способности.

**Гипергеометрическое распределение.** Рассмотрим алгоритм $a$, допускающий $m = n(a, \mathbb{X})$ ошибок на полной выборке. Вероятность того, что из $m$ ошибок в обучающей выборке $X$ окажется ровно $s$ ошибок, равна *гипергеометрической функции*:

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

где параметр $s$ пробегает от $s_0 = \max\{0, m - k\}$ до $s_1 = \min\{m, \ell\}$, а параметр $m$ принимает значения $0, \ldots, L$. Мы полагаем $C_m^s = h_L^{\ell,m}(s) = 0$ для всех прочих значений $m, s$.

Определим *функцию гипергеометрического распределения* следующим образом:

$$H_L^{\ell,m}(z) = \sum_{s=s_0}^{\lfloor z \rfloor} h_L^{\ell,m}(s).$$

Рассмотрим множество $A = \{a\}$, состоящее из одного алгоритма. Тогда $\mu X = a$ для любой выборки $X \in [\mathbb{X}]^\ell$. Это значит, что вероятность переобучения $Q_\varepsilon$ преобразовалась в вероятность больших уклонений между частотами ошибок в выборках $X, \bar{X}$. Допустим, что число ошибок $n(a, \mathbb{X})$ нам известно, и получим точное выражение для $Q_\varepsilon$.

**Теорема 2.1 (FC-оценка [76]).** *Для фиксированного алгоритма $a$, такого что $m = n(a, \mathbb{X})$, любой генеральной выборки $\mathbb{X}$ и любого $\varepsilon \in [0, 1]$ вероятность переобучения определяется левым хвостом гипергеометрического распределения:*

$$Q_\varepsilon(a, \mathbb{X}) = H_L^{\ell,m}\left(\tfrac{\ell}{L}(m - \varepsilon k)\right). \tag{2.6}$$

Гипергеометрическое распределение играет важную роль во многих комбинаторных оценках. Оценка (2.6), примененная совместно с неравенством Буля, позволяет получить верхнюю оценку на $Q_\varepsilon(\mu, \mathbb{X})$, справедливую для любого метода обучения $\mu$:

**Теорема 2.2 (VC-оценка [76]).** *Для любой генеральной выборки $\mathbb{X}$, метода обучения $\mu$ и любого $\varepsilon \in [0, 1]$ вероятность переобучения ограничена суммой FC-оценок по множеству алгоритмов $A$:*

$$Q_\varepsilon(\mu, \mathbb{X}) \leqslant \mathsf{P}\left[\max_{a \in A} \delta(a, X) \geqslant \varepsilon\right] \leqslant \sum_{a \in A} H_L^{\ell,m}\left(\tfrac{\ell}{L}(m - \varepsilon k)\right), \quad m = n(a, \mathbb{X}). \tag{2.7}$$

Назовем две причины завышенности оценки (2.7). Во-первых, большинство алгоритмов из $A$ имеют высокую частоту ошибок, а следовательно, имеют исчезающе малую вероятность реализоваться в результате обучения. Тем не менее, оценка равномерного уклонения игнорирует это свойство метода обучения $\mu$. Во-вторых, неравенство Буля игнорирует тот факт, что алгоритмы с близкими векторами ошибок переобучаются в основном на одних и тех же разбиениях. Более точные оценки должны учитывать свойства метода обучения и сходство между алгоритмами.

## 2.2 Расслоение и связность

**Граф расслоения-связности.** На множестве алгоритмов $A$ естественным образом вводится *отношение частичного 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 $a \neq b$. *Расстоянием* между парой алгоритмов называют хэммингово расстояние между их векторами ошибок: $\rho(a,b) = \sum_{i=1}^{L} |a_i - b_i|$. Если $a < b$ и $\rho(a,b) = 1$, то говорят, что $a$ предшествует $b$, и записывают $a \prec b$. Очевидно, что при этом $n(a,\mathbb{X}) + 1 = n(b,\mathbb{X})$.

**Определение 2.1 (Воронцов, [74]).** *Графом расслоения–связности*[1] *множества алгоритмов $A$ называют направленный граф $\langle A, E\rangle$ с множеством ребер $E = \big\{(a,b)\colon a \prec b\big\}$.*

**Пример 2.2.** На рис. 2.1 показан граф расслоения–связности, порождаемый семейством линейных алгоритмов классификации на выборке длины $L = 10$. Выборка линейно разделима, поэтому в графе имеется нулевой слой. Этот слой состоит 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.

Граф расслоения–связности является многодольным, доли соответствуют слоям $A_m = \{a \in A\colon n(a,\mathbb{X}) = m\}$, ребрами могут соединяться только алгоритмы соседних слоев. 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 $I(b,x_{ab}) = 1$.

**Оценка расслоения-связности.**

**Определение 2.2.** *Верхней связностью $u(a)$ алгоритма $a \in A$ называют число ребер графа расслоения-связности, исходящих из вершины $a$:*

$$u(a) = \#\big\{x_{ab} \in \mathbb{X}\colon a \prec b\big\}. \tag{2.8}$$

[1] В [56] аналогичный граф называют термином 1-inclusion graph.

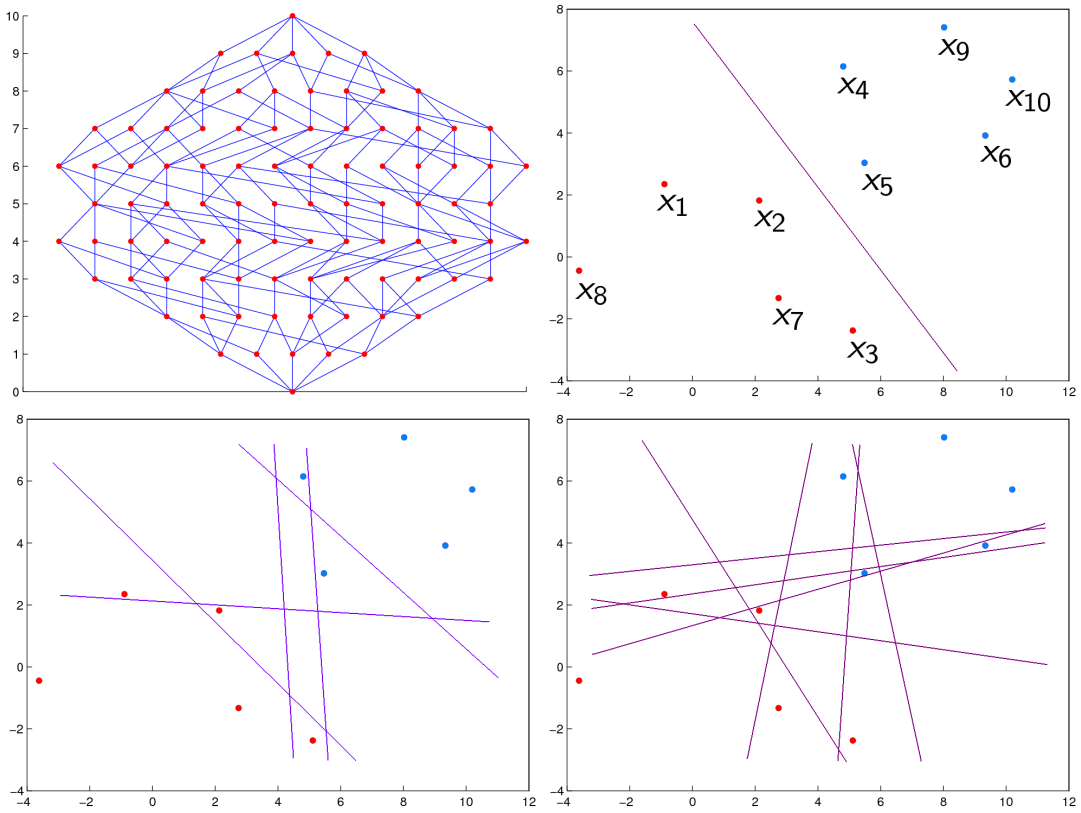



Рис. 2.1: Пример графа расслоения–связности (вверху слева; по вертикальной оси отложены номера слоев), порождаемого семейством линейных алгоритмов классификации на выборке из 10 объектов, по 5 объектов в каждом классе (вверху справа). Первый слой образуется 5 алгоритмами с одной ошибкой (внизу слева), второй слой — 8 алгоритмами с двумя ошибками (внизу справа), и т. д.

**Определение 2.3.** *Неполноценностью $q(a)$ алгоритма $a \in A$ называют 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 $x$:*

$$q(a) = \#\{x \in \mathbb{X} \mid \exists b \in A\colon I(b,x) < I(a,x),\ b \leqslant a\} \tag{2.9}$$

В терминах графа расслоения-связности $q(a)$ равно числу различных объектов $x_{bc}$, соответствующих всевозможным ребрам $(b,c)$ на путях, ведущих к вершине $a$.

**Пример 2.3.** На рис. 2.2 показан пример вычисления верхней связности и неполноценности алгоритма по графу расслоения-связности.

**Теорема 2.3 (SC-оценка [76]).** *Пусть метод обучения $\mu$ является пессимистическим МЭР. Тогда для любого $\varepsilon \in [0,1]$ вероятность переобучения ограничена взвешенной суммой*

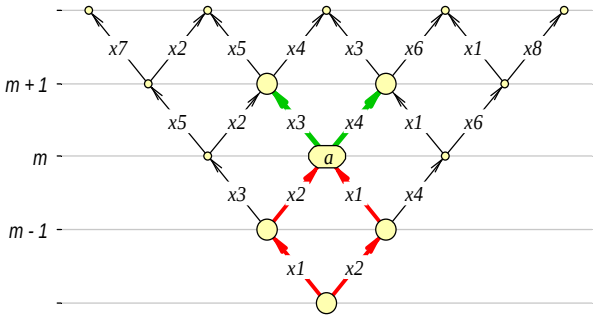



Рис. 2.2: Пример двумерной сети алгоритмов. Для алгоритма $a$ верхняя связность $u(a) = \#\{x3, x4\} = 2$, неполноценность $q(a) = \#\{x1, x2\} = 2$.

*FC-оценок по множеству A:*

$$Q_\varepsilon(\mu, \mathbb{X}) \leqslant \sum_{a \in A} \frac{C_{L-u-q}^{\ell-u}}{C_L^\ell} H_{L-u-q}^{\ell-u,m-q}\left(\tfrac{\ell}{L}(m - \varepsilon k)\right), \tag{2.10}$$

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

SC-оценка (2.10) превращается в VC-оценку (2.7), если положить все $q(a)$ и $r(a)$ равными нулю. Вес $P_a = C_{L-u-q}^{\ell-u}/C_L^\ell$ в сумме (2.10) соответствует верхней оценке на вероятность $\mathsf{P}[\mu X = a]$ получить алгоритм $a$ в результате обучения. Эта величина уменьшается экспоненциально быстро с ростом связности $u(a)$ и неполноценности $q(a)$. Таким образом, эффективное вычисление $Q_\varepsilon(\mu, \mathbb{X})$ требует знаний не про все множество $A$, а лишь про несколько нижних слоев $A$.

Оценка расслоениясвязности (2.10) уточняет VC-оценку (2.7), но в ряде случаев остается завышенной.

**Эксперимент 2.1.** Рассмотрим модельное множество $A = (a_1, a_2)$, состоящее из двух алгоритмов. Векторы ошибок подобраны так, что оба алгоритма допускают по $m$ ошибок на полной выборке, а хэммингово расстояние $\rho(a_1, a_2)$ равняется заранее фиксированному числу $d$. На рис. 2.3 приведена зависимость медианы распределения $Q_\varepsilon(A)$ от хэммингова расстояния между алгоритмами. Видно, что переобучение увеличивается с ростом $\rho(a_1, a_2)$. Вместе с тем и неполноценность, и верхняя связность обоих алгоритмов данного множества равны нулю. Таким образом, SC-оценка не учитывает зависимость вероятности переобучения от $\rho(a_1, a_2)$.

Эффект сходства алгоритмов еще сильнее проявляется на примере следующего модельного множества алгоритмов.

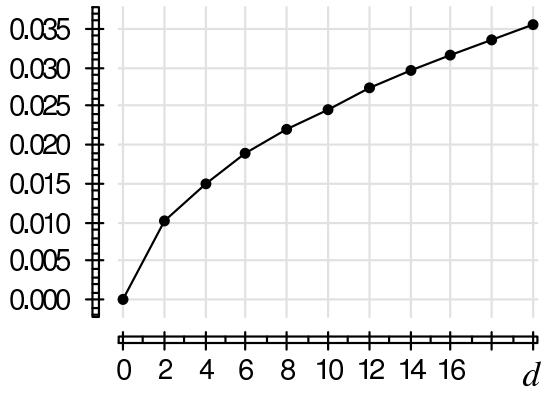



Рис. 2.3: Зависимость медианы распределения $Q_\varepsilon$ от хэммингова расстояния $d = \rho(a_1, a_2)$ между векторами ошибок пары алгоритмов. $L = 100$, $\ell = 50$.

**Определение 2.4.** *Пусть $a_0$ — произвольный алгоритм с $m$ ошибками, $r \leqslant m$ — натуральное число. Центральным слоем хэммингова шара называется модельное множество алгоритмов*

$$B_r^m(a_0) = \{a \in \mathbb{A} \colon \rho(a, a_0) \leqslant r \text{ и } n(a, \mathbb{X}) = m\}.$$

*Данное множество состоит из алгоритмов хэммингова шара радиуса $r$ с центром в $a_0$, допускающих на полной выборке столько же ошибок, сколько и центр шара. Вероятность переобучения $Q_\varepsilon(B_r^m(a_0))$ зависит только от радиуса шара $r$ и числа ошибок $n(a_0, \mathbb{X})$, поэтому в дальнейшем вместо $B_r^m(a_0)$ будет использоваться сокращенная запись $B_r^m$.*

**Эксперимент 2.2.** Исследуем вероятность переобучения $Q_\varepsilon(B_r^m)$ численно с помощью метода Монте-Карло, усреднив переобученность $\delta(\mu X, X)$ не по всем $X \in [\mathbb{X}]^\ell$, а по 10 тыс. случайным подвыборкам. Для сравнения мы рассмотрим еще одно модельное множество $R_n^m$, составленное из $n$ алгоритмов, допускающих по $m$ ошибок на полной выборке. Векторы ошибок всех алгоритмов из $R_n^m$ сгенерированы случайно и независимо. В следующей таблице показано, при каком числе алгоритмов $n$ в $R_n^m$ медианы распределений $Q_\varepsilon(R_n^m)$ и $Q_\varepsilon(B_r^m)$ совпадают.

Из таблицы 2.1 видно, что всего семь алгоритмов со случайными векторами ошибок могут переобучиться так же сильно, как и множество из десятков тысяч алгоритмов с близкими векторами ошибок. Таким образом, вероятность переобучения множества $B_r^m$ существенно зависит от сходства алгоритмов.

Основной задачей данной диссертационной работы является получение оценок, одновременно учитывающих и расслоение алгоритмов по числу ошибок, и сходство алгоритмов внутри одного слоя.

Таблица 2.1: Сравнение $|R_n^m|$ и $|B_r^m|$ при $L = 50$, $\ell = 25$, $m = 10$

| $r$ | $\lvert B_r^m \rvert$ | $\lvert R_n^m \rvert$ | $EOF(\mu, \mathbb{X})$ | $\varepsilon\colon Q_\varepsilon(B_r^m) = 0.5$ |
|---|---|---|---|---|
| 2 | 401 | 2 | 0.079 | 0.320 |
| 4 | 35 501 | 7 | 0.160 | 0.400 |
| 6 | 1 221 101 | 39 | 0.240 | 0.400 |
| 8 | 20 413 001 | 378 | 0.319 | 0.400 |

## 2.3 Основные выводы и постановка задачи

В комбинаторной теории оценок обобщающей способности изучается функционал вероятности переобучения. Этот функционал основан на принципе полного скользящего контроля и зависит как от задачи, так и от метода обучения. Как исходный функционал вероятности переобучения, так и все его оценки зависят только от наблюдаемых величин, таких как конечная генеральная выборка объектов или матрица ошибок алгоритмов. В большинстве комбинаторных оценок в качестве метода обучения используется минимизация эмпирического риска. Данный метод обучения позволяет исследовать такие явления, как сходство и расслоение алгоритмов, также изучаемые в классической модели SLT.

Основной комбинаторной оценкой является оценка расслоения-связности. Она является менее завышенной по сравнению с классическими оценками и успешно применяется для повышения качества логических алгоритмов классификации. Вместе с тем данная оценка не учитывает сходство алгоритмов внутри слоя алгоритмов с равным числом ошибок. Анализ и устранение причин завышенности оценки расслоения-связности являются основными целями данной диссертационной работы.

# Глава 3. Теоретико-групповой подход

Основная задача данной главы — получение новых комбинаторных методов вывода оценок вероятности переобучения.

## 3.1 Рандомизированный метод обучения и РМЭР

При минимизации эмпирического риска может возникать неоднозначность — несколько алгоритмов из $A(X) \equiv \operatorname{Arg}\min\limits_{a \in A} n(a, X)$ могут иметь одинаковое число ошибок на обучающей выборке. В [8] для устранения неоднозначности и получения точных верхних оценок вероятности переобучения использовалась *пессимистичная* минимизация эмпирического риска (2.2) — предполагалось, что в случае неоднозначности выбирается алгоритм с наибольшим числом ошибок на генеральной выборке $\mathbb{X}$. Это не устраняет неоднозначность окончательно. Возможны ситуации, когда несколько алгоритмов имеют наименьшее число ошибок на обучающей выборке $X$ и одинаковое число ошибок на генеральной выборке $\mathbb{X}$. В таких случаях на множестве алгоритмов вводился линейный порядок, и среди неразличимых алгоритмов выбирался алгоритм с бóльшим порядковым номером. Введение приоритетности алгоритмов является искусственным приемом, не имеющим адекватных аналогов среди известных методов обучения.

*Рандомизированный метод обучения* произвольному множеству алгоритмов $A \subseteq \mathbb{A}$ и произвольной обучающей выборке $X \in [\mathbb{X}]^\ell$ ставит в соответствие функцию распределения весов на множестве алгоритмов:

$$\mu : 2^{\mathbb{A}} \times [\mathbb{X}]^\ell \to \{f : \mathbb{A} \to [0, 1]\}. \tag{3.1}$$

Естественно полагать, что эта функция нормирована и может быть интерпретирована как вероятность получить каждый алгоритм в результате обучения.

Детерминированный метод обучения является частным случаем рандомизированного, когда функция распределения весов $f(a)$ принимает единичное значение ровно на одном алгоритме и нулевое на всех остальных.

Заметим, что вместо определения (3.1) можно задать то же самое отображение

эквивалентным способом:

$$\mu : 2^{\mathbb{A}} \times [\mathbb{X}]^{\ell} \times \mathbb{A} \to [0, 1].$$

*Рандомизированный метод минимизации эмпирического риска (РМЭР)* выбирает произвольный алгоритм из множества $A(X)$ случайно и равновероятно:

$$\mu(A, X, a) = \frac{[a \in A(X)]}{|A(X)|}. \tag{3.2}$$

Поскольку в задаче статистического обучения появляется второй независимый источник случайности, определение вероятности переобучения $Q_\varepsilon(A)$ приходится модифицировать. Наиболее естественный вариант модификации — усреднение по множеству $A(X)$:

$$Q_\varepsilon(A) = \mathsf{E}\frac{1}{|A(X)|}\sum_{a \in A(X)} \big[\delta(a, X) \geqslant \varepsilon\big]. \tag{3.3}$$

Переставим местами знаки суммирования, $\mathsf{E}\sum = \sum \mathsf{E}$. Получим сумму по всем алгоритмам $a \in A$, каждое слагаемое которой обозначим через $Q_\varepsilon(a, A)$ и назовем *вкладом алгоритма* $a$ в вероятность переобучения:

$$Q_\varepsilon(A) = \sum_{a \in A} Q_\varepsilon(a, A), \quad Q_\varepsilon(a, A) = \mathsf{E}\frac{\big[a \in A(X)\big]}{|A(X)|}\big[\delta(a, X) \geqslant \varepsilon\big].$$

Аналогичным образом введем *вероятность реализации алгоритма* $a$:

$$P(a, A) = \mathsf{E}\frac{\big[a \in A(X)\big]}{|A(X)|}. \tag{3.4}$$

В частном случае, когда для всех $X \in [\mathbb{X}]^{\ell}$ множество $A(X)$ состоит из единственного алгоритма, вероятность переобучения (3.3) и вероятность реализации (3.4) принимают привычный вид:

$$Q_\varepsilon(A) = \mathsf{E}\big[\delta(\mu X, X) \geqslant \varepsilon\big], \quad P(a, A) = \mathsf{E}\big[a = \mu X\big].$$

Рандомизированный МЭР $\mu_r$ занимает промежуточное положение между *оптимистичным* и *пессимистичным* методами:

$$\mu_o X = \arg\min_{a \in A(X)} n(a, \bar{X}) \text{ — оптимистичный ММЭР;}$$

$$\mu_p X = \arg\max_{a \in A(X)} n(a, \bar{X}) \text{ — пессимистичный ММЭР.}$$

Приводим без доказательства следующее утверждение: для произвольного множества алгоритмов $A \subseteq \mathbb{A}$ и каждого $\varepsilon \in (0, 1]$ справедлива цепочка неравенств:

$$Q_\varepsilon(\mu_o, A) \leqslant Q_\varepsilon(\mu_o, A) \leqslant Q_\varepsilon(\mu_p, A). \tag{3.5}$$

## 3.2 Перестановки объектов

Введем симметрическую группу $S_L$ всех $L!$ перестановок, действующую на выборке $\mathbb{X} = \{x_1, \ldots, x_L\}$. Для произвольной перестановки $\pi \in S_L$ обозначим через $\pi x$ образ объекта $x \in \mathbb{X}$ под действием перестановки $\pi$. Действие перестановок на объектах естественным образом переносится на подмножества объектов, на алгоритмы как бинарные векторы ошибок длины $L$ и на множества алгоритмов. Эти действия определены следующим образом:

- действие перестановки $\pi \in S_L$ на подмножество объектов:
$$\pi X = \big\{\pi x \colon x \in X\big\};$$
- действие перестановки $\pi \in S_L$ на алгоритм:
$$\pi \mathbf{a} = \big(I(\pi a, x_i)\big)_{i=1}^{L} = \big(I(a, \pi^{-1} x_i)\big)_{i=1}^{L};$$
- действие перестановки $\pi \in S_L$ на множество алгоритмов:
$$\pi A = \big\{\pi a \colon a \in A\big\}.$$

Заметим, что действие одной и той же перестановки $\pi$ сначала на выборку $\mathbb{X}$, а затем на алгоритм $a$ восстанавливает исходный вектор ошибок алгоритма $a$. Благодаря такому определению действие на алгоритм обладает рядом полезных свойств.

**Лемма 3.1.** *Свойства действия произвольной перестановки $\pi \in S_L$:*

*1) $I(\pi a, \pi x) = I(a, x)$ для любых $a \in A$ и $x \in \mathbb{X}$;*

*2) $n(\pi a, \mathbb{X}) = n(a, \mathbb{X})$ для любого $a \in A$;*

*3) $n(\pi a, \pi X) = n(a, X)$ для любых $a \in A$ и $X \subseteq \mathbb{X}$;*

*4) $\delta(\pi a, \pi X) = \delta(a, X)$ для любых $a \in A$ и $X \subseteq \mathbb{X}$;*

*5) $[a \in A(X)] = [\pi a \in (\pi A)(\pi X)]$ для любых $a \in A$ и $X \subseteq \mathbb{X}$;*

*6) $|A(X)| = |(\pi A)(\pi X)|$ для любых $A$ и $X \subseteq \mathbb{X}$;*

*7) $\rho(a, a') = \rho(\pi a, \pi a')$ для любых $a, a' \in A$, где $\rho(a, a')$ — расстояние Хэмминга векторами между ошибок алгоритмов $a$ и $a'$:*
$$\rho(a, a') = \sum_{x \in \mathbb{X}} |I(a, x) - I(a', x)|.$$

**Доказательство.** Свойство 1) следует из определения:

$$I(\pi a, \pi x) = I(a, \pi^{-1}\pi x) = I(a, x).$$

Свойство 2) следует из свойства 1):

$$n(\pi a, \mathbb{X}) = \sum_{i=1}^{L} I(\pi a, x_i) = \sum_{i=1}^{L} I(\pi a, \pi x_i) = \sum_{i=1}^{L} I(a, x_i) = n(a, \mathbb{X}).$$

Свойство 3) также следует из свойства 1):

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

Свойство 4) следует из свойства 3):

$$\begin{aligned}\delta(a, X) &= \frac{L - n(a, X)}{k} - \frac{n(a, X)}{\ell} = \\ &= \frac{L - n(\pi a, \pi X)}{k} - \frac{n(\pi a, \pi X)}{\ell} = \delta(\pi a, \pi X).\end{aligned}$$

Свойство 5) следует из определения (2.1) и свойства 1):

$$\begin{aligned}a_0 \in A(X) \;\Leftrightarrow\; & a_0 \in \arg\min_{a \in A} n(a, X) \;\Leftrightarrow \\ & \forall a \in A \to n(a_0, X) \leqslant n(a, X) \;\Leftrightarrow \\ & \forall a \in A \to n\big(\pi a_0, \pi X\big) \leqslant n\big(\pi a, \pi X\big) \;\Leftrightarrow \\ & \forall a' \in \pi A \to n\big(\pi a_0, \pi X\big) \leqslant n\big(a', \pi X\big) \;\Leftrightarrow \\ & \pi a_0 \in \arg\min_{a' \in \pi A} n(a', \pi X) \;\Leftrightarrow\; \pi a_0 \in (\pi A)(\pi X).\end{aligned}$$

Свойство 6) следует из свойства 5):

$$\begin{aligned}|A(X)| = \sum_{a \in A} [a \in A(X)] &= \sum_{a \in A} [\pi a \in (\pi A)(\pi X)] = \\ &= \sum_{a' \in \pi A} [a' \in (\pi A)(\pi X)] = |(\pi A)(\pi X)|.\end{aligned}$$

Свойство 7) следует из свойства 1):

$$\begin{aligned}\rho(\pi a, \pi a') &= \sum_{x \in \mathbb{X}} |I(\pi a, x) - I(\pi a', x)| = \\ &= \sum_{x' \in \mathbb{X}} |I(\pi a, \pi x') - I(\pi a', \pi x')| = \\ &= \sum_{x' \in \mathbb{X}} |I(a, x') - I(a', x')| = \rho(a, a').\end{aligned}$$

■

В дальнейшем нам будет нужно доказывать утверждения, аналогичные лемме 3.1, но для более сложных функций. Чтобы упростить эту задачу, введем следующую классификацию функций.

- Симметричной функцией *первого рода* будем называть $g\colon \mathbb{A} \times [\mathbb{X}]^\ell \to \mathbb{R}$, такую что для всех $\pi \in S_L$ выполнено $g(a, X) = g(\pi a, \pi X)$;
- Симметричной функцией *второго рода* будем называть $G\colon 2^{\mathbb{A}} \times [\mathbb{X}]^\ell \to 2^{\mathbb{A}}$, такую что для всех $\pi \in S_L$ выполнено $\pi G(A, X) = G(\pi A, \pi X)$;
- Симметричной функцией *третьего рода* будем называть $f\colon 2^{\mathbb{A}} \times [\mathbb{X}]^\ell \to \mathbb{R}$, такую что для всех $\pi \in S_L$ выполнено $f(A, X) = f(\pi A, \pi X)$.

Лемма 3.1 утверждает, что функции $n(a, X)$ и $\nu(a, X)$ являются симметричными функциями первого рода, а $A(X)$ как функция $A$ и $X$ является симметричной функцией второго рода.

Две следующие теоремы позволяют строить новые симметричные функции из уже имеющихся:

**Теорема 3.2.** *Пусть $g_1, g_2, \ldots, g_p$ — симметричные функции первого рода, $f_1, f_2, \ldots, f_p$ — симметричные функции третьего рода, $F\colon \mathbb{R}^p \to \mathbb{R}$ — произвольная функция многих переменных. Тогда $F(g_1, g_2, \ldots, g_p)$ — вновь симметричная функция первого рода, $F(f_1, f_2, \ldots, f_p)$ — симметричная функция третьего рода.*

**Доказательство.** Проведя элементарные выкладки, получим

$$F(\pi a, \pi X) \equiv F(g_1(\pi a, \pi X), \ldots, g_p(\pi a, \pi X)) = F(g_1(a, X), \ldots, g_p(a, X)) \equiv F(a, X),$$

и аналогично для функций третьего рода. ■

**Теорема 3.3.** *Пусть $g$ — симметричная функция первого рода, $G$ — симметричная функция второго рода. Тогда*

$$f(A, X) \equiv |G(A, X)| \text{ и } f(A, X) \equiv \sum_{a \in G(A,X)} g(a, X)$$

*являются симметричными функциями третьего рода.*

**Доказательство.** Заметим, что для любого $A \subset \mathbb{A}$ выполнено $|A| = |\pi A|$, поскольку $\pi$, как элемент группы, является биекцией. Следовательно,

$$|G(A, X)| = |\pi G(A, X)| = |G(\pi A, \pi X)|$$

.

Для функции $f(A,X) \equiv \sum\limits_{a \in G(A,X)} g(a,X)$ запишем цепочку равенств:

$$f(\pi A, \pi X) = \sum_{a \in G(\pi A, \pi X)} g(a, \pi X) = \sum_{a \in \pi G(A,X)} g(a, \pi X) =$$
$$= \sum_{a \in G(A,X)} g(\pi a, \pi X) = \sum_{a \in G(A,X)} g(a,X) = f(A,X).$$

■

Из приведенных выше теорем следует, что вклад каждого разбиения в вероятность переобучения РМЭР является симметричной функцией третьего рода.

$$Q_\varepsilon(A,X) \equiv \frac{1}{|A(X)|} \sum_{a \in A(X)} [\delta(a,X) \geqslant \varepsilon] = Q_\varepsilon(\pi A, \pi X).$$

Это утверждение, как и большинство теорем следующего параграфа, оказывается верно не только для РМЭР, но и для более широкого класса *корректных* методов обучения.

**Определение 3.1.** *Рандомизированный метод обучения* $\mu : 2^{\mathbb{A}} \times [\mathbb{X}]^\ell \times \mathbb{A} \to [0,1]$ *называется корректным, если при любых* $A \in 2^{\mathbb{A}}$, $X \in [\mathbb{X}]^\ell$, $a, b \in A$ *и* $\pi \in S_L$ *выполнены следующие условия:*

1) *нормировка:*
$$\sum_{a \in A} \mu(A,X,a) = 1; \tag{3.6}$$

2) *неразличимость алгоритмов с одинаковой частотой ошибок на обучении:*
$$n(a,X) = n(b,X) \;\to\; \mu(A,X,a) = \mu(A,X,b); \tag{3.7}$$

3) *инвариантность результата обучения относительно замены множества алгоритмов* $A$ *на* $\pi A$*:*
$$\mu(A,X,a) = \mu\big(\pi A, \pi X, \pi a\big). \tag{3.8}$$

Первое условие означает «вероятностную» нормировку весов алгоритмов и обеспечивает нулевую «вероятность» алгоритмам, не принадлежащих множеству $A$. Второе условие означает, что при любом разбиении $\mathbb{X} = X \sqcup \bar{X}$, $X \in [\mathbb{X}]^\ell$ вероятность получить алгоритм в результате обучения зависит только от количества ошибок алгоритма на обучении. Третье условие означает, что результат обучения не изменится, если подействовать перестановкой $\pi$ одновременно и на множество объектов $[\mathbb{X}]^\ell$, и на множество алгоритмов $\mathbb{A}$.

**Теорема 3.4.** *Рандомизированный МЭР* $\mu(A, X, a) = \frac{[a \in A(X)]}{|A(X)|}$ *является корректным рандомизированным методом обучения.*

**Доказательство.** Первое условие проверяется явно:

$$\sum_{a \in A} \mu(A, X, a) = \sum_{a \in A(X)} \frac{1}{|A(X)|} = 1.$$

Для доказательства второго утверждения достаточно заметить, что два алгоритма $a_1$ и $a_2$ с равным числом ошибок на обучении могут лежать в множестве $A(X)$ только одновременно. Следовательно, вероятность получить каждый из алгоритмов в результате обучения равна либо нулю, либо $\frac{1}{|A(X)|}$.

Для проверки третьего условия достаточно доказать, что

$$a_0 \in \operatorname{Arg}\min_{a \in A} n(a, X) \quad \Leftrightarrow \quad \pi a_0 \in \operatorname{Arg}\min_{a \in \pi A} n(a, \pi X).$$

Используя лемму 3.1, проведем цепочку равносильных утверждений:

$$\begin{aligned} a_0 \in \operatorname{Arg}\min_{a \in A} n(a, X) \quad &\Leftrightarrow \\ &\Leftrightarrow \quad \forall a \in A \to n(a_0, X) \leqslant n(a, X) \quad \Leftrightarrow \\ &\Leftrightarrow \quad \forall a \in A \to n\big(\pi a_0, \pi X\big) \leqslant n\big(\pi a, \pi X\big) \quad \Leftrightarrow \\ &\Leftrightarrow \quad \forall a' \in \pi A \to n\big(\pi a_0, \pi X\big) \leqslant n\big(a', \pi X\big) \quad \Leftrightarrow \\ &\Leftrightarrow \quad \pi a_0 \in \operatorname{Arg}\min_{a \in \pi A} n(a, \pi X). \end{aligned}$$

Теорема доказана. ■

## 3.3 Группа симметрии множества алгоритмов

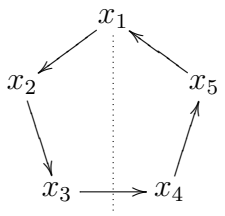
**Определение 3.2.** *Группой симметрии множества алгоритмов* $A$ *называют множество всех перестановок, действие которых на* $A$ *не меняет его:*

$$\operatorname{Sym}(A) = \{\pi \in S_L \colon \pi A = A\}.$$

Если подействовать любой из перестановок $\pi \in \operatorname{Sym}(A)$ на строки матрицы ошибок множества $A$, то получится то же самое множество столбцов; переставив столбцы, можно получить исходную матрицу ошибок. Очевидно, что множество $\operatorname{Sym}(A)$ является группой.

**Пример 3.1.** Рассмотрим множество алгоритмов, заданное матрицей ошибок

$$\begin{array}{c} \\ x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \end{array} \begin{array}{c} \begin{array}{ccccc} a_1 & a_2 & a_3 & a_4 & a_5 \end{array} \\ \begin{pmatrix} 1 & 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 & 1 \\ 1 & 0 & 0 & 1 & 1 \\ 1 & 1 & 0 & 0 & 1 \end{pmatrix} \end{array}$$




Группа симметрий данного множества алгоритмов совпадает с группой симметрий правильного пятиугольника и называется *диэдральной группой*. Образующими элементами группы являются циклическая перестановка $\pi_1 = (x_1, x_2, x_3, x_4, x_5)$ и осевая симметрия $\pi_2 = (x_2, x_5)(x_3, x_4)$.

Пусть далее $G \subseteq \mathrm{Sym}(A)$ — произвольная подгруппа группы $\mathrm{Sym}(A)$.

Для любой перестановки $\pi \in G$ и любого алгоритма $a \in A$ алгоритм $\pi a$ снова лежит в $A$. В таких случаях говорят, что группа $G$ *действует* на множестве $A$.

*Орбитой* алгоритма $a \in A$ называется множество алгоритмов $Ga = \{\pi a \colon \pi \in G\}$. Орбита также целиком лежит в $A$. Орбиты двух различных алгоритмов $Ga$ и $Ga'$ либо совпадают, либо не пересекаются. Следовательно, множество $A$ разбивается на непересекающиеся подмножества — орбиты:

$$A = \bigsqcup_{\omega \in \Omega(A)} \omega = \bigsqcup_{\omega \in \Omega(A)} Ga_\omega,$$

где $\Omega(A)$ — множество всех орбит в $A$, $a_\omega$ — произвольный представитель орбиты $\omega$.

Из свойства 2) леммы 3.1 следует, что алгоритмы одной орбиты обязательно лежат в одном слое. Обратное, вообще говоря, неверно.

**Лемма 3.5.** *Алгоритмы из одной орбиты имеют равные вероятности реализации и равные вклады в вероятность переобучения: для любой перестановки $\pi$ из $G$*

$$P(a, A) = P(\pi a, A), \quad Q_\varepsilon(a, A) = Q_\varepsilon(\pi a, A).$$

**Доказательство.** Воспользуемся определением вероятности реализации (3.4), свойствами 5), 6) из леммы 3.1, и свойством $A = \pi A$:

$$P(a, A) = \mathsf{E}\frac{[a \in A(X)]}{|A(X)|} = \mathsf{E}\frac{[\pi a \in (\pi A)(\pi X)]}{|(\pi A)(\pi X)|} = \mathsf{E}\frac{[\pi a \in A(\pi X)]}{|A(\pi X)|}.$$

Под знаком $\mathsf{E}$ можно всюду заменить $\pi X$ на $X$, так как результат не зависит от порядка суммирования разбиений:

$$P(a, A) = \mathsf{E}\frac{\big[\pi a \in A(X)\big]}{|A(X)|} = P(\pi a, A).$$

Воспользуемся определением вероятности реализации (3.4), свойствами 4), 5), 6) из леммы 3.1, и свойством $A = \pi A$:

$$\begin{aligned} Q_\varepsilon(a, A) &= \mathsf{E}\frac{\big[a \in A(X)\big]}{|A(X)|}\big[\delta(a, X) \geqslant \varepsilon\big] = \\ &= \mathsf{E}\frac{\big[\pi a \in (\pi A)(\pi X)\big]}{|(\pi A)(\pi X)|}\big[\delta(\pi a, \pi X) \geqslant \varepsilon\big] = \\ &= \mathsf{E}\frac{\big[\pi a \in A(\pi X)\big]}{|A(\pi X)|}\big[\delta(\pi a, \pi X) \geqslant \varepsilon\big]. \end{aligned}$$

Вновь заменяя $\pi X$ на $X$ под знаком $\mathsf{E}$, получим:

$$Q_\varepsilon(a, A) = \mathsf{E}\frac{\big[\pi a \in A(X)\big]}{|A(X)|}\big[\delta(\pi a, X) \geqslant \varepsilon\big] = Q_\varepsilon(\pi a, A).$$

■

**Разложение вероятности переобучения по орбитам множества алгоритмов.** Из теоремы о равном вкладе алгоритмов одной орбиты немедленно следует формула разложения вероятности переобучения по орбитам. Она является основным инструментом получения точных оценок для РМЭР.

**Теорема 3.6.** *Для любой генеральной выборки $\mathbb{X}$, любого множества алгоритмов $A$ с попарно различными векторами ошибок и любого $\varepsilon \in [0, 1]$ справедлива формула разложения вероятности переобучения по орбитам множества $A$:*

$$Q_\varepsilon(A) = \sum_{\omega \in \Omega(A)} |\omega|\, \mathsf{E}\frac{\big[a_\omega \in A(X)\big]}{|A(X)|}\big[\delta(a_\omega, X) \geqslant \varepsilon\big], \tag{3.9}$$

*где $\Omega(A)$ — множество всех орбит в $A$, $a_\omega$ — произвольный представитель орбиты $\omega$.*

**Доказательство.** Перегруппируем слагаемые в (3.3) по орбитам множества $A$, затем применим лемму 3.5 о равном вкладе алгоритмов одной орбиты:

$$\begin{aligned} Q_\varepsilon(A) &= \sum_{\omega \in \Omega(A)} \sum_{a \in \omega} \mathsf{E}\frac{\big[a \in A(X)\big]}{|A(X)|}\big[\delta(a, X) \geqslant \varepsilon\big] = \\ &= \sum_{\omega \in \Omega(A)} |\omega|\, \mathsf{E}\frac{\big[a_\omega \in A(X)\big]}{|A(X)|}\big[\delta(a_\omega, X) \geqslant \varepsilon\big]. \end{aligned}$$

■

**Разложение вероятности переобучения по орбитам множества разбиений.** В [30] отмечено, что по аналогии с действием группы $\mathrm{Sym}(A)$ на множестве $A$ можно рассматривать действие этой же группы на множестве $[\mathbb{X}]^\ell$ всех $\ell$-элементных подмножеств генеральной выборки $\mathbb{X}$.

*Орбитой* выборки $X \in [\mathbb{X}]^\ell$ называется множество $GX = \{\pi X \colon \pi \in G\}$. Множество всех выборок длины $\ell$ разбивается на непересекающиеся орбиты:

$$[\mathbb{X}]^\ell = \bigsqcup_{\tau \in \Omega[\mathbb{X}]^\ell} \tau = \bigsqcup_{\tau \in \Omega[\mathbb{X}]^\ell} GX_\tau,$$

где $\Omega[\mathbb{X}]^\ell$ — множество всех орбит в $[\mathbb{X}]^\ell$, $X_\tau$ — представитель орбиты $\tau$.

**Теорема 3.7 (Толстихин, [30]).** *Для любой генеральной выборки $\mathbb{X}$, любого множества алгоритмов $A$ с попарно различными векторами ошибок и любого $\varepsilon \in [0,1]$ справедлива формула разложения вероятности переобучения одновременно и по орбитам множества $A$, и по орбитам множества $[\mathbb{X}]^\ell$:*

$$Q_\varepsilon(A) = \sum_{\omega \in \Omega(A)} \frac{|\omega|}{C_L^\ell} \sum_{\tau \in \Omega[\mathbb{X}]^\ell} |\{X \in \tau \colon a_\omega \in A(X)\}| \frac{\big[\delta(a_\omega, X_\tau) \geqslant \varepsilon\big]}{|A(X_\tau)|}. \tag{3.10}$$

*где $\Omega[\mathbb{X}]^\ell$ — множество всех орбит в $[\mathbb{X}]^\ell$, $\Omega(A)$ — множество всех орбит в $A$, $a_\omega$ — представитель орбиты $\omega$, $X_\tau$ — представитель орбиты $\tau$.*

Это разложение одновременно использует и орбиты на множестве алгоритмов, и орбиты на множестве разбиений выборки.

В настоящей работе выводится еще одно разложение вероятности переобучения, в котором используются лишь орбиты на множестве разбиений выборки. Оно оказывается проще, чем (3.10), и вместе с тем оказывается полезным при выводе формул вероятности переобучения для целого ряда модельных семейств.

Представим вероятность переобучения в виде суммы вкладов разбиений:

$$Q_\varepsilon(A) = \mathsf{E}\, Q(A, X), \quad Q_\varepsilon(A, X) = \frac{1}{|A(X)|} \sum_{a \in A(X)} \big[\delta(a, X) \geqslant \varepsilon\big],$$

где $Q_\varepsilon(A, X)$ — вклад разбиения $X \sqcup \bar{X}$ в вероятность переобучения. Поскольку разбиениям $X \sqcup \bar{X}$ взаимно однозначно соответствуют выборки $X \in [\mathbb{X}]^\ell$, далее будем говорить также о *вкладе выборки* $X$ в вероятность переобучения.

**Лемма 3.8.** *Пусть для некоторой функции $f \colon 2^{\mathbb{A}} \times [\mathbb{X}]^\ell \to \mathbb{R}$ для всех $A \subset \mathbb{A}$, $X \in [\mathbb{X}]^\ell$ и всех $\pi \in \mathrm{Sym}(A)$ выполнено условие $f(A, X) = f(A, \pi X)$. Тогда справедливо следующее*

*разложение:*

$$\sum_{X\in[\mathbb{X}]^\ell} f(A,X) = \sum_{\tau\in\Omega([\mathbb{X}]^\ell)} |\tau| f(A,X_\tau).$$

**Доказательство** с очевидностью следует из группировки равных слагаемых. ■

**Лемма 3.9.** *Выборки из одной орбиты имеют равные вклады в вероятность переобучения:* $Q_\varepsilon(A,X) = Q_\varepsilon(\pi A, X)$ *для любой перестановки* $\pi$ *из* $G$.

**Доказательство.** Заметим, что выражение $\frac{[\delta(a,X)\geqslant\varepsilon]}{|A(X)|}$ является симметричной функцией первого рода. Тогда по теореме 3.3 выражение $Q_\varepsilon(X,A)$ является симметричной функцией третьего рода. Следовательно,

$$Q_\varepsilon(A,X) = Q_\varepsilon(\pi A, \pi X) = Q_\varepsilon(A, \pi X).$$

Для завершения доказательства осталось воспользоваться леммой 3.8. ■

**Теорема 3.10.** *Для любой генеральной выборки* $\mathbb{X}$*, любого множества алгоритмов* $A$ *с попарно различными векторами ошибок и любого* $\varepsilon \in [0,1]$ *справедлива формула разложения вероятности переобучения по орбитам множества* $[\mathbb{X}]^\ell$*:*

$$Q_\varepsilon(A) = \frac{1}{C_L^\ell} \sum_{\tau\in\Omega[\mathbb{X}]^\ell} \frac{|\tau|}{|A(X_\tau)|} \sum_{a\in A(X_\tau)} \big[\delta(a,X_\tau) \geqslant \varepsilon\big], \tag{3.11}$$

*где* $\Omega[\mathbb{X}]^\ell$ *— множество всех орбит в* $[\mathbb{X}]^\ell$*,* $X_\tau$ *— представитель орбиты* $\tau$.

Доказательство аналогично доказательству теоремы 3.6.

В качестве примера применения полученных формул рассмотрим множество $\mathbb{A} = \{0,1\}^L$, состоящее из всех возможных бинарных векторов ошибок.

**Теорема 3.11.** *Вероятность переобучения РМЭР, примененного к множеству всех алгоритмов* $\mathbb{A} = \{0,1\}^L$*, дается формулой:*

$$Q_\varepsilon(\mathbb{A}) = \frac{1}{2^k} \sum_{m=\lceil\varepsilon k\rceil}^{k} C_k^m.$$

**Доказательство.** Для всех перестановок $\pi \in S_L$ выполнено $\pi\mathbb{A} = \mathbb{A}$. Следовательно, $\mathrm{Sym}(A) = S_L$. Заметим, что для каждой пары обучающих выборок $X$, $X'$ возможно указать перестановку $\pi \in S_L$, такую что $X' = \pi X$. Такую ситуацию называют «транзитивным действием группы $S_L$ на множестве $[\mathbb{X}]^\ell$». Для нас это означает, что имеется лишь

одна орбита $\tau = [\mathbb{X}]^\ell$. Выбрав произвольную выборку $X$ в качестве ее представителя и воспользовавшись теоремой 3.10, получим

$$Q_\varepsilon(\mathbb{A}) = \frac{1}{|\mathbb{A}(X)|} \sum_{a \in \mathbb{A}(X)} [\delta(a, X) \geqslant \varepsilon].$$

Множество $\mathbb{A}(X)$ состоит из всех алгоритмов, не допускающих ошибок на $X$. Следовательно, $|\mathbb{A}(X)| = 2^k$. Для завершения доказательства осталось заметить, что переобученными в $\mathbb{A}(X)$ будут те и только те алгоритмы, у которых не менее $\lceil \varepsilon k \rceil$ ошибок на контрольной выборке. ■

Завершая параграф, интересно рассмотреть частный случай, когда все алгоритмы из $A$ имеют равное число ошибок на полной выборке.

**Следствие 3.1.** *Пусть все $a \in A$ имеют равное число ошибок на полной выборке: $n(a, \mathbb{X}) = m$. Тогда вероятность переобучения РМЭР записывается в виде*

$$Q_\varepsilon(A) = \frac{1}{C_L^\ell} \sum_{\tau \in \Omega([\mathbb{X}]^\ell)} |\tau| \left[ \min_{a \in A} n(a, X_\tau) \leqslant \frac{\ell}{L}(m - \varepsilon k) \right]. \tag{3.12}$$

**Доказательство.** Отметим, что в рассматриваемом случае для любой обучающей выборки $X$ все алгоритмы из множества $A(X)$ либо переобучены, либо нет. Действительно, согласно определению $A(X)$ они имеют равное число ошибок на обучении. Число ошибок на полной выборке одинаково по определению множества $A$. Следовательно, все алгоритмы из $A$ имеют равное число ошибок на контрольной выборке, а также равные уклонения частот. Применим формулу (3.11) и получим следующее выражение для вероятности переобучения:

$$Q_\varepsilon(A) = \frac{1}{C_L^\ell} \sum_{\tau \in \Omega([\mathbb{X}]^\ell)} |\tau| \left[ \delta(a, X_\tau) > \varepsilon \right].$$

Для получения формулы (3.12) осталось выразить уклонение частот $\delta(a, X_\tau)$ через число ошибок лучшего алгоритма на обучении и количество ошибок на полной выборке $m$. ■

## 3.4 Покрытия множества алгоритмов

Допустим, что исходное множество алгоритмов $A$ представлено в виде разбиения на непересекающиеся подмножества $A = A_1 \sqcup A_2 \sqcup \cdots \sqcup A_t$ так, что в каждое $A_i$ попали лишь алгоритмы с близкими векторами ошибок. В данной ситуации будем называть множества $A_i$ *кластерами* алгоритмов. Покажем, что задачу оценивания вероятности переобучения всего множества $A$ можно свести к оцениванию вероятности переобучения отдельных кластеров.

**Лемма 3.12.** *Пусть множество алгоритмов $A$ произвольным образом представлено в виде разбиения на непересекающиеся подмножества $A = A_1 \sqcup A_2 \sqcup \cdots \sqcup A_t$. Тогда вероятность переобучения пессимистического метода минимизации эмпирического риска оценивается сверху следующим выражением:*

$$Q_\varepsilon(A) \leqslant \sum_{i=1}^{t} Q_\varepsilon(A_t). \tag{3.13}$$

**Доказательство.** Заметим, что достаточно доказать утверждение для $t = 2$ (общее утверждение получается по индукции). Обозначим через $\mu(A, X)$ алгоритм, выбранный пессимистическим методом минимизации эмпирического риска из множества $A$ по обучающей выборке $X$. Рассмотрим произвольное разбиение $X \in [\mathbb{X}]^\ell$ и покажем следующее:

$$[\delta(\mu(A, X), X) \geqslant \varepsilon] \leqslant [\delta(\mu(A_1, X), X) \geqslant \varepsilon] + [\delta(\mu(A_2, X), X) \geqslant \varepsilon]. \tag{3.14}$$

Для разбиения $X$ и множеств $A_1, A_2, A$ множества $A_1(X), A_2(X), A(X)$ определены согласно (2.1). Обозначим через $n_1(X), n_2(X)$ и $n(X)$ число ошибок на обучающей выборке для алгоритмов из $A_1(X)$, $A_2(X)$ и $A(X)$, соответственно. Очевидно, что $n_1(X) \geqslant n(X)$ и $n_2(X) \geqslant n(X)$, но по крайней мере одно из этих неравенств обязательно обращается в равенство. Рассмотрим два случая: в первом случае одно неравенство строгое, во втором оба неравенства обращаются в равенство.

**Случай 1.** Пусть для определенности $n_1(X) > n(X)$. Тогда $A_2(X) = A(X)$, а следовательно, $\mu(A_2, X) = \mu(A, X)$. Отсюда немедленно следует (3.14).

**Случай 2.** Из $n_1(X) = n_2(X) = n(X)$ следует, что $A(X) = A_1(X) \cup A_2(X)$. Таким образом, либо $\mu(A, X) \in A_1(X)$, либо $\mu(A, X) \in A_2(X)$ (в зависимости от того, в какое из этих двух множеств попал алгоритм с наибольшим числом ошибок на полной выборке). Значит, вновь выполнено (3.14). ∎

Для оценки $Q_\varepsilon(A_i)$ предположим, что в каждом $A_i$ алгоритмы допускают равное число ошибок на полной выборке. Тогда, согласно следующей лемме, можно расширить $A_i$ до произвольного множества $B$ с известной оценкой $Q_\varepsilon(B)$.

**Лемма 3.13 (Толстихин, [30]).** *Рассмотрим вложенные множества алгоритмов: $A_i \subseteq B \subseteq \mathbb{A}$. Допустим, что все алгоритмы $b \in B$ допускают по $m = n(b, \mathbb{X})$ ошибок на полной выборке. Тогда для минимизации эмпирического риска для всех $\varepsilon > 0$ выполнено неравенство $Q_\varepsilon(A_i) \leqslant Q_\varepsilon(B)$.*

**Доказательство.** Докажем утверждение для частного случая $B = A_i \cup \{b\}$. Рассмотрим произвольное разбиение $X \in [\mathbb{X}]^\ell$. Нас интересуют только разбиения с $\mu(B, X) = b$, потому что вклад остальных разбиений в вероятность переобучения не изменился. Пусть $a = \mu(A_i, X)$ — алгоритм, выбранный на разбиении $X$ методом обучения из множества $A_i$. Поскольку $\mu$ является минимизацией эмпирического риска, получим $n(b, X) \leqslant n(a, X)$. Поскольку по условию алгоритмы $a$ и $b$ имеют равное число ошибок на полной выборке, уклонение частоты $\delta(b, X) \geqslant \delta(a, X)$. Следовательно, вклад каждого разбиения от добавления алгоритма $b$ мог только увеличиться. ■

## 3.5 Теоремы о порождающих и запрещающих множествах (ПЗМ)

Первый подход, позволивший получать точные оценки вероятности переобучения в рамках слабой вероятностной аксиоматики, основан на выделении порождающих и запрещающих объектов [75].

**Гипотеза 3.1.** Пусть множество $A$, выборка $\mathbb{X}$ и детерминированный метод обучения $\mu$ 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

$$\big[\mu X{=}a\big] = \big[X_a \subseteq X\big]\big[X'_a \subseteq \bar{X}\big] \quad \text{для всех } X \in [\mathbb{X}]^\ell. \tag{3.15}$$

Множество $X_a$ называется *порождающим*, множество $X'_a$ — *запрещающим* для алгоритма $a$. Гипотеза 3.1 означает, что метод $\mu$ выбирает алгоритм $a$ тогда и только тогда, когда в обучающей выборке $X$ находятся все порождающие объекты и ни одного запрещающего. Все остальные объекты $\mathbb{X}\backslash X_a\backslash X'_a$ называются *нейтральными* для алгоритма $a$.

Для произвольного алгоритма $a \in A$ введем следующие обозначения:

$L_a = L - |X_a| - |X'_a|$ — число нейтральных объектов в генеральной выборке;

$\ell_a = \ell - |X_a|$ — число нейтральных объектов в обучающей выборке;

$m_a = n(a, \mathbb{X}\backslash X_a\backslash X'_a)$ — число ошибок алгоритма $a$ на нейтральных объектах;

$s_a(\varepsilon) = \frac{\ell}{L}\big(n(a, \mathbb{X}) - \varepsilon k\big) - n(a, X_a)$ — наибольшее число ошибок алгоритма $a$ на нейтральных обучающих объектах $X \setminus X_a$, при котором имеет место большое уклонение частот ошибок, $\delta(a, X) \geqslant \varepsilon$.

Введем функцию гипергеометрического распределения:

$$H_L^{\ell,m}(z) = \sum_{s=0}^{\lfloor z \rfloor} \frac{C_m^s C_{L-m}^{\ell-s}}{C_L^\ell}.$$

**Теорема 3.14.** *Если справедлива гипотеза 3.1, то вероятность получить в результате обучения алгоритм $a$ равна $P_a(A) = \mathsf{P}[\mu X{=}a] = C_{L_a}^{\ell_a}/C_L^\ell$, вероятность переобучения равна*

$$Q_\varepsilon(A) = \sum_{a \in A} P_a H_{L_a}^{\ell_a, m_a}\big(s_a(\varepsilon)\big).$$

Данный результат позволил получить формулы вероятности переобучения для широкого класса модельных семейств алгоритмов, в частности для монотонных и унимодальных сетей.

Теорема 3.14 получена для детерминированных методов обучения, для которых результатом обучения является один алгоритм $a \in A$. В следующих параграфах мы приводим два важных обобщения этого результата: во-первых, на случай произвольного разложения множества алгоритмов на подмножества; во-вторых, на случай рандомизированного метода обучения.

### 3.5.1 ПЗМ для разложения множества алгоритмов на подмножества

В данном параграфе мы обобщим метод порождающих и запрещающих множеств [75] так, чтобы он был применим к произвольному разложению множества алгоритмов на подмножества.

**Гипотеза 3.2.** Пусть множество алгоритмов $A$ представлено в виде разбиения на непересекающиеся подмножества $A = A_1 \sqcup A_2 \sqcup \cdots \sqcup A_t$. Пусть выборка $\mathbb{X}$ и метод обучения $\mu$ таковы, что для каждого $i = 1, \ldots, t$ можно указать пару непересекающихся подмножеств $X_i \subset \mathbb{X}$ и $X_i' \subset \mathbb{X}$, удовлетворяющую условию

$$[\mu(A, X) \in A_i] \leqslant [X_i \subset X][X_i' \subset \bar{X}] \;\text{ для всех } X \in [\mathbb{X}]^\ell.$$

Пусть, кроме этого, все алгоритмы $a \in A_i$ не допускают ошибок на $X_i$ и ошибаются на всех объектах из $X_i'$.

Множество $X_i$ будем называть *порождающим*, множество $X_i'$ — *запрещающим* для $A_i$. Гипотеза 3.2 означает, что результат обучения может принадлежать $A_i$ только в том

случае, если в обучающей выборке $X$ находятся все порождающие объекты и ни одного запрещающего. Все остальные объекты $\mathbb{Y}_i \equiv \mathbb{X}\backslash X_i\backslash X'_i$ будем называть *нейтральными* для $A_i$.

Пусть $L_i = L - |X_i| - |X'_i|$, $\ell_i = \ell - |X_i|$, $k_i = k - |X'_i|$. Пусть $Q'_\varepsilon(A_i)$ есть вероятность переобучения на множестве нейтральных объектов $\mathbb{Y}_i$:

$$Q'_\varepsilon(A_i) = \frac{1}{C^{\ell_i}_{L_i}} \sum_{Y \in [\mathbb{Y}_i]^{\ell_i}} [\delta(\mu(A_i, Y), Y) \geqslant \varepsilon],$$

где $[\mathbb{Y}_i]^{\ell_i}$ — множество разбиений $\mathbb{Y}_i$ на обучающую выборку $Y$ длины $\ell_i$ и контрольную выборку $\bar{Y}$ длины $k_i = L_i - \ell_i$.

**Теорема 3.15.** *Пусть выполнена гипотеза 3.2, а на разбиение $A = A_1 \sqcup A_2 \sqcup \cdots \sqcup A_t$ наложено дополнительное ограничение: внутри каждого кластера $A_i$ все алгоритмы допускают равное число ошибок (обозначаемое через $m_i$). Тогда вероятность переобучения $Q_\varepsilon(A)$ ограничена сверху следующей оценкой:*

$$Q_\varepsilon(A) \leqslant \sum_{i=1}^{t} P_i\, Q'_{\varepsilon_i}(A_i), \tag{3.16}$$

*где $P_i = \frac{C^{\ell_i}_{L_i}}{C^{\ell}_{L}}$, $\varepsilon_i = \frac{L_i}{\ell_i k_i}\frac{\ell k}{L}\varepsilon + \left(1 - \frac{\ell L_i}{L \ell_i}\right)\frac{m_i}{k_i} - \frac{|X'_i|}{k_i}$, $Q'_\varepsilon(A_i)$ — определенная выше вероятность переобучения на множестве нейтральных объектов.*

**Доказательство** во многом повторяет доказательство аналогичной теоремы из [75]. Распишем определение вероятности переобучения:

$$\begin{aligned} Q_\varepsilon(A) &= \frac{1}{C^{\ell}_{L}} \sum_{X \in [\mathbb{X}]^{\ell}} [\delta(\mu(A, X), X) \geqslant \varepsilon] = \\ &= \frac{1}{C^{\ell}_{L}} \sum_{i=1}^{t} \sum_{X \in [\mathbb{X}]^{\ell}} [\mu(A, X) \in A_i][\delta(\mu(A_i, X), X) \geqslant \varepsilon] \leqslant \\ &\leqslant \frac{1}{C^{\ell}_{L}} \sum_{i=1}^{t} \sum_{X \in [\mathbb{X}]^{\ell}} [X_i \subset X][X'_i \subset \bar{X}][\delta(\mu(A_i, X), X) \geqslant \varepsilon]. \end{aligned}$$

Пусть $Y = X\backslash X_i$. Тогда $\sum\limits_{X \in [\mathbb{X}]^{\ell}}$ при условии $[X_i \subset X][X'_i \subset \bar{X}]$ можно заменить на суммирование по $Y \in [\mathbb{Y}_i]^{\ell_i}$.

$$Q_\varepsilon(A) \leqslant \frac{C^{\ell_i}_{L_i}}{C^{\ell}_{L}} \sum_{i=1}^{t} \frac{1}{C^{\ell_i}_{L_i}} \sum_{Y \in [\mathbb{Y}_i]^{\ell_i}} [\delta(\mu(A_i, X), X) \geqslant \varepsilon], \text{ где } X = Y \sqcup X_i. \tag{3.17}$$

Выразим условие $\delta(\mu(A_i, X), X) \geqslant \varepsilon$ в терминах $Y$. Обозначим $a = \mu(A_i, X)$, и пусть $n(a, Y) = s$. Тогда, используя условие $n(a, X_i) = 0$ и $n(a, X'_i) = |X'_i|$ из гипотезы 3.2,

получим $n(a,X) = s$, $n(a,\bar{X}) = m_i - s$, $n(a,\bar{Y}) = m_i - |X'_i| - s$. Следовательно, условия переобучения для $X$ и $Y$ запишутся следующим образом:

$$[\delta(\mu(A_i,X),X) \geqslant \varepsilon] = \Big[s \leqslant \frac{\ell}{L}(m_i - \varepsilon k)\Big],$$
$$[\delta(\mu(A_i,Y),Y) \geqslant \varepsilon_i] = \Big[s \leqslant \frac{\ell_i}{L_i}\big(m_i - |X'_i| - \varepsilon_i k_i\big)\Big].$$

Пусть $\varepsilon_i = \frac{L_i}{\ell_i k_i}\frac{\ell k}{L}\varepsilon + \big(1 - \frac{\ell L_i}{L\ell_i}\big)\frac{m_i}{k_i} - \frac{|X'_i|}{k_i}$. Непосредственной проверкой убеждаемся, что $[\delta(\mu(A_i,X),X) \geqslant \varepsilon] = [\delta(\mu(A_i,Y),Y) \geqslant \varepsilon_i]$. Подставляя это в (3.17), получаем утверждение теоремы. ■

Покажем, как для произвольного разбиения $A = A_1 \sqcup A_2 \sqcup \cdots \sqcup A_t$ построить систему порождающих и запрещающих множеств. Следуя [75], введем на $A$ отношение частичного 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 $a \prec b$.

Для отдельного алгоритма $a \in A$ порождающие и запрещающие множества определены в [75]:

$$\begin{aligned} X_a &= \{x \in X \colon \exists b \in A \colon a \prec b, I(a,x) < I(b,x)\}, \\ X'_a &= \{x \in X \colon \exists b \in A \colon b < a, I(b,x) < I(a,x)\}. \end{aligned} \tag{3.18}$$

Для кластера $A_i$ положим

$$X_i = \bigcap_{a \in A_i} X_a, \quad X'_i = \bigcap_{a \in A_i} X'_a. \tag{3.19}$$

**Лемма 3.16.** *Множества $X_i$ и $X'_i$, определенные в (3.19), являются, соответственно, порождающим и запрещающим множествами для кластера $A_i$ в смысле гипотезы 3.2.*

**Доказательство.** Для произвольного разбиения $X \in [\mathbb{X}]^\ell$ обозначим $a = \mu X$, и пусть $a \in A_i$. В [75] показано, что определенные в (3.18) множества $X_a$ и $X'_a$ являются порождающим и запрещающим множествами для алгоритма $a$, т. е. из условия $\mu X = a$ следует, что $X_a \subset X$ и $X'_a \subset \bar{X}$. Из определения $X_i$ и $X'_i$ следует, что $X_i \subset X_a$ и $X'_i \subset X'_a$. Следовательно, $X_i \subset X$ и $X'_i \subset \bar{X}$.

Условие «все алгоритмы $a \in A_i$ не допускают ошибок на $X_i$ и ошибаются на всех объектах из $X'_i$» также следует из определения $X_i$ и $X'_i$. ■

### 3.5.2 ПЗМ для рандомизированного метода обучения

В случае рандомизированного МЭР результатом обучения является подмножество $A(X) \subseteq A$. Таким образом, множество алгоритмов $A$ порождает множество подмножеств алгоритмов, получающихся в результате обучения

$$\mathfrak{A}(A) = \big\{A(X)\colon X \in [\mathbb{X}]^\ell\big\}.$$

**Гипотеза 3.3.** Пусть множество $A$ и выборка $\mathbb{X}$ таковы, что для каждого $\alpha \in \mathfrak{A}(A)$ можно указать пару непересекающихся подмножеств $X_\alpha \subset \mathbb{X}$ и $X'_\alpha \subset \mathbb{X}$, удовлетворяющую условию

$$\big[A(X){=}\alpha\big] = \big[X_\alpha \subseteq X\big]\big[X'_\alpha \subseteq \bar{X}\big] \quad \text{для всех } X \in [\mathbb{X}]^\ell. \tag{3.20}$$

Следующая теорема является непосредственным обобщением теоремы 3.14 для РМЭР.

**Теорема 3.17.** *Если справедлива гипотеза 3.3, то вероятность переобучения РМЭР есть*

$$Q_\varepsilon(A) = \sum_{a\in A}\sum_{\alpha\in\mathfrak{A}(A)} \frac{[a\in\alpha]}{|\alpha|}\frac{C^{\ell_\alpha}_{L_\alpha}}{C^\ell_L} H^{\ell_\alpha, m^a_\alpha}_{L_\alpha}\big(s^a_\alpha(\varepsilon)\big), \tag{3.21}$$

*где введены следующие обозначения:*

$$\begin{aligned} L_\alpha &= L - |X_\alpha| - |\bar{X}_\alpha|; \quad \ell_\alpha = \ell - |X_\alpha|; \\ m^a_\alpha &= n(a, \mathbb{X}\backslash X_\alpha \backslash X'_\alpha); \\ s^a_\alpha(\varepsilon) &= \tfrac{\ell}{L}\big(n(a,\mathbb{X}) - \varepsilon k\big) - n(a, X_\alpha). \end{aligned}$$

**Доказательство.** Рассмотрим функционал $Q_\varepsilon(A)$. Введем под знак суммирования по $X$ два вспомогательных суммирования: первое — по всем $\alpha \in \mathfrak{A}(A)$ при условии $\alpha = A(X)$, второе — по всем значениям $s$ числа ошибок алгоритма $a$ на подвыборке $X\backslash X_\alpha$. Очевидно, значение $Q_\varepsilon(A)$ от этого не изменится:

$$\begin{aligned} Q_\varepsilon(A) &= \mathsf{E}\sum_{a\in A(X)} \frac{1}{|A(X)|}[\delta(a,X) \geqslant \varepsilon] = \\ &= \mathsf{E}\sum_{\alpha\in\mathfrak{A}(A)}\sum_{a\in\alpha} \frac{[\alpha = A(X)]}{|\alpha|}[\delta(a,X) \geqslant \varepsilon] = \\ &= \mathsf{E}\sum_{\alpha\in\mathfrak{A}(A)}\sum_{a\in\alpha}\sum_{s=0}^{\ell_\alpha} \frac{[\alpha = A(X)]}{|\alpha|}[n(a, X\backslash X_\alpha) = s][\delta(a,X) \geqslant \varepsilon]. \end{aligned} \tag{3.22}$$

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

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

Следовательно,

$$[\delta(a,X) \geqslant \varepsilon] = \big[s \leqslant \frac{\ell}{L}(n(a,\mathbb{X}) - \varepsilon k) - n(a,X_\alpha)\big] = [s \geqslant s^a_\alpha(\varepsilon)].$$

Подставим полученное выражение в (3.22), затем заменим $[\alpha = A(X)]$ правой частью равенства (3.20) и переставим знак суммирования $\mathsf{E} \equiv \frac{1}{C^\ell_L} \sum\limits_{X \in [\mathbb{X}]^\ell}$ так, чтобы он оказался первым справа.

$$Q_\varepsilon(A) = \sum_{\alpha \in \mathfrak{A}(A)} \sum_{a \in \alpha} \sum_{s=0}^{\ell_a} \frac{1}{|\alpha|} \underbrace{\mathsf{E}\big[X_\alpha \subseteq X\big]\big[X'_\alpha \subseteq \bar{X}\big]\big[n(a, X \backslash X_\alpha) = s\big]}_{N(\alpha,a)} \big[s \leqslant s^a_\alpha(\varepsilon)\big]. \tag{3.23}$$

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

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

$$\mathbf{a} = \Big( \underbrace{X_\alpha; \overbrace{1,\ldots,1}^{s}; \underbrace{0,\ldots,0}_{X \backslash X_\alpha}}_{X}; \underbrace{X'_\alpha; \overbrace{1,\ldots,1}^{m^a_\alpha - s}; \underbrace{0,\ldots,0}_{\bar{X} \backslash X'_\alpha}}_{\bar{X}} \Big).$$

Число ошибок алгоритма $a$ на объектах, не попадающих ни в $X_\alpha$, ни в $X'_\alpha$, равно $m^a_\alpha$. Существует $C^s_{m^a_\alpha}$ способов выбрать из них $s$ объектов, которые попадут в $X \backslash X_\alpha$. Для каждого из этих способов имеется ровно $C^{\ell_\alpha - s}_{L_a - m^a_\alpha}$ способов выбрать $\ell_\alpha - s$ объектов, на которых алгоритм $a$ не допускает ошибку, и которые также попадут в $X \backslash X_\alpha$. Тем самым однозначно определяется состав выборки $X \backslash X_\alpha$, а следовательно, и состав выборки $\bar{X} \backslash X'_\alpha$. Таким образом, $N(\alpha,a) = C^s_{m^a_\alpha} C^{\ell_\alpha - s}_{L_\alpha - m^a_\alpha} / C^\ell_L$. Подставим это выражение в (3.23) и выделим в нем формулу гипергеометрической функции вероятности:

$$\begin{aligned} Q_\varepsilon(A) &= \sum_{\alpha \in \mathfrak{A}(A)} \sum_{a \in \alpha} \frac{1}{|\alpha|} \frac{C^{\ell_\alpha}_{L_\alpha}}{C^\ell_L} \sum_{s=s_0}^{\ell_\alpha} \big[s \leqslant s^a_\alpha(\varepsilon)\big] \frac{C^s_{m^a_\alpha} C^{\ell_\alpha - s}_{L_\alpha - m^a_\alpha}}{C^{\ell_\alpha}_{L_\alpha}} = \\ &= \sum_{a \in A} \sum_{\alpha \in \mathfrak{A}(A)} \frac{[a \in \alpha]}{|\alpha|} \frac{C^{\ell_\alpha}_{L_\alpha}}{C^\ell_L} H^{\ell_\alpha, m^a_\alpha}_{L_\alpha}\big(s^a_\alpha(\varepsilon)\big). \end{aligned}$$

Теорема доказана. ∎

**Следствие 3.2.** *Пусть во множестве $A$ найдется алгоритм $a_0$, такой что для любого $a \in A$ вектор ошибок алгоритма $a_0$ содержится в векторе ошибок алгоритма $a$. Обозначим через $X_0$ множество объектов, на которых ошибается алгоритм $a_0$. Пусть система порождающих*

*и запрещающих множеств такова, что для всех* $\alpha \in \mathfrak{A}(A)$ *выполнено* $X_0 \cap X_\alpha = \varnothing$ *и* $X_0 \cap X'_\alpha = \varnothing$. *Тогда*

$$m^a_\alpha = n(a_0, \mathbb{X}), \quad s^a_\alpha(\varepsilon) = \tfrac{\ell}{L}\big(n(a, \mathbb{X}) - \varepsilon k\big).$$

**Доказательство.** Зафиксируем обучающую выборку $X \in [\mathbb{X}]^\ell$, и пусть $\alpha = A(X)$. Докажем, что из $a \in \alpha$ следует $n(a, X_\alpha) = 0$. Пусть $a$ ошибается на объекте $x$. Нам необходимо доказать, что $x \notin X_\alpha$. Допустим обратное, тогда по определению запрещающих объектов $x \in X_\alpha$ обязан лежать в обучении. Условие $X_0 \cap X_\alpha = \varnothing$ означает, что $a_0$ не ошибается на $x$. Следовательно, алгоритм $a$ делает как минимум на одну ошибку больше, чем $a_0$ на обучении. Противоречие.

Второе утверждение заключается в том, что из $a \in \alpha$ следует $n(a, \mathbb{X}) = n(a_0, X_0) + n(a, X'_\alpha)$. Запишем число ошибок алгоритма $a$ в виде $n(a, \mathbb{X}) = n(a, X_0) + n(a, X \setminus X_0)$. Из условий теоремы следует, что $n(a, X_0) = n(a_0, X_0)$. Следовательно, для доказательства достаточно показать, что $n(a, X \setminus X_0) = n(a, X'_\alpha)$. Отметим, что из условия $X_0 \cap X'_\alpha$ следует, что $X'_\alpha \subset X \setminus X_0$, а значит, $n(a, X \setminus X_0) \geqslant n(a, X'_\alpha)$. Осталось доказать, что каждая ошибка $a \in X \setminus X_0$ алгоритма $a$ принадлежит $X'_\alpha$. Это следует из того, что алгоритмы $a_0$ и $a$ оба лежат в $A(X)$, а значит неразличимы на обучающей выборке.

Из доказанных выше утверждений следует, что

$$\begin{aligned} m^a_\alpha &= n(a, \mathbb{X} \setminus X_\alpha \setminus X'_\alpha) = n(a_0, X_0) = n(a_0, \mathbb{X}); \\ s^a_\alpha(\varepsilon) &= \frac{\ell}{L}\big(n(a, \mathbb{X}) - \varepsilon k\big) - n(a, X_\alpha) = \frac{\ell}{L}\big(n(a, \mathbb{X}) - \varepsilon k\big). \end{aligned}$$

■

**Следствие 3.3.** *Полученная формула легко объединяется с теоремой о разбиении множества алгоритмов на орбиты:*

$$Q_\varepsilon(A) = \sum_{\omega \in \Omega(A)} \sum_{\alpha \in \mathfrak{A}(A)} [a_\omega \in \alpha] \frac{|\omega|}{|\alpha|} \frac{C^{\ell_\alpha}_{L_\alpha}}{C^\ell_L} H^{\ell_\alpha, m^{a_\omega}_\alpha}_{L_\alpha}\big(s^{a_\omega}_\alpha(\varepsilon)\big). \tag{3.24}$$

**Доказательство.** Доказательство немедленно следует из леммы 3.5 о равном вкладе алгоритмов одной орбиты в вероятность переобучения. ■

## 3.6 Основные выводы

Теоремы 3.6 и 3.10, полученные в данном параграфе, являются основным инструментом для вывода оценок вероятности переобучения РМЭР для симметричных семейств алгоритмов. Оценки (3.9) и (3.11) являются точными равенствами и, следовательно, неулучшаемы.

Помимо этого, метод порождающих и запрещающих множеств (ПЗМ), предложенный К. В. Воронцовым, был обобщен по двум направлениям: во-первых, на случай разложения множества алгоритмов на кластеры, во-вторых, на случай РМЭР.

# Глава 4. Точные оценки вероятности переобучения для РМЭР

В данной главе изучается ряд модельных семейств алгоритмов, непосредственно заданных с помощью бинарной матрицы ошибок. Эти семейства моделируют идеализированные свойства реальных семейств алгоритмов: размерность и разреженность семейства, расслоение алгоритмов по числу ошибок, связность и сходство алгоритмов, и др.

Изучение модельных множеств интересно не только в теории, но также может быть полезно на практике. В частности, в работе П. Ботова [3] модельные множества используются для аппроксимации реального семейства классификаторов с помощью унимодальной несимметричной сети алгоритмов малой высоты и размерности, для которой известны точные комбинаторные формулы вероятности переобучения. Эксперименты на решающих деревьях и реальных задачах классификации показывают, что такой подход повышает обобщающую способность получаемых алгоритмов классификации.

В настоящей работе предлагается использовать модельные семейства в качестве объемлющего множества $B$ (см. лемму 3.13) для эффективного вычисления оценки (3.16), основанной на разложениии и покрытии множества алгоритмов.

## 4.1 Монотонные и унимодальные цепи

**Определение 4.1.** *Цепью алгоритмов называется семейство алгоритмов, элементы которого можно выстроить в такую последовательность, что любые два соседних алгоритма последовательности различаются лишь на одном объекте.*

Точные оценки вероятности переобучения ПМЭР для монотонных и унимодальных цепей были получены ранее К. В. Воронцовым. Ниже аналогичные результаты будут получены для РМЭР.

### 4.1.1 Монотонная цепь

Монотонная цепь является фундаментальным объектом в теории комбинаторного обучения. Это одно из простейших модельных семейств, одновременно обладающее свойствами расслоения и связности.

**Определение 4.2.** *Множество алгоритмов* $\{a_0, \ldots, a_D\}$ *называется монотонной цепью, если оно является цепью в смысле определения 4.1, и кроме этого число ошибок является монотонной функцией номера алгоритма в цепи:*

$$n(a_i, \mathbb{X}) = n(a_0, \mathbb{X}) + i, \text{ при } i = 0, \ldots, D.$$

Монотонная цепь алгоритмов — это простейшая модель однопараметрического *связного семейства алгоритмов*, предполагающая, что при непрерывном удалении некоторого параметра от оптимального значения число ошибок на полной выборке только увеличивается.

**Пример 4.1.** Пусть $A$ — семейство *линейных алгоритмов классификации*, т. е. семейство параметрических отображений из $\mathbb{X} = \mathbb{R}^n$ в $\mathbb{Y} = \{-1, +1\}$ вида

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

Параметр $w \in \mathbb{R}^n$ задает направляющий вектор гиперплоскости, разделяющей пространство $\mathbb{R}^n$ на два полупространства, соответствующих классам $-1$ и $+1$. Пусть функция потерь имеет вид $I(a, x) = \big[a(x, w) \neq y(x)\big]$, где $y(x)$ — истинная классификация объекта $x$. Пусть 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 $n(a_0, \mathbb{X}) = 0$ в силу линейной разделимости.

Мы воспользуемся теоремой 3.2 о порождающих и запрещающих множествах для рандомизированного метода обучения. Для этого мы вначале установим структуру множества $\mathfrak{A}(A) = \big\{A(X) \colon X \in [\mathbb{X}]^\ell\big\}$, затем построим систему порождающих и запрещающих множеств для каждого $\alpha \in \mathfrak{A}(A)$ и, наконец, воспользуемся теоремой 3.2.

**Лемма 4.1.** *Для монотонной цепи длины* $D$ *множество* $\mathfrak{A}(A)$ *состоит из* $D + 1$ *элемента:* $\mathfrak{A}(A) = \{\alpha_0, \ldots, \alpha_D\}$, *причем для всех* $i$ *выполнено* $\alpha_i = \{a_0, \ldots, a_i\}$.

**Доказательство.** Доказательство следует непосредственно из определения РМЭР 3.2 и монотонности числа ошибок в цепи. ■

Чтобы выписать структуру порождающих и запрещающих множеств, необходимо зафиксировать нумерацию объектов выборки. Сделаем это так, как показано ниже:

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

При такой нумерации каждый из алгоритмов $a_t$, $t = 1, \ldots, D$, допускает ошибку на объектах $x_1, \ldots, x_t$. Нумерация остальных объектов не имеет значения, так как на этих объектах алгоритмы неразличимы.

**Лемма 4.2.** *Система порождающих и запрещающих множеств монотонной цепи устроена следующим образом:*

$$[A(X){=}\alpha_t] = [x_1, \ldots, x_D \in \bar{X}], \text{ при } t = D,$$
$$[A(X){=}\alpha_t] = [x_{t+1} \in X][x_1, \ldots, x_t \in \bar{X}], \text{ при } t \leqslant D.$$

**Доказательство.** Рассмотрим два случая.

1. Если $t = D$, то $\alpha_t$ совпадает со всем множеством алгоритмов. Следовательно, $A(X) = \alpha_t$ тогда и только тогда, когда все объекты $\{x_1, \ldots, x_D\}$ будут находиться в контрольной подвыборке $\bar{X}$. В этом случае

$$[A(X){=}\alpha_t] = [x_1, \ldots, x_D \in \bar{X}].$$

2. Во всех остальных случаях $A(X) = \alpha_t$ тогда и только тогда, когда все объекты $\{x_1, \ldots, x_t\}$ будут находиться в контрольной подвыборке $\bar{X}$, а объект $x_{t+1}$ — в обучающей подвыборке $X$. В этом случае

$$[A(X){=}\alpha_t] = [x_{t+1} \in X][x_1, \ldots, x_t \in \bar{X}].$$

■

Отметим, что в случае монотонной цепи структура множества $\mathfrak{A}(A)$ оказалась идентичной самому множеству $A$. Кроме этого, система порождающих и запрещающих множеств рандомизированного МЭР оказалась такой же, как и у детерминированного МЭР. Данное совпадение вызвано простой структурой рассматриваемого множества алгоритмов и не имеет места в общем случае.

**Теорема 4.3.** *Для монотонной цепи из $D$+1 алгоритмов вероятность переобучения РМЭР равна*

$$Q_\varepsilon(A) = \sum_{d=0}^{D} \sum_{t=d}^{D} \frac{1}{1+t} \frac{C_{L'}^{\ell'}}{C_L^\ell} H_{L'}^{\ell',m}\big(s(\varepsilon)\big), \tag{4.1}$$

*где* $L' = L{-}t{-}F,\ \ell' = \ell{-}F,\ F = [t{\neq}D],\ s(\varepsilon) = \left\lfloor \frac{\ell}{L}(m{+}d{-}\varepsilon k) \right\rfloor$.

**Доказательство.** Воспользовавшись леммами 4.1 и 4.2, получим, что для всех $t = 0, \ldots, D$ выполнено $|\alpha_t| = t{+}1$, $L_t = L{-}t{-}1$, $\ell_t = \ell{-}[t = D]$, $m_t^d = m{+}d{-}d = m$, $s_t^d(\varepsilon) = \frac{\ell}{L}(m{+}d{-}\varepsilon k)$, где для упрощения обозначений вместо двойных индексов $L_{\alpha_t}$, $\ell_{\alpha_t}$, $m_{\alpha_t}^d$, $s_{\alpha_t}^d(\varepsilon)$ использованы одинарные: $L_t$, $\ell_t$, $m_t^d$, и $s_t^d(\varepsilon)$. Далее заметим, что система порождающих и запрещающих множеств, построенных в лемме 4.2, удовлетворяет условиям следствия 3.2 теоремы 3.17. Подставив посчитанные значения $L_t$, $\ell_t$, $m_t^d$, $s_t^d(\varepsilon)$ в (3.21), получим утверждение настоящей теоремы. ■

В приведенном доказательстве мы не рассматривали отдельно случай $D \leqslant k$ и $D > k$. Эти эффекты уже учтены корректно благодаря тому, что мы доопределили нулем биномиальные коэффициенты в гипергеометрическом распределении. Также отметим, что в случае монотонной цепи группа симметрии оказалась тривиальной, и потому не учитывалась при вычислениях.

### 4.1.2 Унимодальная цепь

Унимодальная цепь является более реалистичной моделью однопараметрического *связного семейства* по сравнению с монотонной цепью. Если мы имеем лучший алгоритм $a_0$ с оптимальным значением некоторого вещественного параметра, то отклонение значения этого параметра как в бóльшую, так и в меньшую сторону приводит к увеличению числа ошибок.

**Определение 4.3.** *Множество алгоритмов*

$$\{a_0, a_1, \ldots, a_D, a'_1, \ldots, a'_D\}$$

*называется унимодальной цепью, если выполнены два условия:*

1) *левая ветвь* $\{a_0, a_1, \ldots, a_D\}$ *и правая ветвь* $\{a_0, a'_1, \ldots, a'_D\}$ *являются монотонными цепями;*

2) *пересечение множества ошибок алгоритмов* $a_D$ *и* $a'_D$ *равно множеству ошибок алгоритма* $a_0$.

Параметр $D$ будем называть *длиной ветвей* унимодальной цепи.

Рассмотрим унимодальную цепь длины $D$ и пронумеруем объекты генеральной выборки $\mathbb{X}$ так, как показано ниже:

$$
\begin{array}{c}
\\
x_1 \\
x_2 \\
\\
x_D \\
\hline
x'_1 \\
x'_2 \\
\\
x'_D
\end{array}
\begin{array}{c}
\begin{array}{cccccccccc}
a_0 & a_1 & a_2 & \cdots & a_D & a'_1 & a'_2 & \cdots & a'_D
\end{array} \\
\left(\begin{array}{cccccccccc}
0 & 1 & 1 & \cdots & 1 & 0 & 0 & \cdots & 0 \\
0 & 0 & 1 & \cdots & 1 & 0 & 0 & \cdots & 0 \\
\cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\
0 & 0 & 0 & \cdots & 1 & 0 & 0 & \cdots & 0 \\
\hline
0 & 0 & 0 & \cdots & 0 & 1 & 1 & \cdots & 1 \\
0 & 0 & 0 & \cdots & 0 & 0 & 1 & \cdots & 1 \\
\cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\
0 & 0 & 0 & \cdots & 0 & 0 & 0 & \cdots & 1
\end{array}\right)
\end{array}
$$

Нумерация остальных объектов не имеет значения, так как на этих объектах алгоритмы неразличимы.

**Лемма 4.4.** *Группа симметрии унимодальной цепи при $D \geqslant 1$ содержит в качестве своей подгруппы группу перестановок $S_2$. Данная подгруппа действует на алгоритмы цепи перестановкой левой и правой ветви.*

**Доказательство.** Рассмотрим группу $S_2$, состоящую из нулевой перестановки и из перестановки, действующей по правилу $(x_1 \leftrightarrow x'_1, \ldots, x_D \leftrightarrow x'_D)$. Данная группа удовлетворяет условию леммы. ■

Из леммы 4.4 следует, что алгоритмы разных ветвей с равным числом ошибок лежат в одной орбите действия группы симметрии. Орбита $\omega_0 = \{a_0\}$ содержит единственный алгоритм. Для остальных орбит $\omega_d = \{a_d, a'_d\}$ договоримся выбирать алгоритм $a_d$ из левой ветви в качестве представителя орбиты.

**Лемма 4.5.** *Для унимодальной цепи длины $D$ выполнено $\mathfrak{A}(A) = \{\alpha_{t_1,t_2})\}$, где $t_1, t_2 = 0, \ldots, D$. При этом $\alpha_{t_1,t_2} = \{a_0, a_1, \ldots, a_{t_1}, a'_1, \ldots, a'_{t_2}\}$.*

**Доказательство.** Доказательство следует непосредственно из определения РМЭР 3.2 и монотонности числа ошибок в каждой ветви унимодальной цепи. ■

Множества порождающих и запрещающих объектов для унимодальной цепи строятся по аналогии с леммой 4.2 для монотонной цепи. Мы не будем выписывать их в явном виде

и непосредственно перейдем к доказательству теоремы о вероятности переобучения РМЭМ для унимодальной цепи.

**Теорема 4.6.** *Для унимодальной цепи с ветвями длины $D$ вероятность переобучения РМЭР равна*

$$Q_\varepsilon(A) = \sum_{d=0}^{D} \sum_{t_1=d}^{D} \sum_{t_2=0}^{D} \frac{|\omega_d|}{1+t_1+t_2} \frac{C_{L'}^{\ell'}}{C_L^\ell} H_{L'}^{\ell',m}\big(s(\varepsilon)\big), \tag{4.2}$$

*где* $\omega_d = [d=0] + 2\cdot[d>0]$, $L' = L - S - F$, $S = t_1 + t_2$, $F = [t_1 \neq D] + [t_2 \neq D]$, $\ell' = \ell - F$, $s(\varepsilon) = \big\lfloor \frac{\ell}{L}(m+d-\varepsilon k)\big\rfloor$.

**Доказательство.** Воспользуемся леммами 4.4 и 4.5. Рассмотрим произвольное $\alpha \equiv \alpha_{t_1,t_2} \in \mathfrak{A}(A)$. Легко заметить, что при $t_1, t_2$ строго меньше $D$ множеством порождающих объектов будет $X'_\alpha = \{x_{t_1+1}, x_{t_2+1}\}$. Условие $t_i = D$ уменьшает количество порождающих объектов в $X'_\alpha$ на единицу. Множество запрещающих объектов $X_\alpha = \{x_1, \ldots, x_{t_1}, x'_1, \ldots, x'_{t_2}\}$. Введя обозначение $F = [t_1 \neq D] + [t_2 \neq D]$, получим

$$L_\alpha = L - t_1 - t_2 - F, \qquad \ell_\alpha = \ell - F;$$
$$m^a_\alpha = m + d - d = m, \quad s^a_\alpha(\varepsilon) = \lfloor \frac{\ell}{L}(m+d-\varepsilon k)\rfloor.$$

Наконец, мощность множества $\alpha_{t_1,t_2}$ равна $|\alpha_{t_1,t_2}| = \frac{1}{1+t_1+t_2}$, а $[a_d \in \alpha_{t_1,t_2}] = [d \leqslant t_1]$.

Подставляя эти значения в общую формулу из теоремы 3.17 о порождающих и запрещающих объектах, получаем утверждение доказываемой теоремы. ■

## 4.2 Многомерные семейства алгоритмов

### 4.2.1 Пучок монотонных цепей

**Определение 4.4.** *Пучком из $h$ монотонных цепей называется множество алгоритмов, полученное объединением $h$ монотонных цепей равной длины, с общим первым алгоритмом. Как и в случае унимодальной цепи, предполагается, что множества объектов, на которых ошибаются алгоритмы ветвей, не пересекаются.*

Связка из $2h$ монотонных цепей является моделью $h$-параметрического семейства алгоритмов, в котором разрешено изменять любой из $h$ параметров при фиксированных остальных, а одновременное изменение нескольких параметров не допускается. Данное семейство можно также рассматривать как обобщение трех частных случаев, рассмотренных

в [9]: монотонной цепи ($h = 1$), унимодальной цепи ($h = 2$) и единичной окрестности лучшего алгоритма ($D = 1$).

**Лемма 4.7.** *Группа симметрии связки из $h$ монотонных цепей содержит в качестве подгруппы симметрическую группу $S_h$, действующую на ветви связки всевозможными перестановками.*

**Доказательство.** Достаточно рассмотреть произвольную перестановку $\pi \in S_h$ и указать перестановку $g\colon \mathbb{X} \to \mathbb{X}$, действующую на ветви связки в соответствии с $\pi$. Такая перестановка строится в явном виде по аналогии с леммой 4.4. ■

Из леммы 4.7 следует, что алгоритмы разных ветвей с равным числом ошибок лежат в одной орбите действия группы симметрии. Орбита $\omega_0 = \{a_0\}$ содержит единственный алгоритм. Для остальных орбит $\omega_d = \{a_d^1, a_d^2, \ldots, a_d^h\}$ договоримся выбирать алгоритм $a_d^1$ из первой ветви в качестве представителя орбиты.

**Лемма 4.8.** *Для связки из $h$ цепей длины $D$ выполнено $\mathfrak{A}(A) = \{\alpha_{t_1,\ldots,t_h})\}$, где $t_i = 0, \ldots, D$ для всех $i$. При этом $\alpha_{t_1,\ldots,t_h} = \{a_j^i\}$, где $i = 1, \ldots, h$, $j = 1, \ldots, t_i$.*

**Доказательство.** Доказательство следует непосредственно из определения РМЭР 3.2 и монотонности числа ошибок в каждой ветви унимодальной цепи. ■

Множества порождающих и запрещающих объектов для связки цепей строятся по аналогии с леммой 4.2 для монотонной цепи. Мы вновь не будем выписывать их в явном виде и непосредственно перейдем к доказательству теоремы о вероятности переобучения РМЭМ для связки цепей. Для этого нам понадобится *комбинаторный коэффициент* $R_{D,h}^d(S,F)$, который зависит от параметров $S$ и $F$, от числа монотонных цепей $h$ и от их длины $D$, а также от $d$ — минимального значения параметра $S$. Коэффициент $R_{D,h}^d(S,F)$ равен числу способов представить число $S$ в виде суммы $h$ неотрицательных пронумерованных слагаемых, $S = t_1 + \ldots + t_h$, каждое из которых не превосходит $D$. При этом ровно $F$ слагаемых не должны равняться $D$, а на первое слагаемое накладывается дополнительное ограничение $t_1 \geqslant d$.

**Теорема 4.9.** *Пусть в связке из $h$ монотонных цепей лучший алгоритм допускает $m$ ошибок на полной выборке, длина каждой ветви без учета лучшего алгоритма равна $D$. Тогда при обучении рандомизированным методом вероятность переобучения может быть записана в виде:*

$$Q_\varepsilon(A) = \sum_{d=0}^{D} \sum_{S=d}^{hD} \sum_{F=0}^{h} \frac{|\omega_d| R_{D,h}^d(S,F)}{1+S} \frac{C_{L'}^{\ell'}}{C_L^\ell} H_{L'}^{\ell',m}\big(s(\varepsilon)\big), \tag{4.3}$$

*где* $L' = L - S - F$, $\ell' = \ell - F$, $s(\varepsilon) = \left\lfloor \frac{\ell}{L}(m + d - \varepsilon k) \right\rfloor$; $|\omega_h| = 1$ *при* $h = 0$ *и* $|\omega_d| = h$ *при* $d \geqslant 1$.

**Доказательство.** Воспользуемся леммами 4.7 и 4.8. Для $\alpha_{\mathbf{t}}$, где все $t_i$ строго меньше $D$, множеством порождающих объектов будет $X'_\alpha = \{x^1_{t_1+1}, \ldots, x^h_{t_h+1}\}$. Условие $t_i = D$ уменьшает количество порождающих объектов в $X'_\alpha$ на единицу. Множество запрещающих объектов $X_\alpha = \{x^1_1, \ldots, x^1_{t_1}, x^2_1, \ldots, x^2_{t_2}, \ldots, x^h_1, \ldots, x^h_{t_h}\}$.

Введем обозначения $S = \sum\limits_{i=1}^{p} t_i$, $F = \sum\limits_{i=1}^{p} [t_i \neq D]$. Тогда $L_\alpha = L - S - F$, $\ell_\alpha = \ell - F$, $m^a_\alpha = n(a_0, \mathbb{X})$, $s^a_\alpha = \left\lfloor \frac{\ell}{L}(m + d - \varepsilon k) \right\rfloor$. Наконец, $|\alpha_{\mathbf{t}}| = \frac{1}{1+t_1+\ldots+t_h}$, $[a^1_d \in \alpha_{\mathbf{t}}] = [d \leqslant t_1]$.

Подставляя эти значения в общую формулу из теоремы 3.17 о порождающих и запрещающих множествах, получим следующее выражение для вероятности переобучения:

$$Q_\varepsilon(A) = \sum_{d=0}^{D} \sum_{t_1=d}^{D} \sum_{t_2=0}^{D} \ldots \sum_{t_h=0}^{D} \frac{|\omega_d|}{1+S} \frac{C^{\ell'}_{L'}}{C^{\ell}_{L}} H^{\ell',m}_{L'}\big(s(\varepsilon)\big).$$

Теперь от суммирования по параметрам $t_i$ можно перейти к суммированию по множеству возможных значений $S$ и $F$:

$$Q_\varepsilon(A) = \sum_{d=0}^{D} \sum_{S=d}^{hD} \sum_{F=0}^{h} |\omega_h| \frac{R^d_{D,h}(S,F)}{1+S} \frac{C^{\ell'}_{L'}}{C^{\ell}_{L}} H^{\ell',m}_{L'}\big(s(\varepsilon)\big),$$

где $R^d_{D,h}(S,F)$ — определенный выше комбинаторный коэффициент. ■

**Следствие 4.1.** *Для единичной окрестности из* $h$ *алгоритмов вероятность переобучения равна*

$$Q_\varepsilon(A) = \sum_{d=0}^{1} \sum_{S=d}^{h} \frac{|\omega_d| C^{S-d}_{h-d}}{1+S} \frac{C^{\ell'}_{L'}}{C^{\ell}_{L}} H^{\ell',m}_{L'}\big(s(\varepsilon)\big), \tag{4.4}$$

*где* $L' = L - h$, $\ell' = \ell + S - h$.

На рис. 4.1 и рис. 4.2 представлены результаты численных экспериментов, в которых сравнивались вероятности переобучения для различных вариантов минимизации эмпирического риска. Из четырех кривых на каждом графике верхняя (жирная) соответствует пессимистической минимизации эмпирического риска, нижняя — оптимистической. Две почти сливающиеся кривые между ними соответствуют РМЭР. Одна из кривых вычислена по доказанным формулам, а вторая построена методом Монте-Карло по $10^5$ случайных разбиений, при равновероятном выборе лучшего алгоритма в случаях неопределенности. Различия этих двух кривых находятся в пределах погрешности метода Монте-Карло.

На рис. 4.3 и рис. 4.4 представлены зависимости вероятности переобучения от числа $h$ ветвей в связке и от их длины $D$. Графики построены для РМЭР. Рис. 4.4 показывает,

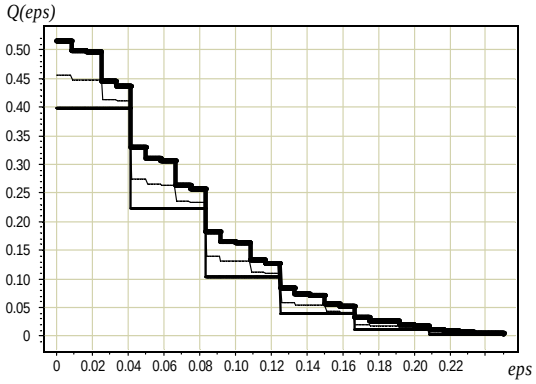



Рис. 4.1: Зависимость $Q_\varepsilon(A)$ от $\varepsilon$ для монотонной цепи при $L = 100$, $\ell = 60$, $D = 40$, $m = 20$.

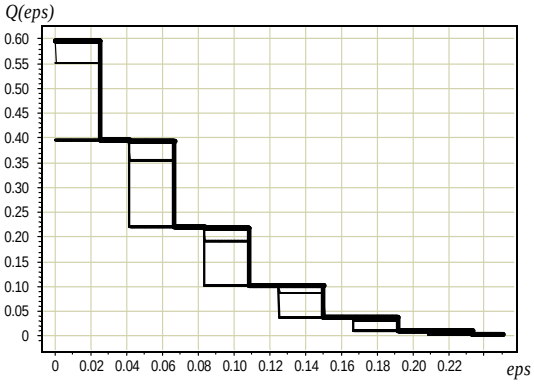



Рис. 4.2: Зависимость $Q_\varepsilon(A)$ от $\varepsilon$ для единичной окрестности при $L = 100$, $\ell = 60$, $h = 10$, $m = 20$.

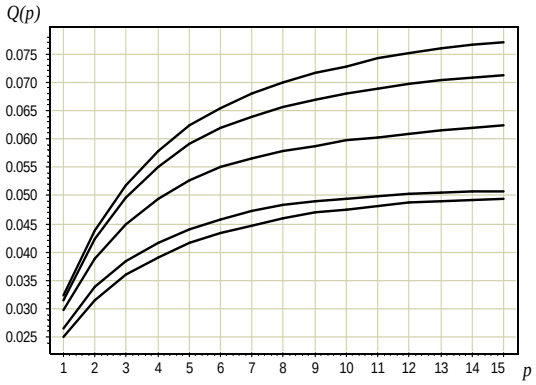



Рис. 4.3: Зависимость $Q_\varepsilon(A)$ от $h$ для связки из монотонных цепей при $L = 300$, $\ell = 150$, $m = 15$, $D = 1, 2, 3, 5, 10$, $\varepsilon = 0.05$.

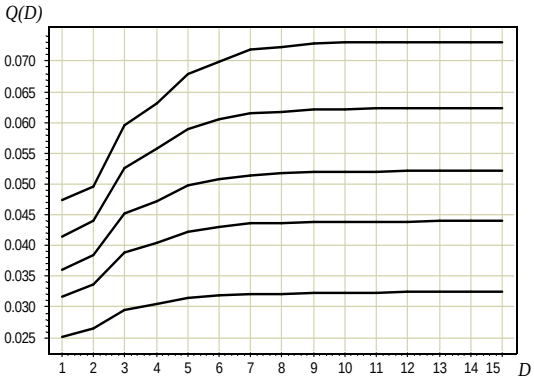



Рис. 4.4: Зависимость $Q_\varepsilon(A)$ от $D$ для связки из $h = 1, 2, 3, 5, 10$ монотонных цепей при $L = 300$, $\ell = 150$, $m = 15$, $\varepsilon = 0.05$.

что при увеличении длин цепей $D$ вероятность переобучения практически перестает расти уже при $D = 7$. Это связано с *эффектом расслоения* — лишь алгоритмы из нижних слоев имеют существенно отличную от нуля вероятность быть выбранными методом минимизации эмпирического риска. Добавление «слишком плохих» алгоритмов не увеличивает вероятность переобучения. Рис. 4.3 показывает, что при увеличении числа цепей $(h)$ в связке вероятность переобучения продолжает расти. Однако скорость роста сублинейна по $h$ благодаря *эффекту связности* — все алгоритмы находятся на хэммингове расстоянии не более $D$ от лучшего алгоритма.

### 4.2.2 Многомерная монотонная сеть алгоритмов

Введем целочисленный вектор индексов $\mathbf{d} = (d_1, \ldots, d_h) \in \mathbb{Z}^h$. Обозначим $\|\mathbf{d}\| = \max\limits_{j=1,\ldots,h} |d_j|$, $|\mathbf{d}| = |d_1| + \ldots + |d_h|$. На множестве векторов индексов введем покомпонентное отношение сравнения: $\mathbf{d} < \mathbf{d}'$, если $d_j \leqslant d'_j$, $j = 1, \ldots, h$, и хотя бы одно из неравенств строгое.

**Определение 4.5.** *Множество алгоритмов $A = \{a_{\mathbf{d}}\}$, где $\mathbf{d} \geqslant 0$ и $\|\mathbf{d}\| \leqslant D$, называется монотонной $h$-мерной сетью алгоритмов длины $D$, если существуют $h \in \mathbb{N}$ и упорядоченные наборы объектов $X_j = \{x_j^1, \ldots, x_j^D\} \subset \mathbb{X}$ для всех $j = 1, \ldots, h$, а также множества $U_1 \subset \mathbb{X}$ и $U_0 \subset \mathbb{X}$, такие что:*

1) *набор $\{U_0, U_1, \{X_j\}_{j=1}^h\}$ является разбиением множества $\mathbb{X}$ на непересекающиеся подмножества;*
2) *$a_d(x_j^i) = [i \leqslant d_j]$, где $x_j^i \in X_j$;*
3) *$a_d(x_0) = 0$ при всех $x_0 \in U_0$;*
4) *$a_d(x_1) = 1$ при всех $x_1 \in U_1$.*

Монотонная сеть алгоритмов — это модель параметрического *связного семейства алгоритмов*, предполагающая, что при непрерывном удалении каждой компоненты вектора параметров от оптимального значения число ошибок на полной выборке только увеличивается.

Обозначим $|U_1| = m$. Из определения следует, что $n(a_{\mathbf{d}}, \mathbb{X}) = m + |d|$. Алгоритм $a_{\mathbf{0}}$ является *лучшим в сети*. Множество алгоритмов с равным числом ошибок $t+m = n(a_{\mathbf{d}}, \mathbb{X})$ называются *$t$-слоем* сети.

**Пример 4.2.** Монотонная двумерная сеть при $m = 0$ и $L = 4$:

| | $a_{0,0}$ | $a_{1,0}$ | $a_{2,0}$ | $a_{0,1}$ | $a_{1,1}$ | $a_{2,1}$ | $a_{0,2}$ | $a_{1,2}$ | $a_{2,2}$ |
|---|---|---|---|---|---|---|---|---|---|
| $x_1$ | 0 | **1** | **1** | 0 | **1** | **1** | 0 | **1** | **1** |
| $x_2$ | 0 | 0 | **1** | 0 | 0 | **1** | 0 | 0 | **1** |
| $x_3$ | 0 | 0 | 0 | **1** | **1** | **1** | **1** | **1** | **1** |
| $x_4$ | 0 | 0 | 0 | 0 | 0 | 0 | **1** | **1** | **1** |

Число алгоритмов в $h$-мерной монотонной сети с ветвями длины $D$ равно $(D+1)^h$. *Укороченной* $h$-мерной монотонной сетью $\tilde{A} \subset A$ назовем первые $D$ слоев из $A$. Таким

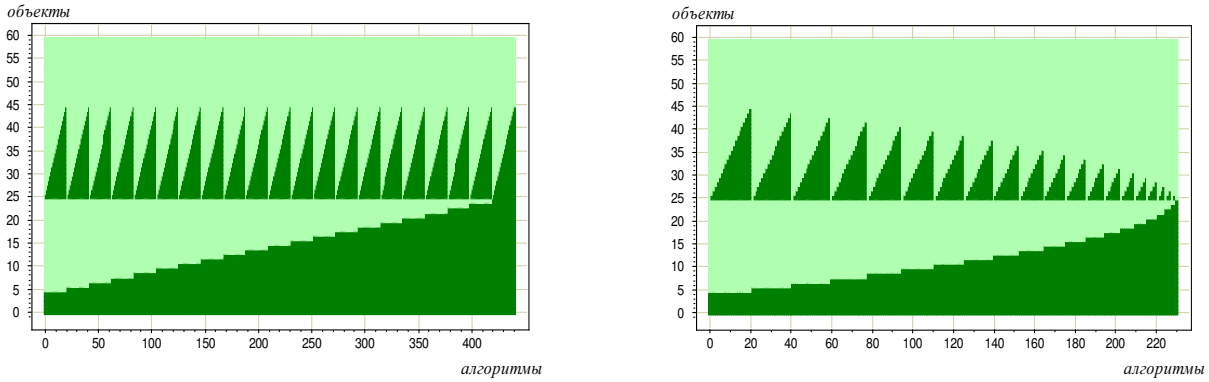



Рис. 4.5: Матрица ошибок монотонной сети (слева) и укороченной монотонной сети (справа) при $D = 20$, $h = 2$, $m = 5$, $L = 60$.

образом,

$$\tilde{A} = \{a_{\mathbf{d}} \in A,\ |\mathbf{d}| \leqslant D\}.$$

Число алгоритмов в $\tilde{A}$ равно $C^h_{D+h}$.

Пример матрицы ошибок монотонной сети приведен на рис. 4.5.

Впервые монотонные сети произвольной размерности были изучены П. Ботовым в [2, 3, 46]. Там же были получены формулы для вероятности переобучения *пессимистического* метода минимизации эмпирического риска.

Численные эксперименты показывают, что при разумных сочетаниях параметров вероятность переобучения РМЭР для укороченной сети $\tilde{A}$ и для простой сети $A$ различаются крайне мало. Поэтому в дальнейшем мы ограничимся исследованием неукороченных монотонных сетей. Для этого класса семейств алгоритмов будут получены явные формулы вероятности переобучения РМЭР.

**Лемма 4.10.** *Группа симметрии монотонной сети размерности $h$ содержит в качестве подгруппы группу $S_h$ всевозможных перестановок множеств $X_1, \ldots, X_h$.*

**Доказательство.** Все алгоритмы $h$-мерной монотонной сети длины $D$ индексированы множеством вектор-индексов $\mathbf{d} \in \{0, \ldots, D\}^h$. Здесь число ошибок алгоритма $a_{\mathbf{d}} = m + |\mathbf{d}|$.

Рассмотрим алгоритм $a_{\mathbf{d}} \in A$ и произвольную $\pi \in S_h$. По данному выше определению действия $\pi$ на $\mathbb{X}$ получаем, что $\pi a_{\mathbf{d}} = a_{\pi \mathbf{d}}$, где действие $\pi$ на вектор $\mathbf{d}$ определяется соответствующей перестановкой его координат. Множество $\{0, \ldots, D\}^h$ сохраняется при применении к нему произвольной перестановки координат $\pi \in S_h$. Поэтому $\forall \mathbf{d} \in \{0, \ldots, D\}^h$ выполнено $\pi \mathbf{d} \in \{0, \ldots, D\}^h$. А следовательно, $a_{\pi \mathbf{d}} \in A$. ■

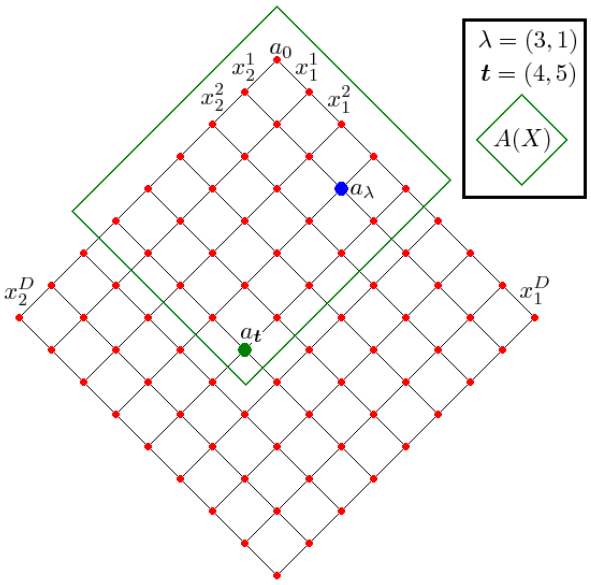



Рис. 4.6: Строение множества $A(X)$ для двумерной монотонной сети; $h = 2$, $D = 8$.

Пусть $Y_h^D$ — множество целочисленных неотрицательных невозрастающих последовательностей из $h$ элементов, каждый из которых не превосходит $D$; пусть $|S_h\mathbf{d}|$ — число различных слов, состоящих из символов $d_1, \ldots, d_h$.

**Лемма 4.11.** *Пусть $A = \{a_{\mathbf{d}}\}$, $\|d\| \leqslant D$ — монотонная сеть длины $D$ с размерностью $h$. Тогда множество орбит $A$ под действием $S_h$ индексировано всевозможными векторами $\lambda \in Y_h^D$. Число алгоритмов в орбите $\omega_\lambda$, где $\lambda = (\lambda_1, \ldots, \lambda_h)$, равно числу различных слов длины $h$, состоящих из символов $\lambda_1, \ldots, \lambda_h$: $|\omega_\lambda| = |S_h\lambda|$.*

**Доказательство.** Напомним, что вместо действия $S_h$ на $A = \{a_{\mathbf{d}}\}$ можно рассматривать действие $S_h$ на вектор индексов $\mathbf{d}$, где новое действие определяется соответствующей перестановкой координат вектора $\mathbf{d}$.

Рассмотрим орбиту произвольного алгоритма $a_{\mathbf{d}}$. Возьмем перестановку $\pi \in S_h$, упорядочивающую координаты $\mathbf{d}$ в порядке невозрастания. Положим $\lambda = \pi\mathbf{d}$. Построенная таким образом $\lambda$ лежит в множестве $Y_h^D$. При этом различным $\lambda_1$ и $\lambda_2$ будут соответствовать различные орбиты действия группы $S_h$ на $\{a_{\mathbf{d}}\}$.

Взаимно-однозначное соответствие между словами длины $h$ из символов $\lambda_1, \ldots, \lambda_h$ и количеством элементов орбиты $|\omega_\lambda|$ очевидно. ■

**Лемма 4.12.** *Для монотонной сети множество $\mathfrak{A}(A) \equiv \{A(X) \colon X \in [\mathbb{X}]^\ell\}$ устроено следующим образом:*

$$\mathfrak{A}(A) = \{\alpha_{\mathbf{t}} : \mathbf{t} \in [0, \ldots, D]^h\},$$

*где* $\alpha_{\mathbf{t}} = \{a_{\mathbf{d}} \,|\, \mathbf{d} \leqslant \mathbf{t}\}$.

**Доказательство.** Рассмотрим пример структуры множества $A(X)$, приведенный на рис. 4.6. Из определения монотонной сети следует, что в $A(X)$ всегда найдется «крайний» алгоритм $a_{\mathbf{t}}$, такой что $A(X) = \{a_{\mathbf{d}} \,|\, \mathbf{d} \leqslant \mathbf{t}\}$. И наоборот, для любого $\mathbf{t}$ легко построить такую выборку $X_{\mathbf{t}}$, для которой $A(X) = \{a_{\mathbf{d}} \,|\, \mathbf{d} \leqslant \mathbf{t}\}$. Следовательно, $\mathfrak{A}(A) = \{\alpha_{\mathbf{t}} : \mathbf{t} \in [0, \ldots, D]^h\}$, и множества $\alpha_{\mathbf{t}}$ устроены так, как утверждается в настоящей лемме. ■

**Лемма 4.13.** *Пусть для произвольного вектора индексов* $\mathbf{t} \geqslant \mathbf{0}$ *множество* $J(\mathbf{t})$ *обозначает множество тех индексов* $j \in \{1, \ldots, h\}$, *для которых* $t_j < D$ *(строго). Тогда для* $\alpha_{\mathbf{t}}$ *множество порождающих и запрещающих множеств устроено следующим образом:*

$$\bar{X}_{\mathbf{t}} = \bigcup_{j \in J(\mathbf{t})} x_j^{t_j+1}, \quad \bar{X}'_{\mathbf{t}} = \bigcup_{j=1}^{h} \bigcup_{i=1}^{t_j} x_j^i.$$

*Построенные таким образом* $\bar{X}_{\mathbf{t}}$ *и* $\bar{X}'_{\mathbf{t}}$ *являются порождающим и запрещающим множеством для* $\alpha_{\mathbf{t}}$.

**Доказательство.** Утверждение леммы следует непосредственно из строения множеств $\bar{X}_{\mathbf{t}}$, $\bar{X}'_{\mathbf{t}}$ и определения РМЭР (3.2). ■

**Теорема 4.14.** *С учетом симметрий монотонной сети вероятность переобучения записывается в виде*

$$Q_\varepsilon(A) = \sum_{\mathbf{d} \in Y_h^D} \sum_{\substack{\mathbf{t} \geqslant \mathbf{d}, \\ \|\mathbf{t}\| \leqslant D}} \frac{|S_h\mathbf{d}|}{V(\mathbf{t})} \frac{C_{L'}^{\ell'}}{C_L^\ell} H_{L'}^{\ell',m}\big(s(\varepsilon)\big), \tag{4.5}$$

*где* $Y_h^D$ *— множество целочисленных неотрицательных невозрастающих последовательностей из* $h$ *элементов, каждый из которых не превосходит* $D$, $|S_h\mathbf{d}|$ *— число различных слов, состоящих из символов* $d_1, \ldots, d_h$.

**Доказательство.** Для доказательства формулы (4.5) необходимо воспользоваться теоремой 3.3 и леммами 4.10, 4.11, 4.12 и 4.13. ■

Расчет по формуле (4.5) требует $O(h \cdot D^h \cdot C_{D+h}^h)$ операций. Оценим, насколько больше операций потребуется, если не учитывать симметрии.

**Теорема 4.15.** *Без учета симметрий вероятность переобучения РМЭР, примененного к монотонной сети* $A = \{a_{\mathbf{d}}\}$ *размерности* $h$, $\|\mathbf{d}\| \leqslant D$, *дается выражением:*

$$Q_\varepsilon(A) = \sum_{\substack{\mathbf{d} \geqslant \mathbf{0}, \\ \|\mathbf{d}\| \leqslant D}} \sum_{\substack{\mathbf{t} \geqslant \mathbf{0}, \\ \|\mathbf{t}\| \leqslant D}} \frac{[\mathbf{t} \geqslant \mathbf{d}]}{V(\mathbf{t})} \frac{C_{L'}^{\ell'}}{C_L^\ell} H_{L'}^{\ell',m}\big(s(\varepsilon)\big), \tag{4.6}$$

*где* $V(\mathbf{t}) = \prod_{j=1}^{h}(t_j+1)$, $\ell' = \ell - \sum_{j=1}^{h}[t_j \neq D]$, $k' = k - |\mathbf{t}|$, $L' = \ell' + k'$, $s(\varepsilon) = \frac{\ell}{L}\big[m + |\mathbf{d}| - \varepsilon k\big]$.

**Доказательство.** Для доказательства формулы (4.6) необходимо воспользоваться теоремой 3.17 и леммами 4.12 и 4.13. ■

Вычисление вероятности переобучения по новой формуле, не учитывающей симметрии, требует $O(h \cdot D^{2h})$ операций. Рассмотрим отношение $D^h/C^h_{h+D}$, показывающее, во сколько раз увеличился объем вычислений. Данная величина максимальна при $D \gg h$. Это соответствует случаю сетей большой длины, на которых группа симметрии действует наиболее эффективно. В этом случае учет симметрий дает выигрыш в $h!$ раз, что в точности соответствует количеству элементов в группе симметрий. В остальных случаях (сети больших размерностей и малой длины) выигрыш оказывается меньше.

### 4.2.3 Многомерная унимодальная сеть алгоритмов

Унимодальная сеть является более реалистичной моделью связного параметрического семейства по сравнению с монотонной сетью. Если мы имеем лучший алгоритм $a_0$ с оптимальным значением вектора вещественных параметров, то отклонение значений компонент этого вектора как в бóльшую, так и в меньшую сторону приводит к увеличению числа ошибок.

**Определение 4.6.** *Множество алгоритмов* $A = \{a_{\mathbf{d}}\}$, *где* $\|\mathbf{d}\| \leqslant D$, *называется унимодальной* $h$*-мерной сетью алгоритмов, если существует* $h \in \mathbb{N}$ *и упорядоченные наборы объектов* $X_j = \{x_j^1, x_j^2, \ldots, x_j^D\} \subset \mathbb{X}$, $Y_j = \{y_j^1, y_j^2, \ldots, y_j^D\} \subset \mathbb{X}$, *для всех* $j = 1, \ldots, h$, *а также множества* $U_1 \subset \mathbb{X}$ *и* $U_0 \subset \mathbb{X}$, *такие что выполнены условия:*

1) *Набор* $\big\{U_0, U_1, \{X_j\}_{j=1}^h, \{Y_j\}_{j=1}^h\big\}$ *является разбиением множества* $\mathbb{X}$ *на непересекающиеся множества;*

2) $a_d(x_j^i) = [d_j > 0]\big[i \leqslant |d_j|\big]$, *где* $x_j^i \in X_j$;

3) $a_d(y_j^i) = [d_j < 0]\big[i \leqslant |d_j|\big]$, *где* $y_j^i \in Y_j$;

4) $a_d(x_0) = 0$ *при всех* $x_0 \in U_0$;

5) $a_d(x_1) = 1$ *при всех* $x_1 \in U_1$.

Заметим, что данное определение отличается от определения монотонной сети отсутствием ограничения $\mathbf{d} \geqslant 0$. Число алгоритмов в $h$-мерной унимодальной сети с ветвями

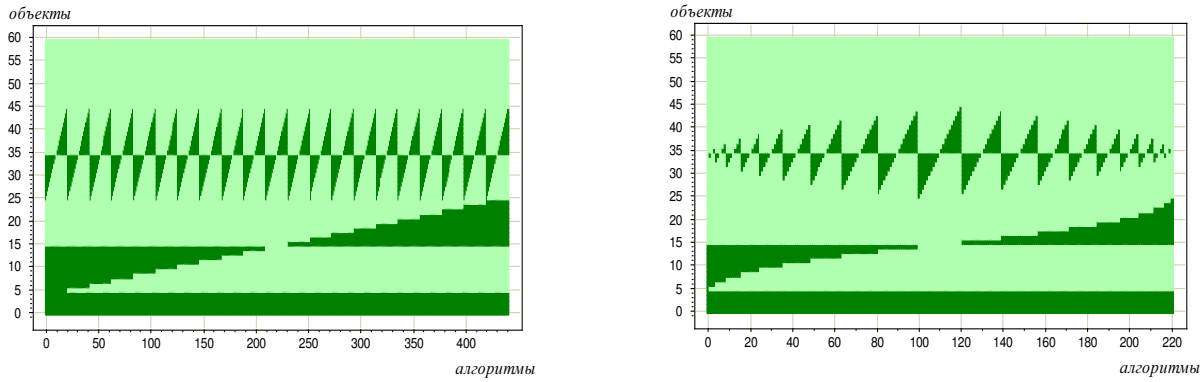



Рис. 4.7: Матрица ошибок унимодальной сети (слева) и укороченной унимодальной сети (справа) при $D = 10$, $h = 2$, $m = 5$, $L = 60$.

длины $D$ составляет $(2D + 1)^h$. *Укороченной* $h$-мерной унимодальной сетью $\tilde{A}$ назовем множество первых $D$ слоев из $A$:

$$\tilde{A} = \{a_{\mathbf{d}} \in A \colon n(a_{\mathbf{d}}, \mathbb{X}) \leqslant m + D\}.$$

На рис. 4.7 показаны примеры матрицы ошибок унимодальных сетей.

Формула для вероятности переобучения *пессимистического* метода минимизации эмпирического риска на укороченных унимодальных сетях также была получена в [2]. Ниже рассматриваются неукороченные унимодальные сети и случай рандомизированного МЭР.

**Лемма 4.16.** *Группа симметрии унимодальной сети размерности $h$ содержит в качестве подгруппы группу* $\mathrm{Sym}(A) = (S_2)^h \times S_h$. *Группа $S_h$ действует на множестве пар $(X_j, Y_j)_{j=1}^h$ всеми возможными перестановками; $j$-тая группа $S_2$ переставляет объекты множества $X_j$ и $Y_j$ местами, сохраняя относительный порядок объектов.*

**Доказательство.** Заметим, что все алгоритмы $h$-мерной унимодальной сети длины $D$ индексированы множеством вектор-индексов $\mathbf{d} \in \{-D, \ldots, D\}^h$. При этом число ошибок алгоритма $a_{\mathbf{d}}$ равно $n(a_{\mathbf{d}}, \mathbb{X}) = m + |\mathbf{d}|$.

Рассмотрим алгоритм $a_{\mathbf{d}} \in A$ и произвольную $\pi = (z_1, \ldots, z_h) \times \pi_0 \in \mathrm{Sym}(A)$, где $z_j \in S_2$, $\pi_0 \in S_h$. По данному выше определению действия $\pi$ на $\mathbb{X}$ получаем, что $\pi a_{\mathbf{d}} = a_{\pi \mathbf{d}}$, где действие $\pi$ на вектор $\mathbf{d}$ определяется перестановкой его координат с помощью $\pi_0$ и инверсией знаков для всех $j$, таких что $z_j \neq id$ — транспозиция. Множество $\{-D, \ldots, D\}^h$ сохраняется при применении к нему произвольной перестановки координат $\pi \in (S_2)^h \times S_h$. Поэтому $\forall \mathbf{d} \in \{-D, \ldots, D\}^h$ выполнено $\pi \mathbf{d} \in \{-D, \ldots, D\}^h$, следовательно, $a_{\pi \mathbf{d}} \in A$. ■

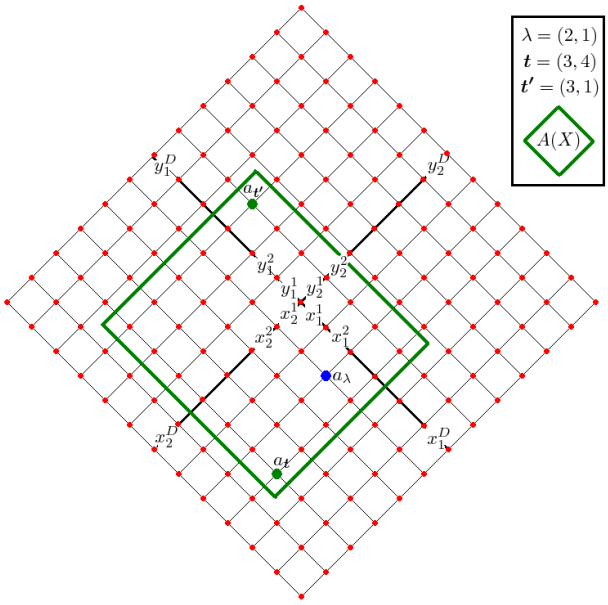



Рис. 4.8: Строение множества $A(X)$ для двумерной унимодальной сети.

**Лемма 4.17.** *Пусть $A = \{a_{\mathbf{d}}\}$, $\|d\| \leqslant D$ — унимодальная сеть длины $D$ и размерности $h$. Тогда множество орбит $A$ под действием $\mathrm{Sym}(A)$ индексировано всевозможными элементами из $Y_h^D$. Пусть $\lambda = (\lambda_1, \ldots, \lambda_h) \in Y_h^D$. Обозначим через $|S_h\lambda|$ число различных слов длины $h$, состоящих из символов $\lambda_1, \ldots, \lambda_h$. Пусть $|\lambda > 0|$ — число строго положительных компонент вектора $\lambda$.*

*Тогда число алгоритмов в орбите $\omega_\lambda$ равно $|S_h\lambda| \cdot 2^{|\lambda>0|}$.*

**Доказательство.** Доказательство полностью повторяет рассуждения леммы 4.11. Множитель $2^{|\lambda>0|}$ соответствует возможности сменить знак у всех ненулевых компонент вектора $\mathbf{d}$. ∎

**Теорема 4.18.** *Вероятность переобучения РМЭР, примененного к унимодальной сети $A = \{a_{\mathbf{d}}\}$ размерности $h$, $\|d\| \leqslant D$, дается выражением:*

$$Q_\varepsilon(A) = \sum_{\mathbf{d} \in Y_h^D} \sum_{\substack{\mathbf{t} \geqslant \mathbf{d}, \\ \|\mathbf{t}\| \leqslant D}} \sum_{\substack{\mathbf{t}' \geqslant 0, \\ \|\mathbf{t}'\| \leqslant D}} \frac{|S_h\mathbf{d}\,| \cdot 2^{|\mathbf{d}>0|}}{T(\mathbf{t}+\mathbf{t}')} \frac{C_{L'}^{\ell'}}{C_L^\ell} H_{L'}^{\ell',m}(s_0), \tag{4.7}$$

*где $\ell' = \ell - \sum_{j=1}^h \big([t_j \neq D] + [t'_j \neq D]\big)$, $k' = k - |\mathbf{t}| - |\mathbf{t}'|$, а остальные обозначения совпадают с обозначениями теоремы 4.14.*

**Доказательство.** Напомним, что через $\mathfrak{A}(A) = \big\{A(X) \colon X \in [\mathbb{X}]^\ell\big\}$ обозначалось множество подмножеств алгоритмов, получающихся в результате обучения. Пусть $A$ — унимодальная сеть. Тогда для произвольной обучающей выборки $X$ множество $A(X)$ устроено специфи-

ческим образом. На рис. 4.8 показано, что в $A(X)$ всегда найдется такая пара алгоритмов $(a_{\mathbf{t}_1}, a_{\mathbf{t}_2})$, что

$$A(X) = \{a_{\mathbf{d}} \,|\, \mathbf{t}_1 \leqslant \mathbf{d} \leqslant \mathbf{t}_2\}.$$

Следовательно, $\mathfrak{A}(A) = \{\alpha_{\mathbf{t},\mathbf{t}'} : \mathbf{t}, \mathbf{t}' \in [0, \ldots, D]^h\}$.

Обозначим через $J(\mathbf{t})$ множество тех индексов $j \in \{1, \ldots, h\}$, для которых $t_j < D$. Положим

$$X_{\mathbf{t}} = \bigcup_{j \in J(\mathbf{t})} x_j^{t_j+1}, \quad X'_{\mathbf{t}} = \bigcup_{j=1}^{h} \bigcup_{i=1}^{t_j} x_j^i;$$

$$Y_{\mathbf{t}'} = \bigcup_{j \in J(\mathbf{t}')} y_j^{t'_j+1}, \quad Y'_{\mathbf{t}'} = \bigcup_{j=1}^{h} \bigcup_{i=1}^{t'_j} y_j^i.$$

Множества $X_{\mathbf{t}} \cup Y_{\mathbf{t}'}$ и $X'_{\mathbf{t}} \cup Y'_{\mathbf{t}'}$ являются, соответственно, порождающим и запрещающим множествами для $\alpha_{\mathbf{t},\mathbf{t}'}$. Применив теорему о порождающих и запрещающих множествах, получим формулу (4.7). ■

### 4.2.4 Разреженные монотонные и унимодальные сети

В предыдущих параграфах рассматривались семейства алгоритмов, реализующиеся при непрерывном изменении компонент вектора вещественных параметров. На практике возможны ситуации, в которых наблюдаемое семейство будет собственным подмножеством рассмотренных выше монотонных и унимодальных сетей. В данном параграфе рассматриваются только такие подмножества, в которых наложено ограничение на минимальное расстояние между ближайшими алгоритмами в семействе. Эти случаи соответствуют изменению каждой компоненты вектора вещественных параметров с постоянным шагом.

**Определение 4.7.** *Пусть $\rho \in \mathbb{N}$ — целочисленный параметр; $A = \{a_{\mathbf{d}}\}$ — $h$-мерная монотонная сеть длины $\rho D$; $m \equiv n(a_0, \mathbb{X})$. Разреженной $h$-мерной монотонной сетью $\ddot{A}$ плотности $\rho$ и длины $D$ будем называть подмножество $A$, заданное условием:*

$$\ddot{A} = \left\{a_{\mathbf{d}} \in A \,\middle|\, \mathbf{d} \in (\rho\mathbb{Z})^h\right\}.$$

Отметим, что при $\rho > 1$ граф смежности разреженной монотонной сети состоит из изолированных точек.

**Пример 4.3.** На рис. 4.9 выделено подмножество двумерной монотонной сети с параметром $D = 8$, соответствующее разреженной монотонной сети с параметрами $\rho = 2$, $D = 4$.

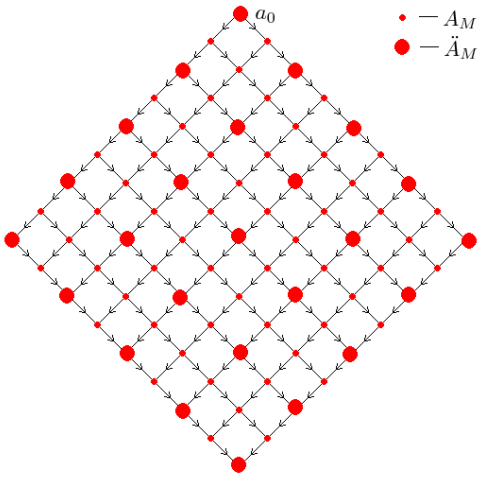



Рис. 4.9: Двумерная разреженная монотонная сеть при $\rho = 2$, $D = 4$.

Заметим, что для разреженной монотонной сети не удается выписать систему порождающих и запрещающих множеств непосредственно в виде (3.20). При этом структура вывода формулы вероятности переобучения сохраняется прежней — вначале используется теорема 3.6 о разложении вероятности переобучения по орбитам действия группы симметрии, затем определяется структура множества $\mathfrak{A}(A)$ и, после подсчета числа элементов в получившихся множествах, выписывается финальная оценка.

**Теорема 4.19.** *Вероятность переобучения РМЭР, примененного к разреженной монотонной сети $\ddot{A}_M = \{a_{\mathbf{d}}\}$ размерности $h$, $\|d\| \leqslant D$, дается выражением:*

$$Q_\varepsilon(\ddot{A}_M) = \sum_{\lambda \in Y_*^{h,D}} \sum_{\substack{\mathbf{t} \geqslant \rho\lambda, \\ \|\mathbf{t}\| \leqslant \rho D}} \frac{|S_h\lambda|}{T\big(\lfloor \mathbf{t}/\rho \rfloor\big)} \frac{C_{L'}^{\ell'}}{C_L^\ell} H_{L'}^{\ell',m}(s_0), \tag{4.8}$$

*где $Y_h^D$ определено в теореме 4.14, $|S_h\lambda|$ — мощность орбиты действия симметрической группы $S_h$ на $\lambda$, $T(\mathbf{t}) = \prod_j (t_j + 1)$, $\ell' = \ell - \sum_{j=1}^h [t_j \neq \rho D]$, $k' = k - |\mathbf{t}|$, $L' = \ell' + k'$, $s_0 = \frac{\ell}{L}[m + \rho|\lambda| - \varepsilon k]$, $H_{L'}^{\ell',m}(s)$ — функция гипергеометрического распределения.*

**Доказательство.** Воспользуемся теоремой 3.6 о разложении вероятности переобучения по орбитам множества алгоритмов:

$$Q_\varepsilon(A_M) = \frac{1}{C_L^\ell} \sum_{\lambda \in Y_*^{h,D}} |S_h\lambda| \sum_{X \in [\mathbb{X}]^\ell} \frac{[a_\lambda \in A_M(X)]}{|A_M(X)|} \left[\delta(a_\lambda, X) \geqslant \varepsilon\right].$$

**Шаг 1.** Зафиксируем $X \in [\mathbb{X}]^\ell$. Обозначим через $t_j$ максимальный индекс из $\{0, \ldots, \rho D\}$, при котором все объекты $\{x_j^1, \ldots, x_j^{t_j}\}$ содержатся в $\bar{X}$, а $x_j^{t_j+1}$, при его наличии, лежит в $X$. Положим $\mathbf{t} = \{t_j\}_{j=1}^h$. Тогда условие $a_\lambda \in A_M(X)$ перепишется как $\mathbf{t} \geqslant \rho\lambda$.

Действительно, заметим, что для всех $a \in A_M$ и $X \in [\mathbb{X}]^\ell$ выполнено $n(a,X) \geqslant n(a_0,X)$. Следовательно, алгоритм $a_\lambda$ может быть выбран, только если объекты $x_j^i$ при всех $j = 1,\ldots,h$ и $i \leqslant \rho\lambda_j$ лежат в контроле. В терминах $\mathbf{t}$ это записывается как $\mathbf{t} \geqslant \rho\lambda$.

Обозначим множество разбиений на обучение и контроль с фиксированным значением параметра $\mathbf{t}$ через $[\mathbb{X}]^\ell_{\mathbf{t}}$. Тогда

$$Q_\varepsilon(A_M) = \frac{1}{C_L^\ell} \sum_{\lambda \in Y_h^D} |S_h\lambda| \sum_{\substack{\mathbf{t}\geqslant\rho\lambda,\\ \|\mathbf{t}\|\leqslant\rho D}} \sum_{X\in[\mathbb{X}]^\ell_{\mathbf{t}}} \frac{1}{|A_M(X)|} \left[\delta(a_\lambda, X) \geqslant \varepsilon\right].$$

**Шаг 2.** Пусть $X \in [\mathbb{X}]^\ell_{\mathbf{t}}$. Заметим, что алгоритм $a_{\mathbf{d}} \in A_M(X)$ тогда и только тогда, когда $\rho\mathbf{d} \leqslant \mathbf{t}$. Следовательно,

$$|A_M(X)| = (\lfloor t_1/\rho\rfloor+1)(\lfloor t_2/\rho\rfloor+1)\ldots(\lfloor t_h/\rho\rfloor+1).$$

Обозначим $T(\mathbf{v}) = \prod_j (v_j+1)$. Тогда $|A(X)| = T\big(\lfloor \mathbf{t}/\rho\rfloor\big)$.

**Шаг 3.** Обозначим через $s = |U_1 \cap X|$ число объектов из $U_1$, лежащих в обучении. Тогда $\delta(a_\lambda, X) = \frac{m-s+\rho|\lambda|}{k} - \frac{s}{\ell}$, и условие $\delta(a_\lambda, X) \geqslant \varepsilon$ запишется в виде $s \leqslant \frac{\ell}{L}[m + \rho|\lambda| - \varepsilon k] \equiv s_0$. Множество всех разбиений из $[\mathbb{X}]^\ell_{\mathbf{t}}$ с фиксированным параметром $s$ обозначим через $[\mathbb{X}]^\ell_{\mathbf{t},s}$. Тогда

$$Q_\varepsilon(A_M) = \frac{1}{C_L^\ell} \sum_{\lambda \in Y_h^D} |S_h\lambda| \sum_{\substack{\mathbf{t}\geqslant\rho\lambda,\\ \|\mathbf{t}\|\leqslant\rho D}} \frac{1}{T\big(\lfloor \mathbf{t}/\rho\rfloor\big)} \sum_{s=0}^{s_0} \big|[\mathbb{X}]^\ell_{\mathbf{t},s}\big|.$$

**Шаг 4.** Вычислим мощность множества $[\mathbb{X}]^\ell_{\mathbf{t},s}$.

Введем обозначения $\ell' = \ell - \sum_{j=1}^h [t_j \neq \rho D]$, $k' = k - |\mathbf{t}|$, $L' = \ell' + k'$. Тогда простое комбинаторное вычисление показывает, что $\big|[\mathbb{X}]^\ell_{\mathbf{t},s}\big| = C_m^s C_{L'-m}^{k'-s}$. Следовательно,

$$Q_\varepsilon(A_M) = \frac{1}{C_L^\ell} \sum_{\lambda \in Y_h^D} |S_h\lambda| \sum_{\substack{\mathbf{t}\geqslant\rho\lambda,\\ \|\mathbf{t}\|\leqslant\rho D}} \frac{1}{T\big(\lfloor \mathbf{t}/\rho\rfloor\big)} \sum_{s=0}^{s_0} C_m^s C_{L'-m}^{k'-s}.$$

Напомним, что $H_{L'}^{\ell',m}(z) = \sum\limits_{s=0}^{\lfloor z\rfloor} \frac{C_m^s C_{L'-m}^{\ell'-s}}{C_{L'}^{\ell'}}$ — функция гипергеометрического распределения [8]. Тогда

$$Q_\varepsilon(A_M) = \sum_{\lambda \in Y_h^D} \sum_{\substack{\mathbf{t}\geqslant\rho\lambda,\\ \|\mathbf{t}\|\leqslant\rho D}} \frac{|S_h\lambda|}{T\big(\lfloor \mathbf{t}/\rho\rfloor\big)} \frac{C_{L'}^{\ell'}}{C_L^\ell} H_{L'}^{\ell',m}(s_0).$$

■

**Определение 4.8.** *Пусть $\rho \in \mathbb{N}$ — целочисленный параметр; $A = \{a_{\mathbf{d}}\}$, где $\|d\| \leqslant \rho D$ — $h$-мерная унимодальная сеть; $m \equiv n(a_0, \mathbb{X})$. Разреженной $h$-мерной унимодальной сетью*

*$\ddot{A}$ плотности $\rho$ будем называть следующее подмножество $A$:*

$$\ddot{A} = \left\{ a_{\mathbf{d}} \in A \,\middle|\, \mathbf{d} \in (\rho\mathbb{Z})^h \right\}.$$

**Теорема 4.20.** *Вероятность переобучения РМЭР, примененного к разреженной унимодальной сети $\ddot{A} = \{a_{\mathbf{d}}\}$ размерности $h$, $\|d\| \leqslant D$, дается выражением:*

$$\begin{aligned} Q_\varepsilon(A) &= \sum_{\lambda \in Y_h^D} \sum_{\substack{\mathbf{t} \geqslant \rho\lambda, \\ \|\mathbf{t}\| \leqslant \rho D}} \sum_{\substack{\mathbf{t}' \geqslant 0, \\ \|\mathbf{t}'\| \leqslant \rho D}} \mathbb{S}(\lambda, \mathbf{t}, \mathbf{t}'); \\ \mathbb{S}(\lambda, \mathbf{t}, \mathbf{t}') &= \frac{|S_h\lambda| \cdot 2^{|\lambda>0|}}{T\big(\lfloor \mathbf{t}/\rho \rfloor + \lfloor \mathbf{t}'/\rho \rfloor\big)} \frac{C_{L'}^{\ell'}}{C_L^\ell} H_{L'}^{\ell',m}(s_0), \end{aligned} \tag{4.9}$$

*где $\ell' = \ell - \sum_{j=1}^h \big([t_j \neq \rho D] + [t'_j \neq \rho D]\big)$, $k' = k - |\mathbf{t}| - |\mathbf{t}'|$, а остальные обозначения совпадают с обозначениями теоремы 4.19.*

**Доказательство.**

**Шаг 1.** Выберем в качестве представителя $a_\lambda$ орбиты $\omega_\lambda$ алгоритм, не допускающий ошибок на множестве $Y = \bigcup_{j=1}^h Y_j$. Этого можно добиться, взяв произвольный $a_{\mathbf{d}} \in \omega_\lambda$ и поменяв знаки у всех $d_j < 0$ с помощью транспозиции $z_j$.

Введя обозначения $\mathbf{t}$ и $[\mathbb{X}]_{\mathbf{t}}^\ell$ так же, как и на первом шаге вывода формулы для монотонной сети, получим

$$Q_\varepsilon(\ddot{A}) = \frac{1}{C_L^\ell} \sum_{\lambda \in Y_h^D} |S_h\lambda| \cdot 2^{|\lambda>0|} \sum_{\substack{\mathbf{t} \geqslant \rho\lambda, \\ \|\mathbf{t}\| \leqslant \rho D}} \sum_{X \in [\mathbb{X}]_{\mathbf{t}}^\ell} \frac{1}{|\ddot{A}(X)|} \left[\delta(a_\lambda, X) \geqslant \varepsilon\right].$$

**Шаг 2.** Обозначим с помощью $t'_j$ максимальный индекс из $\{0, \ldots, \rho D\}$, при котором все объекты $\{y_j^1, \ldots, y_j^{t'_j}\}$ содержатся в $\bar{X}$, а $y_j^{t'_j+1}$, при его наличии, лежит в $X$. Положим $\mathbf{t}' = \{t'_j\}_{j=1}^h$. Заметим, что вектор $\mathbf{t}'$ играет для набора $\{Y_j\}$ ту же роль, что $\mathbf{t}$ для $\{X_j\}$. Обозначим через $[\mathbb{X}]_{\mathbf{t},\mathbf{t}'}^\ell$ множество разбиений с фиксированными параметрами $\mathbf{t}$ и $\mathbf{t}'$.

Пусть $X \in [\mathbb{X}]_{\mathbf{t},\mathbf{t}'}^\ell$. Заметим, что $[a_{\mathbf{d}} \in \ddot{A}(X)] = [-\mathbf{t}' \leqslant \rho\mathbf{d} \leqslant \mathbf{t}]$. Следовательно, $|\ddot{A}(X)| = T\big(\lfloor \mathbf{t}/\rho \rfloor + \lfloor \mathbf{t}'/\rho \rfloor\big)$.

**Шаг 3.** Обозначим через $s = |U_1 \cap X|$ число объектов из $U_1$, лежащих в обучении. Пусть $s_0 \equiv \frac{\ell}{L}[m + \rho|\lambda| - \varepsilon k]$. Повторяя рассуждения аналогичного шага теоремы 4.19, получим

$$Q_\varepsilon(\ddot{A}) = \frac{1}{C_L^\ell} \sum_{\lambda \in Y_h^D} |S_h\lambda| \cdot 2^{|\lambda>0|} \sum_{\substack{\mathbf{t} \geqslant \lambda, \\ \|\mathbf{t}\| \leqslant D}} \sum_{\substack{\mathbf{t}' \geqslant 0, \\ \|\mathbf{t}'\| \leqslant D}} \frac{1}{T(\mathbf{t}, \mathbf{t}')} \sum_{s=0}^{s_0} \big|[\mathbb{X}]_{\mathbf{t},\mathbf{t}',s}^\ell\big|.$$

**Шаг 4.** Посчитаем мощность множества $[\mathbb{X}]_{\mathbf{t},\mathbf{t}',s}^\ell$.


Рис. 4.10: Зависимость $Q_\varepsilon(\ddot{A})$ от разреженности $\rho$ монотонной сети при $L = 150$, $\ell = 90$, $\varepsilon = 0.05$, $D = 3$, $m = 5$, $h = 1, 2, 3, 4$.


Рис. 4.11: Зависимости вероятности переобучения $Q_\varepsilon(\ddot{A})$ для разреженной монотонной и унимодальной сетей от $\rho$ при $L = 150$, $\ell = 90$, $\varepsilon = 0.05$, $D = 3$, $m = 5$, $h = 1(2), 2(4)$.

Обозначим $\ell' = \ell - \sum_{j=1}^{h}\big([t_j \neq \rho D] + [t'_j \neq \rho D]\big)$, $k' = k - |\mathbf{t}| - |\mathbf{t}'|$, $L' = \ell' + k'$. Тогда $\big|[\mathbb{X}]^{\ell}_{\mathbf{t},\mathbf{t}',s}\big| = C^s_m C^{k'-s}_{L'-m}$. Воспользовавшись определением функции гипергеометрического распределения, получим:

$$Q_\varepsilon(A) = \sum_{\lambda\in Y_h^D}\sum_{\substack{\mathbf{t}\geqslant\rho\lambda,\\ \|\mathbf{t}\|\leqslant\rho D}}\sum_{\substack{\mathbf{t}'\geqslant 0,\\ \|\mathbf{t}'\|\leqslant\rho D}} \frac{|S_h\lambda|\cdot 2^{|\lambda>0|}}{T\big(\lfloor\mathbf{t}/\rho\rfloor + \lfloor\mathbf{t}'/\rho\rfloor\big)}\frac{C^{\ell'}_{L'}}{C^{\ell}_{L}} H^{\ell',m}_{L'}(s_0).$$

■

Приведем результаты численных расчетов, иллюстрирующих поведение вероятности переобучения монотонной и унимодальной разреженных сетей. Расчеты выполнены с помощью доказанных выше формул (4.8), (4.9).

На рис. 4.10 изображена зависимость вероятности переобучения $h$-мерной монотонной сети от разреженности $\rho$ (при $h = 1, 2, 3, 4$). При увеличении размерности вероятность переобучения возрастает. При увеличении разреженности $\rho$ вероятность переобучения падает и вскоре выходит на константу, соответствующую вероятности переобучения лучшего алгоритма семейства $a_0$. Это связано с тем, что с уменьшением плотности семейства возрастает роль *явления расслоения* [9, 8].

На рис. 4.11 приведены результаты сравнения разреженных $h$-мерных унимодальных сетей с разреженными $2h$-мерными монотонными сетями при $h = 1$ и $h = 2$. Тонкая серая кривая соответствует вероятности переобучения для унимодальной сети. Полученные результаты подтверждают гипотезу [2] о связи вероятности переобучения для унимодальных сетей с вероятностью переобучения монотонных сетей удвоенной размерности.

## 4.3 Плотные семейства

Во всех предыдущих параграфах рассматривались модельные семейства алгоритмов, в которых число алгоритмов увеличивалось полиномиально по параметру $h$. В данном параграфе будут рассмотрены более плотные модельные семейства алгоритмов, не обладающие параметром размерности. В дальнейшем эти семейства будут использованы в качестве объемлющих множеств $B$ (см. лемму 3.13) для эффективного вычисления оценки (3.16).

### 4.3.1 Слой хэммингова шара

Напомним, что расстояние между алгоритмами $\rho(a, a')$ определялось как расстояние Хэмминга между их векторами ошибок:

$$\rho(a, a') = \sum_{x \in \mathbb{X}} |I(a, x) - I(a', x)|.$$

**Определение 4.9.** *Центральным слоем хэммингова шара называется множество*

$$B_r(a_0) = \{a \in \{0,1\}^L \colon n(a, \mathbb{X}) = n(a_0, \mathbb{X}),\ \text{и } \rho(a, a_0) \leqslant r\}.$$

Данное множество получается сечением *шара алгоритмов* $\{a \in \{0,1\}^L \colon \rho(a, a_0) \leqslant r\}$ и *слоя алгоритмов* $A_m = \{a \in \{0,1\}^L \colon n(a, \mathbb{X}) = m\}$ при $m = n(a_0, \mathbb{X})$.

Центральный слой хэммингова шара алгоритмов не имеет явного аналога среди реальных семейств алгоритмов. Однако изучение этого модельного семейства представляет значительный теоретический интерес, поскольку слой хэммингова шара является максимально связным множеством булевых векторов. Это делает его привлекательным примером для изучения влияния эффекта сходства на вероятность переобучения.

Впервые шар алгоритмов и его центральный слой были изучены в работе [30]. В частности, были доказаны леммы о группе симметрий данных множеств и о структуре орбит. Кроме этого, с помощью теоремы 3.7 были получены точные формулы вероятности переобучения РМЭР для этих семейств. В настоящей работе приводится более простой вариант доказательства формулы вероятности переобучения центрального слоя шара, основанный на теореме 3.10.

Положим без ограничения общности, что алгоритм $a_0$ ошибается на первых $m = n(a_0, \mathbb{X})$ объектах генеральной выборки $\mathbb{X}$. В дальнейшем множество, состоящее из пер-

вых $m$ объектов $\mathbb{X}$, мы будем обозначать $X^m$. Множество, состоящее из последних $L-m$ объектов будем обозначать $X^{L-m}$.

**Лемма 4.21 (Толстихин, [30]).** *Группа $S_m \times S_{L-m}$, где $S_m$ и $S_{L_m}$ — симметрические группы перестановок, действующие на множествах $X^m$ и $X^{L-m}$, соответственно, является подгруппой группы симметрий множества алгоритмов $B_r(a_0)$.*

**Лемма 4.22 (Толстихин, [30]).** *Орбиты $\tau \in \Omega([\mathbb{X}]^\ell)$ действия группы $S_m \times S_{L-m}$ на множестве $[\mathbb{X}]^\ell$ индексированы параметром $i = |X \cap X^m| = n(a_0, X^\ell)$ — числом ошибок алгоритма $a_0$ на обучении. Мощность орбиты $\tau_i$ записывается в виде $|\tau_i| = C_L^\ell h_L^{\ell,m}(i)$.*

**Теорема 4.23.** *Вероятность переобучения множества алгоритмов, получаемого сечением шара алгоритмов центральным $m$-слоем, дается в виде*

$$Q_\mu(\varepsilon, A) = H_L^{\ell,m}\big(\frac{\ell}{L}(m - \varepsilon k) + \lfloor r/2 \rfloor\big) \cdot [m \geqslant \varepsilon k],$$

*$H_L^{\ell,m}(s)$ — функция гипергеометрического распределения.*

**Доказательство.** Заметим, что утверждение лемм 4.21 и 4.22 верно и для сечения шара центральной плоскостью. Поскольку все алгоритмы имеют равное число ошибок на полной выборке, применим следствие 3.1 из теоремы о разложении вероятности переобучения по орбитам разбиений выборки:

$$Q_\mu(\varepsilon, A) = \sum_{i=0}^{m} h_L^{\ell,m}(i) \left[\min_{a \in A} n(a, X_i) \leqslant \frac{\ell}{L}(m - \varepsilon k)\right].$$

Напомним, что по определению $i = |X \cap X^m|$. Пусть $r' = \left\lfloor \frac{r}{2} \right\rfloor$. Тогда выполнено следующее утверждение:

$$\min_{a \in A} n(a, X_i) = \begin{cases} 0, \text{ при } i \leqslant r', \\ i - r', \text{ при } i > r'. \end{cases}$$

Следовательно

$$Q_\mu(\varepsilon, A) = \sum_{i=0}^{\lfloor s_d(\varepsilon) \rfloor + r'} h_L^{\ell,m}(i) = H_L^{\ell,m}\big(\frac{\ell}{L}(m - \varepsilon k) + \lfloor r/2 \rfloor\big) \cdot [m \geqslant \varepsilon k],$$

■

### 4.3.2 Слой интервала булева куба

Центральный слой хэммингова шара $B_r(a_0)$ можно использовать в качестве объемлющего множества для вычисления вероятности переобучения каждого элемента разложения

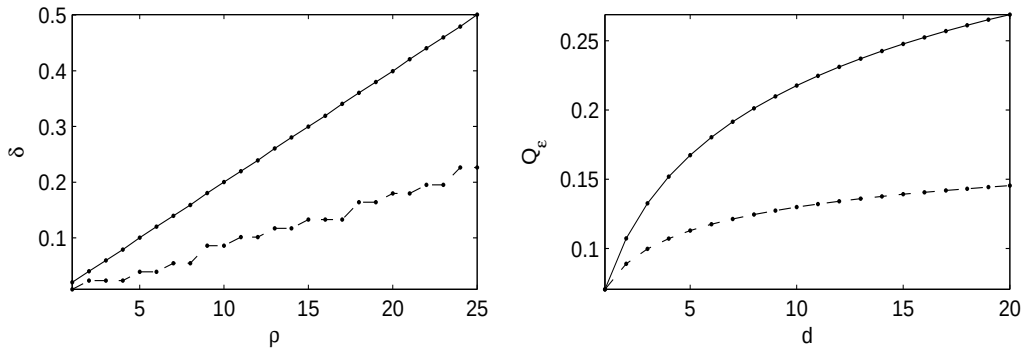



Рис. 4.12: Переобучение центрального слоя шара $B^m_{2\rho}$ (сплошная кривая) и слоя интервала $\hat{B}^{m-\rho}_{2\rho,\rho}$ (пунктирная кривая) при $L = 200$, $\ell = 100$, $m = 50$. Рисунок слева отображает среднее уклонение частот ошибок на обучении и контроле в зависимости от параметра $\rho$. Рисунок справа отображает зависимость средней вероятности переобучения $\bar{Q}_\varepsilon(B, d)$ от параметра $d$ при $\rho = 5$, $\varepsilon = 0.1$.

$A = A_1 \sqcup \cdots \sqcup A_t$ оценки 3.16. Отметим, что оценка $Q_\varepsilon(A_i) \leqslant Q_\varepsilon(B_r(a_0))$ является сильно завышенной. Это связано с тем, что аппроксимация множества $A_i$ с помощью центрального слоя хэммингова шара не позволяет учесть объекты выборки, лежащие глубоко внутри своего класса и одинаково классифицируемые всеми алгоритмами кластера $A_i$. Следующее модельное множество исправляет этот недостаток.

**Определение 4.10.** *Пусть все объекты генеральной выборки $\mathbb{X}$ разделены на три непересекающихся множества: надежно классифицируемые объекты $X_0$, ошибочно классифицируемые объекты $X_1$ и пограничные объекты $X_r$. Пусть $|X_r| = r$ и $|X_1| = m$, $\rho \in \mathbb{N}$ — параметр, $\rho \leqslant r$. Рассмотрим алгоритм $a_0$, допускающий $m$ ошибок на $X_1$ и $\rho$ ошибок на $X_r$. Слоем интервала булева куба будем называть множество алгоритмов $\hat{B}^m_{r,\rho} \subset \mathbb{A}$, удовлетворяющее следующим условиям:*

- *$\hat{B}^m_{r,\rho}$ содержит все алгоритмы, допускающие ровно $\rho$ ошибок на объектах из $X_r$;*
- *ни один алгоритм из $\hat{B}^m_{r,\rho}$ не ошибается на объектах из $X_0$;*
- *все алгоритмы из $\hat{B}^m_{r,\rho}$ ошибаются на всех объектах из $X_1$.*

На рис. 4.12 слева сравниваются вероятности переобучения центрального слоя хэммингова шара и слоя интервала булева куба. Видно, что слой интервала дает меньшую оценку вероятности переобучения. Следовательно, аппроксимация кластера $A_i$ с помощью слоя интервала булева куба дает более точную оценку вероятности переобучения.

**Теорема 4.24.** *Вероятность переобучения РМЭР для $\hat{B}^m_{r,\rho}$ дается следующей формулой:*

$$Q_\varepsilon(\hat{B}^m_{r,\rho}) = \frac{1}{C^\ell_L} \sum_{i=0}^{\min(m,\ell)} \sum_{j=0}^{\min(r,\ell-i)} C^i_m C^j_r C^{\ell-i-j}_{L-m-r} \Big[ \frac{m+\rho-t(i,j)}{k} - \frac{t(i,j)}{\ell} \geqslant \varepsilon \Big], \tag{4.10}$$

*где* $t(i,j) = i + \max(0, \rho - r - j)$.

**Доказательство.** Рассмотрим три симметрические группы перестановок $S_m$, $S_r$ и $S_{L-m-r}$, действующие на множествах $X_1$, $X_r$ и $X_0$, соответственно. Группой симметрий множества алгоритмов $\hat{B}^m_{r,\rho}$ является декартово произведение $S_m \times S_r \times S_{L-m-r}$. Орбиты действия $\mathrm{Sym}(\hat{B}^m_{r,\rho})$ на $[\mathbb{X}]^\ell$ индексируются двумя параметрами, $i = |X \cap X_1|$ и $j = |X \cap X_r|$, где $X$ — обучающая выборка. Мощность орбиты $\tau_{i,j}$ дается, соответственно, выражением $|\tau_{i,j}| = C^i_m C^j_r C^{\ell-i-j}_{L-m-r}$.

Поскольку все алгоритмы множества $B^m_{r,\rho}$ допускают равное число ошибок, то для каждой выборки $X \in [\mathbb{X}]^\ell$ все алгоритмы из $A(X)$ имеют одинаковое число ошибок и на обучении, и на контроле. Легко подсчитать, что для произвольной выборки $X \in \tau_{i,j}$ и $a \in A(X)$ выполнено $n(a,X) = i + \max(0, \rho - r - j)$ и $n(a,\bar{X}) = m + \rho - n(a,X)$. Для доказательства теоремы осталось подставить эти значения и мощность орбиты $|\tau_{i,j}|$ в теорему 3.10 о разложение вероятности переобучения по орбитам действия группы симметрии на разбиениях выборки. ■

## 4.4 Основные выводы

В данной главе получены точные оценки вероятности переобучения РМЭР для девяти модельных семейств алгоритмов. При выводе оценок используется разработанный выше математический инструментарий: разложение вероятности переобучения по орбитам действия группы симметрий на множестве алгоритмов или на множестве разбиений выборки, а также теорема о порождающих и запрещающих множествах для РМЭР.

# Глава 5. Вычислительные эксперименты на реальных данных

## 5.1 Эффективное вычисление SC-оценки

В данном параграфе подробно разбирается вопрос о вычислении SC-оценки (2.10) по известной матрице ошибок алгоритмов $A$. Вопрос о том, как эффективно сгенерировать эту матрицу для реального семейства алгоритмов, мы рассмотрим в одном из следующих параграфов.

Напомним, что для вычисления SC-оценки необходимо для каждого алгоритма семейства найти верхнюю связность (определение 2.2) и неполноценность (определение 2.3).

Будем называть алгоритм $s \in A$ *истоком* в множестве $A$, если $s \leqslant a$ для всех $a \in A$. Очевидно, что для вычисления неполноценности алгоритмов $q(a)$ достаточно сравнивать алгоритм $a$ лишь с истоками. На практике количество истоков семейства на порядки меньше полного числа алгоритмов, поэтому предварительный поиск всех истоков позволяет эффективнее вычислять неполноценность алгоритмов.

Для вычисления верхней связности алгоритма $a \in A$ нужен эффективный способ находить все алгоритмы, отличающиеся от $a$ на одном объекте. Это легко сделать, если для каждого алгоритма построить полиномиальный хэш его вектора ошибок: $h(a) = \sum_{i=1}^{L} p^i I(a, x_i)$, где $p$ — простое число, $p \neq 2$. Этот хэш обладает полезным свойством: если два алгоритма $a, b \in A$ различаются на одном объекте, то $h(a) - h(b) = p^i$ для некоторого $i$. Построим для множества $A$ хэш-контейнер (или бинарное дерево поиска), используя данный полиномиальный хэш в качестве ключа. Тогда для поиска «соседей» алгоритма $a$ достаточно просто проверить все числа из $\{h(a) \pm p^i\}_{i=1}^{L}$ на принадлежность хэш-контейнеру. При реализации можно вычислять не сам хэш, а величину $h(a) \mod 2^{64}$, используя целочисленный тип данных и разрешая переполнения. Коллизию ключей в хэш-контейнере можно просто игнорировать. В отдельных случаях это может привести к неточному вычислению верхней связности некоторых алгоритмов, но для практических целей такой точности всегда достаточно.

**Алгоритм 5.1.1** Эффективное вычисление SC-оценки

**Вход:** Матрица ошибок алгоритмов $A$; вектор $\varepsilon_1, \ldots, \varepsilon_t$;

**Выход:** Вектор SC-оценок, посчитанных в точках $E = \{\varepsilon_1, \ldots, \varepsilon_t\}$.

1: $S := \text{НайтиИстоки}(A)$;
2: **для всех** $\varepsilon \in E$ положить $Q(\varepsilon) := 0$;
3: **для всех** $a \in A$
4: $\quad X_q := X_r := \varnothing$;
5: $\quad$ **для всех** $s \in S$ **таких, что** $s < a$
6: $\quad\quad X_r := X_r \cup \{x \in \mathbb{X}\colon I(a,x) = 1 \text{ и } I(s,x) = 0\}$;
7: $\quad$ **для всех** $b \in A\colon a \prec b$
8: $\quad\quad X_q := X_q \cup \{x_{ab}\}$;
9: $\quad H(0) := 0$; $m_a := n(a, \mathbb{X}$;
10: $\quad$ **для** $s = 1, \ldots, m_a$
11: $\quad\quad H(s) := H(s-1) + C^s_{m_a - |X_r|} C^{\ell - |X_q| - s}_{L - |X_q| - m_a} / C^\ell_L$;
12: $\quad$ **для всех** $\varepsilon \in E$ **таких, что** $m_a \geqslant \varepsilon k$
13: $\quad\quad Q(\varepsilon) := Q(\varepsilon) + H(\lfloor \frac{\ell}{L}(m - \varepsilon k) \rfloor)$;
14: **Вернуть** $Q$

**Алгоритм 5.1.2** НайтиИстоки($A$)

**Выход:** Множество истоков $S \subset A$.

1: $S := \varnothing$
2: **для всех** $a \in A$
3: $\quad$ **если** $\exists s \in S\colon s < a$ **то перейти к шагу** 2;
4: $\quad$ исключить из $S$ все $s \in S$ такие что $a < s$;
5: $\quad S := S \cup \{a\}$;
6: **Вернуть** $S$

Полезно реализовать вычисление оценки $Q_\varepsilon(A) \leqslant \eta(\varepsilon)$ сразу для вектора значений $\varepsilon$. Это помогает эффективнее обращать оценку $Q_\varepsilon(A) \leqslant \eta(\varepsilon)$, найдя $\max\limits_{\varepsilon \in [0,1)} \eta(\varepsilon) \geqslant 0.5$.

Наконец, отметим, что при вычислении биномиальных коэффициентов полезно составить таблицу $\log n!$ при $n = 1, \ldots, L$ и использовать ее на шаге 11 алгоритма 5.1.1. Отметим, что биномиальный коэффициент $C^\ell_L$ выходит за границы чисел двойной точности уже при $L = 1030$, $\ell = L/2$. Вместе с тем каждое слагаемое вида $C^s_{m_a - r} C^{\ell - q - s}_{L - q - m_a} / C^\ell_L$ никогда не превышает единицу. Поэтому важно сперва вычислить логарифм всего выражения, и лишь затем взять его экспоненту.

## 5.2 Применение комбинаторных оценок к логическим алгоритмам

В работах [16, 76] был предложен критерий предсказанной информативности, использующий SC-оценку для повышения качества логических алгоритмов классификации.

В данной диссертационной работе предложен более эффективный метод вычисления предсказанной информативности [35]. В отличие от предыдущих работ, не производится никакого дополнительного перебора и оценивания правил — оценки вычисляются только по тем правилам, которые уже были построены в процессе перебора.

**Логические закономерности.** Рассматривается стандартная постановка задачи классификации. Задано множество объектов $\mathbb{X} = (x_i)_{i=1}^{L}$, описанных $n$ действительными признаками, $x_i = (x_i^1, \ldots, x_i^n)$; каждому объекту $x_i$ соответствует ответ $y_i$ из множества $Y = \{-1, 1\}$.

*Логическим правилом* называется конъюнкция пороговых предикатов (термов) вида

$$r(x_i) \equiv r(x_i; c^1, \ldots, c^n) = \prod_{j \in \omega} \left[ x_i^j \lesseqgtr_j c^j \right], \tag{5.1}$$

где $\omega \subseteq \{1, \ldots, n\}$ — подмножество признаков, $\lesseqgtr_j$ — одна из операций сравнения $\{\leqslant, \geqslant\}$, $c^j$ — порог по $j$-му признаку. Говорят, что правило $r$ *выделяет* объект $x$, если $r(x) = 1$.

*Логическая закономерность* — это правило, выделяющее достаточно много ($p$) объектов выбранного класса $y$ (положительных примеров) и приемлемо мало ($n$) объектов всех остальных классов (отрицательных примеров). Для поиска закономерностей класса $y$ по обучающей выборке $X \subset \mathbb{X}$ решается задача двухкритериальной оптимизации:

$$p(r, X) = \sum_{x_i \in X} r(x_i) \, [y_i = y] \to \max_r;$$
$$n(r, X) = \sum_{x_i \in X} r(x_i) \, [y_i \neq y] \to \min_r .$$

Обычно эта задача сводится к максимизации выбранного скалярного критерия информативности $H(p, n)$. В частности, это может быть точный тест Фишера [66], энтропийный критерий, индекс Джини, тест $\chi^2$, тест $\omega^2$ и другие [15]. В обзоре [55] приведено более 20 критериев, но ни один из них не является безусловно лучшим.

Для поиска закономерностей применяются методы дискретной оптимизации: жадные

алгоритмы с последующий редукцией правил [53], поиск в ширину [18], генетические алгоритмы [78], асимптотически оптимальные алгоритмы [10] и другие.

Выбор функционала информативности и метода его оптимизации является эвристикой.

**Переобучение закономерностей.** На практике часто приходится наблюдать эффект переобучения закономерностей — на независимой контрольной выборке пропорция числа положительных $p'$ и отрицательных $n'$ примеров, как правило, смещается в нежелательную сторону: $n'/p' > n/p$. Для сокращения переобучения в [16, 76] предлагается использовать функционал *предсказанной информативности*. Это обычный функционал информативности $H$, в который вместо величин $p$, $n$ на известной обучающей выборке подставляются оценки соответствующих величин $p'$, $n'$ на неизвестной контрольной выборке,

$$\tilde{H}(p,n) = H(p - \delta', n + \delta''),$$

где $\delta'$ и $\delta''$ — поправки на переобучение, получаемые из комбинаторных оценок вероятности переобучения. Преимущество данного подхода в том, что он совместим с любыми функционалами информативности и любыми алгоритмами поиска закономерностей, поэтому его можно встраивать в стандартные библиотеки. Эксперименты на 6 реальных задачах классификации из репозитория UCI показывают, что максимизация предсказанной информативности улучшает обобщающую способность двух типов композиций закономерностей — взвешенного голосования и решающего списка (голосования по старшинству) [76].

В то же время, недостатком предложенного в [76] алгоритма является относительно низкая численная эффективность. Чтобы вычислить вероятность переобучения для заданного набора признаков, приходится перебирать все конъюнкции, находящиеся в некоторой окрестности выбранной оптимальной конъюнкции. При этом размер окрестности увеличивается экспоненциально с ростом числа признаков (ранга конъюнкции). Кроме того, в [76] предполагается, что значения каждого признака попарно различны на объектах выборки.

В работе [35] предлагается ряд упрощений, повышающих численную эффективность и расширяющих границы применимости метода максимизации предсказанной информативности. Во-первых, в оценках вероятности переобучения не учитывается связность, что позволяет применять их для признаков любых типов. Во-вторых, вместо полного перебора конъюнкций по специально построенной окрестности применяется сокращенный перебор только по тем конъюнкциям, для которых в процессе поиска закономерностей было вычислено значение информативности. Данный подход приводит к улучшению обобщающей способности логических закономерностей по сравнению с [76].

**Оценки вероятности переобучения для логических закономерностей.** Правило $r$ класса $y \in Y$ индуцирует на $\mathbb{X}$ бинарный вектор ошибок $\big(I(r,x_i)\big)_{i=1}^{L}$, где $I(r,x_i) = \big[r(x_i) \neq [y_i{=}y]\big]$ — индикатор ошибки правила $r$ на объекте $x_i$. Как и в прошлых параграфах, *число* и *частота* ошибок правила $r$ на выборке $X \subseteq \mathbb{X}$ обозначаются, соответственно, через $m(r,X) = \sum\limits_{x_i \in X} I(r,x_i)$ и $\nu(r,X) = \frac{m(r,X)}{|X|}$. *Методом обучения* называется отображение, которое произвольной обучающей выборке $X \subseteq \mathbb{X}$ ставит в соответствие некоторое правило $r = \mu X$.

Метод обучения $\mu$ называется *монотонным*, если $\mu X = \arg\min\limits_{r} K(r,X)$, где критерий $K(r,X)$ — строго монотонная функция вектора ошибок: для любой пары правил $r, v$ и любой выборки $X \subset \mathbb{X}$ если $I(r,x_i) \leqslant I(v,x_i)$ для всех $x_i \in X$ и хотя бы одно из неравенств строгое, то $K(r,X) < K(v,X)$.

В работе [16] доказано, что SC-оценка (2.10) выполнена не только для ПМЭР, но и для любого монотонного метода обучения. Кроме этого, оказывается, что если функция $H(p,n)$ строго монотонно возрастает по $p$ и строго монотонно убывает по $n$, то критерий $K(r,X) = -H\big(p(r,X), n(r,X)\big)$ является монотонным, а максимизация информативности — монотонным методом обучения. Все используемые на практике критерии обладают свойством монотонности в указанном смысле.

Пусть $R$ — некоторое множество правил. Тогда, согласно SC-оценке, для любого монотонного метода обучения и любой выборки $\mathbb{X}$ справедлива оценка

$$Q_\varepsilon(\mu,\mathbb{X}) \leqslant \sum_{r \in R} \frac{C_{L-u-q}^{\ell-u}}{C_L^\ell} H_{L-u-q}^{\ell-u,m-q}\left(\tfrac{\ell}{L}(m-\varepsilon k)\right) \equiv \eta(\varepsilon), \tag{5.2}$$

где $m = n(r,\mathbb{X})$, $u = u(r)$ — верхняя связность (2.8), $q = q(r)$ — неполноценность (2.9) правила $r$.

Пусть $\varepsilon(\eta)$ — функция, обратная к $\eta(\varepsilon)$. Тогда справедливо утверждение, эквивалентное неравенству (5.2): с вероятностью не менее $1-\eta$

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

Для контроля переобучения правил эту оценку необходимо обобщить, чтобы она учитывала ошибки первого и второго рода. Введем множества положительных и отрицательных примеров:

$$\mathbb{X}' = \{x_i \in \mathbb{X} \colon y_i = y\}; \quad \mathbb{X}'' = \{x_i \in \mathbb{X} \colon y_i \neq y\}.$$

Также введем индикаторы ошибки I и II рода:

$$I'(r,x) = [r(x_i) = 0][y_i = y];$$
$$I''(r,x) = [r(x_i) = 1][y_i \neq y].$$

Число и частоту ошибок относительно этих индикаторов обозначим через $m'(r,X)$, $m''(r,X)$, $\nu'(r,X)$ и $\nu''(r,X)$ соответственно.

Следующие формулы являются обобщением оценки (5.2). Для любого монотонного метода обучения справедливы оценки вероятности переобучения по ошибкам первого и второго рода:

$$Q'_\varepsilon(\mu,\mathbb{X}) \leqslant \sum_{r\in R} \frac{C^{\ell-u}_{L-u-q}}{C^\ell_L} H^{\ell-u,m'-q'}_{L-u-q}\left(\tfrac{\ell}{L}(m'-\varepsilon k)\right) \equiv \eta''(\varepsilon);$$
$$Q''_\varepsilon(\mu,\mathbb{X}) \leqslant \sum_{r\in R} \frac{C^{\ell-u}_{L-u-q}}{C^\ell_L} H^{\ell-u,m''-q''}_{L-u-q}\left(\tfrac{\ell}{L}(m''-\varepsilon k)\right) \equiv \eta''(\varepsilon);$$

где $u = u(r)$ — верхняя связность относительно индикатора ошибки $I$, $q = q(r)$, $q' = q'(r)$, $q'' = q''(r)$ — неполноценность правила $r$ относительно индикаторов ошибки $I$, $I'$, $I''$, соответственно; $m' = m'(r,\mathbb{X}')$ и $m'' = m''(r,\mathbb{X}'')$ — число ошибок $r$ на $\mathbb{X}$ относительно индикаторов ошибки $I'$, $I''$.

**Критерий предсказанной информативности.** Пусть $H(p,n)$ — критерий информативности, монотонный по $p$ и $n$. Обозначим через $\varepsilon'(\eta)$ и $\varepsilon''(\eta)$ функции, обратные к $\eta'(\varepsilon)$ и $\eta''(\varepsilon)$. В новых обозначениях число положительных и отрицательных примеров во всей выборке $\mathbb{X}$ равны, соответственно, $p = |\mathbb{X}'| - m'(r,\mathbb{X})$ и $n = m''(r,\mathbb{X})$ Возьмем в качестве поправок на переобучение медианные оценки частоты ошибок на контроле, получаемые при $\eta = 0.5$:

$$\tilde{H}(p,n) = H\big(p - L\varepsilon'(0.5),\ n + L\varepsilon''(0.5)\big). \tag{5.3}$$

Полученная оценка не накладывает никаких ограничений на то, как именно выбирается множество правил $R$. Оценки расслоения–связности, использованные в [76], довольно жестко предполагали, что $R$ — это множество всех правил, получаемых при фиксации набора признаков $\omega$, фиксации знаков неравенств $\lessgtr_j$ и варьировании порогов $c^j$. В этом случае максимизация предсказанной информативности $\tilde{H}(p,n)$ может использоваться только в качестве критерия отбора признаков $\omega$.

Теперь же можно ввести более общее представление процесса поиска закономерностей, считая, что он разбит на *стадии*. На каждой стадии просматривается некоторое множество

**Алгоритм 5.2.1** ComBoost (Committee Boosting).

**Вход:** $X$ — обучающая выборка; $T, l_0, l_1$ — параметры;

**Выход:** композиция правил $a_T = (r_1, \ldots, r_T)$.

инициализировать выборку $X'$ и отступы: $X' := X$; $M_i := 0$, $i = 1, \ldots, \ell$;

**для всех** $t = 1, \ldots, T$

обучить правила $r_t^y$, $y \in Y$ по выборке $X'$;

$(r_t, y_t) := \arg\min\limits_{(r_t^y, y)\colon y \in Y} \sum\limits_{x_i \in X'} [a_t(x_i) \neq y_i]$;

обновить значения отступов: $M_i := M_i + y_t y_i r_t(x_i)$, $i = 1, \ldots, \ell$;

упорядочить выборку $X$ по возрастанию $M_i$;

$X' := \{x_i \in X\colon \ell_0 < i \leqslant \ell_1\}$.

правил $R$ и из них выбирается лучшее. Критерий $\tilde{H}$ предсказывает, какую информативность выбранное правило будет иметь на новых данных. Для этого используется все множество правил $R$, учитывается его сложность и расслоение. Таким образом, критерий $\tilde{H}$ позволяет правильно отранжировать правила, полученные на разных стадиях, но не позволяет сделать правильный выбор внутри каждой стадии. В работе [35] для вычисления поправок на переобучение правила $r$ в качестве множества $R$ использовались все правила того же целевого класса, что и $r$, построенные алгоритмом поиска закономерностей для признаков, входящих в состав $r$.

**Композиция закономерностей.** *Простое голосование* — это один из стандартных способов построения композиции вида

$$a_t(x) = \operatorname{sign}\left(\sum_{r \in R_{+1}} r(x) - \sum_{r \in R_{-1}} r(x)\right), \tag{5.4}$$

состоящей из $t = |R_{-1}| + |R_{+1}|$ логических закономерностей, где $R_y$ — множество закономерностей класса $y$. Для обучения композиции (5.4) используется комитетный бустинг ComBoost [24]. В отличие от других разновидностей бустинга, он не взвешивает объекты выборки, а только отбирает подвыборки. Поэтому к методу обучения базовых закономерностей применимы комбинаторные оценки переобучения, существующие только для бинарных функций потерь. Другое важное преимущество ComBoost в том, что, благодаря явной оптимизации распределения отступов, он стремится набрать минимальное достаточное число базовых закономерностей.

На шаге 3 алгоритма 5.2.1 для каждого класса $y$ применяется алгоритм 5.2.2 поиска информативных правил, аналогичный алгоритму ТЭМП [18].

**Алгоритм 5.2.2** Усеченный поиск в ширину.

**Вход:** $X$ — обучающая выборка;

$\Theta$ — семейство термов;

$M$ — максимальный ранг конъюнкции;

$S_1$ — параметр ширины поиска;

**Выход:** $R$ — набор правил.

---

инициализация: $R := \varnothing$, $R_0 := \{\varnothing\}$;

**для** $m = 1, \ldots, M$

$R_m := \varnothing$;

**для всех** $r \in R_{m-1}$

нарастить правило $r$ термами $t$:

$R_m := R_m \bigcup \{r \wedge t \colon t \in \Theta$ допустим для $r\}$;

выбрать в $R_m$ целевые классы;

по критерию $\tilde{H}$ оставить в $R_m$ не более $S_1$ лучших правил за каждый класс;

сохранить правила: $R := R \cup R_m$;

**Вернуть** $R$;

---

На шаге 5 алгоритма 5.2.2 допустимыми для добавления считаются термы, не содержащие признаков, которые уже вошли в правило $r$.

**Результаты экспериментов.** В эксперименте [35] на семи реальных задачах классификации из репозитория UCI сравнивались три варианта ComBoost с точным тестом Фишера в качестве критерия информативности:

A: без поправок на переобучение;

B: с поправками по предложенному методу (5.3);

C: с поправками по эмпирической оценке $Q(\varepsilon)$, вычисляемой методом Монте-Карло по случайному подмножеству разбиений.

Для упрощения комбинаторных оценок верхняя связность искусственно полагалась равной нулю во всех экспериментах.

Результаты эксперимента приведены в таблице 5.1. Во всех задачах, кроме australian, варианты B и C дают лучшее качество классификации тестовых данных. Для упрощения комбинаторные оценки вычислялись неточно, и в некоторых случаях вариант B лучше

Таблица 5.1: Средняя частота ошибок (в процентах) на обучающей и тестовой выборке по различным задачам и различным методом контроля переобучения.

| | ComBoost-C | | ComBoost-B | | ComBoost-A | |
|---|---|---|---|---|---|---|
| задача | обуч. | тест | обуч. | тест | обуч. | тест |
| australian | 6.8 | 14.0 | 9.9 | 14.9 | 6.2 | **13.8** |
| echo-card | 0.1 | 2.3 | 0.2 | **0.9** | 0.1 | 2.4 |
| german | 13.1 | **25.4** | 18.3 | 27.6 | 12.9 | 26.0 |
| heart dis. | 8.0 | 18.9 | 11.1 | **18.5** | 7.6 | 19.3 |
| hepatitis | 3.0 | 19.9 | 7.8 | **18.0** | 1.8 | 21.4 |
| labor | 0.6 | **8.9** | 1.1 | 11.9 | 0.5 | 10.9 |
| liver | 11.3 | **31.4** | 33.0 | 42.7 | 8.3 | 32.3 |

варианта C. На пяти из семи задач вариант B дал лучшие результаты, чем в [76], несмотря на то, что он не учитывает эффект связности. Во всех задачах вариант B имеет существенно меньшую переобученность — разность частоты ошибок между тестовой и обучающей выборками.

## 5.3 Проблема сэмплирования алгоритмов

Для вычисления SC-оценки с помощью алгоритма 5.1.1 требуется в явном виде построить матрицу ошибок алгоритмов $A$. В прошлом параграфе этой проблемы удалось избежать, поскольку используемый метод поиска логических закономерностей просматривал в процессе своей работы большое число промежуточных решений. Именно множество этих решений и было использовано для вычисления SC-оценки.

В остальных случаях множество $A$ приходится генерировать. Как было показано в [17], Для широко используемого класса линейных алгоритмов классификации все множество $A$ может состоять из огромного числа алгоритмов (до $10^9$, и даже выше), поэтому на практике возникает проблема выбора небольшого подмножества (вплоть до $10^4$ алгоритмов), позволяющего достаточно точно восстановить оценку переобучения (2.3). В работе [29] для построения множества алгоритмов предлагается метод случайных блужданий по графу расслоения-связности. Основные вычислительные затраты метода были связаны с трудоемкостью поиска всех соседей алгоритма на графе расслоения-связности. В данной диссертационной работе предлагается более эффективный способ

организовать такое случайное блуждание [72]. Для этого вместо поиска всех соседей алгоритма предлагается производить поиск вдоль случайно выбранного направления.

Пусть объекты $\{x_i\}_{i=1}^L$ описаны вещественными признаками: $x_i \in \mathbb{R}^d$, $i = 1, \ldots, L$. Пусть каждому объекту $x_i$ однозначно соответствует один из двух возможных классов $y_i \in \{-1, +1\}$. Рассмотрим множество несмещенных линейных классификаторов $a(x; w) = \operatorname{sign}\langle w, x\rangle$, где $w \in \mathbb{R}^d$ задает вектор весов признаков. Пусть $\mathbb{X} = \{(x_i, y_i)\}_{i=1}^L$ — генеральная выборка объектов. Будем говорить, что алгоритмы $(w_1, w_2)$ являются *соседями*, если они по-разному классифицируют лишь один объект (т. е. $\exists! x \in \mathbb{X}$, такой что $\operatorname{sign}(\langle w_1, x\rangle) \cdot \operatorname{sign}(\langle w_2, x\rangle) = -1$).

Для произвольного алгоритма $w_0$ опишем метод поиска его соседей. В дальнейшем этот метод будет использован, чтобы организовать случайное блуждание по графу расслоения-связности.

**Поиск соседних алгоритмов вдоль заданного направления.** *Преобразование двойственности* $D$ отображает точку $x \in \mathbb{R}^d$ в гиперплоскость $D(x) = \{w \in \mathbb{R}^d \colon \langle w, x\rangle = 0\}$, и наоборот, гиперплоскость $h = \{x \in \mathbb{R}^d \colon \langle w, x\rangle = 0\}$ отображается в точку $D(h) = w$. Каждая гиперплоскость $h_i = D(x_i)$ разделяет $\mathbb{R}^d$ на два полупространства:

$$h_i^+ = \{w \in \mathbb{R}^d \colon \operatorname{sign}\langle w, x_i\rangle = y_i\},$$
$$h_i^- = \{w \in \mathbb{R}^d \colon \operatorname{sign}\langle w, x_i\rangle = -y_i\},$$

таких что в $h_i^+$ все алгоритмы классифицируют объект $x_i$ правильно, а в $h_i^-$ — неправильно. Применив преобразование $D$ ко всей генеральной выборке $\mathbb{X} \subset \mathbb{R}^d$, получим набор гиперплоскостей $\mathbb{H} \equiv \{D(x_i)\}_{i=1}^L$. Эти гиперплоскости разбивают пространство всех алгоритмов $\mathbb{R}^d$ на конусы, называемые *ячейками конфигурации* [39]. При этом оказывается, что в каждой ячейке конфигурации все алгоритмы имеют равный вектор ошибок. Тем самым задача поиска соседних алгоритмов сводится к поиску соседних ячеек конфигурации.

Для поиска алгоритма, соседнего с $w_0$, предлагается выбрать произвольный вектор $u \in \mathbb{R}^d$ и рассмотреть однопараметрическое семейство алгоритмов $\{w_0 + tu \colon t \geqslant 0\}$. Этому семейству соответствует луч в пространстве алгоритмов, выходящий из $w_0$ в направлении $u$. Пересечение этого луча с плоскостью $h_i \in \mathbb{H}$ определяется условием $\langle w_0 + tu, x_i\rangle = 0$, т. е. происходит при значении $t_i = -\frac{\langle w_0, x_i\rangle}{\langle u, x_i\rangle}$. Пусть $t_{(1)}$ и $t_{(2)}$ являются первым и вторым минимальным положительным значением из множества $\{t_i\}$, $i = 1, \ldots, L$. Легко понять, что алгоритм $w' = w_0 + \frac{1}{2}(t_{(1)} + t_{(2)})u$ является соседним с $w_0$.

**Алгоритм 5.3.1** Случайное блуждание на графе расслоения-связности

**Вход:** начальная точка $w_0$; выборка $\mathbb{X} \subset \mathbb{R}^d$; параметры $N, m, n \in \mathbb{N}$, $p \in (0, 1]$;

**Выход:** множество алгоритмов $A$ с попарно-различными векторами ошибок

Инициализация: $v_i = w_0$, $i = 1, \ldots, N$;
$A := \varnothing$;
**пока** $|A| < n$
  **для всех** $i \in 1, \ldots, N$
    найти соседа $v'_i$ для $v_i$ вдоль случайного направления $u \in \mathbb{R}^d$;
    **если** $n(v'_i, \mathbb{X}) > n(v_i, \mathbb{X})$ **то**
      с вероятностью $(1 - p)$ **перейти к шагу** 4;
    **иначе если** $n(v'_i, \mathbb{X}) > n(w_0, \mathbb{X}) + m$ **то**
      **перейти к шагу** 4;
    $v_i := v'_i$;
    $A := A \cup v_i$;
**Вернуть** $A$

**Случайное блуждание на графе расслоения-связности.** Техника случайных блужданий [41, 68] является общепринятым методом сэмплирования вершин из больших графов. В нашем случае случайные блуждания будут использованы, чтобы оценить вероятность переобучения по формулам (2.3), (2.7) или (2.10). При блужданиях будет применяться описанная выше процедура поиска случайного соседа алгоритма $w \in A$.

Алгоритм 5.3.1 управляется следующими параметрами: $n$ — критерий останова по числу алгоритмов, $m$ — ограничение на число рассматриваемых слоев алгоритмов, $N$ — число одновременных блужданий, $p$ — вероятность перехода к алгоритму с большим числом ошибок. Вычислительная сложность алгоритма равна $O(Ldn)$.

Рис. 5.1 соответствует $n = 2000$ шагам простого блуждания ($p = 0$). На нижнем графике показано число ошибок $n(v_i, \mathbb{X})$ как функция от номера шага. На верхнем графике показана цветовая карта хэммингова расстояния $\rho(v_i, v_j)$. В качестве начального приближения использовался алгоритм, настроенный логистической регрессией. Естественно ожидать, что данный алгоритм имеет меньше ошибок, чем случайно выбранный алгоритм, поэтому при простом случайном блуждании число ошибок быстро смещается вверх. Этот эффект является нежелательным, поскольку наибольший вклад в SC-оценку вносят алгоритмы с наименьшим числом ошибок.

На рис. 5.2 аналогичные результаты представлены для случайного блуждания, в ко-

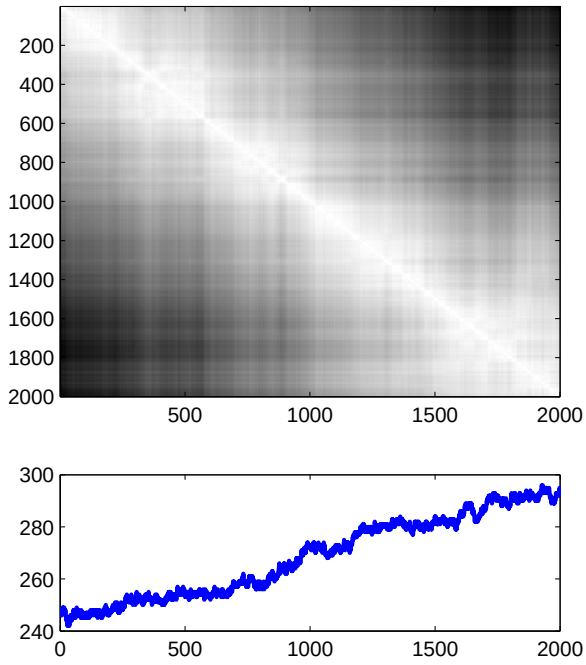



Рис. 5.1: Карта хэммингова расстояния между алгоритмами (верхний график) и профиль числа ошибок (нижний график) для простого блуждания ($p = 1.0$)

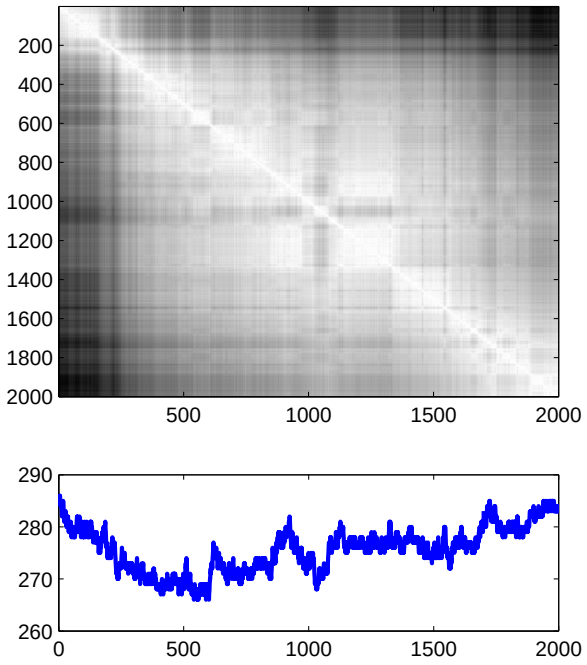



Рис. 5.2: Карта хэммингова расстояния между алгоритмами (верхний график) и профиль числа ошибок (нижний график) для блуждания при $p = 0.5$.

тором переход к алгоритму с большим числом ошибок производится с вероятностью $p = 0.5$. Благодаря этому случайное блуждание остается в нижних слоях графа расслоения-связности.

## 5.4 Прогноз кривых обучения логистической регрессии

В данном параграфе приводятся результаты численного эксперимента из работы [72]. Основные цели эксперимента заключались в следующем:

- проверить, что метод случайного блуждания по графу расслоения-связности дает репрезентативную выборку алгоритмов, достаточную для вычисления оценок переобучения;

Таблица 5.2: Описание задач

| Задача | #Объектов | #Признаков | Задача | #Объектов | #Признаков |
|---|---|---|---|---|---|
| Sonar | 208 | 60 | Statlog | 2310 | 19 |
| Glass | 214 | 9 | Wine | 4898 | 11 |
| Liver dis. | 345 | 6 | Waveform | 5000 | 21 |
| Ionosphere | 351 | 34 | Pageblocks | 5473 | 10 |
| Wdbc | 569 | 30 | Optdigits | 5620 | 64 |
| Australian | 690 | 6 | Pendigits | 10992 | 16 |
| Pima | 768 | 8 | Letter | 20000 | 16 |
| Faults | 1941 | 27 | | | |

- проверить, что комбинаторное определение вероятности переобучения $Q_\varepsilon(\mu, \mathbb{X})$ (2.3) и средней ошибки $\bar{\nu}_\ell(\mu, \mathbb{X})$ (2.5) дают разумные оценки, сравнимые с фактическим переобучением используемых на практике методов обучения (таких как, например, логистическая регрессия).

В эксперименте было использовано 15 задач из репозитория UCI [40]. Описание задач приводится в таблице 5.2. Для многоклассовых задач целевые метки были вручную сгруппированы в два класса. К каждому признаку задачи применялось линейное преобразование так, чтобы все значения признака попали в интервал $[0, 1]$.

В эксперименте сравниваются спрогнозированные и фактические кривые обучения логистической регрессии. Для этого исходная задача $\mathbb{X}$ разбивается на обучающую выборку $\mathbb{X}_L$ и контрольную выборку $\mathbb{X}_K$. Обучающая выборка $\mathbb{X}_L$ используется для настройки логистической регрессии и вычисления оценок переобучения. Затем эти оценки сравниваются с реальной частотой ошибок на контрольной выборке $\mathbb{X}_K$.

Параметр $L$ варьировался в пределах от 5% до 95% от размера исходной задачи с шагом в 5%. Для каждого значения $L$ генерируется $M = 100$ случайных разбиений $\mathbb{X} = \mathbb{X}^i_L \cup \mathbb{X}^i_K$, $i = 1, \ldots, M$, по которым с помощью метода Монте-Карло оценивается частота ошибок на обучении $\nu_L(\mu_{\text{LR}}, \mathbb{X})$ (формула (2.4)) и частота ошибок на контроле $\bar{\nu}_L(\mu_{\text{LR}}, \mathbb{X})$ (формула (2.5)), где в качестве метода обучения $\mu_{\text{LR}}$ выступает логистическая регрессия:

$$\hat{\nu}_L(\mu_{\text{LR}}, \mathbb{X}) = \frac{1}{M}\sum_{i=1}^{M} \nu(\mu_{\text{LR}}\mathbb{X}^i_L, \mathbb{X}^i_L), \qquad \hat{\bar{\nu}}_L(\mu_{\text{LR}}, \mathbb{X}) = \frac{1}{M}\sum_{i=1}^{M} \nu(\mu_{\text{LR}}\mathbb{X}^i_L, \mathbb{X}^i_K).$$

По каждой обучающей выборке $\mathbb{X}_L$ производится сэмплирование множества алгоритмов, а также оценка средних ошибок ПМЭР (ошибки на обучении $\nu_\ell(\mu, \mathbb{X}_L)$ и ошибки

на контроле $\bar{\nu}_\ell(\mu, \mathbb{X}_L)$, где метод обучения $\mu$ соответствует ПМЭР). Для сэмплирования множества алгоритмов по выборке $\mathbb{X}_L$ запускается алгоритм 5.3.1 с параметрами $n = 8\,192$, $N = 64$, $m = 15$, $p = 0.8$, а в качестве начального приближения используется алгоритм $\mu_{\text{LR}}\mathbb{X}_L$, настроенный логистической регрессией. Для вычисления $\nu_\ell(\mu, \mathbb{X}_L)$ и $\bar{\nu}_\ell(\mu, \mathbb{X}_L)$ вновь используется метод Монте-Карло, оценивающий определения (2.4) и (2.5) по $M' = 4\,096$ разбиениям $\mathbb{X}_L = X_\ell^j \cup X_k^j$, $j = 1, \ldots, M'$, при $\frac{\ell}{L} = 0.8$:

$$\hat{\nu}_\ell(\mu, \mathbb{X}_L) = \frac{1}{M'} \sum_{j=1}^{M} \nu(\mu X_\ell^j, X_\ell^j), \qquad \hat{\bar{\nu}}_\ell(\mu, \mathbb{X}_L) = \frac{1}{M'} \sum_{j=1}^{M} \nu(\mu X_\ell^j, X_k^j).$$

Затем эти оценки усредняются по всем разбиениям $\mathbb{X} = \mathbb{X}_L^i \cup \mathbb{X}_K^i$.

Эти четыре значения (фактическая частота ошибок логистической регрессии на обучении $\nu_L(\mu_{\text{LR}}, \mathbb{X})$ и на контроле $\bar{\nu}_L(\mu_{\text{LR}}, \mathbb{X})$, частота ошибок ПМЭР на обучении $\nu_\ell(\mu, \mathbb{X}_L)$, частота ошибок ПМЭР на контроле $\bar{\nu}_\ell(\mu, \mathbb{X}_L)$) приводятся на рисунках 5.3 и 5.4 как функции числа объектов на обучении. Рисунки приводятся в порядке возрастания числа объектов в задачах. Отметим, что частота ошибок ПМЭР на контроле может быть как выше, так и ниже фактической частоты ошибок логистической регрессии, поскольку $\mu$ и $\mu_{\text{LR}}$ являются разными методами обучения. Тем не менее из графиков следует вывод, что оценка $\bar{\nu}_\ell(\mu, \mathbb{X}_L)$, полученная по обучающей выборке $\mathbb{X}_L$, дает достаточно точный прогноз фактической частоты ошибок $\bar{\nu}_L(\mu_{\text{LR}}, \mathbb{X})$ и кривой обучения логистической регрессии на контрольной выборке $\mathbb{X}_K$.

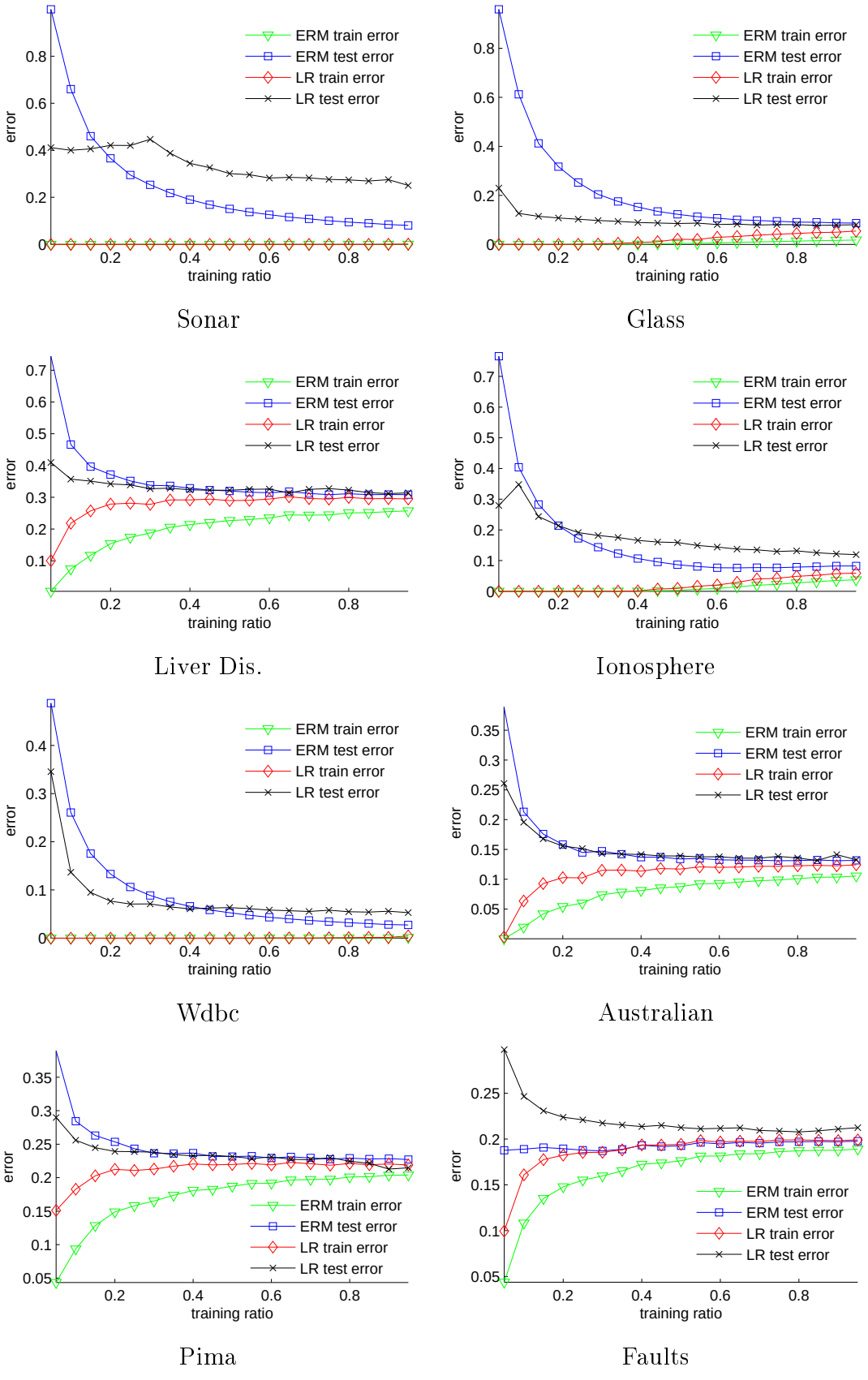



Рис. 5.3: Кривые обучения логистической регрессии и ПМЭР. Частота ошибок логистической регрессии оценена методом Монте-Карло на разбиениях исходной задачи $\mathbb{X} = \mathbb{X}_L \cup \mathbb{X}_K$. Частота ошибок ПМЭР оценена по разбиениям обучающей выборки $\mathbb{X}_L = X_\ell \cup X_k$. Продолжение на с. 92.

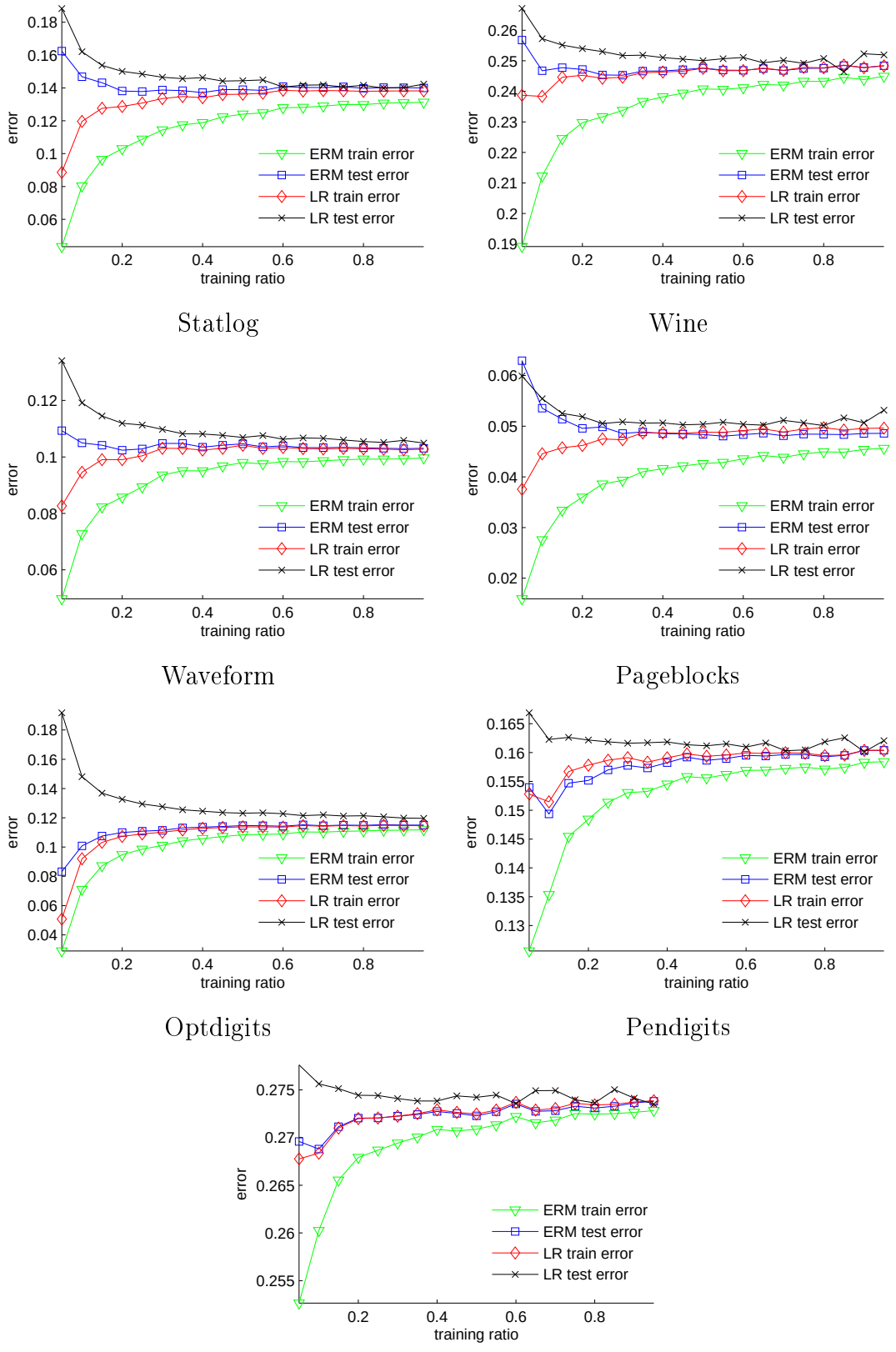



Рис. 5.4: Кривые обучения логистической регрессии и ПМЭР. Начало на с. 91.

## 5.5 Экспериментальное сравнение комбинаторных оценок

Проведем экспериментальное сравнение новой оценки (3.16) с тремя комбинаторными оценками (VC- и SC-оценками из [77], ES-оценкой из [29]) и двумя PAC-Bayes оценками из [57]. Основная цель данного эксперимента — сравнить завышенность перечисленных оценок.

В качестве исходных данных были использованы те же задачи из репозитория UCI, что и в прошлом параграфе. Описание задач приведено в таблице 5.2.

К каждой задаче была применена процедура пятикратной кросс-валидации, которая запускалась 100 раз для усреднения результатов. Таким образом, для каждой задачи генерировалось $M = 500$ разбиений $\mathbb{X} = \mathbb{X}^i_L \sqcup \mathbb{X}^i_K$, $i = 1, \ldots, M$ и вычислялась оценка Монте-Карло для среднего уклонения частот ошибок логистической регрессии:

$$\hat{\delta}_L(\mu_{\text{LR}}, \mathbb{X}) = \frac{1}{M} \sum_{i=1}^{M} \nu(\mu_{\text{LR}}\mathbb{X}^i_L, \mathbb{X}^i_K) - \nu(\mu_{\text{LR}}\mathbb{X}^i_L, \mathbb{X}^i_L).$$

После этого каждая обучающая выборка $\mathbb{X}_L$ использовалась для вычисления комбинаторной оценки на уклонение частот ошибок МЭР. Множества алгоритмов $A$, из которого МЭР выбирал лучший алгоритм, генерировалось с помощью случайных блужданий по графу расслоения-связности линейных классификаторов (алгоритм 5.3.1). В качестве начального приближения для случайного блуждания использовался алгоритм $\mu_{\text{LR}}\mathbb{X}_L$, настроенный логистической регрессией по обучающей выборке $\mathbb{X}_L$. Далее вновь использовался метод Монте-Карло: генерировались $M' = 4096$ случайных разбиений $\mathbb{X}_L = X^j_\ell \sqcup X^j_k$, $j = 1, \ldots, M'$ (при $\frac{\ell}{L} = 0.8$) и вычислялась следующая величина:

$$\hat{\delta}_\ell(\mu, \mathbb{X}_L) = \frac{1}{M'} \sum_{j=1}^{M'} \nu(\mu X^j_\ell, X^j_k) - \nu(\mu X^j_\ell, X^j_\ell),$$

где $\mu$ — метод минимизации эмпирического риска. В заключение эта величина усреднялась по всем разбиениям $\mathbb{X} = \mathbb{X}^i_L \sqcup \mathbb{X}^i_K$. Величины $\hat{\delta}_L(\mu_{\text{LR}}, \mathbb{X})$ и $\hat{\delta}_\ell(\mu, \mathbb{X}_L)$ соответствуют реальному переобучению логистической регрессии и его идеальной оценке в рамках комбинаторного подхода.

В таблице 5.3 сравниваются следующие величины: $\hat{\delta}_L(\mu_{\text{LR}}, \mathbb{X})$ и $\hat{\delta}_\ell(\mu, \mathbb{X}_L)$; четыре верхние комбинаторные оценки величины $\hat{\delta}_\ell(\mu, \mathbb{X}_L)$, обозначенные как VC, SC, ES и AF; две PAC-Bayes оценки (обозначены как DD и DI). В отличие от комбинаторных оценок,

Таблица 5.3: Сравнение фактического переобучения и различных оценок

| | Ошибка | Переобучение | | Комбинаторные оценки | | | | PAC-Bayes | |
|---|---|---|---|---|---|---|---|---|---|
| Task | Тест | $\hat{\delta}_L(LR)$ | $\hat{\delta}_\ell(\mu)$ | VC | SC | ES | AF | DI | DD |
| glass | 0.076 | 0.030 | 0.067 | 0.191 | 0.127 | 0.124 | 0.106 | 1.268 | 0.740 |
| Liver dis. | 0.315 | 0.017 | 0.046 | 0.249 | 0.192 | 0.146 | 0.161 | 1.207 | 1.067 |
| Ionosphere | 0.126 | 0.079 | 0.042 | 0.138 | 0.099 | 0.087 | 0.084 | 1.219 | 1.149 |
| Australian | 0.136 | 0.014 | 0.023 | 0.130 | 0.101 | 0.081 | 0.086 | 1.145 | 0.678 |
| pima | 0.227 | 0.007 | 0.021 | 0.151 | 0.117 | 0.090 | 0.098 | 0.971 | 0.749 |
| faults | 0.210 | 0.011 | 0.008 | 0.091 | 0.070 | 0.046 | 0.060 | 1.110 | 1.054 |
| statlog | 0.142 | 0.004 | 0.008 | 0.072 | 0.060 | 0.043 | 0.051 | 1.102 | 0.746 |
| wine | 0.250 | 0.002 | 0.003 | 0.061 | 0.047 | 0.032 | 0.040 | 0.776 | 0.637 |
| waveform | 0.105 | 0.003 | 0.003 | 0.043 | 0.033 | 0.023 | 0.023 | 0.561 | 0.354 |
| pageblocks | 0.051 | 0.001 | 0.003 | 0.030 | 0.022 | 0.016 | 0.018 | 0.739 | 0.186 |
| Optdigits | 0.121 | 0.006 | 0.003 | 0.043 | 0.034 | 0.023 | 0.026 | 1.068 | 0.604 |

ограничивающих уклонение частот ошибок, PAC-Bayes оценки справедливы непосредственно для частоты ошибок на контроле (приведена в столбце «Ошибка тест»). DD-оценка учитывает размерность пространства признаков и является более точной по сравнению с универсальной DI-оценкой, справедливой для любого числа признаков.

Комбинаторная SC-оценка соответствует оценке расслоения-связности из [77]. VC-оценка также приведена в [77], и она, в отличие от SC-оценки, не учитывает ни расслоение, ни связность. ES-оценка [29] основана на более тонком учете расслоения, при котором каждый алгоритм сравнивается со всем множеством найденных истоком графа расслоения-связности.

AF-оценка получена из оценки метода порождающих и запрещающих множеств (3.16). Чтобы полностью конкретизировать (3.16), необходимо уточнить следующее: метод разбиения исходного множества алгоритмов $A$ на кластеры, способ выбора порождающих и запрещающих множеств для каждого кластера, способ оценивания вероятности переобучения каждого кластера. В AF-оценке порождающие и запрещающие множества выбирались в соответствии с (3.16), для оценки вероятности переобучения каждого кластера использовалась формула (4.10), а представление множества алгоритмов $A$ в виде $A = A_1 \sqcup \cdots \sqcup A_t$ производилось с помощью иерархической кластеризации при выборе расстояния дальнего соседа. Исходная метрика на $A$ определялась как хэммингово расстояние между векторами

ошибок алгоритмов.

Из экспериментальных данных следует, что переобучение логистической регрессии хорошо приближается комбинаторной оценкой $\hat{\delta}_\ell(\mu, \mathbb{X}_L)$ для МЭР. Все четыре комбинаторные оценки существенно точнее обеих PAC-Bayes оценок. Среди комбинаторных оценок наименее завышенной оказывается ES-оценка. За ней следует предложенная нами AF-оценка. Каждая из этих оценок существенно уточняет SC-оценку расслоения-связности. Улучшение точности в ES- и AF-оценках основано на двух различных эффектах — более тонком учете расслоения в ES-оценке и учете сходства алгоритмов в AF-оценке. Представляется возможным, что объединение ES- и AF-оценок позволит добиться еще большего качества комбинаторных оценок вероятности переобучения.

В заключение отметим, что в данном эксперименте VC-оценка посчитана по малому подмножеству алгоритмов $A$, полученному с помощью случайных блужданий. Это «локальная» версия VC-оценки. Обычная VC-оценка основана на VC-размерности $d$ всего пространства алгоритмов и превышает единицу на всех рассмотренных задачах.

# Заключение

Основные результаты диссертационной работы.

1. Предложен теоретико-групповой метод вывода оценок вероятности переобучения для рандомизированного метода минимизации эмпирического риска.
2. Доказаны точные оценки вероятности переобучения рандомизированного метода минимизации эмпирического риска для девяти модельных семейств, включая монотонные и унимодальные сети, слой хэммингова шара и ряд других.
3. Получена общая оценка вероятности переобучения, основанная на разложении и покрытии множества алгоритмов.
4. В экспериментах на реальных данных показано, что новая оценка вероятности переобучения является более точной по сравнению с другими оценками вероятности переобучения.

Возможным направлением дальнейших исследований является повышение точности комбинаторных оценок вероятности переобучения путем более аккуратного учета эффекта расслоения, а также применение полученных в данной работе оценок для улучшения обобщающей способности логических алгоритмов классификации.

# Список литературы


1. Айзерман М. А., Браверман Э. М., Розоноэр Л. И. Метод потенциальных функций в теории обучения машин. — М.: Наука, 1970. — 320 с.
2. 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.
3. Ботов П. В. Уменьшение вероятности переобучения итерационных методов статистического обучения. — 2011. — С. 44–47.
4. Вапник В. Н., Червоненкис А. Я. О равномерной сходимости частот появления событий к их вероятностям // ДАН СССР. — 1968. — Т. 181, № 4. — С. 781–784.
5. 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.
6. Вапник В. Н., Червоненкис А. Я. Теория распознавания образов. — М.: Наука, 1974.
7. Воронцов К. В. Слабая вероятностная аксиоматика и надежность эмпирических предсказаний // Математические методы распознавания образов-13. — М.: МАКС Пресс, 2007. — С. 21–25.
8. Воронцов К. В. Комбинаторный подход к проблеме переобучения // Всеросс. конф. Математические методы распознавания образов-14. — М.: МАКС Пресс, 2009. — С. 18–21.
9. Воронцов К. В. Точные оценки вероятности переобучения // Доклады РАН. — 2009. — Т. 429, № 1. — С. 15–18.
10. Дюкова Е. В., Инякин А. С. Об асимптотически оптимальном построении тупиковых покрытий целочисленной матрицы // Математические вопросы кибернетики. — Т. 17. — М.: Физматлит., 2008. — С. 247–262.
11. Журавлев Ю. И. Корректные алгебры над множествами некорректных (эвристических) алгоритмов. Часть I // Кибернетика. — 1977. — № 4. — С. 5–17.
12. Журавлев Ю. И. Корректные алгебры над множествами некорректных (эвристических) алгоритмов. Часть II // Кибернетика. — 1977. — № 6. — С. 21–27.
13. Журавлев Ю. И. Корректные алгебры над множествами некорректных (эвристических) алгоритмов. Часть III // Кибернетика. — 1978. — № 2. — С. 35–43.
14. Журавлев Ю. И. Об алгебраическом подходе к решению задач распознавания или

классификации // Проблемы кибернетики. — 1978. — Т. 33. — С. 5–68.

15. Загоруйко Н. Г. Прикладные методы анализа данных и знаний. — Новосибирск: ИМ СО РАН, 1999. — 270 с.
16. Ивахненко А. А., Воронцов К. В. Критерии информативности пороговых логических правил с поправкой на переобучение порогов // 15-я всероссийская конференция «Математические методы распознавания образов», Петрозаводск. — 2011. — С. 48–51.
17. 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.
18. Лбов Г. С. Методы обработки разнотипных экспериментальных данных. — Новосибирск: Наука, 1981. — 160 с.
19. Мазуров Вл. Д. Комитеты системы неравенств и задача распознавания // Кибернетика. — 1971. — № 3. — С. 140–146.
20. Мазуров Вл. Д. Метод комитетов в задачах оптимизации и классификации. — М.: Наука, 1990. — 250 с.
21. Матросов В. Л. Корректные алгебры ограниченной емкости над множествами некорректных алгоритмов // ДАН СССР. — 1980. — Т. 253, № 1. — С. 25–30.
22. Матросов В. Л. Емкость алгебраических расширений модели алгоритмов вычисления оценок // ЖВМиМФ. — 1984. — Т. 24, № 11. — С. 1719–1730.
23. Матросов В. Л. Емкость алгоритмических многочленов над множеством алгоритмов вычисления оценок // ЖВМиМФ. — 1985. — Т. 25, № 1. — С. 122–133.
24. Маценов А. А. Комитетный бустинг: минимизация числа базовых алгоритмов при простом голосовании // Математические методы распознавания образов-13. — М.: МАКС Пресс, 2007. — С. 180–183.
25. Пыткеев Е. Г., Хачай М. Ю. Топологические свойства измеримых структур и достаточные условия равномерной сходимости частот к вероятностям. — 2012. — № 2. — С. 89–98.
26. Рудаков К. В. Универсальные и локальные ограничения в проблеме коррекции эвристических алгоритмов // Кибернетика. — 1987. — № 2. — С. 30–35.
27. Рудаков К. В. Об алгебраической теории универсальных и локальных ограничений для задач классификации // Распознавание, Классификация, Прогноз. — 1988. — Т. 1. — С. 176–200.

28. 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.

29. Соколов Е. А., Воронцов К. В. Минимизация вероятности переобучения для композиций линейных классификаторов малой размерности // Межд. конф. Интеллектуализация обработки информации ИОИ-9. — М.: МАКС Пресс, 2012. — С. 85–85.

30. Толстихин И. О. Вероятность переобучения плотных и разреженных семейств алгоритмов // Межд. конф. Интеллектуализация обработки информации ИОИ-8. — М.: МАКС Пресс, 2010. — С. 83–86.

31. Фрей А. И. Точные оценки вероятности переобучения для симметричных семейств алгоритмов // Всеросс. конф. Математические методы распознавания образов-14. — М.: МАКС Пресс, 2009. — С. 66–69.

32. Фрей А. И. Точные оценки вероятности переобучения для симметричных семейств алгоритмов // Труды 52-й научной конференции МФТИ «Современные проблемы фундаментальных и прикладных наук». Часть II. Управление и прикладная математика. — М.: МФТИ, 2009. — С. 106–109.

33. Фрей А. И. Вероятность переобучения плотных и разреженных многомерных сеток алгоритмов // Межд. конф. Интеллектуализация обработки информации ИОИ-8. — М.: МАКС Пресс, 2010. — С. 87–90.

34. Фрей А. И. Метод порождающих и запрещающих множеств для рандомизированного метода минимизации эмпирического риска // Всеросс. конф. Математические методы распознавания образов-15. — М.: МАКС Пресс, 2011. — С. 60–63.

35. Фрей А. И., Ивахненко А. А., Решетняк И. М. Применение комбинаторных оценок вероятности переобучения в простом голосовании пороговых конъюнкций // Межд. конф. Интеллектуализация обработки информации ИОИ-9. — М.: МАКС Пресс, 2012. — С. 86–89.

36. Фрей А. И., Толстихин И. О. Комбинаторные оценки вероятности переобучения на основе кластеризации и покрытий множества алгоритмов // Machine learning and data analysis. — 2013. — Т. 1(6). — С. 751–767.

37. Хачай М. Ю. О длине обучающей выборки для комитетного решающего правила. — 2000. — № 2. — С. 219–223.

38. Хачай М. Ю. О вычислительной сложности задачи о минимальном комитете и смежных задач // Доклады РАН. — 2006. — Т. 406, № 6. — С. 742–745.

39. Agarwal Pankaj K., Sharir Micha. Arrangements and their applications // Handbook of

Computational Geometry. — Elsevier Science Publishers B.V. North-Holland, 1998. — P. 49–119.

40. UCI machine learning repository: Rep. / University of California, Irvine, School of Information and Computer Sciences. — Executor: A. Asuncion, D.J. Newman: 2007.
41. Avrachenkov K., Ribeirom B., Towsley D. Improving random walk estimation accuracy with uniform restarts // Proc. of WAW 2010. — 2010.
42. Bartlett P., Bousquet O., Mendelson S. Localized rademacher complexities // COLT: 15th Annual Conference on Computational Learning Theory. — Springer, Berlin, 2002. — P. 44–58.
43. Bartlett P., Bousquet O., Mendelson S. Local rademacher complexities // The Annals of Statistics. — 2005. — V. 33, no. 4. — P. 1497–1537.
44. Bartlett Peter L., Mendelson Shahar, Philips Petra. Local complexities for empirical risk minimization // COLT: 17th Annual Conference on Learning Theory / Ed. by John Shawe-Taylor, Yoram Singer. — Springer-Verlag, 2004. — P. 270–284.
45. Bax E. Similar classifiers and VC error bounds. — No. CalTech-CS-TR97-14. — 1997. — P. 19.
46. Botov P. V. Exact estimates of the probability of overfitting for multidimensional modeling families of algorithms // Pattern Recognition and Image Analysis. — 2011. — V. 21, no. 1. — P. 52–65.
47. Bottou Léon, Cortes Corinna, Vapnik Vladimir. On the effective VC dimension. — 1994.
48. Boucheron S., Lugosi G., Massart P. Concentration inequalities using the entropy method // The Annals of Probability. — 2003. — V. 31, no. 3. — P. 1583–1614.
49. Boucheron Stephane, Bousquet Olivier, Lugosi Gabor. Theory of classification: A survey of some recent advances // ESAIM: Probability and Statistics. — 2005. — V. 9. — P. 323–375.
50. Bousquet O. A bennett concentration inequality and its application to suprema of empirical processes // Comptes Rendus Mathematique. — 2002. — V. 334, no. 6. — P. 495–500.
51. Bousquet Olivier. Concentration Inequalities and Empirical Processes Theory Applied to the Analysis of Learning Algorithms: Ph. D. thesis / Olivier Bousquet. — Ecole Polytechnique, France. — 2002. — 351 p.
52. Bousquet Olivier, Boucheron Stéphane, Lugosi Gábor. Introduction to statistical learning theory // Advanced Lectures on Machine Learning. — Springer, 2004. — P. 169–207.
53. Cohen William W., Singer Yoram. A simple, fast and effective rule learner // Proc. of the 16 National Conference on Artificial Intelligence. — 1999. — P. 335–342.

54. Frei A. I. Accurate estimates of the generalization ability for symmetric set of predictors and randomized learning algorithms // Pattern Recognition and Image Analysis. — 2010. — V. 20, no. 3. — P. 241–250.

55. Fürnkranz Johannes, Flach Peter A. Roc 'n' rule learning-towards a better understanding of covering algorithms // Machine Learning. — 2005. — V. 58, no. 1. — P. 39–77.

56. Haussler D., Littlestone N., Warmuth M. K. Predicting $\{0,1\}$-functions on randomly drawn points // Inf. Comput. — 1994. — V. 115. — P. 248–292.

57. Jin Chi, Wang Liwei. Dimensionality dependent pac-bayes margin bound. // NIPS / Ed. by Peter L. Bartlett, Fernando C. N. Pereira, Christopher J. C. Burges et al. — 2012. — P. 1043–1051.

58. Klein T. Une inegalite de concentration a gauche pour les processus empiriques // C.R. Acad. Sci. Paris. — 2002. — V. 334. — P. 500–505.

59. Koltchinskii V. Oracle Inequalities in Empirical Risk Minimization and Sparse Recovery Problems: École d'Été de Probabilités de Saint-Flour XXXVIII-2008. Lecture Notes in Mathematics. — Springer, 2011.

60. Koltchinskii Vladimir. Rademacher penalties and structural risk minimization // IEEE Transactions on Information Theory. — 2001. — V. 47, no. 5. — P. 1902–1914.

61. Koltchinskii Vladimir. Local rademacher complexities and oracle inequalities in risk minimization // The Annals of Statistics. — 2006. — V. 34, no. 6. — P. 2593–2656.

62. Koltchinskii Vladimir, Panchenko Dmitry. Rademacher processes and bounding the risk of function learning // High Dimensional Probability, II / Ed. by D. E. Gine, J.Wellner. — Birkhauser, 1999. — P. 443–457.

63. Langford John. Tutorial on practical prediction theory for classification // Journal of Machine Learning Research. — 2005. — V. 6. — P. 273–306.

64. Langford John, Shawe-Taylor John. PAC-Bayes and margins // Advances in Neural Information Processing Systems 15. — MIT Press, 2002. — P. 439–446.

65. Lugosi Gabor. On concentration-of-measure inequalities. — Machine Learning Summer School, Australian National University, Canberra. — 2003.

66. Martin J. Kent. An exact probability metric for decision tree splitting and stopping // Machine Learning. — 1997. — V. 28, no. 2-3. — P. 257–291.

67. McAllester David A. Some pac-bayesian theorems // Proceedings of the eleventh annual conference on Computational learning theory / ACM. — 1998. — P. 230–234.

68. Ribeiro B., Towsley D. Estimating ans sampling graphs with multidimensional random walks

// 10th Conf. on Internet Measurement. — 2010. — P. 390–403.

69. Seeger Matthias. PAC-Bayesian generalization error bounds for Gaussian process classification // Journal of Machine Learning Research. — 2002. — V. 3. — P. 233–269.
70. Talagrand Michel. New concentration inequalities in product spaces // Inventiones mathematicae. — 1996. — V. 126, no. 3. — P. 505–563.
71. Vapnik Vladimir. Statistical Learning Theory. — Wiley, New York, 1998.
72. Vorontsov K., Frey A. I., Sokolov E. Computable combinatorial overfitting bounds // Machine learning and data analysis. — 2013. — V. 1(6). — P. 724–733.
73. Vorontsov K. V. Combinatorial probability and the tightness of generalization bounds // Pattern Recognition and Image Analysis. — 2008. — V. 18, no. 2. — P. 243–259.
74. 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. — V. 19, no. 3. — P. 412–420.
75. Vorontsov K. V. Exact combinatorial bounds on the probability of overfitting for empirical risk minimization // Pattern Recognition and Image Analysis. — 2010. — V. 20, no. 3. — P. 269–285.
76. 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. — P. 66–73.
77. Vorontsov K. V., Ivahnenko A. A., Reshetnyak I. M. Generalization bound based on the splitting and connectivity graph of the set of classifiers // Pattern Recognition and Image Analysis: new information technologies (PRIA-10). — 2010.
78. Yankovskaya A. E., Tsoy Y. R. Selection of optimal set of diagnostic tests with use of evolutionary approach in intelligent systems // 5-th EUSFLAT Conference New Dimensions in Fuzzy Logic and Related Technologies. — V. 2. — 2007. — P. 267–270.