Точные оценки вероятности переобучения

К. В. Воронцов (www.ccas.ru/voron)

10 марта 2009 г.

Аннотация

Предлагается новый подход к проблеме обобщающей способности обучаемых алгоритмов, который даёт точные оценки вероятности переобучения и не опирается на неравенство Буля. Точные оценки выводятся для метода минимизации эмпирического риска и семейств алгоритмов специального вида: монотонных и унимодальных цепочек, единичной окрестности оптимального алгоритма, двухэлементного семейства. Есть основания полагать, что предлагаемый подход удастся распространить и на семейства более общего вида.

Получение достаточно точных верхних оценок обобщающей способности до сих пор остаётся открытой проблемой в теории статистического обучения. В течение последних сорока лет, начиная с работ В. Н. Вапника и А. Я. Червоненкиса [3], неоднократно предпринимались попытки существенно уточнить оценки, см. обзоры [21, 17]. Однако наиболее точные из известных оценок всё ещё сильно завышены [11, 14, 18]. Завышенность приводит к необоснованному требованию увеличивать длину обучающей выборки, а в методе структурной минимизации риска — к чрезмерному упрощению алгоритмов [12]. Наиболее интересные для практики случаи, когда выборки достаточно малы, а алгоритмы достаточно сложны, оказываются за границами применимости теории, поскольку оценки вероятности переобучения в этих случаях тривиально превышают единицу. Завышенные оценки лишь на качественном уровне описывают связь переобучения со сложностью семейства алгоритмов, и не всегда подходят для точных количественных предсказаний и управления процессом обучения. Остаётся открытым вопрос, не связано ли переобучение с какими-то более тонкими и пока не изученными явлениями.

Предшествующие эксперименты [5, 22] позволили выявить и количественно сравнить основные причины завышенности классических оценок. Было показано, что вероятность переобучения существенно зависит не только от сложности семейства (числа различных алгоритмов в нём), но ещё и от степени их различности. Основной вывод заключался в том, что для получения точных оценок необходимо одновременно учесть два эффекта: степень сходства алгоритмов в семействе и расслоение семейства по уровням частоты ошибок. Пренебрежение одним из этих эффектов сводит на нет все усилия, направленные на учёт второго. Данный вывод косвенно подтверждается и тем, что известные попытки учесть эти эффекты по отдельности [11, 14, 7, 19] не дали радикального улучшения точности. На сегодняшний день не существует теории, которая учитывала бы оба эффекта. Основной причиной завышенности принято считать неравенство Буля (union bound), применяемое при доказательстве практически всех известных на сегодняшний день сложностных оценок.

В данной работе развивается новый подход, в котором неравенство Буля вообще не используется. Основная идея заключается в следующем. Допустим, для каждого алгоритма семейства можно указать число *эталонных* объектов, которые обязаны присутствовать в обучающей выборке, и число *шумовых* объектов, которых не должно быть в обучающей выборке, чтобы в результате обучения был получен именно этот алгоритм. Доказано, что тогда можно в явном виде выписать вероятности получения каждого из алгоритмов, а также точные значения вероятности переобучения. Несмотря на кажущуюся ограниченность исходных предположений, данный подход позволил вывести точные оценки для ряда конкретных случаев. В первую очередь он хорошо подходит для изучения связных семейств [19], которые наиболее интересны с практической точки зрения.

Раздел 1 вводит основные определения и обозначения. В разделе 2 объясняется необходимость введения слабой вероятностной аксиоматики, которую можно также называть комбинаторной или перестановочной. В разделе 3 кратко перечисляются основные результаты предшествовавших исследований и экспериментов, которые привели к рассматриваемым далее постановкам задач. В разделе 4 формулируются и доказываются две основные теоремы, с помощью которых в следующих разделах выводятся точные оценки вероятности переобучения для четырёх частных видов семейств: монотонных цепочек алгоритмов (раздел 5), унимодальных цепочек алгоритмов (раздел 6), единичной окрестности оптимального алгоритма (раздел 7) и двухэлементного семейства алгоритмов (раздел 8).

1 Вероятность переобучения

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

На практике под «алгоритмами» понимаются функции, реализующие отображения из множества объектов $\mathbb{X}$ в некоторое множество ответов, и используемые для решения задач классификации, регрессии, прогнозирования. Однако для целей данного исследования нет необходимости конкретизировать понятие «алгоритма»; достаточно считать алгоритмы элементами некоторого абстрактного множества $\mathbb{A}$, предполагая лишь, что существует способ определить, допускает ли алгоритм a ошибку на объекте x. Такое понимание «алгоритма» существенно расширяет класс рассматриваемых задач.

Числом ошибок алгоритма a на выборке $X \subseteq \mathbb{X}$ называется величина

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

Частотой ошибок или *эмпирическим риском* алгоритма a на выборке X называется величина $\nu(a, X) = \frac{1}{|X|} n(a, X)$. Она принимает значения из отрезка $[0, 1]$.

Обозначим через $\mathbb{X}_L^\ell$ множество всех ℓ-элементных подмножеств генеральной выборки $\mathbb{X}$. Очевидно, $|\mathbb{X}_L^\ell| = C_L^\ell$.

Методом обучения называется отображение $\mu\colon \mathbb{X}_L^\ell \to \mathbb{A}$, которое произвольной *обучающей выборке* $X \in \mathbb{X}_L^\ell$ ставит в соответствие некоторый алгоритм $a = \mu X$ из $\mathbb{A}$.

Метод обучения μ называется методом *минимизации эмпирического риска*, если

$$\mu X = \arg\min_{a \in \mathbb{A}} n(a, X). \tag{1.1}$$

Вектором ошибок алгоритма a будем называть L-мерный бинарный вектор $(a)_{\mathbb{X}} = \big(I(a, x_i)\big)_{i=1}^{L}$. Поскольку в дальнейшем нас будут интересовать не сами алгоритмы, а, главным образом, их векторы ошибок, для краткости будем использовать обозначение a вместо $(a)_{\mathbb{X}}$ и говорить «вектор a».

Обозначим через A множество векторов ошибок, порождаемых алгоритмами вида $a = \mu X$ на всевозможных обучающих подвыборках X:

$$A = \big\{(\mu X)_{\mathbb{X}} \colon X \in \mathbb{X}_L^\ell\big\}.$$

Заметим, что мощность множества алгоритмов $\big\{\mu X \colon X \in \mathbb{X}_L^\ell\big\}$ не превосходит C_L^ℓ. Она может оказаться и строго меньше C_L^ℓ, поскольку метод μ может строить по различным выборкам совпадающие алгоритмы. Мощность множества векторов A может оказаться ещё меньше, поскольку различные алгоритмы могут порождать совпадающие векторы ошибок. В общем случае $|A| \leqslant C_L^\ell$.

Уклонением частот ошибок алгоритма a на двух выборках X и $\bar{X} = \mathbb{X} \backslash X$ называется разность частот $\delta(a, X) = \nu(a, \bar{X}) - \nu(a, X)$. *Переобученностью* метода μ на выборке X будем называть уклонение частот ошибок алгоритма $a = \mu X$:

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

Будем говорить, что метод μ *переобучен* на выборке X, если $\delta_\mu(X) \geqslant \varepsilon$, где ε — положительный вещественный параметр, называемый *порогом переобучения*.

Обычно термин «переобучение» вводится неформально и обозначает нежелательное явление, когда алгоритм, настроенный по обучающей выборке, заметно хуже ведёт себя на новых контрольных данных. Здесь этому термину придаётся более строгий формальный смысл. Нашей основной задачей будет получение точных оценок *вероятности переобучения* $\mathsf{P}_{\mathbb{X}}\big\{\delta_\mu(X) \geqslant \varepsilon\big\}$.

2 Слабая вероятностная аксиоматика

Статистическая теория обучения основана на предположении, что выборка $\mathbb{X}$ — простая, то есть что её элементы выбраны из некоторой (как правило, бесконечной) генеральной совокупности случайно, независимо, согласно одной и той же (как правило, неизвестной) вероятностной мере.

Мы будем придерживаться более слабой вероятностной аксиоматики [22].

Пусть $\mathbb{X}$ — произвольное конечное множество объектов. Предполагается, что все его C_L^ℓ разбиений на наблюдаемую обучающую выборку X длины ℓ и скрытую контрольную выборку $\bar{X}$ длины $k = L - \ell$ реализуются с равной вероятностью. Данное предположение фактически эквивалентно стандартной гипотезе о независимости элементов выборки $\mathbb{X}$. Отметим, что существование вероятностной меры на всём пространстве объектов не предполагается, и даже само это пространство не вводится.

В слабой аксиоматике событиями являются подмножества разбиений выборки $\mathbb{X}$. Точнее, для произвольного предиката $\beta \colon \mathbb{X}_L^\ell \to \{\textit{истина}, \textit{ложь}\}$ вероятность события

$\beta(X)$ определяется как доля разбиений, при которых $\beta(X)$ истинно:

$$\mathsf{P}\left[\beta(X)\right] = \frac{1}{C_L^\ell} \sum_{X \in \mathbb{X}_L^\ell} \left[\beta(X)\right].$$

Здесь и далее квадратные скобки (нотация Айверсона [6]) означают преобразование логического выражения в число 0 или 1 по правилам $[\textit{истина}] = 1$, $[\textit{ложь}] = 0$.

Цель данной работы — получение точных оценок *вероятности переобучения* для метода минимизации эмпирического риска μ:

$$Q_\varepsilon(\mu, \mathbb{X}) = \mathsf{P}\left[\delta_\mu(X) \geqslant \varepsilon\right]. \tag{2.1}$$

Введение слабой аксиоматики мотивируется следующими соображениями.

Во-первых, в задачах анализа данных выборки могут быть только конечными, будь то уже известные наблюдаемые данные, или скрытые данные, которые станут известны в будущем. В некоторых задачах число предсказаний k настолько мало́, что вводить «вероятность ошибки» как предел частоты ошибок при $k \to \infty$ просто некорректно. Слабая аксиоматика позволяет получать содержательные результаты, справедливые при любых конечных ℓ и k, чисто комбинаторными методами, причём во многих случаях оценки получаются точными. Понятие «вероятность ошибки» в слабой аксиоматике вообще не определяется. Качество алгоритмов характеризуется частотой их ошибок на конечных выборках. Понятие переобученности определяется как отклонение частоты ошибок в двух подвыборках, а не как отклонение частоты ошибок от её вероятности. Заметим, что такой подход не является новым в статистической теории. Основополагающие работы Вапника и Червоненкиса [3] также основывались на оценках уклонения частот в двух подвыборках.

Во-вторых, вероятности, определяемые через «долю разбиений выборки», легко оценивать эмпирически. Для этого можно применять, в частности, метод Монте-Карло, заменяя среднее по всем разбиениям средним по случайному подмножеству разбиений. Эта методика напоминает скользящий контроль [10, 13], но отличается от него тем, что оценивается не эмпирическое среднее частоты ошибок на контроле, а эмпирическое распределение переобученности δ_μ. Именно это позволило в [22] выделить и численно сравнить четыре основных фактора завышенности классических оценок Вапника-Червоненкиса. Вообще, в слабой аксиоматике яснее прослеживается связь теоретических оценок с эмпирическими методиками типа перестановочных тестов или скользящего контроля.

В-третьих, оценки вида $Q_\varepsilon(\mu, \mathbb{X}) \leqslant \eta(\varepsilon)$ при необходимости легко переносятся из слабой аксиоматики в сильную (колмогоровскую). Для этого достаточно взять математическое ожидание по полной выборке $\mathbb{X}$ от обеих частей неравенства:

$$\mathsf{P}_{\mathbb{X}}\big\{\delta_\mu(X) \geqslant \varepsilon\big\} = \mathsf{E}_{\mathbb{X}} Q_\varepsilon(\mu, \mathbb{X}) \leqslant \mathsf{E}_{\mathbb{X}} \eta(\varepsilon),$$

разумеется, дополнительно предположив, что выборка $\mathbb{X}$ — простая. Если оценка $\eta(\varepsilon)$ не зависит от полной выборки $\mathbb{X}$, то она непосредственно переносится из слабой аксиоматики в сильную. Если оценка зависит от некоторой функции полной выборки $T(\mathbb{X})$, то значение этой функции надо либо интерпретировать как априорное знание, либо оценивать по наблюдаемой части выборки. Во всех этих случаях вид оценки не меняется при переходе от слабой аксиоматики к сильной. Поэтому вполне допустимо полностью оставаться в рамках слабой аксиоматики.

В слабой аксиоматике особую роль играет *гипергеометрическая функция вероятности*

$$h_L^{\ell,m}(s) = C_m^s C_{L-m}^{\ell-s} / C_L^\ell,$$

выражающая долю способов, которыми можно выбрать ℓ объектов без возвращений из генеральной выборки длины L, содержащей m ошибок, получив при этом s ошибок. Функция $h_L^{\ell,m}(s)$ имеет три целочисленных параметра L, $\ell = 0,\ldots,L$, $m = 0,\ldots,L$ и аргумент s, принимающий целые значения от $s_0 = \max\{0, m-k\}$ до $s_1 = \min\{m,\ell\}$. При всех других целых значениях m и s договоримся доопределять биномиальные коэффициенты C_m^s и функцию $h_L^{\ell,m}(s)$ нулём.

Введём стандартным образом *гипергеометрическую функцию распределения*

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

Следующее утверждение является общеизвестным классическим фактом. Здесь мы его лишь переформулируем в слабой аксиоматике.

Лемма 2.1. *Пусть алгоритм a допускает m ошибок на генеральной выборке, $n(a,\mathbb{X}) = m$. Тогда вероятность того, что алгоритм a допускает s ошибок на выборке X, описывается гипергеометрической функцией вероятности:*

$$\mathsf{P}\big[n(a,X) = s\big] = \mathsf{P}\big[n(a,\bar{X}) = m - s\big] = h_L^{\ell,m}(s)\,,$$

а вероятность большого уклонения частот ошибок алгоритма a описывается гипергеометрической функцией распределения:

$$\mathsf{P}\big[\delta(a,X) \geqslant \varepsilon\big] = H_L^{\ell,m}\left(\tfrac{\ell}{L}(m - \varepsilon k)\right). \tag{2.2}$$

Замечание 2.1. Значение $\left\lfloor \frac{\ell}{L}(m-\varepsilon k)\right\rfloor$ равно наибольшему числу ошибок алгоритма a на наблюдаемой выборке, при котором уклонение частот $\delta(a,X)$ превышает ε.

Замечание 2.2. При $\ell, k \to \infty$ правая часть (2.2) стремится к нулю. Другими словами, величина уклонения $\delta(a,X)$ стремится по вероятности к нулю. Таким образом, лемма 2.1 выражает закон больших чисел в слабой аксиоматике. Известно много верхних оценок скорости сходимости в законе больших чисел — неравенства Чебышёва, Хёффдинга, Чернова и др. [16]. Однако все они являются завышенными оценками некоторого асимптотического аналога точного равенства (2.2).

Замечание 2.3. В правой части (2.2) стоит величина $m = n(a,X) + n(a,\bar{X})$, зависящая от числа ошибок в скрытой выборке $n(a,\bar{X})$, которое как раз и требуется предсказать, зная число ошибок в наблюдаемой выборке. Это кажущееся противоречие разрешается довольно просто. Переформулируем оценку (2.2) в эквивалентном виде: «с вероятностью $\eta(\varepsilon) = H_L^{\ell,m}\left(\frac{\ell}{L}(m-\varepsilon k)\right)$ справедливо неравенство $\delta(a,X) \geqslant \varepsilon(\eta)$», где $\varepsilon(\eta)$ — функция, обратная к $\eta(\varepsilon)$. Записав это неравенство как уравнение относительно $n(a,\bar{X})$ и решив его численно, получаем искомую точную верхнюю оценку либо для $n(a,\bar{X})$, либо для уклонения частот $\delta(a,X)$. Аналогичная техника обращения активно использовалась в [15, 14] при получении оценок обобщающей способности, правда, для биномиального распределения, а не для гипергеометрического.

Замечание 2.4. При больших значениях L, ℓ, m гипергеометрическая функция вероятности довольно точно аппроксимируется биномиальной $h_L^{\ell,m}(s) \to C_\ell^s p^s (1-p)^s$, где $p = \frac{m}{L}$ есть вероятность ошибки. В ряде работ оценки обобщающей способности строятся именно на основе биномиальных распределений [15, 14].

3 Предшествующие результаты и эксперименты

Классические оценки Вапника-Червоненкиса [20] легко переформулируются в слабой аксиоматике. По сути дела, в исходных работах [2, 3, 4, 1] они именно так и выводились[1]. В явном виде переформулировка была сделана в [22]:

$$Q_\varepsilon \leqslant |A| \max_{m=0,\ldots,L} H_L^{\ell,m}\left(\tfrac{\ell}{L}(m - \varepsilon k)\right), \tag{3.1}$$

где $|A|$ — число различных векторов ошибок, порождаемых методом обучения μ на выборке $\mathbb{X}$, называемое также *коэффициентом разнообразия* (shattering coefficient). Хорошо известно, что на практике эта оценка чрезвычайно завышена. Основной причиной завышенности принято считать применение *неравенства Буля* (union bound) в процессе получения этой оценки. Многочисленные оценки, по разным направлениям улучшающие данный результат, также не обходятся без неравенства Буля.

Для выяснения причин завышенности (3.1) функционал Q_ε и некоторые его верхние оценки, взятые с промежуточных шагов вывода (3.1), были измерены экспериментально методом Монте-Карло по случайному подмножеству разбиений [22]. Измерения выполнялись на семи реальных задачах классификации из репозитория UCI. В качестве метода обучения μ использовался алгоритм индукции правил (rule induction). Были выделены четыре фактора завышенности, два из которых оказались наиболее существенными.

Эффект расслоения. В каждой задаче семейство *расслаивается* по уровням частоты ошибок, причём основная масса алгоритмов концентрируется в области наихудших частот (около 50%). Лишь малая доля алгоритмов имеют высокие шансы быть полученными в результате обучения, остальная часть семейства фактически не задействуется. В то же время, понятие VC-размерности и другие распространённые меры сложности основаны на подсчёте количества алгоритмов в семействе, без учёта вероятностей их получения. Пренебрежение эффектом расслоения может ухудшать оценку Q_ε в 10^2–10^5 раз.

Эффект сходства. Для каждого алгоритма в семействе существует определённое количество похожих на него алгоритмов. Чем больше в семействе схожих алгоритмов, тем сильнее завышено неравенство Буля. Пренебрежение эффектом сходства может ухудшать оценку Q_ε в 10^3–10^4 раз.

Остальные факторы завышенности носят технический характер, вместе имеют порядок 10^1–10^2 и относительно легко устраняются.

---

[1] Дальнейшее развитие статистической теории обучения пошло по пути отказа от комбинаторной техники вывода оценок в пользу более продвинутого математического аппарата теории вероятностей: изопериметрических неравенств, теории эмпирических процессов. При этом проблема завышенности оценок «заметалась под ковёр»: количество промежуточных оценок возрастало, и становилось всё труднее проследить, на каком именно из знаков $\leqslant$ происходит наибольшая потеря точности. Эта тенденция хорошо видна по последним наиболее заметным работам, в том числе обзорного характера [21, 9, 14, 8, 17].

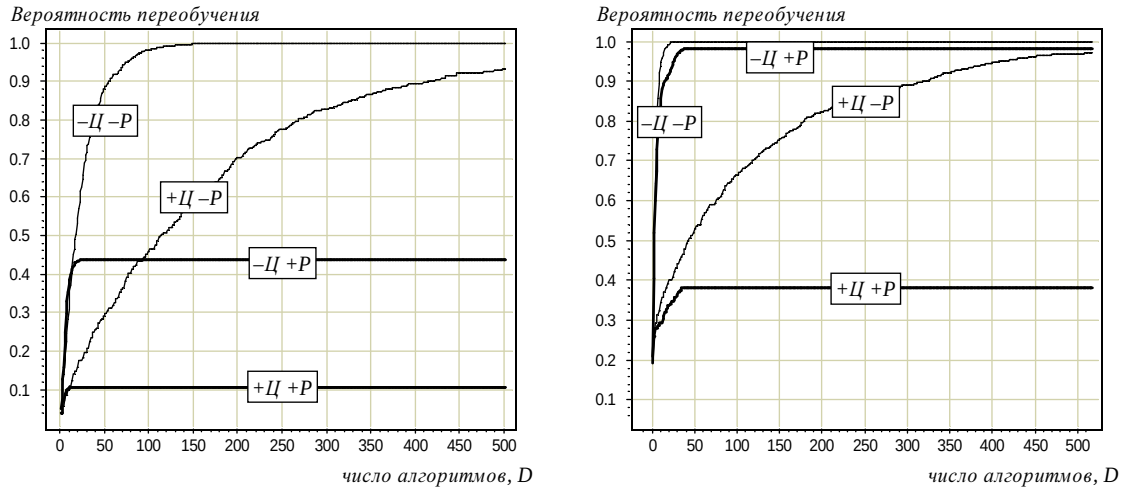

Рис. 1. Зависимость вероятности переобучения Q_ε от числа алгоритмов D при $m = 10$ («лёгкая задача», левый график) и $m = 50$ («трудная задача», правый график). Условные обозначения: $+Ц$ — наличие цепочки, $-Ц$ — отсутствие цепочки, $+Р$ — наличие расслоения, $-Р$ — отсутствие расслоения.

Явление расслоения и связанные с ним оценки переобучения (shell bounds) изучались в работах Дж. Лэнгфорда [15, 14]. К сожалению, они довольно громоздки, требуют имитационного моделирования методом Монте-Карло и не дают радикального выигрыша в точности по сравнению с оценками Вапника-Червоненкиса. Другой подход связан с введением *алгоритмической функции везения* (algorithmic luckiness function), с помощью которой все алгоритмы семейства ранжируются по их предпочтительности относительно заданной выборки; затем, следуя классической VC-теории, применяется неравенство Буля и оцениваются мощности покрытия (covering numbers) [11].

Влияние сходства алгоритмов на вероятность переобучения почти не изучалось, за исключением [7, 19], где также не удалось добиться радикального улучшения оценок. Все эти результаты свидетельствуют о том, что учёт расслоения и сходства по отдельности не даёт существенного выигрыша в точности оценок.

Разработка нового подхода началась с поиска простого, но практически важного примера, для которого эффекты расслоения и сходства можно было бы учесть совместно. Таким примером стала *цепочка алгоритмов* — последовательность векторов ошибок, в которой каждый последующий вектор отличается от предыдущего только на одном объекте. Цепочка является простейшим частным случаем *связного семейства* алгоритмов [19]. Связным называется такое семейство, в котором для каждого алгоритма можно указать другой алгоритм, у которого вектор ошибок отличается только на одном объекте. Связные семейства часто встречаются на практике. К ним относятся многие алгоритмы классификации с непрерывной по параметрам разделяющей поверхностью: линейные классификаторы, машины опорных векторов с непрерывными ядрами, нейронные сети с непрерывными функциями активации, решающие деревья с пороговыми условиями, и многие другие. Цепочки возникают, в частности, при непрерывном изменении одного из параметров, либо при непрерывном изменении вектора параметров вдоль некоторой непрерывной траектории.

Вторая серия экспериментов [23] проводилась на модельных данных. Строились

цепочки двух типов. В *цепочке с расслоением* задавался лучший алгоритм a_0, допускающий m ошибок на генеральной выборке. Каждый следующий вектор ошибок a_d, $d = 1, \ldots, D$, генерировался из a_{d-1} путём инверсии одной случайно выбранной координаты. В *цепочке без расслоения* число ошибок на генеральной выборке, чередуясь, принимало значения m и $m+1$. Для каждой цепочки строилась соответствующая ей *не-цепочка* из векторов a'_d с таким же числом ошибок, $n(a'_d, \mathbb{X}) = n(a_d, \mathbb{X})$, но случайным образом перепутанными координатами; тем самым разрушалась структура сходства алгоритмов. Итого, строилось четыре последовательности векторов ошибок с одинаковыми параметрами D и m. Их сопоставление позволило разделить влияние сходства и расслоения на вероятность переобучения. Оценки Q_ε вычислялись методом Монте-Карло по 1000 случайных разбиений при $\ell = k = 100$, $m \in \{10, 50\}$, $\varepsilon = 0.05$. Зависимости Q_ε от числа алгоритмов в цепочке D показаны на рис. 1. Видно, что свойство сходства (наличие цепочки) заметно снижает темп роста этой зависимости, свойство расслоения опускает уровень горизонтальной асимптоты, особенно для лёгких задач (левый график), и только при наличии у семейства обоих свойств достигаются приемлемые численные значения вероятности переобучения. Для трудных задач (правый график) свойство расслоения в отдельности уже практически не влияет на оценку, однако в паре со свойством сходства, опять-таки, приводит к существенному её улучшению. Заметим, что по Вапнику-Червоненкису зависимость $Q_\varepsilon(D)$ линейно возрастает и вообще не имеет горизонтальной асимптоты.

Известные в теории статистического обучения подходы не дают точных оценок вероятности переобучения для цепочек с расслоением. В то же время, такие оценки могут быть получены путём несложных комбинаторных рассуждений. Их обобщение и привело к основному результату, описанному в следующем разделе.

4 Общие оценки вероятности переобучения

В данном разделе выводятся две точные оценки вероятности переобучения, основанные на предположении, что для каждого вектора ошибок можно в явном виде указать множество разбиений $X \cup \bar{X} = \mathbb{X}$, при которых алгоритм с данным вектором ошибок является результатом обучения.

Гипотеза 4.1. *Пусть для каждого вектора ошибок $a \in A$ можно указать пару непересекающихся подмножеств $X_a \subset \mathbb{X}$ и $X'_a \subset \mathbb{X}$ таких, что*

$$\big[\mu X = a\big] = \big[X_a \subseteq X\big]\big[X'_a \subseteq \bar{X}\big]. \tag{4.1}$$

Гипотеза 4.1 означает, что для каждого алгоритма a можно указать множество *производящих* (эталонных, опорных) объектов X_a, которые обязаны присутствовать в обучающей выборке X, и множество *разрушающих* (мешающих, шумовых) объектов X'_a, которых не должно быть в обучающей выборке, чтобы метод μ предпочёл алгоритм a. Все остальные объекты $\mathbb{X}\backslash(X_a \cup X'_a)$ будем называть *нейтральными* для алгоритма a. Их присутствие в обучающей выборке не влияет на результат обучения. В следующих разделах будут приведены нетривиальные примеры семейств, для которых гипотеза 4.1 выполняется.

Введём следующие обозначения:

$L_a = L - |X_a| - |X'_a|$ — число объектов, нейтральных для алгоритма a;

$\ell_a = \ell - |X_a|$ — число обучающих объектов, нейтральных для алгоритма a;
$k_a = k - |X'_a|$ — число контрольных объектов, нейтральных для алгоритма a.

Лемма 4.1. *Для любой выборки X справедливо тождество*

$$\sum_{a\in A} \big[\mu X{=}a\big] = 1. \tag{4.2}$$

Доказательство с очевидностью вытекает из того, что для любой выборки X метод μ выбирает один и только один алгоритм.

Тождество (4.2) может использоваться для проверки того, что условия в правой части (4.1) сформулированы корректно.

Лемма 4.2. *Если гипотеза 4.1 справедлива, то вероятность получить в результате обучения вектор ошибок a равна*

$$P_a = \mathsf{P}\big[\mu X{=}a\big] = \frac{C^{\ell_a}_{L_a}}{C^{\ell}_{L}}.$$

Доказательство. Согласно гипотезе 4.1

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

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

Вероятность переобучения Q_ε легко записать, зная для каждого вектора ошибок a из A вероятность P_a получить его в результате обучения и вероятность $Q_a(\varepsilon)$ большого уклонения частот при условии, что получен вектор ошибок a. По формуле полной вероятности

$$Q_\varepsilon = \sum_{a\in A} P_a Q_a(\varepsilon).$$

Условная вероятность $Q_a(\varepsilon)$ даётся леммой 2.1, если учесть, что при фиксированном алгоритме a подмножества X_a и X'_a не участвуют в разбиениях. Рассматривая L_a нейтральных объектов и всевозможные их разбиения на ℓ_a обучающих и k_a контрольных, получим:

$$Q_a(\varepsilon) = H^{\ell_a, m_a}_{L_a}\left(s_a(\varepsilon)\right);$$
$$m_a = n(a,\mathbb{X}) - n(a,X_a) - n(a,X'_a);$$
$$s_a(\varepsilon) = \tfrac{\ell}{L}\big(n(a,\mathbb{X}) - \varepsilon k\big) - n(a,X_a);$$

где m_a — число ошибок алгоритма a на нейтральных объектах, $s_a(\varepsilon)$ — наибольшее число ошибок алгоритма a на подвыборке нейтральных обучающих объектов $X \setminus X_a$, при котором имеет место большое уклонение частот ошибок: $\delta(a,X) \geqslant \varepsilon$.

Итак, точная оценка вероятности переобучения Q_ε может быть легко выписана, если для каждого алгоритма известны множества производящих и разрушающих объектов. Более строгий комбинаторный вывод точной оценки Q_ε представлен ниже.

Теорема 4.3. *Если гипотеза 4.1 справедлива, то вероятность переобучения вычисляется по формуле*

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

Доказательство. Рассмотрим функционал Q_ε. Введём в (2.1) под знак суммирования по X ещё два вспомогательных суммирования: первый — по всем векторам ошибок a из A при условии $\mu X = a$, второй — по всем значениям s числа ошибок алгоритма a на подвыборке $X\backslash X_a$. Очевидно, значение Q_ε от этого не изменится:

$$Q_\varepsilon = \mathsf{P}\big[\delta_\mu(X) \geqslant \varepsilon\big] = \mathsf{P}\sum_{a\in A}\big[\mu X{=}a\big]\sum_{s=0}^{\ell_a}\big[n(a, X\backslash X_a) = s\big]\big[\delta(a, X) \geqslant \varepsilon\big]. \tag{4.3}$$

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

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

следовательно,

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

Подставим полученное выражение в (4.3), затем заменим $\big[\mu X = a\big]$ правой частью равенства (4.1) и переставим знаки суммирования (очевидно, P также можно рассматривать как суммирование):

$$Q_\varepsilon = \sum_{a\in A}\sum_{s=0}^{\ell_a} \underbrace{\mathsf{P}\big[X_a \subseteq X\big]\big[X'_a \subseteq \bar{X}\big]\big[n(a, X\backslash X_a) = s\big]}_{N(a)}\big[s \leqslant s_a(\varepsilon)\big]. \tag{4.4}$$

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

Для наглядности представим вектор ошибок a разбитым на шесть блоков:

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

Число ошибок алгоритма a на объектах, не попадающих ни в X_a, ни в X'_a, равно m_a. Существует $C_{m_a}^s$ способов выбрать из них s объектов, которые попадут в $X\backslash X_a$. Для каждого из этих способов имеется ровно $C_{L_a - m_a}^{\ell_a - s}$ способов выбрать $\ell_a - s$ объектов, на которых алгоритм a не допускает ошибку, и которые также попадут в $X\backslash X_a$. Тем самым однозначно определяется состав выборки $X\backslash X_a$, а, значит, и состав выборки $\bar{X}\backslash X'_a$. Таким образом,

$$N(a) = \frac{C_{m_a}^s C_{L_a - m_a}^{\ell_a - s}}{C_L^\ell}.$$

Подставим полученное выражение в (4.4) и выделим в нём формулу гипергеометрической функции вероятности:

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

Теорема доказана. ■

Далеко не во всех случаях условие того, что вектор ошибок a является результатом обучения, выражается в столь простом виде (4.1). Доказанная теорема легко обобщается на более сложный случай, когда для каждого a существует несколько альтернативных вариантов выделить множества эталонных и шумовых объектов, и при каждом из этих вариантов a является результатом обучения.

Гипотеза 4.2. *Пусть для каждого вектора ошибок $a \in A$ можно указать такое множество индексов V_a, и для каждого индекса $v \in V_a$ такую пару непересекающихся подмножеств $X_{av}, X'_{av} \subset \mathbb{X}$ и такой коэффициент $c_{av} \in \mathbb{R}$, что*

$$\big[\mu X{=}a\big] = \sum_{v\in V_a} c_{av} \big[X_{av} \subseteq X\big] \big[X'_{av} \subseteq \bar{X}\big]. \tag{4.5}$$

Введём для каждого вектора ошибок a и индекса v следующие обозначения:

$\ell_{av} = \ell - |X_{av}|; \quad k_{av} = k - |X'_{av}|; \quad L_{av} = \ell_{av} + k_{av};$

$m_{av} = n(a,\mathbb{X}) - n(a,X_{av}) - n(a,X'_{av});$

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

Лемма 4.4. *Если гипотеза 4.2 справедлива, то вероятность получить в результате обучения вектор ошибок a равна*

$$\mathsf{P}\big[\mu X{=}a\big] = \sum_{v\in V_a} c_{av} P_{av}; \qquad P_{av} = \frac{C_{L_{av}}^{\ell_{av}}}{C_L^\ell}.$$

Доказательство. Аналогично доказательству леммы 4.2,

$$\mathsf{P}\big[\mu X{=}a\big] = \sum_{v\in V_a} c_{av}\, \mathsf{P}\big[X_{av} \subseteq X\big] \big[X'_{av} \subseteq \bar{X}\big] = \sum_{v\in V_a} c_{av} P_{av}. \qquad ■$$

Теорема 4.5. *Если гипотеза 4.2 справедлива, то вероятность переобучения вычисляется по формуле*

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

Доказательство. Аналогично доказательству теоремы 4.3 вероятность переобучения приводится к выражению

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

которое отличается от (4.4) введением знака суммирования по v, двойными индексами $_{av}$ вместо одинарных $_a$, и появлением коэффициентов c_{av}. Оставшаяся часть доказательства проводится по аналогии с доказательством теоремы 4.3. ■

В следующих разделах рассматривается применение доказанных формул к некоторым специальным семействам алгоритмов.

5 Монотонная цепочка алгоритмов

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

Определение 5.1. *Последовательность векторов ошибок $A = \{a_0, a_1, \ldots, a_D\}$ называется цепочкой алгоритмов, если $\rho(a_{d-1}, a_d) = 1$ для всех $d = 1, \ldots, D$, где $\rho(a, a')$ — хэммингово расстояние между векторами ошибок:*

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

В работе [23] было экспериментально показано, что вероятность переобучения цепочки существенно ниже, чем у произвольного несвязного множества алгоритмов с такими же частотами ошибок $\nu(a_d, \mathbb{X})$. В данной работе выводятся точные оценки вероятности переобучения для некоторых специальных классов цепочек.

Определение 5.2. *Цепочка алгоритмов $A = \{a_0, a_1, \ldots, a_D\}$ называется монотонной, если $n(a_d, \mathbb{X}) = m + d$ при некотором $m \geqslant 0$. Алгоритм a_0 называется лучшим в цепочке.*

Пример 5.1. Пусть $\mathbb{X}$ — множество точек в $\mathbb{R}^n$; $\mathbb{A}$ — семейство линейных алгоритмов классификации — параметрических отображений из $\mathbb{X}$ в $\{-1, +1\}$ вида

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

с параметром $w \in \mathbb{R}^n$. Пусть функция потерь имеет вид $I(a, x) = \big[a(x, w) \neq y(x)\big]$, где $y(x)$ — истинная классификация объекта x, и множество объектов $\mathbb{X}$ линейно разделимо, т. е. существует $w^* \in \mathbb{R}^n$, при котором алгоритм $a(x, w^*)$ не допускает ошибок на $\mathbb{X}$. Тогда, при некоторых дополнительных предположениях технического характера, множество векторов ошибок $\big\{\big(a(x, w^* + t\delta)\big)_{\mathbb{X}} \colon t \in [0, +\infty)\big\}$ образует монотонную цепочку для любого направляющего вектора $\delta \in \mathbb{R}^n$, за исключением некоторого конечного множества векторов. При этом $m = 0$.

Теорема 5.1. *Пусть $A = \{a_0, a_1, \ldots, a_D\}$ — монотонная цепочка; $L \geqslant m + D$; метод обучения μ является методом минимизации эмпирического риска, причём если минимум в (1.1) достигается на нескольких различных алгоритмах, то выбирается алгоритм с бóльшим числом ошибок на генеральной выборке (тем самым будет получена точная верхняя оценка вероятности переобучения). Тогда:*

1) в случае $D \geqslant k$

$$Q_\varepsilon = \sum_{d=0}^{k} P_d H_{L-d-1}^{\ell-1,m}(s_d(\varepsilon)); \qquad P_d = \frac{C_{L-d-1}^{\ell-1}}{C_L^\ell};$$

2) в случае $D < k$

$$Q_\varepsilon = \sum_{d=0}^{D-1} P_d H_{L-d-1}^{\ell-1,m}\left(s_d(\varepsilon)\right) \;+\; P_D H_{L-D}^{\ell,m}\left(s_d(\varepsilon)\right);$$

$$P_d = \frac{C_{L-d-1}^{\ell-1}}{C_L^\ell},\ d = 0,\ldots,D-1; \qquad P_D = \frac{C_{L-D}^{\ell}}{C_L^\ell},$$

где P_d — вероятность получить алгоритм a_d методом μ; $s_d(\varepsilon) = \frac{\ell}{L}(m + d - \varepsilon k)$.

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

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

$$\begin{array}{rccccccccc}
 & x_1 & x_2 & x_3 & & x_D & & \overbrace{\qquad}^{m} & \\
a_0 = (& 0, & 0, & 0, & \ldots & 0, & 0,\ldots,0, & 1,\ldots,1 &); \\
a_1 = (& 1, & 0, & 0, & \ldots & 0, & 0,\ldots,0, & 1,\ldots,1 &); \\
a_2 = (& 1, & 1, & 0, & \ldots & 0, & 0,\ldots,0, & 1,\ldots,1 &); \\
a_3 = (& 1, & 1, & 1, & \ldots & 0, & 0,\ldots,0, & 1,\ldots,1 &); \\
\ldots & & & & \ldots & & \ldots & \ldots & \\
a_D = (& 1, & 1, & 1, & \ldots & 1, & 0,\ldots,0, & 1,\ldots,1 &);
\end{array}$$

При рассмотрении алгоритма a_d возможны три случая.

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

$$[\mu X{=}a_d] = 0.$$

2. Если $d = D < k$, то метод μ выберет наихудший алгоритм в цепочке a_D тогда и только тогда, когда все объекты $\{x_1,\ldots,x_D\}$ будут находиться в контрольной подвыборке $\bar{X}$. В этом случае

$$[\mu X{=}a_d] = [x_1,\ldots,x_D \in \bar{X}].$$

3. Во всех остальных случаях метод μ выберет алгоритм a_d, если только все объекты $\{x_1,\ldots,x_d\}$ будут находиться в контрольной подвыборке $\bar{X}$, а объект x_{d+1} — в обучающей подвыборке X. В этом случае

$$[\mu X{=}a_d] = [x_{d+1} \in X][x_1,\ldots,x_d \in \bar{X}].$$

Теперь можно применить теорему 4.3.

Если $D \geqslant k$, то алгоритму a_d соответствуют следующие значения параметров (для упрощения обозначений вместо двойных индексов ℓ_{a_d} будем использовать одинарные ℓ_d): $\ell_d = \ell - 1,\ k_d = k - d,\ m_d = m + d - d = m,\ s_d(\varepsilon) = \frac{\ell}{L}(m + d - \varepsilon k)$. Отсюда получаем утверждение теоремы для случая $D \geqslant k$.

Если $D < k$, то алгоритмам $a_0,\ldots,a_{D-1}$ соответствуют те же значения параметров, что и при $D \geqslant k$. Для наихудшего алгоритма a_D отличается только параметр $\ell_D = \ell$. Отсюда получаем утверждение теоремы для случая $D < k$. ■

Замечание 5.1. В ходе доказательства полезно проверить, что вероятности P_d вычислены корректно и в сумме дают единицу. Для случая $D \geqslant k$ проверка сводится к применению известного комбинаторного тождества:

$$\sum_{d=0}^{D} P_d = \sum_{d=0}^{k} P_d + \sum_{d=k+1}^{D} 0 = \frac{1}{C_L^\ell}\big(C_{L-1}^{\ell-1} + C_{L-2}^{\ell-1} + \cdots + C_{\ell-1}^{\ell-1}\big) = 1.$$

Для случая $D < k$ то же самое тождество приходится применить дважды, заметив, что $C_{L-D}^{\ell} = C_{L-D-1}^{\ell-1} + \cdots + C_{\ell-1}^{\ell-1}$:

$$\sum_{d=0}^{D} P_d = \frac{1}{C_L^\ell}\big(C_{L-1}^{\ell-1} + \cdots + C_{L-D}^{\ell-1} + C_{L-D}^{\ell}\big) = 1.$$

6 Унимодальная цепочка алгоритмов

Унимодальная цепочка является более реалистичной моделью однопараметрического связного семейства, по сравнению с монотонной цепочкой. 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 будет приводить, как правило, к увеличению числа ошибок.

Определение 6.1. *Множество векторов ошибок $A = \{a_0, a_1, \ldots, a_D, a'_1, \ldots, a'_{D'}\}$ называется унимодальной цепочкой, если левая ветвь $\{a_0, a_1, \ldots, a_D\}$ и правая ветвь $\{a_0, a'_1, \ldots, a'_{D'}\}$ являются монотонными цепочками. Алгоритм a_0 называется лучшим в цепочке.*

Пример 6.1 (продолжение примера 5.1). Пусть множество объектов $\mathbb{X} \subset \mathbb{R}^n$ линейно разделимо, т. е. существует линейный алгоритм классификации $a(x, w^*)$ с параметром $w^* \in \mathbb{R}^n$, не допускающий ошибок на $\mathbb{X}$. Тогда множество алгоритмов $\big\{a(x, w^* + t\delta)\colon t \in \mathbb{R}\big\}$ образует унимодальную цепочку для почти любого направляющего вектора $\delta \in \mathbb{R}^n$.

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

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

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

$$
\begin{array}{rccccccccccccc}
 & x_1 & x_2 & x_3 & & x_D & x'_1 & x'_2 & x'_3 & & x'_D & & \overbrace{\qquad}^{m} & \\
a_0 = (& 0, & 0, & 0, & \ldots & 0, & 0, & 0, & 0, & \ldots & 0, & 0,\ldots,0, & 1,\ldots,1 &); \\
a_1 = (& 1, & 0, & 0, & \ldots & 0, & 0, & 0, & 0, & \ldots & 0, & 0,\ldots,0, & 1,\ldots,1 &); \\
a_2 = (& 1, & 1, & 0, & \ldots & 0, & 0, & 0, & 0, & \ldots & 0, & 0,\ldots,0, & 1,\ldots,1 &); \\
a_3 = (& 1, & 1, & 1, & \ldots & 0, & 0, & 0, & 0, & \ldots & 0, & 0,\ldots,0, & 1,\ldots,1 &); \\
\ldots & & & & \ldots & & & & & \ldots & & \ldots & \ldots & \\
a_D = (& 1, & 1, & 1, & \ldots & 1, & 0, & 0, & 0, & \ldots & 0, & 0,\ldots,0, & 1,\ldots,1 &); \\
 & & & & & & & & & & & & & \\
a'_1 = (& 0, & 0, & 0, & \ldots & 0, & 1, & 0, & 0, & \ldots & 0, & 0,\ldots,0, & 1,\ldots,1 &); \\
a'_2 = (& 0, & 0, & 0, & \ldots & 0, & 1, & 1, & 0, & \ldots & 0, & 0,\ldots,0, & 1,\ldots,1 &); \\
a'_3 = (& 0, & 0, & 0, & \ldots & 0, & 1, & 1, & 1, & \ldots & 0, & 0,\ldots,0, & 1,\ldots,1 &); \\
\ldots & & & & \ldots & & & & & \ldots & & \ldots & \ldots & \\
a'_D = (& 0, & 0, & 0, & \ldots & 0, & 1, & 1, & 1, & \ldots & 1, & 0,\ldots,0, & 1,\ldots,1 &);
\end{array}
$$

Теорема 6.1. *Пусть $A = \{a_0, a_1, \ldots, a_D, a'_1, \ldots, a'_D\}$ — унимодальная цепочка, $k \leqslant D$ и $2D + m \leqslant L$; метод обучения μ является методом минимизации эмпирического риска, причём если минимум в* (1.1) *достигается на нескольких различных алгоритмах, то выбирается алгоритм с бóльшим числом ошибок на генеральной выборке (тем самым будет получена точная верхняя оценка вероятности переобучения); если же из двух алгоритмов оба имеют равную частоту ошибок как на обучающей выборке, так и на генеральной, то выбирается алгоритм из левой цепочки. Тогда вероятность получить каждый из алгоритмов цепочки в результате обучения есть*

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

вероятность переобучения при $s_d(\varepsilon) = \frac{\ell}{L}(m + d - \varepsilon k)$ выражается в виде

$$
Q_\varepsilon = \frac{C_{L-2}^{\ell-2}}{C_L^\ell} H_{L-2}^{\ell-2,m}\left(s_0(\varepsilon)\right) + \sum_{d=1}^{k}\left(2\frac{C_{L-d-1}^{\ell-1}}{C_L^\ell} H_{L-d-1}^{\ell-1,m}\left(s_d(\varepsilon)\right) - \right.
$$
$$
\left. - \frac{C_{L-2d-2}^{\ell-1}}{C_L^\ell} H_{L-2d-2}^{\ell-1,m}\left(s_d(\varepsilon)\right) - \frac{C_{L-2d-1}^{\ell-1}}{C_L^\ell} H_{L-2d-1}^{\ell-1,m}\left(s_d(\varepsilon)\right)\right).
$$

Доказательство. Введём вспомогательные переменные:

$$
\begin{aligned}
\beta_d &= [x_{d+1} \in X][x_1, \ldots, x_d \in \bar{X}], \quad d = 1, \ldots, D-1; \\
\beta_D &= [x_1, \ldots, x_D \in \bar{X}]; \\
\beta'_d &= [x'_{d+1} \in X][x'_1, \ldots, x'_d \in \bar{X}], \quad d = 1, \ldots, D-1; \\
\beta'_D &= [x'_1, \ldots, x'_D \in \bar{X}].
\end{aligned}
$$

Условия $\beta_1, \ldots, \beta_D$ несовместны, причём одно из них выполнено тогда и только тогда, когда $x_1 \in \bar{X}$. Следовательно,

$$[x_1 \in X] + \beta_1 + \cdots + \beta_D = 1.$$

Аналогично,

$$[x'_1 \in X] + \beta'_1 + \cdots + \beta'_D = 1.$$

Заметим, что выражения для β_d и β'_d совпадают с условиями получения алгоритмов a_d и a'_d соответственно, если левую и правую ветви рассматривать как отдельные монотонные цепочки. В случае унимодальной цепочки эти условия приобретают более сложный вид. Если выполнено условие β_d и одновременно одно из условий $\beta'_{d+1}, \ldots, \beta'_D$, то метод μ предпочтёт соответствующий алгоритм из правой ветви (согласно условию теоремы выбирается наихудший алгоритм из всех, допускающих одинаковое наименьшее число ошибок на X). Аналогично, если выполнено условие β'_d и одновременно одно из условий $\beta_d, \ldots, \beta_D$, то метод μ предпочтёт соответствующий алгоритм из левой ветви. Обратим внимание, что алгоритмы левой ветви (согласно условию теоремы) имеют приоритет. Таким образом, условия получения всех алгоритмов унимодальной цепочки выражаются через вспомогательные переменные:

$$\begin{aligned}
&[\mu X{=}a_0] = [x_1, x'_1 \in X] = \big(1 - \beta_1 - \cdots - \beta_D\big)\big(1 - \beta'_1 - \cdots - \beta'_D\big);\\
&[\mu X{=}a_d] = \beta_d\big(1 - \beta'_{d+1} - \cdots - \beta'_D\big), \quad d = 1, \ldots, D-1;\\
&[\mu X{=}a'_d] = \beta'_d\big(1 - \beta_d - \cdots - \beta_D\big), \quad d = 1, \ldots, D-1;\\
&[\mu X{=}a_D] = \beta_D;\\
&[\mu X{=}a'_D] = \beta'_D\big(1 - \beta_D\big).
\end{aligned}$$

Непосредственной подстановкой нетрудно убедиться, что тождество (4.2) выполнено, следовательно, совокупность условий $[\mu X = a]$ определена корректно.

Найдём вероятности всех алгоритмов цепочки, применив теорему 4.5.

$$\begin{aligned}
P_0 = \mathsf{P}\big[\mu X{=}a_0\big] &= \mathsf{P}[x_1, x'_1 \in X] = \frac{C^{\ell-2}_{L-2}}{C^\ell_L};\\
P_d = \mathsf{P}\big[\mu X{=}a_d\big] &= \mathsf{P}[x_{d+1} \in X][x_1, \ldots, x_d \in \bar{X}] -\\
&\quad - \sum_{t=d+1}^{k-d} \mathsf{P}[x_{d+1}, x'_{t+1} \in X][x_1, \ldots, x_d, x'_1, \ldots, x'_t \in \bar{X}] =\\
&\quad = \frac{1}{C^\ell_L}\left(C^{\ell-1}_{L-d-1} - \sum_{t=d+1}^{k-d} C^{\ell-2}_{L-d-t-2}\right) = \frac{C^{\ell-1}_{L-d-1} - C^{\ell-1}_{L-2d-2}}{C^\ell_L};\\
P'_d = \mathsf{P}\big[\mu X{=}a'_d\big] &= \mathsf{P}[x'_{d+1} \in X][x'_1, \ldots, x'_d \in \bar{X}] -\\
&\quad - \sum_{t=d}^{k-d} \mathsf{P}[x'_{d+1}, x_{t+1} \in X][x'_1, \ldots, x'_d, x_1, \ldots, x_t \in \bar{X}] =\\
&\quad = \frac{1}{C^\ell_L}\left(C^{\ell-1}_{L-d-1} - \sum_{t=d}^{k-d} C^{\ell-2}_{L-d-t-2}\right) = \frac{C^{\ell-1}_{L-d-1} - C^{\ell-1}_{L-2d-1}}{C^\ell_L}.
\end{aligned}$$

Теперь запишем вероятность переобучения, применив теорему 4.5.

$$Q_\varepsilon = \frac{C_{L-2}^{\ell-2}}{C_L^\ell} H_{L-2}^{\ell-2,m}(s_0(\varepsilon)) + \\ + \sum_{d=1}^{k}\left(\frac{C_{L-d-1}^{\ell-1}}{C_L^\ell} H_{L-d-1}^{\ell-1,m}(s_d(\varepsilon)) - \sum_{t=d+1}^{k-d} \frac{C_{L-d-t-2}^{\ell-2}}{C_L^\ell} H_{L-d-t-2}^{\ell-2,m}(s_d(\varepsilon))\right) + \\ + \sum_{d=1}^{k}\left(\frac{C_{L-d-1}^{\ell-1}}{C_L^\ell} H_{L-d-1}^{\ell-1,m}(s_d(\varepsilon)) - \sum_{t=d}^{k-d} \frac{C_{L-d-t-2}^{\ell-2}}{C_L^\ell} H_{L-d-t-2}^{\ell-2,m}(s_d(\varepsilon))\right).$$

Полученное выражение можно упростить, заметив, что

$$\sum_{t=d+1}^{k-d} \frac{C_{L-d-t-2}^{\ell-2}}{C_L^\ell} H_{L-d-t-2}^{\ell-2,m}(s_d(\varepsilon)) = \frac{C_{L-2d-2}^{\ell-1}}{C_L^\ell} H_{L-2d-2}^{\ell-1,m}(s_d(\varepsilon));$$
$$\sum_{t=d}^{k-d} \frac{C_{L-d-t-2}^{\ell-2}}{C_L^\ell} H_{L-d-t-2}^{\ell-2,m}(s_d(\varepsilon)) = \frac{C_{L-2d-1}^{\ell-1}}{C_L^\ell} H_{L-2d-1}^{\ell-1,m}(s_d(\varepsilon));$$

Подставляя эти выражения в формулу для Q_ε, получим требуемую оценку. ■

Замечание 6.1. Нетрудно убедиться, что вероятности P_d, P'_d найдены корректно:

$$P_0 + \sum_{d=1}^{D}\big(P_d + P'_d\big) = \frac{C_{L-2}^{\ell-2}}{C_L^\ell} + \sum_{d=1}^{D} \frac{C_{L-d-1}^{\ell-1} - C_{L-2d-2}^{\ell-1}}{C_L^\ell} + \frac{C_{L-d-1}^{\ell-1} - C_{L-2d-1}^{\ell-1}}{C_L^\ell} = \\ = \frac{C_{L-2}^{\ell-2}}{C_L^\ell} + \frac{1}{C_L^\ell}\sum_{d=1}^{D}\Big(2C_{L-d-1}^{\ell-1} - C_{L-2d-2}^{\ell-1} - C_{L-2d-1}^{\ell-1}\Big) = \\ = \frac{1}{C_L^\ell}\Big(C_{L-2}^{\ell-2} + 2\big(C_{L-2}^{\ell-1} + \cdots + C_{\ell-1}^{\ell-1}\big) - \big(C_{L-3}^{\ell-1} + \cdots + C_{\ell-1}^{\ell-1}\big)\Big) = \\ = \frac{1}{C_L^\ell}\Big(C_{L-2}^{\ell-2} + 2C_{L-1}^{\ell} - C_{L-2}^{\ell}\Big) = 1.$$

7 Единичная окрестность лучшего алгоритма

Другим примером связного семейства является единичная окрестность лучшего алгоритма. Это искусственная постановка задачи, но она интересна по двум причинам. Во-первых, это «экстремальный» случай, когда алгоритмы максимально близки друг к другу, и классические оценки, основанные на неравенстве Буля, максимально завышены. Во-вторых, это первый шаг на пути к общим точным оценкам вероятности переобучения. Следующим шагом должно стать увеличение радиуса окрестности.

Определение 7.1. *Множество векторов ошибок $A = \{a_0, a_1, \ldots, a_D\}$ называется единичной окрестностью алгоритма a_0, если для всех $d = 1, \ldots, D$ векторы ошибок a_d попарно различны, $n(a_d, \mathbb{X}) = n(a_0, \mathbb{X}) + 1$ и $\rho(a_0, a_d) = 1$. Алгоритм a_0 называется лучшим алгоритмом или центром окрестности.*

Теорема 7.1. *Пусть $A = \{a_0, a_1, \ldots, a_D\}$ — единичная окрестность алгоритма a_0; $m = n(a_0, \mathbb{X})$; $L \geqslant m + D$; метод обучения μ является методом минимизации эмпирического риска, причём если минимум в (1.1) достигается на нескольких различных алгоритмах, то выбирается алгоритм с меньшим номером. Тогда*

$$Q_\varepsilon = P_0 H_{L-D}^{\ell-D,m}\left(\tfrac{\ell}{L}(m-\varepsilon k)\right) + \sum_{d=1}^{D} P_d H_{L-d}^{\ell-d+1,m}\left(\tfrac{\ell}{L}(m+1-\varepsilon k)\right);$$

$$P_0 = \frac{C_{L-D}^k}{C_L^k}; \quad P_d = \frac{C_{L-d}^{k-1}}{C_L^k},\ d = 1, \ldots, D;$$

где P_d — вероятность получить алгоритм a_d в результате обучения.

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

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

$$\begin{array}{rccccccc}
 & x_1 & x_2 & x_3 & & x_D & & \overbrace{\qquad}^{m} \\
a_0 = (& 0, & 0, & 0, & \ldots & 0, & 0,\ldots,0, & 1,\ldots,1 \ ); \\
a_1 = (& 1, & 0, & 0, & \ldots & 0, & 0,\ldots,0, & 1,\ldots,1 \ ); \\
a_2 = (& 0, & 1, & 0, & \ldots & 0, & 0,\ldots,0, & 1,\ldots,1 \ ); \\
a_3 = (& 0, & 0, & 1, & \ldots & 0, & 0,\ldots,0, & 1,\ldots,1 \ ); \\
\ldots & & & & \ldots & & \ldots & \ldots \\
a_D = (& 0, & 0, & 0, & \ldots & 1, & 0,\ldots,0, & 1,\ldots,1 \ );
\end{array}$$

Нетрудно видеть, что множество разбиений, при которых метод μ выбирает алгоритм a_d, представляется в следующем виде:

$$\begin{aligned}
&\big[\mu X{=}a_0\big] = \big[x_1, \ldots, x_D \in X\big];\\
&\big[\mu X{=}a_d\big] = \big[x_1, \ldots, x_{d-1} \in X\big]\big[x_d \in \bar{X}\big], \quad d = 1, \ldots, D.
\end{aligned}$$

Параметры для подстановки в формулу теоремы 4.3:

$$\begin{aligned}
&\ell_0 = \ell - D; \quad k_0 = k; \quad m_0 = m; \quad s_0(\varepsilon) = \tfrac{\ell}{L}(m - \varepsilon k);\\
&\ell_d = \ell - d + 1; \quad k_d = k - 1; \quad m_d = m; \quad s_d(\varepsilon) = \tfrac{\ell}{L}(m + 1 - \varepsilon k); \quad d = 1, \ldots, D.
\end{aligned}$$

Подставляя эти параметры в формулу теоремы 4.3, получаем требуемое. ■

Замечание 7.1. Нетрудно убедиться, что вероятности P_d найдены корректно:

$$\sum_{d=0}^{D} P_d = \frac{1}{C_L^k}\big(C_{L-D}^k + \underbrace{C_{L-D}^{k-1} + \cdots + C_{L-1}^{k-1}}_{C_L^k - C_{L-D}^k}\big) = 1.$$

8 Пара алгоритмов

Рассмотрим ещё один «экстремальный» частный случай — когда семейство состоит только из двух алгоритмов, $\mathbb{A} = \{a_1, a_2\}$. Мы покажем, что уже в этом простейшем случае возникает переобучение и проявляются эффекты расслоения и сходства, снижающие вероятность переобучения. Таким образом, на данном примере удобно демонстрировать природу переобучения.

Точная оценка вероятности переобучения для двухэлементного семейства была представлена в [23] без доказательства. Ниже эта оценка выводится как формальное следствие теоремы 4.5.

Теорема 8.1. *Пусть в выборке $\mathbb{X}$ имеется m_0 объектов, на которых оба алгоритма допускают ошибку; m_1 объектов, на которых только a_1 допускает ошибку; m_2 объектов, на которых только a_2 допускает ошибку; и для определённости $m_1 \leqslant m_2$:*

$$a_1 = (\,1,\ldots,1,\ 1,\ldots,1,\ 0,\ldots,0,\ 0,\ldots,0\,);$$
$$a_2 = (\underbrace{1,\ldots,1}_{m_0},\ \underbrace{0,\ldots,0}_{m_1},\ \underbrace{1,\ldots,1}_{m_2},\ 0,\ldots,0\,).$$

Пусть метод обучения μ является методом минимизации эмпирического риска, причём если минимум в (1.1) достигается на обоих алгоритмах, то выбирается a_2.

Тогда для любого $\varepsilon \in [0,1)$ справедлива точная оценка:

$$Q_\varepsilon = \sum_{s_0=0}^{m_0}\sum_{s_1=0}^{m_1}\sum_{s_2=0}^{m_2} \frac{C_{m_0}^{s_0} C_{m_1}^{s_1} C_{m_2}^{s_2} C_{L-m_0-m_1-m_2}^{\ell-s_0-s_1-s_2}}{C_L^\ell} \times$$
$$\times\Big(\big[s_1 < s_2\big]\big[s_0+s_1 \leqslant \tfrac{\ell}{L}(m_0+m_1-\varepsilon k)\big] + \\ + \big[s_1 \geqslant s_2\big]\big[s_0+s_2 \leqslant \tfrac{\ell}{L}(m_0+m_2-\varepsilon k)\big]\Big). \tag{8.1}$$

Доказательство. Введём множества объектов U_1 и U_2, на которых только один из алгоритмов допускает ошибку:

$$U_1 = \big\{x \in \mathbb{X}\colon I(a_1,x)=1,\ I(a_2,x)=0\big\}, \quad |U_1| = m_1;$$
$$U_2 = \big\{x \in \mathbb{X}\colon I(a_1,x)=0,\ I(a_2,x)=1\big\}, \quad |U_2| = m_2.$$

Метод μ выбирает алгоритм a_1 тогда и только тогда, когда в обучающую подвыборку X попадает больше объектов из U_1, чем из U_2; иначе выбирается a_2. Обозначим через X_1 произвольное подмножество U_1, через X_2 — произвольное подмножество U_2.

$$[\mu X{=}a_1] = \sum_{(X_1,X_2)\in V_1} \big[X_1 \cup X_2 \subseteq X\big]\big[(U_1\backslash X_1)\cup(U_2\backslash X_2) \subseteq \bar{X}\big];$$
$$[\mu X{=}a_2] = \sum_{(X_1,X_2)\in V_2} \big[X_1 \cup X_2 \subseteq X\big]\big[(U_1\backslash X_1)\cup(U_2\backslash X_2) \subseteq \bar{X}\big];$$

где V_1 и V_2 — индексные множества, соответственно, для алгоритмов a_1 и a_2:

$$V_1 = \big\{v = (X_1,X_2)\colon X_1 \subseteq U_1,\ X_2 \subseteq U_2,\ |X_1| < |X_2|\big\};$$
$$V_2 = \big\{v = (X_1,X_2)\colon X_1 \subseteq U_1,\ X_2 \subseteq U_2,\ |X_1| \geqslant |X_2|\big\}.$$

Отсюда находятся все параметры для подстановки в формулу теоремы 4.5:

$X_{1v} = X_{2v} = X_1 \cup X_2$;
$X'_{1v} = X'_{2v} = (U_1 \backslash X_1) \cup (U_2 \backslash X_2)$;
$\ell_{1v} = \ell_{2v} = \ell - |X_1| - |X_2|$;
$k_{1v} = k_{2v} = k - m_1 - m_2 + |X_1| + |X_2|$;
$L_{1v} = L_{2v} = L - m_1 - m_2$;
$m_{1v} = m_0 + m_1 - |X_1| - |U_1 \backslash X_1| = m_0$;
$m_{2v} = m_0 + m_2 - |X_2| - |U_2 \backslash X_2| = m_0$;
$s_{1v}(\varepsilon) = \frac{\ell}{L}(m_0 + m_1 - \varepsilon k) - |X_1|$;
$s_{2v}(\varepsilon) = \frac{\ell}{L}(m_0 + m_2 - \varepsilon k) - |X_2|$;

Запишем вероятность получить алгоритм a_1 согласно теореме 4.5 и, вводя обозначения $s_1 = |X_1|$ и $s_2 = |X_2|$, заменим суммирование по подмножествам X_1, X_2 суммированием по значениям мощности этих подмножеств:

$$P_1 = \sum_{(X_1,X_2)\in V_1} \frac{C^{\ell-|X_1|-|X_2|}_{L-m_1-m_2}}{C^\ell_L} = \sum_{s_1=0}^{m_1}\sum_{s_2=0}^{m_2} \big[s_1 < s_2\big] C^{s_1}_{m_1} C^{s_2}_{m_2} \frac{C^{\ell-s_1-s_2}_{L-m_1-m_2}}{C^\ell_L}.$$

Вероятность P_2 получается из P_1 заменой $[s_1 < s_2]$ на $[s_1 \geqslant s_2]$.

Запишем вероятность переобучения согласно теореме 4.5, снова заменяя суммирование по подмножествам X_1, X_2 суммированием по s_1, s_2:

$$Q_\varepsilon = \sum_{s_1=0}^{m_1}\sum_{s_2=0}^{m_2} \big[s_1 < s_2\big] C^{s_1}_{m_1} C^{s_2}_{m_2} \frac{C^{\ell-s_1-s_2}_{L-m_1-m_2}}{C^\ell_L} H^{\ell-s_1-s_2,m_0}_{L-m_1-m_2}\left(\tfrac{\ell}{L}(m_0+m_1-\varepsilon k) - s_1\right) +$$
$$+ \sum_{s_1=0}^{m_1}\sum_{s_2=0}^{m_2} \big[s_1 \geqslant s_2\big] C^{s_1}_{m_1} C^{s_2}_{m_2} \frac{C^{\ell-s_1-s_2}_{L-m_1-m_2}}{C^\ell_L} H^{\ell-s_1-s_2,m_0}_{L-m_1-m_2}\left(\tfrac{\ell}{L}(m_0+m_2-\varepsilon k) - s_2\right).$$

Подставляя сюда гипергеометрическую функцию распределения как сумму

$$H^{\ell-s_1-s_2,m_0}_{L-m_1-m_2}(z) = \sum_{s_0=0}^{m_0} [s_0 \leqslant z] \frac{C^{s_0}_{m_0} C^{\ell-s_0-s_1-s_2}_{L-m_0-m_1-m_2}}{C^{\ell-s_1-s_2}_{L-m_1-m_2}}, \tag{8.2}$$

и упрощая полученное выражение, получаем утверждение теоремы. ■

Замечание 8.1. В выражениях (8.2) и (8.1) для сокращения записи неявно предполагается, что $C^s_m = 0$ при $s \notin \{0, \ldots, m\}$. Можно не делать этого предположения, но тогда под знаки суммирования по s_0, s_1, s_2 необходимо ввести ещё одно условие $\big[0 \leqslant \ell - s_0 - s_1 - s_2 \leqslant L - m_0 - m_1 - m_2\big]$.

Численные эксперименты с данной оценкой для пары алгоритмов уже были выполнены в [23]. Основной вывод заключался в том, что даже в этом простейшем случае возникает переобучение и проявляются эффекты расслоения и сходства, снижающие вероятность переобучения.

9 Выводы и открытые проблемы

В работе предложен новый, чисто комбинаторный, подход к проблеме обобщающей способности. Доказаны две общие теоремы, с помощью которых удалось получить точные оценки вероятности переобучения для четырёх частных семейств алгоритмов. Пока что эти примеры носят демонстрационный характер и не связаны напрямую с приложениями. Однако есть основания полагать, что данный подход позволит находить точные оценки и для более сложных случаев.

Выделение эталонных и шумовых объектов относительно каждого алгоритма представляется вполне естественным для методов обучения, основанных на отборе информативных объектов, в частности, для методов опорных векторов (SVM), радиальных базисных функций (RBF), покрывающих множеств (SCM), ближайших соседей (kNN), вывода на основе прецедентов (CBR).

Естественным обобщением цепочек алгоритмов являются многомерные сетки алгоритмов, с помощью которых, возможно, удастся описывать окрестности оптимальных алгоритмов для широкого класса связных семейств. Их изучение может дать принципиально новые оценки обобщающей способности, в которых размерность пространства будет учтена гораздо аккуратнее, чем в оценках, основанных на неравенстве Буля. Очевидно, размерность определяет «степень связности» семейства — число векторов ошибок в единичной окрестности произвольного алгоритма семейства. Можно даже предположить, что такой подход приведёт к принципиально новым понятиям сложности.

Поскольку подход является новым, некоторые относительно простые вопросы также пока остаются открытыми.

Во-первых, рассмотрены лишь точные верхние оценки вероятности переобучения, предполагающие, что в случаях неоднозначности метод минимизации эмпирического риска выбирает наихудший алгоритм из лучших на обучающей выборке. Разумеется, столь пессимистичная стратегия вряд ли реализуется на практике. Можно также рассмотреть стратегии «лучший из лучших», а также рандомизированную стратегию «случайный из лучших». Предварительные численные эксперименты на цепочках показали, что точные верхние и точные нижние оценки довольно быстро сходятся друг к другу с ростом длины выборки.

Во-вторых, полученные оценки имеют довольно громоздкий вид. Задачи получения приближённых и асимптотических оценок являются чисто техническими и намеренно оставлены за пределами данной работы.

Работа поддержана РФФИ (проект № 08-07-00422) и программой ОМН РАН «Алгебраические и комбинаторные методы математической кибернетики и информационные системы нового поколения».

Список литературы

[1] *Вапник В. Н.* Восстановление зависимостей по эмпирическим данным. — М.: Наука, 1979.

[2] *Вапник В. Н., Червоненкис А. Я.* О равномерной сходимости частот появления событий к их вероятностям // *ДАН СССР.* — 1968. — Т. 181, № 4. — С. 781–784.

[3] *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.

[4] *Вапник В. Н., Червоненкис А. Я.* Теория распознавания образов. — М.: Наука, 1974.

[5] *Воронцов К. В.* Комбинаторные обоснования обучаемых алгоритмов // *ЖВМиМФ.* — 2004. — Т. 44, № 11. — С. 2099–2112.
http://www.ccas.ru/frc/papers/voron04jvm.pdf.

[6] *Грэхем Р., Кнут Д., Паташник О.* Конкретная математика. — М.: Мир, 1998.

[7] *Bax E. T.* Similar classifiers and VC error bounds: Tech. Rep. CalTech-CS-TR97-14: 6 1997.
http://citeseer.ist.psu.edu/bax97similar.html.

[8] *Boucheron S., Bousquet O., Lugosi G.* Theory of classification: A survey of some recent advances // *ESAIM: Probability and Statistics.* — 2005. — no. 9. — Pp. 323–375.
http://www.econ.upf.edu/∼lugosi/esaimsurvey.pdf.

[9] *Bousquet O.* Concentration Inequalities and Empirical Processes Theory Applied to the Analysis of Learning Algorithms: Ph.D. thesis / Ecole Polytechnique, France. — 2002.
http://www.kyb.mpg.de/publications/pss/ps1444.ps.

[10] *Efron B.* The Jackknife, the Bootstrap, and Other Resampling Plans. — SIAM, Philadelphia, 1982.

[11] *Herbrich R., Williamson R.* Algorithmic luckiness // *Journal of Machine Learning Research.* — 2002. — no. 3. — Pp. 175–212.
http://citeseer.ist.psu.edu/article/herbrich02algorithmic.html.

[12] *Kearns M. J., Mansour Y., Ng A. Y., Ron D.* An experimental and theoretical comparison of model selection methods // 8th Conf. on Computational Learning Theory, Santa Cruz, California, US. — 1995. — Pp. 21–30.
http://citeseer.ist.psu.edu/kearns95experimental.html.

[13] *Kohavi R.* A study of cross-validation and bootstrap for accuracy estimation and model selection // 14th International Joint Conference on Artificial Intelligence, Palais de Congres Montreal, Quebec, Canada. — 1995. — Pp. 1137–1145.
http://citeseer.ist.psu.edu/kohavi95study.html.

[14] *Langford J.* Quantitatively Tight Sample Complexity Bounds: Ph.D. thesis / Carnegie Mellon Thesis. — 2002.
http://citeseer.ist.psu.edu/langford02quantitatively.html.

[15] *Langford J., McAllester D.* Computable shell decomposition bounds // Proc. 13th Annu. Conference on Comput. Learning Theory. — Morgan Kaufmann, San Francisco, 2000. — Pp. 25–34.
http://citeseer.ist.psu.edu/langford00computable.html.

[16] *Lugosi G.* On concentration-of-measure inequalities. — Machine Learning Summer School, Australian National University, Canberra. — 2003.
http://citeseer.ist.psu.edu/lugosi98concentrationmeasure.html.

[17] *Philips P.* Data-Dependent Analysis of Learning Algorithms: Ph.D. thesis / The Australian National University, Canberra. — 2005.
http://infoeng.rsise.anu.edu.au/files/petra_philips_thesis.pdf.

[18] *Rückert U., Kramer S.* Towards tight bounds for rule learning // Proc. 21th International Conference on Machine Learning, Banff, Canada. — 2004. — P. 90.
http://www.machinelearning.org/icml2004_proc.html.

[19] *Sill J.* Monotonicity and connectedness in learning systems: Ph.D. thesis / California Institute of Technology. — 1998.
http://etd.caltech.edu/etd/available/etd-09222005-110351/.

[20] *Vapnik V.* Statistical Learning Theory. — Wiley, New York, 1998.

[21] *Vayatis N., Azencott R.* Distribution-dependent Vapnik-Chervonenkis bounds // *Lecture Notes in Computer Science*. — 1999. — Vol. 1572. — Pp. 230–240.
http://citeseer.ist.psu.edu/vayatis99distributiondependent.html.

[22] *Vorontsov K. V.* Combinatorial probability and the tightness of generalization bounds // *Pattern Recognition and Image Analysis*. — 2008. — Vol. 18, no. 2. — Pp. 243–259.
http://www.springerlink.com/content/78537p01838123u7/.

[23] *Vorontsov K. V.* On the influence of similarity of classifiers on the probability of overfitting // Pattern Recognition and Image Analysis: new information technologies (PRIA-9). — Vol. 2. — Nizhni Novgorod, Russian Federation, 2008. — Pp. 303–306.
http://www.ccas.ru/frc/papers/voron08pria-conf-eng.pdf.