Лекции по теории обобщающей способности

К. В. Воронцов

21 декабря 2007 г.

Содержание

1 Теория обобщающей способности

Вопрос о качестве алгоритмов, синтезированных по конечным выборкам прецедентов, является фундаментальной проблемой *вычислительной теории обучения* (computational learning theory, COLT). На первый взгляд проблема кажется вызывающе трудной: необходимо предсказать, как построенный алгоритм будет работать в будущем, на тех объектах, о которых пока ещё ничего не известно. Тем не менее, какие-то выводы можно делать, если предположить, что обучающие и контрольные прецеденты взяты независимо из одной и той же выборки, а восстанавливаемая зависимость не изменилась от момента формирования обучающих данных до момента проверки алгоритма на контрольных данных.

При решении практических задач восстановления зависимостей приходится регулярно сталкиваться с явлением *переобучения*, когда алгоритм выдаёт правильные ответы на обучающей выборке, но при этом демонстрирует удручающе низкое качество на новых объектах, не входивших в состав обучения.

Этот эффект принято связывать с избыточной сложностью алгоритма. Чем больше у алгоритма свободных параметров, тем меньшего числа ошибок на обучении можно добиться путём их оптимизации. Однако по мере нарастания сложности модели «оптимальные» алгоритмы начинают слишком хорошо подстраиваться под конкретные данные, улавливая не только черты восстанавливаемой зависимости, но и ошибки измерения обучающей выборки, и погрешность самой модели. В результате ухудшается качество работы алгоритма вне обучающей выборки, то есть его *способность к обобщению* эмпирических фактов (generalization ability).

Из этого наблюдения можно сделать вывод, что для всякой задачи существует оптимальная сложность модели, при которой достигается наилучшее качество обобщения. Первое формальное обоснование этой гипотезы было дано в *статистической теории восстановления зависимостей по эмпирическим данным*, разработанной В. Н. Вапником и А. Я. Червоненкисом в конце 60-х — начале 70-х [2, 1]. Эта теория получила широкую мировую известность и признание в середине 80-х. В настоящее время она активно развивается и применяется для обоснования различных алгоритмов машинного обучения.

Основным результатом теории являются количественные оценки, связывающие обобщающую способность алгоритмов с некоторыми характеристиками выборки и метода обучения, в частности, с длиной обучающей выборки и сложностью семейства алгоритмов. Эти оценки необходимы не только (и не столько) для того, чтобы предсказывать, насколько хорошо будет работать построенный алгоритм. Главная цель теории — указать пути повышения обобщающей способности обучаемых алгоритмов, в частности, выбрать оптимальную структуру модели.

§1.1 Понятие обобщающей способности

Пусть задано пространство объектов X и множество возможных ответов Y. Допустим, что для любого объекта $x \in X$ и любого алгоритма $a\colon X \to Y$ можно сказать, является ли ответ $a(x)$ ошибочным. Формально говоря, существует бинарная

функция потерь $I(a,x)$, называемая *индикатором ошибки*:

$$I(a,x) = \begin{cases} 1, & \text{если ответ } a(x) \text{ ошибочный;} \\ 0, & \text{если ответ } a(x) \text{ правильный;} \end{cases}$$

В задачах обучения по прецедентам индикатор ошибки определяется через целевую функцию $y^*(x)$. В случае классификации, когда множество Y конечно, обычно полагают $I(a,x) = [a(x) \neq y^*(x)]$. В случае регрессии, когда $Y = \mathbb{R}$, можно задать $I(a,x) = \big[|a(x) - y^*(x)| \geqslant \delta\big]$, где δ — фиксированное неотрицательное число. В дальнейшем нам будет не важно, какого типа задача решается, и как именно определён индикатор ошибки.

Предполагается, что значения индикатора ошибки известны только на конечном множестве объектов $X^L = (x_i)_{i=1}^L$. *Частота ошибок* алгоритма a на произвольной подвыборке $U \subseteq X^L$ по определению есть

$$\nu(a,U) = \frac{1}{|U|} \sum_{x \in U} I(a,x).$$

Методом обучения называется отображение μ, которое произвольной конечной обучающей выборке X^ℓ ставит в соответствие определённый алгоритм $a = \mu(X^\ell)$, $a\colon X \to Y$. Будем полагать, что результат обучения $\mu(X^\ell)$ зависит только от состава обучающей выборки, но не от порядка элементов в выборке. Этому требованию удовлетворяет большинство применяемых на практике методов.

1.1.1 Функционалы полного скользящего контроля

Малая частота ошибок на обучающей выборке ещё не гарантирует, что построенный алгоритм будет столь же редко ошибаться на новых объектах. Обобщающая способность метода μ характеризуется частотой ошибок алгоритма на контрольных данных, не участвовавших в процессе обучения. Разобьём *полную* выборку X^L на две непересекающихся подвыборки: *обучающую* X^ℓ и *контрольную* X^k, $L = \ell + k$.

Опр. 1.1. *Переобученностью алгоритма $a = \mu(X^\ell)$ на паре выборок (X^ℓ, X^k) будем называть разность $\delta(\mu, X^\ell, X^k) = \nu(a, X^k) - \nu(a, X^\ell)$.*

На первый взгляд, функционал качества обучения можно было бы определить как частоту ошибок на контроле $\nu(\mu(X^\ell), X^k)$, либо как величину переобученности $\delta(\mu, X^\ell, X^k)$. Однако эти характеристики зависят от способа разбиения выборки X^L на две части, и потому не вполне адекватно характеризуют качество обучения.

Слабая вероятностная интерпретация. Обозначим через (X_n^ℓ, X_n^k), $n = 1, \ldots, N$ всевозможные разбиения выборки X^L на обучающую и контрольную подвыборки длиной ℓ и k соответственно. Число всех разбиений N равно $C_L^\ell = \frac{L!}{\ell!\,k!}$. Будем предполагать, что все разбиения имеют одинаковые шансы реализоваться на практике.

Гипотеза 1.1. *На множестве разбиений $\{1, \ldots, N\}$ задано равномерное распределение вероятностей.*

При данном предположении частота ошибок на обучении $\nu_n^\ell = \nu(\mu(X_n^\ell), X_n^\ell)$ и частота ошибок на контроле $\nu_n^k = \nu(\mu(X_n^\ell), X_n^k)$ являются функциями от номера разбиения n, следовательно, могут рассматриваться как случайные величины.

Функционал полного скользящего контроля (complete cross-validation) определяется как средняя частота ошибок на контрольных подвыборках [7, 8]:

$$Q_c(\mu, X^L) = \mathsf{E}_n \nu_n^k = \frac{1}{N} \sum_{n=1}^{N} \nu_n^k.$$

К сожалению, этот функционал не учитывает разброс величины ν_n^k. Возможны ситуации, когда средняя частота ошибок достаточно мала, тем не менее, значения ν_n^k велики для многих разбиений. Чтобы получить гарантированные верхние оценки величины ν_n^k, необходимо знать её функцию распределения. Определим следующий функционал как вероятность того, что частота ошибок на контроле превысит заданное число $\varepsilon \in [0, 1]$:

$$\tilde{Q}_\varepsilon(\mu, X^L) = \mathsf{P}_n \{\nu_n^k > \varepsilon\} = \frac{1}{N} \sum_{n=1}^{N} [\nu_n^k > \varepsilon].$$

В некоторых случаях удобнее оценивать не частоту ошибок на контроле, а величину переобученности. Следующий функционал есть вероятность того, что переобученность превысит заданное число ε:

$$Q_\varepsilon(\mu, X^L) = \mathsf{P}_n \{\delta(\mu, X_n^\ell, X_n^k) > \varepsilon\} = \frac{1}{N} \sum_{n=1}^{N} [\nu_n^k - \nu_n^\ell > \varepsilon]. \tag{1.1}$$

Функционал Q_ε является кусочно-постоянной невозрастающей функцией параметра ε. Пусть имеется его оценка сверху $Q_\varepsilon \leqslant \eta(\varepsilon)$, где $\eta(\varepsilon)$ — монотонно убывающая функция. Функция $\varepsilon(\eta)$, обратная к $\eta(\varepsilon)$, также монотонно убывающая. Тогда (1.1) эквивалентно утверждению, что *для данного метода μ и выборки X^L с вероятностью, не меньшей $1 - \eta$, выполняется неравенство $\nu_n^k \leqslant \nu_n^\ell + \varepsilon(\eta)$. В этом случае говорят, что обучение состоятельно с точностью ε и надёжностью η.*

Таким образом, имея верхнюю оценку функционала Q_ε, легко получить верхнюю оценку и для частоты ошибок на контрольной выборке. Если оценка $\eta(\varepsilon)$ не зависит от выборки X^L, то верхняя граница ν_n^k будет справедлива при произвольных исходных данных. Тогда можно будет *предсказывать* частоту ошибок на произвольной контрольной выборке, зная только частоту ошибок на обучении, значения параметров ε, η, и, быть может, некоторые свойства метода обучения μ. Получение таких оценок и является основной задачей вычислительной теории обучения.

Сильная вероятностная интерпретация. Обычно в вычислительной теории обучения вводятся более сильные вероятностные предположения, стандартные для математической статистики:

Гипотеза 1.2. *Пусть X — вероятностное пространство с неизвестной вероятностной мерой P; элементы множества X^L выбраны случайно и независимо согласно данной вероятностной мере P.*

В сильной вероятностной интерпретации обобщающую способность метода μ принято определять либо как вероятность ошибки:

$$P_{err}(\mu) = \mathsf{P}_{X^\ell,x}\big\{I\big(\mu(X^\ell),x\big) = 1\big\},$$

либо как вероятность того, что переобученность превысит допустимый порог ε:

$$P_\varepsilon(\mu) = \mathsf{P}_{X^\ell,X^k}\big\{\delta(\mu,X^\ell,X^k) > \varepsilon\big\}.$$

Существуют и другие определения [1, 5].

Нетрудно показать, что эти функционалы можно получить, если взять матожидание по выборке X^L от функционалов Q_c и Q_ε, определённых выше:

$$P_{err}(\mu) = \mathsf{E}_{X^L} Q_c(\mu, X^L);$$
$$P_\varepsilon(\mu) = \mathsf{E}_{X^L} Q_\varepsilon(\mu, X^L).$$

Это означает, что всякая верхняя оценка Q_c или Q_ε элементарно переносится и на вероятностные функционалы, P_{err} или P_ε соответственно. В частности, как станет видно далее, основные результаты теории Вапника-Червоненкиса могут быть получены как в сильной, так и в слабой вероятностной интерпретации [3].

Слабая вероятностная интерпретация имеет определённые преимущества.

- Сильная интерпретация предъявляет к исходным данным избыточные требования, с трудом поддающиеся эмпирической проверке. А именно: на множестве объектов X должна существовать σ-алгебра событий; все рассматриваемые функции выборок должны быть измеримы; объекты X^L должны выбираться случайно из неизвестного, но фиксированного распределения (генеральной совокупности). Слабая интерпретация не делает никаких предположений об объектах вне выборки X^L. Предполагается только, что элементы неслучайной конечной совокупности X^L появляются в произвольном случайном порядке. Это предположение эквивалентно требованию независимости элементов выборки (напомним, что независимость определяется как инвариантность вероятностной меры, а, следовательно, и функционала качества, относительно произвольных перестановок элементов выборки). Независимость — по сути, единственное требование, общее для обеих интерпретаций.

- Функционалы качества вида Q_c или Q_ε поддаются непосредственному эмпирическому измерению. Для этого суммирование производится только по некоторым из $N = C_L^\ell$ разбиений. Различные методы порождения разбиений приводят к большому разнообразию оценок *скользящего контроля*.

- Переход от слабой интерпретации к сильной при необходимости осуществляется «в одно действие» — для этого достаточно взять матожидание функционала качества по выборке X^L.

Далее возьмём за основу слабую вероятностную интерпретацию, то есть будем предполагать, что справедлива Гипотеза 1.1.

1.1.2 Простой случай: оценка качества отдельного алгоритма

Начнём изучение оценок качества обучения с простейшего случая, когда метод μ на любой выборке выдаёт один и тот же алгоритм: $\mu(X^\ell) = \text{const}(X^\ell) = a$. Несмотря на кажущуюся «вырожденность», этот случай имеет фундаментальное значение и тесно связан с законом больших чисел в теории вероятностей.

Лемма 1.1. *Пусть $m = L\nu(a, X^L)$ — число ошибок алгоритма a на выборке X^L. Тогда случайная величина ν_n^ℓ подчиняется гипергеометрическому распределению:*

$$\mathsf{P}_n\{\nu_n^\ell = s/\ell\} = h(\tbinom{\ell\ s}{L\ m}) = \frac{C_m^s C_{L-m}^{\ell-s}}{C_L^\ell},$$

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

Доказательство.

Будем называть объекты, на которых a допускает ошибку, ошибочными. Имеется C_m^s вариантов выбрать s ошибочных объектов в обучающую подвыборку X_n^ℓ. Для каждого варианта имеется $C_{L-m}^{\ell-s}$ способов сформировать оставшуюся часть обучающей подвыборки из безошибочных объектов. Значит, $C_m^s C_{L-m}^{\ell-s}$ — число разбиений, при которых из m ошибочных объектов ровно s попадают в обучающую подвыборку. Их доля в общем числе разбиений $N = C_L^\ell$ как раз и составляет $h(\tbinom{\ell\ s}{L\ m})$. ■

Введём следующее обозначение для суммы крайних левых членов гипергеометрического распределения, см. Рис. 1:

$$H(\tbinom{\ell\ s}{L\ m}) = \sum_{t=s_0}^{s} h(\tbinom{\ell\ t}{L\ m}).$$

Теорема 1.2. *Пусть $m = L\nu(a, X^L)$ — число ошибок алгоритма a на выборке X^L. Тогда для любого $\varepsilon \in [0, 1)$*

$$Q_\varepsilon(a, X^L) = H\big(\tbinom{\ell\ s_1(\varepsilon)}{L\ m}\big), \qquad s_1(\varepsilon) = \lfloor (m - \varepsilon k)\ell/L \rfloor. \tag{1.2}$$

Доказательство.

При подсчёте числа разбиений перегруппируем слагаемые по значениям s:

$$Q_\varepsilon(a, X^L) = \frac{1}{N}\sum_{n=1}^{N}\big[\nu_n^k - \nu_n^\ell > \varepsilon\big] = \sum_{s=s_0}^{s_1}\frac{1}{N}\sum_{n=1}^{N}\Big[\nu_n^\ell = \frac{s}{\ell}\Big]\Big[\frac{m-s}{k} - \frac{s}{\ell} > \varepsilon\Big].$$

Поскольку $s_1(\varepsilon)$ — как раз максимальное целое, для которого удовлетворяется условие $\frac{m-s}{k} - \frac{s}{\ell} > \varepsilon$, данное выражение можно переписать короче:

$$Q_\varepsilon(a, X^L) = \sum_{s=s_0}^{s_1(\varepsilon)}\frac{1}{N}\sum_{n=1}^{N}\Big[\nu_n^\ell = \frac{s}{\ell}\Big] = \sum_{s=s_0}^{s_1(\varepsilon)} h(\tbinom{\ell\ s}{L\ m}) = H\big(\tbinom{\ell\ s_1(\varepsilon)}{L\ m}\big),$$

что и требовалось доказать. ■

Таким образом, значение функционала Q_ε численно равно вероятности того, что значение гипергеометрической случайной величины попадает в левый хвост распределения $[s_0, s_1(\varepsilon)]$, см. Рис. 1. Эта вероятность тем меньше, чем больше длина выборки ℓ и чем выше порог точности ε.

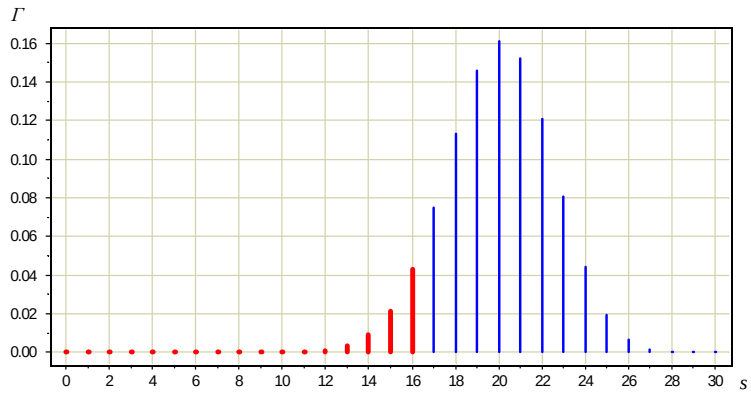

Рис. 1. Гипергеометрическое распределение $h\binom{\ell\ s}{L\ m}$ при $L = 300$, $\ell = 200$, $m = 30$, $\varepsilon = 0.05$. Левый хвост распределения: $[s_0, s_1(\varepsilon)] = [0, 16]$.

Замечание 1.1. Если число ошибок m на полной выборке не известно, то вместо точного равенства (1.2) приходится довольствоваться оценкой сверху:

$$Q_\varepsilon(a, X^L) = \Gamma_L^\ell(s_1(\varepsilon), m) \leqslant \Gamma_L^\ell(s_1(\varepsilon)),$$

где введены следующие обозначения: для произвольного $t = 0, \ldots, \min\{\ell, m\}$

$$\Gamma_L^\ell(t, m) = \sum_{s=s_0}^{t} h\binom{\ell\ s}{L\ m}; \tag{1.3}$$

$$\Gamma_L^\ell(t) = \max_{m=0,\ldots,L} \Gamma_L^\ell\big(t, m\big). \tag{1.4}$$

Замечание 1.2. Известно асимптотическое поведение величины $\Gamma_L^\ell(s_1(\varepsilon))$ [11]:

$$\Gamma_L^\ell(s_1(\varepsilon)) \sim \exp\left(-2\varepsilon^2 \frac{\ell k}{\ell + k}\right), \quad \ell, k \to \infty,$$

откуда следует, что значение Q_ε стремится к нулю при одновременном стремлении ℓ и k к бесконечности. Фактически, равенство (1.2) является аналогом закона больших чисел для слабой вероятностной интерпретации.

1.1.3 Общий случай: оценка качества метода обучения

Рассмотрим теперь общий случай, когда метод μ на разных обучающих выборках может выдавать различные алгоритмы.

Обозначим через A_L^ℓ множество алгоритмов, порождаемых методом μ на всевозможных подвыборках выборки X^L:

$$A_L^\ell(\mu, X^L) = \left\{\mu(X_n^\ell) \;\middle|\; n = 1, \ldots, N\right\}.$$

Очевидно, мощность этого множества не превышает $N = C_L^\ell$. Она может оказаться и меньше N, если некоторые алгоритмы совпадут. Кроме того, некоторые алгоритмы, не совпадающие как отображения $X \to Y$, могут оказаться неразличимыми на объектах выборки X^L.

Опр. 1.2. *Алгоритмы a и a' неразличимы на выборке X^L, если они допускают ошибки на одних и тех же объектах: $I(a, x_i) = I(a', x_i)$ для всех $x_i \in X^L$.*

Неразличимость является отношением эквивалентности на множестве A.

Опр. 1.3. *Коэффициент разнообразия (shatter coefficient) $\Delta^A(X^L)$ множества алгоритмов A на выборке X^L — это число классов эквивалентности, индуцируемых на множестве A отношением неразличимости алгоритмов на выборке X^L.*

В задачах классификации на два класса коэффициент разнообразия совпадает с числом дихотомий (способов разделить выборку на два класса), реализуемых всевозможными алгоритмами из A.

Обозначим коэффициент разнообразия множества алгоритмов $A_L^\ell(\mu, X^L)$ на выборке X^L через $\Delta_L^\ell(\mu, X^L)$.

Теорема 1.3. *Для любого метода μ, выборки X^L, индикатора ошибки I и числа $\varepsilon \in [0, 1)$ справедлива оценка*

$$Q_\varepsilon(\mu, X^L) < \Delta_L^\ell(\mu, X^L) \cdot \Gamma_L^\ell(s_1(\varepsilon)). \tag{1.5}$$

Доказательство.

Введённое отношение неразличимости разбивает множество A_L^ℓ на классы эквивалентности A_{md}, где $m = 0, \ldots, L$ — число ошибок, допускаемых на выборке X^L алгоритмами данного класса; $d = 1, \ldots, D_m$ — порядковый номер класса среди всех классов, алгоритмы которых допускают m ошибок; D_m — число попарно различимых алгоритмов, допускающих ровно m ошибок на выборке X^L.

Запишем функционал качества, суммируя разбиения отдельно по каждому классу эквивалентности:

$$Q_\varepsilon = \sum_{m=0}^{L} \sum_{d=1}^{D_m} \frac{1}{N} \sum_{n=1}^{N} \big[\mu(X_n^\ell) \in A_{md}\big] \big[\nu(\mu(X_n^\ell), X_n^k) > \nu(\mu(X_n^\ell), X_n^\ell) + \varepsilon\big].$$

Значение функционала не изменится, если во внутренней сумме алгоритм $\mu(X_n^\ell)$ заменить на произвольный элемент a_{md} из класса A_{md}. Затем применим тот же приём, что и в доказательстве теоремы 1.2 — перегруппируем слагаемые по значениям числа ошибок s на обучающей выборке:

$$\begin{aligned} Q_\varepsilon &= \sum_{m=0}^{L} \sum_{d=1}^{D_m} \sum_{s=s_0}^{\min\{\ell,m\}} \frac{1}{N} \sum_{n=1}^{N} \Big[\mu(X_n^\ell) \in A_{md}\Big] \Big[\nu(a_{md}, X_n^\ell) = \frac{s}{\ell}\Big] \Big[\frac{m-s}{k} - \frac{s}{\ell} > \varepsilon\Big] = \\ &= \sum_{m=0}^{L} \sum_{d=1}^{D_m} \sum_{s=s_0}^{s_1(\varepsilon)} \underbrace{\frac{1}{N} \sum_{n=1}^{N} \Big[\mu(X_n^\ell) \in A_{md}\Big] \Big[\nu(a_{md}, X_n^\ell) = \frac{s}{\ell}\Big]}_{\gamma(m,s)}. \end{aligned}$$

Оценим сверху внутреннюю сумму $\gamma(m, s)$, заменив $\big[\mu(X_n^\ell) \in A_{md}\big]$ единицей. Применяя рассуждения из доказательства теоремы 1.2, получим $\gamma(m, s) \leqslant h(\ell\, {}^{s}_{L\, m})$. Эта величина не зависит от d, поэтому её можно вынести за знак суммирования

по d. Замечая, что число всех классов эквивалентности равно коэффициенту разнообразия, $\Delta_L^\ell = D_0 + D_1 + \cdots + D_L$, приходим к следующей оценке:

$$Q_\varepsilon \leqslant \sum_{m=0}^{L} D_m \sum_{s=s_0}^{s_1(\varepsilon)} h\binom{\ell\ s}{L\ m} = \sum_{m=0}^{L} D_m \Gamma_L^\ell(s_1(\varepsilon), m) <$$

$$< \left(\sum_{m=0}^{L} D_m\right) \max_m \Gamma_L^\ell(s_1(\varepsilon), m) = \Delta_L^\ell \Gamma_L^\ell(s_1(\varepsilon)).$$

Теорема доказана. ■

Замечание 1.3. Полученная оценка отличается от простого случая (μ = const) появлением коэффициента разнообразия Δ_L^ℓ в качестве сомножителя. Теорема предсказывает ухудшение надёжности во столько раз, сколько классов различимых алгоритмов содержит в себе множество A_L^ℓ. В ходе доказательства дважды делается довольно грубая оценка сверху. В результате оценка (1.5) может оказаться (и в действительности оказывается) сильно завышенной.

1.1.4 Функция роста и ёмкость (VC dimension)

Опр. 1.4. *Функцией роста множества алгоритмов A называется максимальное значение коэффициента разнообразия $\Delta^A(X^L)$ по всем возможным выборкам длины L:*

$$\Delta^A(L) = \max_{X^L} \Delta^A(X^L), \quad L > 0.$$

Функция роста не зависит ни от выборки, ни от метода обучения, и является мерой сложности множества алгоритмов A.

Непосредственно из теоремы 1.3 вытекает следующая

Теорема 1.4 (Вапник, Червоненкис [1]). *При $\ell = k$ для любого метода μ, выборки $X^{2\ell}$, индикатора ошибки I и числа $\varepsilon \in [0, 1)$ справедлива оценка:*

$$Q_\varepsilon^k(\mu, X^{2\ell}) \leqslant \Delta^A(2\ell) \cdot 1.5\, e^{-\varepsilon^2 \ell}. \tag{1.6}$$

Из теоремы следует, что частота ошибок на контроле ν_n^k будет тем меньше, чем меньше частота ошибок на обучении ν_n^ℓ, чем больше длина обучения ℓ, и чем меньше алгоритмов в семействе. Эта оценка завышена ещё сильнее, чем (1.5). В реальных задачах выборка X^L всегда фиксирована; метод обучения μ и индикатор ошибки I, выражающийся через целевую зависимость, также являются фиксированными функциями. Поэтому далеко не все алгоритмы семейства A имеют шансы стать результатом обучения. В общем случае функция роста $\Delta^A(L)$ может принимать существенно бóльшие значения, чем коэффициент разнообразия $\Delta_L^\ell(\mu, X^L)$.

Понятие ёмкости. В теории Вапника-Червоненкиса доказывается, что функция роста $\Delta^A(L)$ либо равна 2^L, либо растёт полиномиально по L, причём промежуточных вариантов не существует. Как следует из Теоремы 1.4, в полиномиальном случае правая часть (1.5) стремится к нулю при $\ell, k \to \infty$, следовательно, обучение асимптотически состоятельно. Разберём этот фундаментальный результат более подробно.

Опр. 1.5. *Если существует число h такое, что $\Delta^A(h) = 2^h$ и $\Delta^A(h+1) < 2^{h+1}$, то оно называется ёмкостью или размерностью Вапника-Червоненкиса (VC-dimension) семейства алгоритмов A. Если такого числа h не существует, то говорят, что семейство A имеет бесконечную ёмкость.*

Если семейство имеет конечную ёмкость h, то его функцию роста можно оценить сверху величиной, зависящей от L полиномиально при $L > h$. Для доказательства этого факта нам понадобятся некоторые вспомогательные построения.

Лемма 1.5. *Функция $\Phi_L^h = C_L^0 + C_L^1 + \cdots + C_L^h$, определённая при целых h и L, таких, что $0 \leqslant h \leqslant L$, однозначно задаётся рекуррентными соотношениями*

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

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

Лемма 1.6 (Вапник, Червоненкис [1]). *Если для любой подвыборки X^{h+1} из X^L выполняется $\Delta^A(X^{h+1}) < 2^{h+1}$, то $\Delta^A(X^L) \leqslant \Phi_L^h$.*

Доказательство.

Доказательство проведём индукцией по h.

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

Для этого при фиксированном h применим индукцию по L.

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

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

1) такие подвыборки X^r из X^L, что алгоритмы семейства A индуцируют на X^{L+1} как X^r, так и (X^r, x_{L+1});

2) все остальные подвыборки.

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

$$\Delta^A(X^L) = K_1 + K_2,$$
$$\Delta^A(X^{L+1}) = 2K_1 + K_2,$$

следовательно, $\Delta^A(X^{L+1}) = \Delta^A(X^L) + K_1$.

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

Имеется две возможности.

1. Допустим, найдётся подвыборка $X^h \subseteq X^L$ такая, что $\Delta^{A'}(X^h) = 2^h$. Это означает, что алгоритмы множества A' индуцируют на X^h, а значит и на (X^h, x_{L+1}),

все возможные подвыборки $X^r \subseteq X^h$. По определению множества A' на (X^h, x_{L+1}) индуцируются также все подвыборки вида (X^r, x_{L+1}). Следовательно

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

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

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

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

Утверждение индукции доказано для $L+1$. ■

Лемма 1.7. *Справедлива оценка* $\Phi_L^h \leqslant 1.5\frac{L^h}{h!}, \quad 0 \leqslant h \leqslant L.$

Доказательство является несложным техническим упражнением [1].

Теорема 1.8 (Вапник, Червоненкис [1]). *Если семейство A имеет конечную ёмкость h, то при $L > h$ функция роста $\Delta^A(L)$ зависит от L полиномиально:*

$$\Delta^A(L) \leqslant \Phi_L^h \leqslant 1.5\frac{L^h}{h!}. \tag{1.7}$$

Доказательство.

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

Пусть $L \geqslant h$. Возьмём произвольную выборку X^L. Для неё выполнено условие леммы 1.6, так как $\Delta^A(h+1) < 2^{h+1}$. Следовательно $\Delta^A(X^L) \leqslant \Phi_L^h$, и в силу произвольности выборки $\Delta^A(L) \leqslant \Phi_L^h$.

Теорема доказана. ■

Функция роста и ёмкость конечного множества алгоритмов. Пусть множество A конечно. Число алгоритмов, попарно неразличимых на выборке X^L, не превышает числа всех алгоритмов, поэтому для функции роста справедлива оценка

$$\Delta^A(L) \leqslant |A|.$$

Ёмкость такого семейства не превышает $\lfloor \log_2 |A| \rfloor$, так как в противном случае функция роста оказалась бы больше $|A|$.

Множества алгоритмов, реализуемых на компьютере, всегда конечны. Если для хранения всех параметров алгоритма используется не более n бит, то число алгоритмов в таком семействе не превышает 2^n, а его ёмкость не превышает $\log_2 2^n = n$. Чтобы эта оценка не была завышенной, для подсчёта необходимого числа бит должно использоваться *максимально экономное кодирование* параметров [4].

Согласно теореме 1.4, чем меньше длина записи алгоритма, тем точнее оценивается частота ошибок на контроле по частоте ошибок на обучении. Отсюда вытекает т. н. принцип *минимума длины описания* (Minimal Description Length, MDL) [9].

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

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

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

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

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

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

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

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

Ёмкость семейства линейных решающих правил. Пусть $X = \mathbb{R}^n$, $Y = \{0,1\}$, A — семейство линейных решающих правил:

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

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

Теорема 1.9. *Ёмкость семейства линейных решающих правил A равна размерности пространства n.*

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

Доказательство.

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

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

Возьмём n векторов $X^n = \{x_1,\ldots,x_n\}$ из X таких, что у i-ого вектора i-ая компонента равна 1, а остальные равны 0. Рассмотрим алгоритм $a \in A$, задаваемый вектором

коэффициентов $w = (w_1, \ldots, w_n)$. Каким бы ни был бинарный вектор $(z_1, \ldots, z_n)$, легко подобрать коэффициенты w_i так, чтобы выполнялось $a(x_i) = [w_i \geqslant 0] = z_i$. Таким образом, мы указали 2^n алгоритмов, различным образом делящих выборку X^n на два класса.

Теперь покажем, что $\Delta^A(n+1) < 2^{n+1}$. Это равносильно высказыванию

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

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

$$x_{n+1} = b_1 x_1 + \cdots + b_n x_n, \tag{1.8}$$

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

Положим $z_i = [b_i \geqslant 0]$ для всех $i = 1, \ldots, n$ и $z_{n+1} = 0$. Рассмотрим произвольный алгоритм $a \in A$ с коэффициентами $w = (w_1, \ldots, w_n)$. Допустим, что $a(x_i) = z_i$ для всех $i = 1, \ldots, n+1$. Умножим обе части равенства (1.8) скалярно на w:

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

Левая часть этого равенства строго меньше нуля, поскольку

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

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

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

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

Теорема доказана. ■

Ёмкость однопараметрического семейства может быть бесконечной, что свидетельствует о нетривиальности понятия ёмкости с одной стороны, и о бесконечных выразительных способностях действительного числа с другой [10].

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

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

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

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

Из определения величины γ_0 следует, что $\pi 10^{-\ell} \leqslant \gamma_0 \leqslant 0.3\pi$, поэтому значение $\sin \gamma_0$ положительно, и правая часть равенства есть просто y_i.

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

1.1.5 Проблема завышенности оценок

К сожалению, непосредственному практическому применению оценки (1.6) препятствует её чрезвычайная завышенность. Чтобы в этом убедиться, достаточно выполнить численный расчёт требуемой длины обучающей выборки ℓ как функции от (h, η, ε). Результаты расчёта приведены в таблице 1.

Таблица 1. Достаточная длина обучающей выборки ℓ как функция от ёмкости h, точности ε и значения функционала качества P_ε^k.

	$P_\varepsilon^k = 0.01$				$P_\varepsilon^k = 1$			
h	$\varepsilon = 0.01$	$\varepsilon = 0.05$	$\varepsilon = 0.1$	$\varepsilon = 0.2$	$\varepsilon = 0.01$	$\varepsilon = 0.05$	$\varepsilon = 0.1$	$\varepsilon = 0.2$
0	60106	2404	601	150	14054	562	140	35
2	295074	9012	1946	408	245330	6963	1423	273
5	673222	19884	4192	848	623320	17823	3664	711
10	1307418	38160	7974	1589	1257471	36095	7444	1452
20	2579359	74855	15572	3082	2529396	72789	15043	2944
50	6401335	185193	38433	7575	6351365	183127	37903	7437
100	12775769	369275	76581	15075	12725798	367208	76051	14937

Правая половина таблицы, соответствующая значению $\eta = 1$, показывает границу применимости оценок Вапника-Червоненкиса. При меньших ℓ верхняя оценка вероятности становится больше 1.

Первая строка таблицы соответствует другому крайнему случаю, когда $h = 0$ и семейство состоит из единственного алгоритма. При этом достигается наилучшая возможная оценка.

Очевидно, требуемая длина выборки существенно превышает количество объектов, с которыми приходится иметь дело на практике. Более того, практические задачи успешно решаются по выборкам в сотни и даже десятки объектов. Фактически, эти случаи остаются за границами применимости теории.

1.1.6 Метод структурной минимизации риска

Имея зависимость Q_ε от ε, легко выразить из неё ε как функцию от ёмкости h, длины обучения ℓ и желаемого значения функционала $\eta = Q_\varepsilon$. Это позволяет записать верхнюю оценку для $\nu(\mu(X^\ell), X^k)$, справедливую с заданной вероятностью.

Утв. 1.10. *При $\ell = k$ для любых μ и y^* с вероятностью $1 - \eta$ справедлива оценка*

$$\nu(\mu(X^\ell), X^k) < \nu(\mu(X^\ell), X^\ell) + \sqrt{\frac{h}{\ell}\left(\ln\frac{2\ell}{h} + 1\right) - \frac{\ln\eta}{\ell}}. \tag{1.9}$$

Первое слагаемое в этой оценке представляет эмпирический риск, убывающий с ростом ёмкости h. Второе слагаемое возрастает с ростом ёмкости, и его можно рассматривать как *штраф за сложность* (complexity penalty). Сумма в общем случае достигает минимума при некотором h.

Для определения оптимальной сложности модели алгоритмов Вапником и Червоненкисом был предложен метод *структурной минимизации риска*. Предположим,

что в семействе A выделена последовательность подсемейств возрастающей ёмкости $A_1 \subset A_2 \subset \ldots \subset A_h = A$. Тогда в ней можно выбрать оптимальное подсемейство, для которого достигается минимальное значение правой части (1.9), и гарантировать заданное качество обучения.

В методе структурной минимизации риска завышенность оценок может приводить к чрезмерному упрощению алгоритмов [6]. Это связано с тем, что первое слагаемое в (1.9) вычисляется точно, а завышенность второго слагаемого, как правило, несколько увеличивается с ростом сложности h.

Поэтому на практике подсемейство оптимальной сложности выбирают непосредственно по критерию скользящего контроля, не прибегая к завышенным верхним оценкам (1.9).

Список литературы

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

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

[3] *Воронцов К. В.* Комбинаторный подход к оценке качества обучаемых алгоритмов // Математические вопросы кибернетики / Под ред. О. Б. Лупанов. — М.: Физматлит, 2004. — Т. 13. — С. 5–36.
http://www.ccas.ru/frc/papers/voron04mpc.pdf.

[4] *Донской В. И.* Колмогоровская сложность классов общерекурсивных функций с ограниченной емкостью // *Таврический вестник информатики и математики.* — 2005. — № 1. — С. 25–34.
http://www.ccas.ru/frc/papers/donskoy05kolmogorov.pdf.

[5] *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.

[6] *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.

[7] *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.

[8] *Mullin M., Sukthankar R.* Complete cross-validation for nearest neighbor classifiers // Proceedings of International Conference on Machine Learning. — 2000.
http://citeseer.ist.psu.edu/309025.html.

[9] *Rissanen J.* Modeling by shortest data description // *Automatica.* — 1978. — Vol. 14. — Pp. 465–471.

[10] *Vapnik V.* The nature of statistical learning theory. — Springer-Verlag, New York, 1995.

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