КИБЕРНЕТИКА

В.И. ДОНСКОЙ

УДК 519.95

СЛОЖНОСТЬ СЕМЕЙСТВ АЛГОРИТМОВ ОБУЧЕНИЯ И ОЦЕНИВАНИЕ НЕСЛУЧАЙНОСТИ ИЗВЛЕЧЕНИЯ ЭМПИРИЧЕСКИХ ЗАКОНОМЕРНОСТЕЙ

Ключевые слова: *машинное обучение, рекурсивные функции, VC-размерность, колмогоровская сложность.*

ВВЕДЕНИЕ

В последние десятилетия интенсивно развиваются подходы к обоснованию и оцениванию методов эмпирического обобщения на основе понятия алгоритмической сложности. Прежде всего, имеется в виду колмогоровский поход и предложенный на его основе как эвристика метод MDL — Minimum Description Length. Догадка о том, что более «простые» решающие правила чаще дают правильные решения, чем «сложные», оправдалась на практике и многие годы воспринималась как «гипотеза простой структурной закономерности». Цель исследований в указанном направлении — понять природу сложности и на основе ее изучения получить методы нахождения оценок качества алгоритмов обучения (эмпирического обобщения). Несмотря на некоторое продвижение в теории, такие оценки до сих пор не получены для многих классов алгоритмов. Это связано, прежде всего, с математическими трудностями вывода логико-комбинаторных оценок и отсутствием общего способа их получения.

В настоящей работе представлен именно общий способ оценивания — так называемый метод *pVCD*, который удалось разработать, ограничив все рассматриваемые семейства моделей эмпирического обобщения до классов, реализуемых на компьютерах, и шире — рассматривая их частично-рекурсивные представления. В рамках алгоритмического подхода введено понятие колмогоровской сложности классов алгоритмов распознавания свойств или извлечения закономерностей. На основе этого понятия предложен метод оценивания неслучайности извлечения эмпирических закономерностей.

1. ЗАДАЧА ОБУЧЕНИЯ КАК ЭМПИРИЧЕСКОЕ ОБОБЩЕНИЕ С ЦЕЛЬЮ ПОСТРОЕНИЯ РЕШАЮЩЕГО ПРАВИЛА (ИЗВЛЕЧЕНИЯ ЗАКОНОМЕРНОСТИ)

Обозначим S произвольное семейство общерекурсивных функций (алгоритмов), состоящее из элементов вида $A: X^n \to \{0,1\}$; $X^n = \{X = (x_1, \dots, x_n):$ $x_i \in \{0,1,\dots,2^M - 1\}\}$ — множество n-мерных векторов $X = (x_1, \dots, x_n)$ с целочисленными неотрицательными переменными. Каждая переменная x_i представляется ровно M битами памяти. При использовании обычного компьютера число M — это его разрядность. Определенное таким образом семейство S является конечным, поскольку не может содержать более $2^{2^{Mn}}$ алгоритмов. Конечность класса является необходимым условием компьютерной реализации. Если область определения значений переменных x_i не ограничивать, то и семейство S может быть неограниченным. В таком случае бесконечное семейство алгоритмов не сможет быть полностью реализовано на обычном конечном компьютере.

Выборка, состоящая из l произвольных элементов множества X^n, обозначается $\widetilde{X}_l = X_1, \ldots, X_l$ и представляет собой упорядоченный набор $n \times l$ ограниченных чисел из расширенного натурального ряда; число l называется длиной выборки. С теоретической точки зрения допустимо считать все рассматриваемые числа и выборки представленными в виде бинарных строк. Множество всех выборок обозначается $\aleph^l$; в общем случае это множество неограниченно; его мощность в конечном случае составляет $\mathrm{card}\, \aleph^l = 2^{Mnl}$. Множество $\{0, 1\}^*$ строк из нулей и единиц любой длины обычным способом представляет числа 0, 1, 2,… Длина слова $p \in \{0, 1\}^*$ обозначается $\mathrm{len}(p)$. Класс частично рекурсивных функций обозначается $P_{p.r.}$.

Обучающей выборкой называется пара $(\widetilde{X}_l, \widetilde{\alpha}_l)$, где $\widetilde{\alpha}_l = (\alpha_1, \cdots, \alpha_l)$, $\alpha_j = F(X_j)$, $j = \overline{1, l}$; $F: X^n \to \{0, 1\}$ — некоторая заранее неизвестная, но предполагаемая существующей классифицирующая функция вида $F: X^n \to \{0, 1\}$. Множество всех возможных обучающих выборок обозначается $\aleph^l \times \{0, 1\}^l$ и представляет собой генеральную совокупность, из которой могут извлекаться обучающие выборки. Задача обучения состоит в нахождении по данной выборке $(\widetilde{X}_l, \widetilde{\alpha}_l)$ функции F или как можно более «близкой» к ней решающей функции (алгоритма или правила) $A^* \in S$. Отыскиваемая функция F как и ее аппроксимация A^*, являются предикатами, определяющими некоторое свойство или закономерность. Именно обобщение свойств выборки (частных наблюдений) с целью выбора решающего правила или нахождения закономерности определяет применяемый метод — эмпирическую индукцию. Изначальная некорректность метода эмпирической индукции, обусловленная неединственностью множества решений задачи обучения, приводит к дополнительной проблеме обоснования выбранного решающего правила.

Семейство S, внутри которого отыскивается решение, определяется условиями, которым должна удовлетворять искомая функция A^*, и выбором модели обучения (и соответствующего класса алгоритмов распознавания), например вычисления оценок, нейронных сетей, деревьев решений или алгебраических корректирующих моделей над перечисленными и/или другими эвристическими алгоритмами. В частности, отыскивается такая функция $A^* \in S$, для которой эмпирический риск $\nu^l(A) = 1/l \sum_{j=1}^{l} |A(X_j) - \alpha_j|$ минимален. От сложности семейства S алгоритмов, применяемых в указанных задачах, зависит обоснование выбора решений. Впервые решающее значение сложности семейств решающих правил в задачах эмпирического обобщения показали В. Н. Вапник и А.Я. Червоненкис [1–3]. Предложенная ими мера сложности, вообще говоря, произвольных вещественнозначных функций — так называемая емкость или VC-размерность является, возможно, одним из наиболее ярких и полезных понятий для развития теории индуктивной математики (и не только). В данной статье приводится алгоритмическое определение сложности семейств S, основанное на идеях А.Н. Колмогорова [5], и рассматривается применение введенной меры алгоритмической сложности и для обоснования машинного обучения, и для оценивания VC-размерности. Статья развивает новое направление в алгоритмической теории обучения, элементы которого впервые появились в работах автора [4, 6].

2. НЕОБХОДИМЫЕ ПОНЯТИЯ ТЕОРИИ ВАПНИКА–ЧЕРВОНЕНКИСА [1–3]

Каждое решающее правило $A \in S$ для произвольной обучающей последовательности $\widetilde{X}_l = X_1, \ldots, X_j, \ldots, X_l$ определяет подпоследовательность X_A, состоящую из тех X_j, для которых $A(X_j) = 1$. Говорят, что алгоритм A индуцирует подпоследовательность X_A на $\widetilde{X}_l$ (и тем самым разбивает $\widetilde{X}_l$ на элементы X_A и их дополнение в $\widetilde{X}_l$). Обозначим $\Delta^S(X_1, \ldots, X_l)$ число различных подпоследовательностей X_A, индуцируемых всеми алгоритмами $A \in S$ (число различных разбиений выборки $\widetilde{X}_l$). Очевидно, $\Delta^S(x_1, \ldots, x_l) \le 2^l$. Число $\Delta^S(X_1, \ldots, X_l)$ называется индексом системы S относительно выборки $X_1, .., X_l$. Функция $m^S(l) = \max\limits_{X_1, \ldots, X_l \in \aleph^l} \Delta^S(X_1, \ldots, X_l)$, где максимум берется по множеству $\aleph^l$ всех последовательностей длины l, называется функцией роста системы S. Функция роста $m^S(l)$ либо тождественно равна 2^l, либо, в противном случае, мажорируется функцией $\sum\limits_{i=0}^{n-1} C_l^i \le 1{,}5 \dfrac{l^{n-1}}{(n-1)!}$, где n — минимальное значение l, при котором $m^S(l) \ne 2^l$. Класс S имеет конечную емкость h, если справедливо неравенство $m^S(l) < 1{,}5 \dfrac{l^h}{h!}$, $l > h$. В случае $m^S(l) \equiv 2^l$ говорят, что емкость класса бесконечна, и используют символьное обозначение $h = \infty$. В зарубежной литературе величину h называют VC-размерностью или VCD (Vapnik–Chervonenkis Dimension):

$$VCD(S) = \begin{cases} h, \ m^S(l) < 1{,}5 \dfrac{l^h}{h!}, \\ \infty, \ m^S(l) \equiv 2^l, \end{cases}$$

$VCD(S) = h$ определяет наибольшее число точек пространства X^n, которое можно разбить любым (из 2^h) возможным способом на два подмножества, используя семейство S, иначе говоря, определяет наибольшую длину последовательности точек, в которой система событий S может индуцировать любую подпоследовательность. Если число событий в системе S конечно, $|S| = N$, то очевидно, что $2^h \le N$; $h = VCD(S) \le \log N$ (здесь и всюду далее используется логарифм по основанию 2).

Центральными результатами теории Вапника–Червоненкиса являются достаточное условие равномерной сходимости (по всему классу S алгоритмов) эмпирических частот ошибок к вероятностям, состоящее в ограниченности $VCD(S) = h < \infty$, и близкое к нему по смыслу необходимое условие. Справедливо неравенство [1–3]

$$P\left(\sup_{A \in S} |\nu^l(A) - P(A)| > \varepsilon \right) \le 6 m^S(2l) e^{-\varepsilon^2 (l-1)/4} .$$

Согласно этому неравенству для того чтобы при заданном $\varepsilon > 0$ эмпирические риски $\nu^l(A)$, $A \in S$, рассматриваемые как частоты событий класса S, сходились (по вероятности) к соответствующим им вероятностям равномерно по классу S, достаточно существования такой конечной величины h (емкости класса или $VCD(S)$), что $m^S(l) \le 1{,}5 \dfrac{l^h}{h!}$ при $l \to \infty$.

Пусть $H^S(l) = M.O.\log\Delta^S(X_1, \dots, X_l)$ — математическое ожидание логарифма индекса S относительно выборки $X_1, \dots, X_l$. Очевидно, что $1 \le \Delta^S(X_1, \dots, X_l) \le 2^l$ и $0 \le \log\Delta^S(X_1, \dots, X_l) \le l$. Поэтому справедлива оценка $0 \le H^S(l) \le l$. Функция $H^S(l)$ называется энтропией системы событий S относительно выборок длины l. Установлено, что для равномерной сходимости частот ошибок обучения к их вероятностям необходимо и достаточно, чтобы отношение $\frac{H^S(l)}{l}$ (энтропия на символ) стремилось к нулю с ростом длины выборки l. Иначе говоря, критерий равномерной сходимости имеет вид $\lim_{l\to\infty} \frac{H^S(l)}{l} = 0$. Очевидно, что $\log m^S(l) \ge H^S(l)$, и условие $\lim_{l\to\infty} \frac{\log m^S(l)}{l} = 0$ является достаточным для существования равномерной сходимости.

3. ОЦЕНИВАНИЕ СЛОЖНОСТИ СЕМЕЙСТВ АЛГОРИТМОВ ЭМПИРИЧЕСКОГО ОБОБЩЕНИЯ НА ОСНОВЕ КОЛМОГОРОВСКОГО ПОДХОДА

Определение 1. Пусть U — такая частично рекурсивная функция, что для каждого алгоритма A из рассматриваемого семейства S и для любой выборки $\widetilde{X}_l$ найдется двоичное слово p, которое обеспечивает выполнение равенства $U(p, \widetilde{X}_l) = \widetilde{y}$, где $\widetilde{y} = A(X_1), \dots, A(X_l)$ — двоичное слово (строка) длины l. При этом каждый алгоритм $A \in S$ полагается определенным на каждой выборке $\widetilde{X}_l$ из $\aleph^l$. Функция U с указанными ниже свойствами имеет место в силу существования универсальной функции двух аргументов для любого семейства частично рекурсивных функций одного аргумента.

1. Сложность алгоритма A относительно выборки $\widetilde{X}_l$ по частично рекурсивной функции U есть $K_U(A \mid \widetilde{X}_l) = \min \operatorname{len}(p) \colon U(p, \widetilde{X}_l) = \widetilde{y}$.

2. Сложность алгоритма A на множестве всех выборок $\aleph^l$ по частично рекурсивной функции U есть $K_{U,\aleph^l}(A) = \max_{\widetilde{X}_l \in \aleph^l} K_U(A \mid \widetilde{X}_l)$.

3. Сложность семейства алгоритмов S на множестве $\aleph^l$ по частично рекурсивной функции U есть $K_{U,\aleph^l}(S) = \max_{A \in S} K_{U,\aleph^l}(A)$.

4. Сложность семейства алгоритмов S на множестве $\aleph^l$ есть $K_l(S) = \min_{U \in P_{p.r.}} K_{U,\aleph^l}(S)$.

Приведенное определение легко поясняется следующим образом. Сложность семейства алгоритмов S на множестве всех возможных выборок $\aleph^l$ длины l — это наименьшая длина двоичного слова p, обеспечивающая определение наиболее сложного (и поэтому — любого) алгоритма $A \in S$. Важно, что слово p обрабатывается одной и той же функцией (программой) U^*, причем согласно свойству 4 — наилучшей в следующем смысле. Программа U^* обеспечивает наибольшее сжатие информации о семействе S в слово p длины $K_l(S)$. Никаких дополнительных требований на программу U^* не накладывается. Поэтому мажоранту сложности $K_l(S)$ можно получить, если точно указать структуру слова p, подлежащего расшифровке, и его длину в битах, а также предоставить алгоритм обработки этого слова, который будет использоваться вместо программы U^* для оценивания сложности сверху.

Замечание. Как уже отмечалось, если снять ограничение $x_i \in \{0,1,\ldots,2^M-1\}$ и полагать, что значения переменных x_i могут быть любыми из расширенного натурального ряда, т.е. $x_i \in \{0,1,2,\ldots\}$, то рассматриваемые семейства S будут бесконечными. Бесконечные семейства функций тем не менее могут иметь конечную емкость h_s (что и требуется для гарантированной возможности решения задачи обучения согласно теории Вапника–Червоненкиса). А колмогоровская сложность $K_l(S)$ бесконечного семейства S может расти с ростом длины l обучающей последовательности.

Теорема 1. Пусть не обязательно конечная система общерекурсивных функций S вида $A: X^n \to \{0,1\}$ имеет ограниченную емкость h_s и колмогоровскую сложность $K_l(S)$. Тогда при конечных значениях $h_s \geq 2$ и $l > h_s$ имеет место двойное неравенство: $h_s \leq K_l(S) < h_s \log l$.

Доказательство. Для семейства функций S сложность $K_l(S)$ определена выше с использованием соотношения $U(p, \widetilde{X}_l) = \widetilde{y}$, в котором булев вектор $\widetilde{y}$ длины l принимает значения, соответствующие различным вариантам разбиения выборки $\widetilde{X}_l$ на два подмножества. Обозначим $\widetilde{y} = A(\widetilde{X}_l)$ результат применения алгоритма A к выборке $\widetilde{X}_l$ ровно l раз. Все возможные варианты разбиений выборки $\widetilde{X}_l$ определяются функциями семейства S: $\widetilde{y} = A(\widetilde{X}_l)$, $A \in S$, причем одинаковые разбиения порождают подклассы эквивалентных в этом смысле на выборке $\widetilde{X}_l$ элементов A из семейства S. Выберем из каждого такого класса эквивалентности по одной функции (алгоритму) и обозначим их $A_0, \ldots, A_i, \ldots, A_{m^S(l)-1}$. Для того чтобы равенство $U(p, \widetilde{X}_l) = A(\widetilde{X}_l)$ при зафиксированной частично рекурсивной функции U выполнялось для всех $A \in S$ на каждой выборке $\widetilde{X}_l$, аргумент p, определяющий номера функций $A_0, \ldots, A_i, \ldots, A_{m^S(l)-1}$, должен принимать при каждом зафиксированном l не менее $m^S(l)$ значений, где $m^S(l)$ — функция роста системы S, определяющая наибольшее число разбиений (наибольшее возможное число различных векторов $\widetilde{y}$) по всем выборкам из $\aleph^l$. Поэтому с учетом того, что U является функцией, должно выполняться неравенство $l(p) \geq]\log m^S(l)[$.

Покажем теперь, что $\min\limits_{U \in P_{p.r.}} K_{U,\aleph^l}(S) =]\log m^S(l)[$. Для этого с учетом последнего нестрогого неравенства, достаточно указать такую функцию $U^* \in P_{p.r.}$, что $K_{U^*,\aleph^l}(S) =]\log m^S(l)[$. Построение функции U^* можно пояснить табл. 1, имеющей в общем случае неограниченное вправо число столбцов.

Таблица 1

Код (номер программы) p	Код (номер) выборки $\widetilde{X}_l$			
	$\widetilde{X}_l^{(0)}$	…	$\widetilde{X}_l^{(j)}$	…
0	…	…	…	…
…	…	…	…	…
i	$\widetilde{y}_{i,0}$	...	$\widetilde{y}_{i,j}$	...
…	…	…	…	…
$m^S(l)-1$	…	…	…	…

Каждая строка табл. 1 с номером i, $0 \leq i \leq m^S(l)-1$, соответствует алгоритму A_i из выбранного выше множества $\{A_1, \ldots, A_i, \ldots, A_{m^S(l)-1}\}$ и числовому значению i кода программы p для этого алгоритма. Значения $\widetilde{y}_{i,j}$, $j = 0,1,2,\ldots$, содержащиеся в этой таблице, являются результатами применения алгоритмов A_i к выборкам $\widetilde{X}_l^{(j)}$, а также двоичными кодами длины l и отождествляются с соответствующими числами расширенного натурального ряда; также интерпретируются числами выборки $\widetilde{X}_l$ и коды p. В силу существования $m^S(l)$ частично рекурсивных функций (алгоритмов) $\{A_1, \ldots, A_i, \ldots, A_{m^S(l)-1}\}$ найдется универсальная функция U^* двух аргументов, обеспечивающая выполнение равенства $U^*(p, \widetilde{X}_l) = A_{i=i(p)}(\widetilde{X}_l)$ для $m^S(l)$ различных значений слова p длины $]\log m^S(l)[$. Поэтому $\min\limits_{U \in P_{p.r.}} K_{U,\aleph^l}(S) =$ $=]\log m^S(l)[$ достигается для функции U^*.

Для класса событий ограниченной емкости $h_s \geq 2$ при $l > h_s$ справедливы соотношения $2^{h_S} \leq m^S(l) < 1{,}5\dfrac{l^{h_S}}{h_S!} < l^{h_S} = 2^{h_S \log l}$, $h_S \leq \log m^S(l) < h_S \log l$. С учетом равенства $\min\limits_{U \in P_{p.r.}} K_{U,\aleph^l}(S) =]\log m^S(l)[$ получаем $h_S \leq K_l(S) < h_S \log l$. ■

Следствие 1. Колмогоровская сложность семейства алгоритмов равна наименьшему целому, большему или равному логарифму функции роста этого семейства: $K_l(S) =]\log m^S(l)[$.

Доказательство. Укажем семейство S, для которого $K_l(S) = 0$. Последнее соотношение имеет место, если для получения равенства $U(p, \widetilde{X}_l) = \widetilde{y}$ наличия слова p вообще не требуется: оно может быть пустым. Например, рассмотрим семейство S, в котором каждый алгоритм $A(X)$ выдает значение суммы по модулю два всех символов входной бинарной строки X. Тогда каждая выборка будет классифицироваться единственным способом, потому $m^S(l) = 1$, $\log m^S(l) = 0$ и $K_l(S) = 0$. Заметим, что при этом алгоритмы в S, вообще говоря, могут быть различными, например, прямое суммирование по модулю; вычисление числа единиц в строке и последующая проверка его четности по младшему двоичному разряду; последовательное инвертирование при прохождении единиц слова X. ■

Следствие 2. Справедливо неравенство $0 \leq K_l(S) \leq l$.

Доказательство. Поскольку $m^S(l) \leq 2^l$, то $K_l(S) =]\log m^S(l)[\leq l$. ■

Теорема 2. Если $\lim\limits_{l \to \infty} \dfrac{K_l(S)}{l} = 0$, то имеет место равномерная сходимость частот ошибок к их вероятностям по всему классу S.

Доказательство. Действительно, $\log m^S(l) \geq H^S(l)$; $\lim\limits_{l \to \infty} \dfrac{\log m^S(l)}{l} =$

$$= \lim_{l \to \infty} \frac{K_l(S)}{l} \geq \lim_{l \to \infty} \frac{H^S(l)}{l} = 0. \quad ■$$

4. МЕТОД ПРОГРАММИРОВАНИЯ КОЛМОГОРОВСКОЙ И ВАПНИКОВСКОЙ ОЦЕНКИ СЛОЖНОСТИ КЛАССОВ РЕШАЮЩИХ ПРАВИЛ

Сложность $K_l(S)$ класса алгоритмов S определяется наименьшей длиной слова (программы) p, по которому с помощью соответствующей частично рекурсивной функции (наилучшему внешнему алгоритму) U^* можно определить слово $\widetilde{y} = A(X_1), \ldots, A(X_l)$ в наиболее «трудном» (на множестве выборок $\aleph^l$ и алгоритмов S) случае. Очевидно, $K_l(S) \leq K_{U,\aleph^l}(S)$ для произвольной функции $U \in P_{p.r.}$, поэтому для оценивания $K_l(S)$ сверху в качестве алгоритма U может быть взята, например, машина Тьюринга MT, вычисляющая $\widetilde{y} = MT(p, \widetilde{X}_l)$, или подходящая программа π на каком-нибудь языке программирования такая, что $\pi(p, \widetilde{X}_l) = \widetilde{y}$ для входа $(p, \widetilde{X}_l)$, и тогда $h_S = VCD(S) \le \le \mathrm{len}(p)$.

Подход к оцениванию VCD на основе соотношения $VCD(S) \le \mathrm{len}(p)$: $U(p, \widetilde{X}_l) = \widetilde{y} = (A(X_1), \ldots, A(X_l))$ называется методом программирования оценки VCD, сокращенно — $pVCD$. Используя соотношение $K_{U^*, \aleph^l}(S) =]\log m^S(l)[$ (следствие 2), получаем $K_{U, \aleph^l}(S) \ge \log m^S(l)$, $U \in P_{p.r.}$; $m^S(l) \le 2^{K_{U,\aleph^l}(S)}$. Подход к оцениванию функции роста $m^S(l)$ на основе соотношения $m^S(l) \le 2^{\mathrm{len}(p)}$, аналогичный методу программирования оценки VCD, называется методом программирования оценки $m^S(l)$, сокращенно — $pm^S(l)$. Вводятся обозначения $\mathrm{len}(p) = pVCD(S)$ и $2^{\mathrm{len}(p)} = pm^S(l)$.

Представим этапы реализации $pVCD$ $(pm^S(l))$.

1. Изучение класса S и определение как можно меньшей совокупности свойств (параметров, структурных особенностей) этого класса, указания значений которых достаточно, чтобы сформировать из них слово p, описывающее любой алгоритм $A \in S$. Предъявить алгоритм U (машину Тьюринга, частично рекурсивную функцию, программу для конечного компьютера) такой, что $\forall A \in S$ $\exists p_A : U(p_A, \widetilde{X}_l) = (A(X_1), \ldots, A(X_l))$.

2. Определение максимальной длины $\mathrm{len}(p_A)$ слова p_A, $A \in S$, как оценки $VCD(S)$ сверху ($2^{\mathrm{len}(p_A)}$ как оценки $m^S(l)$ сверху).

Метод $pVCD$ предполагает конструирование сжатого описания p всего класса S и указания алгоритма U, обрабатывающего вход $(p, \widetilde{X}_l)$. Во многих случаях достаточно очевидности существования такого алгоритма, но может оказаться, что применение $pVCD$ потребует искусства программирования и организации данных p, чтобы получить нетривиальную $pVCD$-оценку.

Сужая круг решающих правил до реализуемых на компьютерах разрядности M, как будет показано ниже, можно получить оценку $pVCD(S)$ с указанием констант.

Лемма 1 (об аддитивности $pVCD$-оценки композиции алгоритмов). Пусть $S_0^r = \{f = f_1 \circ \cdots \circ f_r : f_1 \in S_1, \ldots, f_r \in S_r\}$ — класс композиций алгоритмов зафиксированной структуры $f(f_1, \ldots, f_r)$, принадлежащих семействам $S_1, \ldots, S_r$, для которых известны оценки $pVCD(S_1) = L_1, \ldots, pVCD(S_r) = L_r$. Тогда справедлива оценка $pVCD(S_0^r) = \sum_{j=1,\ldots,r} L_j$.

Доказательство. Поскольку структура композиции неизменна, любой входящий в нее алгоритм определяется совокупностью слов $p_1, \ldots, p_j, \ldots, p_r$, имеющих длины $L_1, \ldots, L_j, \ldots, L_r$. Для обработки этих слов согласно методу программирования оценок и соотношению $U_j(p_j, \widetilde{X}_l) = \widetilde{y}$ указаны алгоритмы U_j, $j = \overline{1, r}$, каждый из которых по слову p_j восстанавливает алгоритм f_j. Поэтому легко указать алгоритм (программу) $U_{S_0^r}$, обрабатывающий конкатенацию $p_0 = p_1 p_2 \cdots p_r$ и соответствующий композиции $f = f_1 \circ \cdots \circ f_r$. Такая программа будет содержать подпрограммы U_j, $j = \overline{1, r}$, которые восстанавливают все алгоритмы $f_1, \dots, f_r$, и переходы между ними, предопределенные зафиксированной структурой композиции и известными длинами $L_1, \dots, L_j, \dots, L_r$ подслов, входящих в конкатенацию $p_0 = p_1 p_2 \cdots p_r$. ■

Следствие 3. *pVCD*-оценка суперпозиции алгоритмов $S_0^r = \{f = f_1 \circ \dots \circ f_r$: $f_1 \in S_1, \dots, f_r \in S_r\}$ имеет, в частности, вид $pVCD(S_0^r) = \log l \sum_{j=1,\dots,r} h_{S_j}$, где $h_{s_1}, \dots, h_{s_r}$ — емкости классов $S_1, \dots, S_r$.

Доказательство вытекает из неравенства $K_l(S) < h_S \log l$.

Замечание. Согласно следствию 1 колмогоровская сложность $K_l(S)$ должна зависеть от длины выборки *l*. Однако при использовании $pVCD(S)$ может быть получена мажоранта сложности, определяемая длиной слова *p* и не зависимая от *l*. Это объясняется тем, что класс *S* может оказаться конечным или что функция $m^S(l)$ растет не быстрее, чем $O(l)$.

5. ПРИМЕР ПРОГРАММИРОВАНИЯ ОЦЕНКИ СЛОЖНОСТИ

Дизъюнктивной нормальной формой (ДНФ) представления булевых функций называется выражение вида $\bigvee_{j=1}^{\mu} (x_{j1}^{\sigma_{j1}} \& x_{j2}^{\sigma_{j2}} \& \dots \& x_{jk_j}^{\sigma_{jk_j}})$, где $x^\delta = x$ при $\delta = 1$ (положительный литерал); $x^\delta = \bar{x}$ при $\delta = 0$ (отрицательный литерал); μ — число конъюнкций в ДНФ; $L = \sum_{j=1}^{\mu} k_j$ — длина ДНФ (суммарное входящее в нее число литералов). Пусть класс $DNF_{L,\mu,n}$ — семейство булевых функций вида $f: \{0,1\}^n \to \{0,1\}$, представимых в виде ДНФ длины не более *L*, содержащих не более чем μ конъюнкций. Используя метод *pVCD*, можно получить оценку $VCD(DNF_{L,\mu,n}) < L + (\mu - 1 + L)]\log(n+1)[$ следующим образом.

Действительно, слово p_f, позволяющее закодировать информацию о любой ДНФ, состоящей из μ конъюнкций над *n* переменными, можно представить конкатенацией μ двоичных слов, сформированных из блоков. Представим фрагмент слова, кодирующего литерал, следующим образом.

Номер переменной x_j, входящей в конъюнкцию, $j \in \{1, \dots, n\}$, или нуль — разделитель блоков	Двоичная цифра 1, если x_j входит в конъюнкцию с инверсией, или нуль — в противном случае

Чтобы представить в двоичном коде один любой номер переменной или нуль, достаточно зарезервировать $]\log(n+1)[$ двоичных разрядов. Поскольку номера переменных начинаются с единицы, нуль можно использовать как признак разделения конъюнкций в строке. Чтобы указать знак литерала — с инверсией или без нее — достаточно одного двоичного разряда. При таком кодировании на каждый литерал в слове p_f будет расходоваться $]\log(n+1)[+1$ бит. На *j*-ю конъюнкцию будет расходоваться $k_j(]\log(n+1)[+1)$ бит для представления литералов, $(\mu-1)]\log(n+1)[$ бит понадобится для разделителей. Поэтому длина слова p_f не превысит

$$(\mu-1)\,]\log(n+1)[+ \sum_{j=1}^{\mu} k_j(]\log(n+1)[+1) = (\mu-1)]\log(n+1)[+ L]\log(n+1)[+ L =$$
$$= L + (\mu - 1 + L)\,]\log(n+1)[.$$

Если ДНФ содержит $m < \mu$ конъюнкций, то последние $\mu - m$ блоков слова p_f заполняются нулями. Пусть дана ДНФ $x_3\bar{x}_5 \vee \bar{x}_2 x_4$ из класса $DNF_{10,2,5}$ —

длины не более 10 и не более чем с двумя конъюнкциями. Пусть $n=5$. Десятичная (для облегчения восприятия) структура слова p_f имеет вид |3|1|5|0|0|2|0|4|1|0|. Расшифровка ДНФ по слову p_f (алгоритм U) поясняется табл. 2.

Таблица 2

Цифра слова p_f	Алгоритм U
3	Взять переменную x_3
1	x_3 берется без инверсии
5	Поскольку цифра не равна нулю, взять в теущую конъюнкцию следующую переменную x_5
0	x_5 берется с инверсией
0	Поскольку вместо номера переменной — нуль, то получена конъюнкция $x_3\bar{x}_5$, и далее начинается описание следующей конъюнкции, если за считанным нулем не последует второй нуль; счетчик выделенных конъюнкций увеличивается на единицу
2	Поскольку цифра не равна нулю, включить в текущую конъюнкцию переменную x_2
0	x_2 берется с инверсией
4	Поскольку цифра не равна нулю, взять в текущую конъюнкцию следующую переменную x_4
1	x_4 берется без инверсии
0	Поскольку вместо номера переменной — нуль, то получена конъюнкция $\bar{x}_2x_4$; счетчик выделенных конъюнкций увеличивается на единицу и становится равным двум. Значение $\mu=2$ свидетельствует об окончании слова p_f и представлении результата расшифровки — $x_3\bar{x}_5 \vee \bar{x}_2x_4$

Поскольку $n=5$ и $]\log(n+1)[\,=3$, двоичное представление слова p_f будет следующим: |011|1|101|0|000|010|0|100|1|000|. Заметим, что знак | сохранен для удобства восприятия структуры слова, но в слове p_f он не содержится.

6. КОЛМОГОРОВСКАЯ СЛОЖНОСТЬ КЛАССОВ РЕШАЮЩИХ ФУНКЦИЙ И ОЦЕНИВАНИЕ ЭМПИРИЧЕСКИХ ЗАКОНОМЕРНОСТЕЙ

Определение 2. Пусть $(\widetilde{X}_l, \widetilde{\alpha}_l)$ — зафиксированная обучающая выборка, S — семейство алгоритмов, используемое для обучения. Выбор решения f^* функциональной системы (если оно существует)

$$\begin{cases} f(X_1)=\alpha_1, \\ f(X_2)=\alpha_2, \\ \dots\dots\dots\dots \\ f(X_l)=\alpha_l, \\ f\in S \end{cases} \tag{1}$$

называется безошибочной настройкой на выборку $\widetilde{X}_l$. Выбор решения функциональной системы (если оно существует)

$$\begin{cases} f(X_{j_1})=\alpha_{j_1}, \\ f(X_{j_2})=\alpha_{j_2}, \\ \dots\dots\dots\dots \\ f(X_{j_k})=\alpha_{j_k}, \\ f\in S \end{cases} \tag{2}$$

называется настройкой на k фиксированных элементов $X_{j_1}, X_{j_2}, \cdots, X_{j_k}$ выборки $\widetilde{X}_l$ и является настройкой на подвыборку $\widetilde{X}_k$ выборки $\widetilde{X}_l$.

В задачах обучения обычно предполагается, что выборка $\widetilde{X}_l$ случайно и независимо извлекается из генеральной совокупности выборок $\aleph^l$. Ниже используется модель извлечения из генеральной совокупности $\aleph^l \times \{0,1\}^l$. В случайно извлеченной паре $(\widetilde{X}_l, \widetilde{\alpha}_l)$ произвольный зафиксированный булев вектор $\widetilde{\alpha}_l$ может появиться с некоторой вероятностью.

Теорема 3. Пусть вероятностная модель извлечения выборки из генеральной совокупности $\aleph^l \times \{0,1\}^l$ такова, что появление любого булевого вектора $\widetilde{\alpha}_l$ в произвольно выбранной паре $(\widetilde{X}_l, \widetilde{\alpha}_l)$ равновероятно. Тогда вероятность $P(S,l,\delta l)$ случайной настройки на какие-нибудь $l-\delta l$ элементов извлеченной выборки $\widetilde{X}_l$ удовлетворяет неравенству $P(S,l,\delta l) < C_l^{\delta l} 2^{-(l-K_l(S)-\delta l)}$, где $K_l(S)$ — колмогоровская сложность семейства S, а δl — число ошибок, допущенных на обучающей выборке $(\widetilde{X}_l, \widetilde{\alpha}_l)$ алгоритмом, выбранным из семейства S в результате обучения.

Доказательство. Семейство S однозначно порождает конечное множество $M_S(\widetilde{X}_l)$ различных способов классификации для любой данной выборки $\widetilde{X}_l$. Мощность множества $M_S(\widetilde{X}_l)$ не превышает $m^S(l)$. Точная настройка на все l элементов выборки может случайно произойти тогда и только тогда, когда способ $\widetilde{\alpha}_l$ классификации последовательности $\widetilde{X}_l$ на два класса содержится во множестве $M_S(\widetilde{X}_l)$. Можно сказать, что точная настройка произойдет тогда, когда входящий в обучающую выборку $(\widetilde{X}_l, \widetilde{\alpha}_l)$ вектор $\widetilde{\alpha}_l$ случайно «попадет» в такую же точку $\widetilde{\alpha}_l$ множества $M_S(\widetilde{X}_l)$. Любой вектор $\widetilde{\alpha}_l$ может появиться в выборке равновероятно по условию теоремы. Поэтому вероятность точной настройки на фиксированную часть выборки длины $l-\delta l$ не превысит $m^S(l) \cdot 2^{\delta l} / 2^l$. Выбрать $l-\delta l$ элементов из l можно $C_l^{\delta l}$ способами. В результате получается оценка $P(S,l,\delta l) < C_l^{\delta l} m^S(l) / 2^{(l-\delta l)}$. Поскольку $K_l(S) =]\log m^S(l)[$, то $2^{K_l(S)} \geq m^S(l)$. Поэтому $P(S,l,\delta l) < C_l^{\delta l} 2^{-(l-K_l(S)-\delta l)}$. ■

Следствие 4. Пусть вероятностная модель извлечения выборки из генеральной совокупности $\aleph^l \times \{0,1\}^l$ такова, что появление любого булевого вектора $\widetilde{\alpha}_l$ в произвольно выбранной паре $(\widetilde{X}_l, \widetilde{\alpha}_l)$ равновероятно. Тогда вероятность $P(S,l,0)$ точной случайной настройки на выборку $(\widetilde{X}_l, \widetilde{\alpha}_l)$ удовлетворяет неравенству $P(S,l,0) < 2^{-(l-K_l(S))}$.

Следствие 5. Пусть вероятностная модель извлечения выборки из генеральной совокупности $\aleph^l \times \{0,1\}^l$ такова, что появление любого булевого вектора $\widetilde{\alpha}_l$ в произвольно выбранной паре $(\widetilde{X}_l, \widetilde{\alpha}_l)$ равновероятно, колмогоровская сложность оценена с помощью метода $pVCD$ и получено неравенство $K_l(S) \leq \mathrm{len}(p)$. Тогда $P(S,l,0) < 2^{-(l-\mathrm{len}(p))}$.

Вероятность неслучайной настройки, т.е. обнаружения закономерности, соответственно оценивается величиной $1-2^{-(l-\mathrm{len}(p))}$.

Если $l-K_l(S) \geq 5$, то $P(S,l,0) < 2^{-5} = 0{,}03125$, и тогда вероятность неслучайного обнаружения закономерности не меньше 0,96. Это вполне приемлемо на практике и позволяет сформулировать следующее правило.

Правило «плюс пять». Для обеспечения надежного извлечения закономерности (решающего правила или алгоритма) из используемого семейства алгоритмов необходимо, чтобы длина обучающей последовательности была хотя бы на пять единиц больше, чем колмогоровская сложность этого семейства.

Применим правило «плюс пять» для класса решающих правил, имеющих вид ДНФ над $n = 100$ переменными длины не более $L = 20$ и не более чем с $\mu = 7$ конъюнкциями. В соответствии с полученной оценкой

$$pVCD(DNF_{L,\mu,n}) < L + (\mu - 1 + L)]\log(n+1)[= 20 + (6+20)7 = 202$$

определяем, что найденная закономерность $DNF_{20,7,100}$, безошибочно классифицирующая всю обучающую выборку длины $l \ge 207$, может считаться неслучайной с вероятностью не менее 0,96.

ЗАКЛЮЧЕНИЕ

Введение понятия колмогоровской сложности $K_l(S)$ семейств алгоритмов, используемых в задачах эмпирического обобщения, позволило получить следующие теоретические и прикладные результаты.

Установлена связь между функцией роста семейства S общерекурсивных функций и колмогоровской сложностью этого семейства: $K_l(S) =]\log m^S(l)[$, а также между емкостью семейства S и его колмогоровской сложностью: $h_S \le K_l(S) < h_S \log l$. Если $m^S(l) \equiv 2^l$, то выполняется равенство $]\log m^S(l)[= l = K_l(S)$. Это означает, что в случае неограниченной емкости класса решающих правил S колмогоровская сложность $K_l(S)$ этого класса равна длине обучающей выборки и достигает своего максимального значения, что указывает на невозможность «сжатия» информации, описывающей семейство S. Показано, что $0 \le K_l(S) \le l$ и качество найденной закономерности при нулевой эмпирической ошибке оценивается вероятностью $2^{-(l-K_l(S))}$.

Разработан метод получения оценок колмогоровской сложности классов алгоритмических решающих правил. Суть этого метода состоит в том, что вместо сложных математических логико-комбинаторных приемов получения оценок используется алгоритмический подход, на основе которого строится информационная строка (слово), позволяющая закодировать любой индивидуальный алгоритм из рассматриваемого класса. Длина этого слова в битах является верхней оценкой колмогоровской и вапниковской сложности изучаемого класса решающих правил.

Особое значение имеет свойство аддитивности *pVCD*-оценок, которое позволяет определить качество найденных при обучении решающих правил, являющихся композициями семейств алгоритмов.

СПИСОК ЛИТЕРАТУРЫ

1. Вапник В.Н. Восстановление зависимостей по эмпирическим данным. — М.: Наука, 1979. — 447 с.
2. Вапник В.Н. Необходимые и достаточные условия равномерной сходимости средних к их математическим ожиданиям // Теория вероятностей и ее применения. — 1981. — **26**, № 3. — С. 543–563.
3. Вапник В.Н. Теория распознавания образов. — М.: Наука, 1974. — 416 с.
4. Донской В.И. Колмогоровская сложность классов общерекурсивных функций с ограниченной емкостью // Таврический вестник информатики и математики. — 2005. — № 1. — С. 25–34.
5. Колмогоров А.Н. Теория информации и теория алгоритмов. — М.: Наука, 1987. — 304 с.
6. Donskoy V.I. The estimations based on the Kolmogorov complexity and machine learning from examples // Proceedings of the Fifth International Conference "Neural Networks and Artificial Intelligence" (ICNNAI'2008). — Minsk: INNS, 2008. — P. 292–297.

Поступила 11.05.2010