УДК 519.95

НЕВЫЧИСЛИМОСТЬ VC-РАЗМЕРНОСТИ СЕМЕЙСТВ КЛАССИФИЦИРУЮЩИХ ФУНКЦИЙ

© В. И. Донской

Таврический национальный университет им. В. И. Вернадского
пр-т Вернадского, 4, Симферополь, Республика Крым, Россия
e-mail: *donskoy@tnu.crimea.ua*

Noncomputability of VC Dimension of Classifier Families.

Donskoy V. I.

Abstract. The following theoretical result is got in the article: Vapnik-Chervonenkis capacity or, speaking otherwise, VC dimension of arbitrary general recursive family of classifiers is noncomputable. VC dimension (VCD) or capacity of families of mappings which decision rules are extracted from, is one of major concepts of machine learning theory. Practice explored that VC dimension succeeded to be found only for a few simple families of classifiers. If to take into account that machine learning implies the use of computers, consideration of VC dimension of families of general recursive functions (algorithms) will be correct. Thus makes sense to examine such families of functions, which defined by neuron networks, decision trees, SVM, and other models in-use in the tasks of machine learning only. Such families, designated $\mathscr{S}$, and functional

$$\mathscr{F} : \mathscr{S} \to VCD(\mathscr{S}),$$

determined on these families and taking on a numerical value equal to the VC-dimension of these families, are examined. By description families of general recursive functions as the lines of arbitrary length, the functional is replaced by function in Turing presentation. Noncomputability of $VCD(S)$ is further proved for arbitrary family $\mathscr{S}$. Kolmogorov complexity of family of general recursive functions $K_l(S)$ is entered to do that, where l is a variable which defines sample length. It is well known that Kolmogorov complexity of arbitrary string is noncomputable. We proved that comlexity $K_l(S)$ is noncomputable as well. Inequality

$$VCD(\mathscr{S}) \le K_l(\mathscr{S}) < VCD(\mathscr{S}) \log l$$

was proven in [3]. Noncomputability of $VCD(S)$ is proved by this inequality usage.

Relation between sample compression, learnability, and VCD was studied in [8]. The compression function takes away from the sample so-called *the compression set*, consisting of no more than k teaching examples (number k is referred to the size of compression).

In the same paper [8] it was proven, that *at the length of sample l and the use of family of classifiers $\mathscr{S}$ there is a scheme of compression of size k, satisfying to inequality*

$$VCD(\mathscr{S}) < k \le VCD(\mathscr{S}) \log l.$$

We proved that the size of compression $k = k(l, \mathscr{S})$ is noncomputable as well.

1. Введение. Постановка задачи

В теории машинного обучения одним из важнейших понятий является VC-*размерность* или *емкость семейств отображений*, из которых извлекаются решающие правила. Если полагать, что машинное обучение подразумевает использование компьютеров, то корректным будет рассмотрение VC-размерности семейств рекурсивных функций (алгоритмов).

Обозначим $\mathfrak{S}$ — семейство общерекурсивных функций (алгоритмов) вида

$$A : X^n \to \{0,1\},\ X^n = \big\{X = (x_1, ..., x_j, ..., x_n) : x_i \in \{0,1,2,\dots,\}\big\}.$$

В теории машинного обучения функции семейства $\mathfrak{S}$ называют *классификаторами*.

(Заметим, что при вычислениях на реальных компьютерах множество значений переменных ограничено: $x_i \in \{0,1,\dots,2^M-1\}$, где M — зафиксированное целое положительное число. Обычно M является разрядностью компьютера — количеством бит, выделяемых для представления одного числа или одного элемента памяти).

В семействе $\mathfrak{S}$ выделим множество подсемейств общерекурсивных классификаторов: $\{\mathscr{S}_1 \subset \mathfrak{S},\ \mathscr{S}_2 \subset \mathfrak{S}, \dots\}$. Будем называть эти подсемейства *классами решающих правил*. Такие классы в рамках парадигмы *машинного обучения* соответствуют семействам общерекурсивных классификаторов, реализуемых, например, нейронными сетями, решающими деревьями, машинами опорных векторов, алгоритмами вычисления оценок [4] и другими алгоритмическими моделями классификации.

Произвольный класс решающих правил (алгоритмов) будем обозначать $\mathscr{S}$, $\mathscr{S} \subset \mathfrak{S}$.

Выборка, состоящая из l произвольных элементов (точек) множества X^n, обозначается $\widetilde{X}_l = X_1, \dots, X_l$ и представляет собой набор $n \times l$ чисел из расширенного натурального ряда. Теоретически и практически допустимо считать все рассматриваемые числа представленными в виде бинарных строк. Множество всех выборок обозначается $\mathscr{X}^l$.

Применение произвольного классификатора $A \in \mathscr{S}$ к l точкам выборки $\widetilde{X}_l$ порождает l двоичных значений – бинарную строку

$$\tilde{y} = (y_1, \dots, y_j, \dots, y_l)\ : y_j = A(X_j) \in \{0,1\},\ j = 1, \dots, l.$$

Будем называть строку $\tilde{y}$ *разбиением выборки* $\widetilde{X}_l$ *на два класса* в соответствии со значениями 0 и 1 функции (алгоритма) A и использовать обозначение $\tilde{y_A} = A(\tilde{X}_l)$.

Применение одного и того же алгоритма к различным выборкам и применение различных алгоритмов к одной и той же выборке дает, вообще говоря, различные

разбиения. Алгоритмы, порождающие одинаковые разбиения любых допустимых выборок, будем называть *подклассом эквивалентных алгоритмов* семейства $\mathscr{S}$.

Определение 1. [1] VC-размерностью или емкостью семейства функций $\mathscr{S} = \{A : X^n \to \{0,1\}\}$, обозначаемой $VCD(\mathscr{S})$, называется наибольшее значение l^* такое, что найдется выборка $\widetilde{X}_{l^*}$, которая может быть разбита всеми 2^{l^*} способами алгоритмами семейства $\mathscr{S}$:

$$\exists \tilde{X}_{l^*} \ : \mid \{\tilde{y} : \tilde{y} = A(\tilde{X}_{l^*}), A \in \mathscr{S}\} \mid = 2^{l^*},$$

но никакая выборка длины большей, чем l^*, разбита всеми способами быть не может. Если же при любом l найдется выборка, разбиваемая всеми 2^l способами, то VC-размерность семейства $\mathscr{S}$ полагается неограниченной (∞).

Определение 2. Пусть $\mathscr{B}(\mathfrak{S})$ — множество всех подмножеств семейства $\mathfrak{S}$ и $\mathscr{K} \in \mathscr{B}(\mathfrak{S})$. Назовем отображение

$$\mathscr{F} : \mathscr{S} \to VCD(\mathscr{S}), \quad \mathscr{F} \in \mathscr{K},$$

функционалом комбинаторной размерности (VC*-размерности*).

Целью дальнейшего изложения является выяснение вопроса: является ли функционал комбинаторной размерности вычислимым?

2. Представление функционала комбинаторной размерности в виде функции, предназначенной для вычисления на машине Тьюринга

Функционал $\mathscr{F}$ ставит в соответствие некоторому подклассу общерекурсивных функций $\mathscr{S}$ число $VCD(\mathscr{S})$. Переходя к эквивалентному тьюринговскому языку представления рекурсивных функций, покажем, что *интересующие нас* семейства $\mathscr{S}$ при зафиксированном (но любом!) значении параметров могут быть представлено словом $W(\mathscr{S})$ на ленте машины Тьюринга. При таком представлении функционал $\mathscr{F}$ интерпретируется заданием функции, которая по слову $W(\mathscr{S})$ должна, если функционал вычислим, выдавать число $VCD(\mathscr{S})$. Указанное представление упростит рассмотрение вопроса о вычислимости функционала $\mathscr{F}$.

Приведем примеры конструирования слова $W(\mathscr{S})$ для некоторых используемых при машинном обучении классов $\mathscr{S}$.

2.1. **Семейство классификаторов, представляемых бинарными решающими деревьями (БРД).** Программирование слова $W_{\mu\text{БРД}}$ для представления семейства $\mathscr{S}_{\mu\text{БРД}}$ БРД с μ листьями основано на представлении каждой из $\mu - 1$ вершин ветвления словом-атомом, состоящим из двух частей – префикса и окончания атома:

Код номера переменной или значение решающей функции (0 или 1)

Номер следующего атома в конкатенации или значение решающей функции (0 или 1)

Префикс атома может иметь $n + 1$ значение, если 0 и 1 резервируются для значений классифицирующей функции, а значениями $2, 3, \ldots, n + 1$ кодируются номера переменных-признаков $1, 2, \ldots, n$. Окончание атома может иметь μ значений: 0 и 1 резервируются также, как в префиксе. Остальные $\mu - 2$ значений соответствуют направленным рёбрам дерева, являющимися указателями на решающие вершины дерева (атомы списка). Указатель на одну (начальную вершину дерева) не требуется: нужны указатели только на $\mu-2$ оставшиеся внутренние вершины. Всего получается μ значений для окончания атома.

Слово $W_{\mu\text{БРД}}$ будет конкатенацией вида: <атом><атом>...<атом>. Если значения префиксов и окончаний всех атомов слова $W_{\mu\text{БРД}}$ зафиксировать, то будет задан некоторый единственный алгоритм $A \in \mathscr{S}_{\text{БРД}}$. Если же значения всех префиксов и окончаний считать пробегающими все допустимые значения, то слово $W_{\text{БРД}}$ будет представлять все семейство $\mathscr{S}_{\mu\text{БРД}}$.

Таким образом получается описание любого семейства БРД-классификаторов для любого сколь угодно большого (но конечного) μ.

2.2. **Семейство классификаторов $\mathscr{S}_{NN}$, представляемых нейронными сетями.** Описание слова $W_{\mu NN}$ определяется следующим образом.

Будем использовать гёделевы номера машин Тьюринга для представления рекурсивных функций ядер нейронных сетей. Узлы нейронной сети будут представляться атомами, состоящими из описания списка входов, строкой-описанием функции ядра и описанием выхода. Каждый узел имеет номер. Каждый вход узла имеет идентификатор, состоящий из номера узла и номера его входа. Каждый выход узла снабжается указателем на некоторый вход какого-либо узла. Свободные входы (на которые не направлен никакий указатель) предназначаются для приема описания классифицируемых объектов. Свободный выход предназначается для значений, выдаваемых нейронным классификатором.

Слово $W_{\mu NN}$ будет являться конкатенацией описаний узлов.

Если разрешить в слове $W_{\mu NN}$ любые допустимые значения параметров, то будет получено описание класса $\mathscr{S}_{\mu NN}$. Для любых возможных значений μ получается описание семейства $\mathscr{S}_{NN}$.

2.3. **Семейство классификаторов $\mathscr{S}_{k-NN}$ — по методу ближайших соседей.** Для формирования слова $W_{\mathscr{S}_{k-NN}}$ в каждом допустимом случае используется сама входная выборка, описание числа k, описание рекурсивной функции расстояния и описание рекурсивной функции вычисления $\arg\min$.

3. Колмогоровская сложность и вычислимость VC-размерности

Определение 3. [6] Колмогоровская сложность слова (строки) x *при заданном способе описания* – вычислимой функции (декомпрессоре) D есть

$$KS_D(x) = \min\{l(p)|D(p) = x\},$$

если существует хотя бы одно двоичное слово p такое, что $D(p) = x$. Иначе полагается, что значение сложности не ограничено. Будем говорить, что в таком случае колмогоровская сложность не определена.

Здесь и далее $l(p)$ обозначает длину слова p в битах.

Определение 4. Условная колмогоровская сложность слова x при заданном слове y есть

$$KS_D(x|y) = \min\{l(p)|D(p, y) = x\};$$

если y – пустое слово, то $KS_D(x|y) = KS_D(x)$

Определение 5. Говорят, что декомпрессор D_1 (слова x) не хуже декомпрессора D_2, если $KS_{D_1}(x|y) \leq KS_{D_2}(x|y) + O(1)$. Декомпрессор называют оптимальным, если он не хуже любого другого декомпрессора.

Теорема 1. (*Соломонова-Колмогорова*) [6]. *Существуют оптимальные декомпрессоры.*

Эта теорема позволяет использовать в определении колмогоровской сложности произвольный оптимальный декомпрессор.

Определение 6. [6] Колмогоровской сложностью $KS(x)$ слова (строки) x называют сложность $KS_U(x)$ при способе описания U, являющемся произвольным оптимальным декомпрессором. Соответственно, условной колмогоровской сложностью $KS(x|y)$ слова x при заданном слове y называют сложность $KS_U(x|y)$.

Доказан и хорошо известен следующий факт теории колмогоровской сложности [5, 2, 9]:

Теорема 2. *Колмогоровская сложность $KS(x|y)$ не является вычислимой функцией.*

Определение 7. [3] Пусть U — такая частично-рекурсивная функция, что для каждого алгоритма $A \in \mathscr{S}$ и для любой выборки $\widetilde{X}_l$ найдется двоичное слово p, которое обеспечивает выполнение равенства $U(p, \widetilde{X}_l) = \widetilde{y}_A$, где $\widetilde{y}_A = A(X_1), ..., A(X_l)$ — двоичное слово (строка) длины l. При этом каждый алгоритм $A \in \mathscr{S}$ полагается определенным на каждой выборке $\widetilde{X}_l$ из X^l. Функция U с указанными свойствами существует 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
$$K_U(A|\widetilde{X}_l) = \min\{len(p) : U(p, \widetilde{X}_l) = \widetilde{y}_A\}.$$
2. Сложность алгоритма A на множестве $\mathscr{X}^l$ по частично-рекурсивной функции U есть
$$K_{U,\mathscr{X}^l}(A) = \max_{\widetilde{X}_l \in \mathscr{X}^l} K_U(A|\widetilde{X}_l)$$
3. Сложность семейства алгоритмов $\mathscr{S}$ на множестве $\mathscr{X}^l$ по частично-рекурсивной функции U есть
$$K_{U,\mathscr{X}^l}(\mathscr{S}) = \max_{A \in \mathscr{S}} K_{U,\mathscr{X}^l}(A).$$
4. Сложность семейства алгоритмов $\mathscr{S}$ на множестве $\mathscr{X}^l$ есть
$$K_l(\mathscr{S}) = \min_{U \in P_{p.r.}} K_{U,\mathscr{X}^l}(\mathscr{S}).$$

В приведенном определении сложность семейства алгоритмов $\mathscr{S}$ на множестве всех возможных выборок $\mathscr{X}^l$ длины l – это наименьшая длина двоичного слова p, по которому можно восстановить самый сложный (и любой) алгоритм $A \in \mathscr{S}$. Важно, что слово p обрабатывается одной и той же функцией (программой) U^*, причем, согласно 4°, наилучшей в следующем смысле. Программа U^* обеспечивает наибольшее сжатие информации о семействе $\mathscr{S}$ в слово p длины $K_l(\mathscr{S})$. Мажоранту сложности $K_l(\mathscr{S})$ можно получить, если точно указать структуру слова p, подлежащего расшифровке, и его длину в битах, а также представить алгоритм обработки этого слова, который будет использоваться вместо программы U^* для оценивания сложности сверху.

Теорема 3. *Колмогоровская сложность $K_l(\mathscr{S})$ произвольного семейства общерекурсивных функций $\mathscr{S}$ невычислима.*

Доказательство. В определении колмогоровской сложности $K_l(\mathscr{S})$ содержится невычислимое (в силу теоремы 2) выражение

$$K_U(A|\widetilde{X}_l) = \min\{len(p) : U(p, \widetilde{X}_l) = \widetilde{y}_A\}.$$

Это приводит к невычислимости $K_l(\mathscr{S})$. □

Теорема 4. [3] *Пусть система частично-рекурсивных функций $\mathscr{S}$ вида $A : X^n \longrightarrow \{0, 1\}$ имеет ограниченную емкость $h_{\mathscr{S}} = VCD(\mathscr{S})$ и колмогоровскую сложность $K_l(\mathscr{S})$. Тогда при конечных значениях $h_{\mathscr{S}} \geq 2$ и $l > h_{\mathscr{S}}$ имеет место двойное неравенство*

$$h_{\mathscr{S}} \leq K_l(\mathscr{S}) < h_{\mathscr{S}} \log l. \tag{1}$$

Теорема 5. *VC-размерность произвольного рекурсивного семейства $\mathscr{S}$ невычислима.*

Доказательство. Предположим, что $h_{\mathscr{S}}$ вычислима. Из неравенств (1) следует

$$K_l(\mathscr{S}) = h_{\mathscr{S}} + j,$$

где j – константа из целочисленного отрезка $0, 1, 2, \ldots, [h_{\mathscr{S}}(logl - 1)]$. Тогда $K_l(\mathscr{S})$, как сумма двух вычислимых слагаемых – $h_{\mathscr{S}}$ и константы, – также должна быть вычислимой. Но это приводит к противоречию: в силу теоремы 3, колмогоровская сложность $K_l(\mathscr{S})$ вычислимой не является. □

Связь между сжатием обучающей выборки, обучаемостью и VCD была изучена в работе Флойда и Вармута [8]. Функция сжатия отбирает из обучающей выборки так называемое *множество сжатия*, состоящее из не более чем k обучающих примеров (число k называют размером сжатия).

В этой же работе [8] было доказано, что *при длине обучающей выборки l и использовании семейства классификаторов $\mathscr{S}$ существует схема сжатия размера k, удовлетворяющая неравенству*

$$VCD(\mathscr{S}) < k \leq VCD(\mathscr{S})logl. \tag{2}$$

Теорема 6. *Размер сжатия $k = k(l, \mathscr{S})$ при использовании для обучения рекурсивного семейства $\mathscr{S}$ и длине обучающей выборки, равной l, невычислим.*

Доказательство. Предположим, размер сжатия k является вычислимым. Но тогда, с учетом неравенства (2),

$$VCD(\mathscr{S}) = k - j \geq 0, \tag{3}$$

где j — некоторая константа. Учитывая, что $VCD(\mathscr{S})$ непустого семейства $\mathscr{S}$ принимает положительные целочисленные значения, можно сделать вывод, что $VCD(\mathscr{S})$ вычислима. Действительно, при таком предположении размер k – вычисл*и*м, константа j – вычисл*и*ма и $k \dot{-} j$ – вычисл*и*мая функция (здесь $\ll \dot{-} \gg$ – рекурсивная функция – усеченная разность,– заменяющая обычное вычитание в формуле (3)). Но сделанное предположение противоречит теореме 5: $VCD(\mathscr{S})$ является невычислимой. □

Заключение

В статье получен следующий теоретический результат: емкость Вапника-Червоненкиса или, говоря иначе, VC-размерность произвольного общерекурсивного семейства классификаторов невычислима.

Направление дальнейших исследований связано с повышением точности оценок VC-размерности на основе метода $pVCD$ [3].

Список литературы

1. *Вапник В. Н., Червоненкис А. Я.* Теория распознавания образов. М.: Наука, 1974. — 416 с.
   V. N. Vapnik *Statistical Learning Theory.* Wiley, New York, 1998.

2. *Вьюгин В. В.* Колмогоровская сложность и алгоритмическая случайность / В. В. Вьюгин. — М.: МФТИ, 2012. — 131 с.

   V. V. V'yugin *Kolmogorov Complexity and Algorithmic Randomness.* MPTI, Moscow, 2012.

3. *Донской В. И.* Сложность семейств алгоритмов обучения и оценивание неслучайности извлечения эмпирических закономерностей / В. И. Донской // Кибернетика и системный анализ, 2012. — № 2. — С. 86–96.

   V. I. Donskoy. Complexity of families of learning algorithms and estimation of empirical pattern extraction nonrandomness // *Cybernetics and System Analysis*, 2012, 2, pp. 86–96.

4. *Журавлев Ю. И.* Алгоритмы распознавания, основанные на вычислении оценок / Ю. И. Журавлев, В. В. Никифоров // Кибернетика, 1971. — № 3. — С. 1–11.

   Yu. I. Zhuravlev. Recognition algorithms based on estimates calculation // *Cybernetics*, 1971, 3, pp. 1–11.

5. *Звонкин А. К., Левин Л. А.* Сложность конечных объектов и обоснование понятий информации и случайности с помощью теории алгоритмов / А. К.|;Звонкин, Л. А. Левин // Успехи математических наук, 1970. — Т. 25:6(156). — С. 85–127.

   A. K. Zvonkin, L. A. Levin. The complexity of finite objects and the development of the concepts of information and randomness by means of the theory of algorithms // *Uspekhi Mat. Nauk*, 1970, 25:6(156), pp. 85–127.

6. *Колмогоров А. Н.* Теория информации и теория алгоритмов // А. Н. Колмогоров. — М.: Наука, 1987. — 304 с.

   A. N. Kolmogorov. Selected Works. Volume III: *Information Theory and the Theory of Algorithms*. Math. and its Applications, Volume 27, 1993.

7. *Успенский В. А., Верещагин Н. К., Шень А.* Колмогоровская сложность и алгоритмическая случайность / В. А. Успенский, Н. К. Верещагин, А. Шень. — М.:МЦНМО, 2010. — 556 с.

   V. A. Uspensky, N. K. Vereshchagin, A. Shen. *Kolmogorov complexity and algorithmic randomness*. MCCME, Moskow, 2010.

8. *Floyd S., Warmuth M.* Sample Compression, learnability, and the Vapnik-Chervonenkis dimension / Sally Floyd, Manfred Warmuth // J. Machine Learning. – 1995. — Vol. 21. — Iss. 3. — P. 269–304.

9. *Li M., Vitanyi P.* An introduction to Kolmogorov complexity and its applications / Ming Li, Paul M. B. Vitanyi. — New York: Springer-Verlag, 1997. — 637 p.

Статья поступила в редакцию 02.06.2014