УДК 519.95

КОЛМОГОРОВСКАЯ СЛОЖНОСТЬ И ЕЕ ПРИМЕНЕНИЕ В МАШИННОМ ОБУЧЕНИИ

© В. И. Донской

Таврический национальный университет им. В.И. Вернадского
факультет математики и информатики
пр-т Вернадского, 4, г. Симферополь, 95007, Украина
e-mail: *donskoy@tnu.crimea.ua*

Abstract. The materials represented in this article carry, mainly, surveying character. The aim of the paper is complete enough presentation of possibilities of mathematical apparatus of algorithmic complexity and probability for application in machine learning. Nevertheless, some new results are presented: theorems about exact compressors and decompressors, approach to determination of the moment of stopping of learning procedure on the basis of complexity analogue of the Bayes rule et al.

Введение

Машинное обучение — важнейшее из направлений теоретических исследований и создания приложений в современной информатике. Сложность решения задач в этой области для математиков определяется прежде всего неполнотой начальной информации и неоднозначностью получаемых решений. Это объясняет появление множества подходов и попыток не только к построению теории машинного обучения и соответствующих алгоритмов, но и даже к самому определению обучаемости [5].

Хорошо обоснованные и приемлемые для практического использования математические результаты в этой области получаются не всегда. Но замечательной особенностью развития машинного обучения является то, что практика зачастую опережает теорию, и появляются успешно работающие обученные программы и автоматы.

Использование колмогоровской сложности и алгоритмической случайности [9, 10] в теории машинного обучения позволяет синтезировать алгоритмы обучения, основываясь на идеях отождествления эмпирического обобщения данных с их максимальным сжатием [18, 21].

Представляемые в данной статье материалы носят, главным образом, обзорный характер. *Целью работы* работы является достаточно полное представление возможностей математического аппарата алгоритмической сложности и случайности для применения в машинном обучении. Тем не менее, в статье содержатся и *новые результаты*: теремы о точных компрессорах и декомпрессорах, подход к определению момента остановки процедуры обучения на основе сложностного аналога правила Байеса и другие.

1. Основные понятия колмогоровской сложности

Определение 1. [10] Колмогоровская сложность слова 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 2. 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 3. 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*) [10]. *Существуют оптимальные декомпрессоры.*

Доказательство. Покажем, что найдется такая частично рекурсивная функция-декомпрессор A, что для любой другой частично рекурсивной функции-декомпрессор $D = D(p, y)$ будет выполнено неравенство

$$KS_A(x|y) \leq KS_D(x|y) + c_D.$$

Здесь c_D — константа, не зависящая от x и y. Используя универсальную частично рекурсивную функцию U с подходящим номером n, для любого декомпрессора D можно записать равенство

$$D(p, y) = U(n, (p, y)) = x.$$

Колмогоровская сложность относительно декомпрессора D есть

$$KS_D(x|y) = l(p).$$

Далее, осуществляя группировку аргументов, можно определить функцию A следующим образом:

$$A((n, p), y) = U(n, (p, y)) = x.$$

Здесь пара слов (n, p) рассматривается как их конкатенация, длина которой есть $l(np) = l(p)+l(n)$. Тогда $A((n, p), y) = D(p, y) = x$ для любого допустимого декомпрессора D. Поэтому для любого номера функции n, определяющего декомпрессор D, найдется константа $c_D \geq l(n)$, зависящая только от выбора этого декомпрессора, такая, что

$$KS_A(x|y) = l(np) \leq l(n) + l(p) + \delta = KS_D(x|y) + c_D,$$

где константа δ определяет дополнительное число бит, которое может потребоваться для того, чтобы входящий в конкатенацию np номер используемой универсальной функции n мог быть отделен от аргумента p. Это можно сделать разными способами независимо от слова p, например, при помощи специального так называемого самоограничивающего кодирования. Подробнее это будет показано ниже при разборе определения колмогоровской сложности, данного Витаньи и Ли [25]. □

Замечание. Конкатенация xy двух строк x и y не может рассматриваться как пара (xy), поскольку в конкатенации, вообще говоря, не содержится информация о нужном разделении строки xy на две подстроки. Поэтому конкатенация дополняется информацией, обеспечивающей её правильное разделение.

Определение 4. Функция $f(x)$ называется перечислимой сверху, если существуют вычислимая функция $F(x, k)$, определенная для всех слов x и всех натуральных чисел k, для которой $F(x, 0) \geq F(x, 1) \geq F(x, 2) \geq ...$ и $f(x) = \lim_{k\to\infty} F(x, k)$ для каждого значения x. При любом k значение $F(x, k)$ является верхней оценкой для $f(x)$. Функция $f(x)$ называется перечислимой снизу, если существует аналогичная нижняя оценка $L(x, k)$.

Теорема 2. *Функция KS перечислима сверху, причём $\big|\{x : KS(x) < n\}\big| < 2^n$ для всех n.*

Доказательство. Покажем, что множество пар $\{\langle n, x\rangle : KS(x) < n\}$, где n — натуральное число, а x — двоичное слово, перечислимо. Если $KS(x) < n$, то существует фигурирующая в определении KS вычислимая функция — декомпрессор D. Используя установленный стандартный порядок двоичных слов, можно организовать вычисления, начиная с $k = 0$, в соответствии с этим порядком. Т. е. перебирать слова p по мере роста их длины, соблюдая условие $KS(x) < n$. Будут перебираться все слова, длина которых не превышает n. Как только окажется, что $D(p) = x$, перечисляющий алгоритм будет выдавать пару $\langle l(p) + k, x\rangle$ и увеличивать k на единицу. Если первая выдача будет парой $\langle l(p) + 0, x\rangle$, то выдаваемая перечисляющая последовательность будет иметь вид $\langle l(p) + 0, x\rangle, \langle l(p) + 1, x\rangle, \langle l(p) + 2, x\rangle...$. Поскольку перебираются все слова длины не больше n, то сумма этих длин $\sum_{i=0}^{n-1} 2^{-i} = 2^n - 1 < 2^n$. Поэтому

$|\{x : KS(x) < n\}| < 2^n$. Определим функцию $F(x,k) = l(p) + n - k$ как последовательность оценок сверху сложности $KS(x)$, полагая $F(k,x) = \infty$ при $k > n$. Тогда $F(x,0) \geq F(x,1) \geq F(x,2) \geq ...$ и $KS(x) = \lim_{k\to\infty} F(x,k)$, поскольку это предельное соотношение соответствует неравенству $k > n$ для любого заданного n. □

Лемма 1. *Для любой вычислимой функции $f(x)$ имеет место неравенство $KS(f(x)) \leq KS(x) + O(1)$ для всех тех значений x, когда $f(x)$ определена.*

Доказательство. Пусть D — оптимальный декомпрессор в определении $KS(x) = KS_D(x) = \min\{l(p) : D(p) = x\}$. Возьмем в качестве другого декомпрессора композицию вычислимых функций $f \circ D$ и рассмотрим

$$KS_{f\circ D}(f(x)) = \min\{l(p) : f(D(p)) = f(x))\} = \min\{l(p) : D(p) = x\} = KS(x).$$

$$KS(f(x)) \leq KS_{f\circ D}(f(x)) + O(1).$$

□

Теорема 3. *Любая частично рекурсивная (вычислимая) функция $L(x)$ такая, что $L(x) \leq KS(x)$ в тех точках, в которых $L(x)$ определена, ограничена некоторой константой C, то есть $L(x) \leq C$ для всех x.*

Доказательство. Предположим, что существует вычислимая функция $L(x)$, являющаяся оценкой снизу колмогоровской сложности: $L(x) \leq KS(x)$. Определим функцию $A(n)$, которая ставит в соответствие натуральному числу n минимальное в порядке перечисления значение x такое, что $L(x) \geq n$. Функция $A(n)$ будет вычислимой в силу предположения, что $L(x)$ вычислима. Тогда $L(A(n)) \leq KS(A(n))$ по сделанному предположению, что $L(x) \leq KS(x)$. Согласно определению функции $A(n)$, имеет место неравенство $L(A(n)) \geq n$. Согласно лемме 1, $KS(A(n)) \leq KS(n) + c_1$. Получается цепочка неравенств:

$$n \leq L(A(n)) \leq KS(A(n)) \leq KS(n) + c_1 \leq logn + c_2,$$

где c_1 и c_2 — некоторые константы. Но следующее из этой цепочки неравенство $n \leq \log n + c_2$ не выполняется для всех n, больших некоторого значения n_0. Полученное противоречие доказывает теорему. □

Замечание. Теорема 3 доказывает несуществование именно функции — нижней оценки колмогоровской сложности для произвольного x.

Теорема 4. *Колмогоровская сложность KS не является вычислимой функцией.*

Доказательство. Предположив, что KS вычислима, получим, что вычислима функция $f(x) = KS(x) - 1$, и тогда $f(x) \leq KS(x)$ для всех непустых строк x. Но такой нижней оценки для колмогоровской сложности не существует согласно теореме 3. □

Теорема 5. *Колмогоровская сложность* $KS_D(x) = \min\{l(p)|D(p) = x\}$ *конечной строки* x *определена тогда и только тогда, когда существует машина Тьюринга* T_C *(компрессор) такая, что* $T_C(x) = p$.

Доказательство. Действительно, если существует машина Тьюринга D такая, что $D(p) = x$, то существует система подстановок Маркова M_D, алгоритмически эквивалентная МТ D (реализующая тот же самый алфавитный оператор). Применение M_D к слову p даст $x = M_D(p)$. Зафиксируем выполненную при этом последовательность марковских подстановок:

$$\tilde{S}(M_D, y, x) = \{s_1, ..., s_j, ..., s_\mu : s_j = \lambda_j \to \rho_j\},$$

где λ_j — левая часть подстановки (замещаемое подслово), а ρ_j — правая часть подстановки (замещающее подслово), вместе с последовательностью $k_1, ..., k_j, ..., k_\mu$ номеров символов текущего обрабатываемого слова, начиная с которых реализуются подстановки. Тогда компрессор T_C может быть композицией машин Тьюринга двух типов: подвода головки к символу с номером k_j (обозначим эти машины T_j^1) и заменой подслова ρ_j на подслово λ_j (обозначим их T_j^2). Применение к слову x последовательно машин $T_\mu^1, T_\mu^2, ..., T_j^1, T_j^2, .., T_1^1, T_1^2$ даёт композицию T_C такую, что $T_C(x) = p$ (машина T_1^2 должна быть снабжена заключительным состоянием). Аналогично доказывается, что если для строки x существует машина Тьюринга T_C (компрессор) такая, что $T_C(x) = p$, где p – некоторая строка, то можно указать соответствующую ей машину-декомпрессор D_{T_C} такую, что $D_{T_C}(p) = x$, и тогда колмогоровская сложность $KS_D(x) = \min\{l(p)|D(p) = x\}$ будет определена. □

Определение 5. Назовем точной колмогоровской сложностью строки x

$$KC(x) = \min_{\{D|D(p)=x\}} \min\{l(p)|D(p) = x\}.$$

Как видно из последнего определения, точная колмогоровская сложность определяется наилучшим декомпрессором.

Теорема 6. *Точная колмогоровская сложность не является вычислимой функцией.*

Доказательство. Если бы KC была вычислима, то она была бы нижней оценкой колмогоровской сложности KS: $KC(x) \leq KS(x)$. Но таковой оценки не существует по теореме 3. □

Определение 6. Пусть x — конечная строка, и множество её компрессоров $\mathfrak{T}_C(x) = \{T_C | T_C(x) = p\}$ не является пустым. Назовем

$$K_T(x) = \min_{T_C \in \mathfrak{T}_C} \{l(p) | T_C(x) = p\}$$

сжатием строки наилучшим компрессором.

Очевидно, для конечной строки x сжатие удовлетворяет двойному неравенству $0 \leq K_T(x) \leq l(x)$. Значение 0 соответствует пустой строке.

Теорема 7. *Если $l(x) < \infty$, то $KC(x) = K_T(x)$.*

Доказательство. Предположим, что $KC(x) < K_T(x)$. Зафиксируем наилучший декомпрессор D^*, соответствующий значению $KC(x) = l(p^*)$ на слове p^*. Зафиксируем это слово p^* – кратчайшее описание строки x. Используя марковское представление декомпрессора D^*, построим, как это было сделано при доказательстве теоремы 5, алгоритм-компрессор T_{D^*} такой, что $T_{D^*}(x) = p^*$. Но тогда $K_T(x) \leq KC(x)$. Точно также, предположив, что $K_T(x) < KC(x)$, используем наилучший компрессор для построения соответствующего декомпрессора, и получим $K_T(x) \geq KC(x)$. □

В работах [25, 26] исходная колмогоровская сложность $KS(x)$ определяется, на первый взгляд, иначе (и обозначается $C(x)$). Используется понятие *самоограничивающего кода* $\bar{x}$ заданной бинарной строки $x_1x_2...x_n$, который определяется соотношением $\bar{x} = x_1x_1x_2x_2...x_{n-1}x_{n-1}x_n\neg x_n$. В правой части этого соотношения содержится префиксный код, в котором каждая пара символов, кроме последней, одинакова, а последний символ является инверсией предпоследнего. Действительно, пусть x и y – две бинарных строки такие, что x является префиксом строки y, то есть $y = x\tau$ при непустом окончании τ . Обозначим длины этих строк $l(x) = n$ и $l(y) = m$. Убедимся, что код $\bar{x}$ не будет префиксом кода $\bar{y}$:

$$x = x_1x_2...x_n;$$

$$y = x_1x_2...x_ny_{n+1}...y_m;$$

$$\bar{x} = x_1x_1x_2x_2...x_{n-1}x_{n-1}x_n\neg x_n;$$

$$\bar{y} = x_1x_1x_2x_2...x_{n-1}x_{n-1}x_nx_ny_{n+1}y_{n+1}...y_m\neg y_m.$$

Используя такой префиксный код, определяют *стандартный самоограничивающий код* x' для любой строки x согласно соотношению $x' = \overline{l(x)}x$. Это соотношение определяет, что к исходной строке приписывается префикс, являющийся самоограничивающим кодом ее длины, и $l(x') = n + 2]\log n[$, где $n = l(x)$.

Определение 7. [26] Пусть $T_1, T_2, ..., T_i, ...$ — стандартное перечисление машин Тьюринга, а $\phi_1, \phi_2, ..., \phi_i, ...$ — перечисление соответствующих этим машинам частично рекурсивных функций. Колмогоровская сложность строки x по заданной строке y определяется выражением

$$C(x|y) = \min_{p,i}\{l(i^{'}p) : \phi_i(p, y) = x, p \in \{0,1\}^*, i \in \mathbf{N}\}; \qquad C(x) = C(x|\lambda).$$

В этом определении Витаньи и Ли сложность слова x определяется длиной конкатенации номера $i^{'}$ машины-декомпрессора D_i, представленного в самоограничивающем коде, и кода слова x. Пусть

$$(p^*, i^*) = arg\min_{p,i}\{l(i^{'}p) : \phi_i(p, y) = x, p \in \{0,1\}^*, i \in \mathbf{N}\}$$

По слову $i^{'}$, представленному в самоограниченном коде, можно определить описание декомпрессора (машины) i и отделить его от слова p. Затем можно выполнить программу i (промоделировать её) на любом другом допустимом декомпрессоре — машине D. Тогда

$$C(x|y) \leq C_D(x|y) + l(i^{'}),$$

откуда следует, что $C(x|y) = KS(x|y)$ — колмогоровская сложность относительно некоторого оптимального способа описания D_{i^*}.

Теорема 8. (*О сложности конкатенации строк*). *Пусть xy — конкатенация строк x и y. Тогда выполняется неравенство*

$$KS(xy) \leq KS(x) + 2\log KS(x) + KS(y) + c, \tag{1}$$

где c — некоторая константа.

Доказательство. Пусть p и q — такие слова, что $KS(x) = l(p)$ и $KS(y) = l(q)$. Пусть $D^{'}$ — произвольный декомпрессор. Предположим, что имеет место равенство $D^{'}(pq) = xy = D(p)D(q)$. Но $D^{'}(pq)$ не может быть определено однозначно, поскольку разные разбиения слова pq на части $p_1q_1 = p_2q_2 = pq$ могут давать различные результаты декомпрессии. Чтобы разделение конкатенации pq было корректным, можно применить самоограничивающий код $\overline{l(p)}pq$, чем обеспечивается выполнение условия $D^{'}(\overline{l(p)}pq) = D(p)D(q) = xy$. Тогда

$$KS_{D'}(xy) = 2\log l(p) + l(p) + l(q);$$

$$KS_{D'}(xy) = KS(x) + 2\log KS(x) + KS(y).$$

Переходя от декомпрессора $D^{'}$ к оптимальной машине, согласно теореме Соломонова-Колмогорова получаем неравенство (1) с константой c, не зависящей от x и y. □

Теорема 9. (*Колмогорова-Левина о декомпозиции сложности пары строк*) [9].

$$KS(x,y) = KS(x) + KS(y|x) + O(\log KS(x,y)).$$

2. Префиксная сложность

Префиксная сложность является модификацией простой колмогоровской сложности, приспособленной для построения универсальной вероятностной меры на множестве последовательностей. Если S — некоторое множество строк, в котором любая пара строк удовлетворяет условию: одна из них не является префиксом другой, то множество S называют беспрефиксным. Вычислимая функция $U(p,y)$ двух переменных называется *префиксно-корректной* по первому аргументу, если для любого y множество строк p, на которых эта функция определена, является беспрефиксным. Иногда такую функцию называют самоограниченным декомпрессором. Определение распространяется на случай пустой сроки λ: $U(p,\lambda) = U(p)$. Если $U(p) = x$ для некоторой строки x, то множество $\{p : U(p) = x\}$ является беспрефиксным. И тогда компрессор T_C (см. ниже теорему 11) порождает для всех допустимых конечных строк x беспрефиксное множество.

Определение 8. Пусть U — произвольная вычислимая префиксно-корректная функция. Условная префиксная колмогоровская сложность строки x при условии y есть

$$KP_U(x|y) = \begin{cases} \min\{l(p)|U(p,y) = x\}, & \exists p \;\; U(p,y) = x, \\ \infty, \quad \forall p \quad U(p,y) \neq x \end{cases}$$

Теорема 10. *Существует такая* (*универсальная*) *префиксно-корректная функция* $A = A(p,y)$, *что для любой вычислимой префиксно-корректной функции* $U = U(p,y)$ *и для всех* x *и* y *имеет место неравенство*

$$KP_A(x|y) \leq KP_U(p,y) + O(1).$$

Доказательство. Аналогично доказательству теоремы Соломонова-Колмогорова для сложности KS. □

Определение 9. Условной префиксной сложностью $KP(x|y)$ называют условную префиксную сложность $KP_A(x|y)$ по любой зафиксированной универсальной префиксно-корректной функции A.

Определение 10. Назовем точной условной префиксной сложностью

$$KPC(x|y) = \min_{\{U|U(p,y)=x\}} \min\{l(p)|U(p,y) = x\},$$

если множество префиксно-корректных функций $\{U : U(p, y) = x\}$ не пусто, иначе будем говорить, что точная префиксная сложность не определена, и полагать, что $KPC(x|y) = \infty$.

Если точная префиксная сложность определена, то для любой универсальной вычислимой префиксно-корректной функции U и для любой универсальной префиксно-корректной функции A

$$KPC(x|y) \leq KP_A(x|y) \leq KP_U(p, y) + O(1),$$

$$KPC(x|y) \leq KP(x|y).$$

Поэтому точную префиксную сложность $KPC(x|y)$ можно считать условной префиксной сложностью $KP(x|y)$ (по некоторой наилучшей универсальной вычислимой префиксно-корректной функции U_*). Это позволяет освободиться от латентной константы.

В определении префиксной сложности можно использовать в качестве функции U так называемую префиксную машину Тьюринга. Это приводит к эквивалентному понятию и оказывается полезным для дальнейшего изложения.

Префиксной называют машину Тьюринга T, описываемую, например, следующим образом [12]. Предполагается, что у такой машины помимо рабочей ленты есть входная лента, на которой имеется односторонняя читающая головка. Крайняя левая клетка ленты содержит специальный маркер, справа от которого может быть записана любая последовательность нулей и единиц. Изначально читающая головка находится у левого края входной ленты под специальным маркером. Шаги вычислений машины Тьюринга определяются как символом, который «видит» читающая головка, так и символом, который «видит» головка на рабочей ленте. В зависимости от этих символов и текущего состояния машина предпринимает то или иное действие. Это действие состоит в изменении внутреннего состояния, записи нового символа на рабочей ленте, а также может включать в себя сдвиг и влево, и вправо на рабочей ленте и сдвиг только вправо читающей головки входной ленты. Результат работы машины обычным образом записывается на рабочей ленте, которая изначально является пустой. Когда машина останавливается, читающая головка входной ленты находится в точности над первым пробелом, следующим за заданным на входной ленте словом.

Теорема 11. *Областью определения префиксной машины является беспрефиксное множество.*

Доказательство. Пусть S – множество строк, для которых результат работы префиксной машины T определен. Если $x \in S$, то машина T останавливается при условии, что выполнены все необходимые вычисления, на рабочую ленту выдано результирующее слово $z = T(x)$ и на входной ленте прочитаны в точности все символы строки x, но не более. Последнее условие соответствует нахождению входной головки на символе, следующем за последним символом строки x. Рассмотрим две строки: $x \in S$ и $y \in S$. Предположим, что x является префиксом строки y, то есть $y = x\tau$ при непустом окончании τ. Но тогда, начав работу над словом y, машина T сначала произведёт в точности такие же действия, как при работе над словом x, и затем она остановится, не продолжая просмотр окончания τ слова y. Но тогда результат работы машины на слове y не может быть определен. Это противоречие доказывает, что область определения префиксной машины T — беспрефиксное множество. □

В литературе встречаются другие, эквивалентные определения префиксной машины. В работе [19] префиксная машина Тьюринга T определяется так. Эта машина снабжена тремя лентами: однонаправленной входной лентой (только для чтения), однонаправленной выходной лентой (только для записи) и двунаправленной рабочей лентой. Вдоль однонаправленных лент головка перемещается только слева направо. Все ленты — двоичные, пустой символ не используется. Рабочая лента инициализируется нулями. Машина T останавливается на входе p, выдавая $z = T(p)$, если p находится слева от входной головки, и z находится слева от выходной головки. Множество таких слов p образуют префиксный код. Такие коды называют самоограничивающими программами. Префиксная машина всегда предполагает существование способа, позволяющего указать, где именно на ленте ограничивается входное слово.

Теорема 12. *Для любой префиксной МТ можно указать эквивалентную ей обычную МТ.*

Доказательство. Пусть T — произвольная префиксная машина, заданная своей таблицей команд, а x — произвольная входная строка. Рассмотрим подпрограмму-функцию $Input(x, k)$, возвращающую k-й символ входной строки x. Подпрограмма реализуется подтаблицей с конечным множеством дополнительных состояний. Чтобы получить обычную машину Тьюринга T_1, эквивалентную префиксной машине T, достаточно реализовать указанную подпрограмму внутри последовательности вычислений одноленточной машины. Машина T_1 начинает работу, положив $k = 0$, и пропускает (пройдя до конца вправо) входное слово. Эти действия имитируют подготовку входной ленты префиксной машины. Далее она выполняет шаги, логически эквивалентные последовательности вычислений машины T, вне зоны записи любого

входного слова. Аналогом обращения к выделенной входной ленте префиксной машины T будет обращение к подпрограмме $Input(x, k)$. При таком обращении будет происходить следующее:

- вычисление $k := k + 1$;
- запоминание при помощи специального маркера ячейки ленты, на которой прерываются вычисления;
- переход в начальное состояние подтаблицы-подпрограммы;
- считывание символа $x[k]$;
- подвод к ячейке ленты, соответствующей точке возврата;
- возврат в следующее по логике обработки машины T состояние.

□

Замечание. МТ, суммирующая любой начальный отрезок произвольной конечной двоичной последовательности x, применима к любому её префиксу. Но такой сумматор не реализуем на префиксной МТ. Поэтому

Следствие 1. *Префиксные МТ образуют специфический собственный подкласс машин Тьюринга.*

Следствие 2. *Любая префиксно-корректная вычислимая функция вычислима на МТ без маркера конца входа.*

В справедливости последнего следствия можно убедиться иным способом [3].

Для префиксной сложности KP справедлива такая же теорема о несуществовании нетривиальной вычислимой оценки снизу, как и для колмогоровской сложности KS. Из этой теоремы следует, что префиксная сложность не является вычислимой. Её доказательство [11], такое же, как и доказательство аналогичной теоремы для колмогоровской сложности KS.

Лемма 2.

$$KPC(x, y) \leq KPC(x) + KPC(y).$$

Доказательство. Пусть слово x восстанавливается по кратчайшему слову p наилучшей машиной T_1, соответствующей точной префиксной сложности $KPC(x)$, а слово y восстанавливается по кратчайшему слову q наилучшей машиной T_2, соответствующей точной префиксной сложности $KPC(y)$. По следствию 2 обе эти машины могут не использовать маркер конца входа. Тогда $T_1 \circ T_2(pq) = xy$, где $T_1 \circ T_2$ — композиция

машин Тьюринга. Сначала машина T_1 применяется к слову p и выдаёт x. После её работы головка машины T_2 будет обозревать первый символ слова q. Следовательно,

$$KP_{T_1 \circ T_2}(xy) = KPC(x) + KPC(y) = |p| + |q|.$$

Тогда для любой наилучшей машины $KPC(x, y) \leq KP_{T_1 \circ T_2}(x, y)$. □

Приведем без доказательства еще несколько полезных теорем.

Теорема 13. *Любая частично рекурсивная (вычислимая) функция $L(x)$ такая, что $L(x) \leq KP(x)$ в тех точках, в которых $L(x)$ определена, ограничена некоторой константой C, то есть $L(x) \leq C$ для всех x.*

Теорема 14. *Префиксная сложность не является вычислимой.*

Теорема 15. *Обычная и префиксная сложности связаны неравенством $\forall x KS(x) \leq KP(x) + O(1)$, причем разность $KP(x) - KS(x)$ стремится к бесконечности с ростом длины строки x* [11].

Теорема 16. [11]. *Существует всюду определённая вычислимая функция f, оценивающая сверху KS и на бесконечном множестве равная KS.*

Теорема 17. [11] *Существует всюду определённая вычислимая функция f, оценивающая сверху KP и на бесконечном множестве равная KP.*

3. Универсальное распределение

Определение 11. Вещественнозначная функция $f : \mathbf{N} \to \mathbf{R}$ называется перечислимой, если существует МТ, вычисляющая рекурсивную функцию φ такую, что $\varphi(< x, t >) =< p, q >$, где $\frac{p}{q}$ есть t-е рациональное приближение значение $f(x)$. В этом смысле функцию f, допускающую указанную аппроксимацию, называют рекурсивной.

Определение 12. Будем называть функцию $P : \mathbf{N} \to [0, 1]$ вероятностным распределением, если $\sum_{x \in \mathbf{N}} P(x) \leq 1$. Неравенство вводится для удобства, и полагается, что недостающая вероятность $\varepsilon = 1 - \sum_{x \in \mathbf{N}} P(x)$ сосредоточена на неопределенном элементе $u \notin \mathbf{N}$. В этом случае P называют *полумерой*.

Определение 13. Рассмотрим семейство полумер (вероятностных распределений) $\mathbf{P}_\varepsilon$ на $\mathbf{N}$ (эквивалентно – на $\{0, 1\}^*$). Назовем перечислимую снизу полумеру $\mathbf{m} \in \mathbf{P}_\varepsilon$ максимальной, если для любой другой перечислимой снизу полумеры μ для некоторой константы c и для всех x выполнено неравенство $\mu(x) \leq \mathbf{m}(x)$.

Можно сказать, что максимальная полумера $\mathbf{m}$ «выделяет» так много вероятности каждому объекту, как любое другое распределение семейства $\mathbf{P}_\varepsilon$ с точностью до мультипликативного множителя. В этом смысле она является *универсальной относительно априорной неопределенности*. В некоторых случаях использование полумеры $\mathbf{m}$ в пространстве $\{0,1\}^\infty$ приводит к тем же результатам, которые даёт использование истинного неизвестного априорного распределения.

Теорема 18. *Семейство $\mathbf{P}_\varepsilon$ содержит элемент $\mathbf{m}$, который мультипликативно доминирует все элементы из $\mathbf{P}_\varepsilon$. Иначе говоря, для любой полумеры $P \in \mathbf{P}_\varepsilon$ существует константа c такая, что $c\mathbf{m}(x) > P(x)$ для всех $x \in \mathbf{N}$.*

Доказательство. Можно найти в работах [3, 21, 12]. □

Назовем максимальную в указанном смысле перечислимую снизу полумеру *универсальным распределением*.

Теорема 19.

$$-\log \mathbf{m}(x) = KP(x) + O(1).$$

Доказательство. Сначала докажем неравенство $-\log \mathbf{m}(x) \le KP(x) + O(1)$. Перепишем неравенство в эквивалентной форме $2^{-KP(x)} \le c\mathbf{m}(x)$, где $c \ne 0$ – некоторая константа. В силу максимальности полумеры $\mathbf{m}(x)$ достаточно показать, что функция $2^{-KP(x)}$ является a) *перечислимой снизу* b) *полумерой*. Убедимся в справедливости b). Неравенство $\sum_x 2^{-KP(x)} \le 1$ для полумеры действительно выполняется, так как префиксная сложность $KP(x) = l(x)$ – минимальная длина слова – определена для совокупности слов x, образующий префиксный код. А для префиксного кода справедливо неравенство Крафта $\sum_x 2^{-l(x)} \le 1$.

Убедимся в справедливости a). Известно, что функция префиксной сложности $KP(x)$ перечислима сверху: существует вычислимая функция F такая, что $KP(x) < F(x,k)$ для любого натурального k. Тогда $2^{-KP(x)} > L(x,k) = 2^{-F(x,k)}$, следовательно, $2^{-KP(x)}$ перечислима снизу.

Теперь докажем обратное неравенство: $-\log \mathbf{m}(x) \ge KP(x)+O(1)$. Как уже было показано, функция $2^{-KP(x)} = \mu(x) > 0$ является полумерой; $\mathbf{m}(x) > 0$, поскольку $c\mathbf{m}(x) \ge \mu(x) > 0$. Обозначим

$$\eta = \sup_x |\mathbf{m}(x) - \mu(x)| < 1; \quad \delta = \inf_x \mathbf{m}(x) > 0.$$

Тогда $\mu(x) \ge \mathbf{m}(x) - \eta \ge c_1\mathbf{m}(x)$ для любой константы c_1 такой, что $c_1 \le 1 - \eta/\mathbf{m}(x)$. В качестве $c-1$ можно взять $1 - \eta\ /\delta$, получая $\mu(x) \ge (1 - \eta\ /\delta)\mathbf{m}(x)$ или $2^{-KP(x)} \ge (1 - \eta\ /\delta)\mathbf{m}(x)$, и тогда $-KP(x) \ge \log \mathbf{m}(x) + O(1)$ или $-\log \mathbf{m}(x) \ge KP(x) + O(1)$. □

Следствие 3. $-\log \mathbf{m}(x) = KPC(x) + O(1)$.

4. Сжатие и оценки обучаемости

Определение 14. [13] *Алгоритмом Оккама* с параметрами $\alpha \geq 1$ и $\beta : 0 \leq \beta < 1$ над классом (целевых) гипотез G, в котором сложность любой гипотезы (*длина её бинарного описания*) не превышает n, называется алгоритм обучения, который:

(i) выполняется за полиномиальное время от длины выборки и

(ii) в результате обучения выдаёт гипотезу, имеющую сложность, не превышающую $n^\alpha l^\beta$.

В определении 14 не оговаривается, является ли полученная гипотеза согласованной с обучающей выборкой; кроме этого, выбранная гипотеза может даже не принадлежать классу G.

Теорема 20. [13] *Для алгоритма Оккама над классом (целевых) гипотез G, в котором сложность любой гипотезы не превышает n, независимо от распределения вероятностей на признаковом пространстве (ε, δ)-обучаемость* [5] *имеет место при длине выборки l, оцениваемой как*

$$l = O\left(\frac{1}{\varepsilon}\ln\frac{1}{\delta} + \left(n^\alpha\ /\varepsilon\right)^{1/(1-\beta)}\right),$$

где $\alpha \geq 1$ и $\beta : 0 \leq \beta < 1$.

В случае согласованности алгоритма Оккама с обучающей выборкой $\beta = 0$, и тогда

$$l = O\left(\frac{1}{\varepsilon}\left(n^\alpha + \ln\frac{1}{\delta}\right)\right).$$

Теорема 21. (*Occam's Razor теорема*) [13]. *Пусть G и H — классы концептов. Пусть $g \in G$ — целевой концепт и $n(g)$ – длина его бинарного представления $s(g)$. Пусть A — алгоритм обучения и даны константы $\alpha \geq 1$ и $\beta : 0 \leq \beta < 1$. Предположим, что алгоритм A, используя выборку X_l длины l, извлеченную из признакового пространства в соответствии с вероятностным распределением на нём, выдаёт гипотезу $h \in H$, согласованную как минимум с $(1 - \frac{\varepsilon}{2})l$ примерами из X_l, и её строчное бинарное описание $s(h)$ имеет длину, не большую чем $n(g)^\alpha l^\beta$. Тогда, если*

$$l = O\left(\max\left(\frac{1}{\varepsilon}\log\frac{1}{\delta}, \left(\frac{(n(g)^\alpha}{\varepsilon}\right)^{1/(1-\beta)}\right)\right),$$

или, при $\beta = 0$,

$$l = O\left(\frac{n(g)^\alpha}{\varepsilon}\right),$$

то полиномиальная обучаемость [5] *имеет место.*

Оценка длины выборки, которая требуется для PAC обучаемости в сложностной версии *Occam's Razor* теоремы, основанной на длине описания $s(h) \leq n(g)^{\alpha} l^{\beta}$ выбираемого при обучении концепта h, может быть уточнена [20]:

$$l = \max\left(\frac{2}{\varepsilon}\ln\frac{2}{\delta}, \left(\frac{2\ln 2 \cdot n(g)^{\alpha}}{\varepsilon}\right)^{1/(1-\beta)}\right).$$

Константы α и β, фигурирующие в *Occam's Razor* теореме можно интерпретировать следующим образом. Бинарное описание выбранной гипотезы должно иметь длину, не превышающую $n(g)^{\alpha} l^{\beta}$, где α – степень расширения описания целевого концепта, а β – степень сжатия описания выборки.

Попытки уточнения *Occam's Razor* теоремы привели к следующей формуле для длины выборки, необходимой для (ε, δ)-обучаемости и определяемой сжатием описания выбираемого при обучении концепта h [20]:

$$l = \max\left(\frac{2}{\varepsilon}\ln\frac{1}{\delta}, \left(\frac{2\ln 2 \cdot p(n, s, \delta/2)}{\varepsilon}\right)^{1/(1-\beta)}\right),$$

где $p(n, s, \delta/2)$ – характеризующая сжатие описания концепта оценочная функция такая, что $KP(h) < p(n, s, \delta/2) l^{\beta}$; n – размерность признакового пространства, s – верхняя граница возможных длин описаний по допустимым классам концептов. Если можно указать оценку сверху M_h такую, что $p(n, s, \delta/2) l^{\beta} \leq M_h$ для всех допустимых значений параметров функции p, то требуемая длина выборки будет определяться как

$$l = \max\left(\frac{2}{\varepsilon}\ln\frac{2}{\delta}, \left(\frac{2\ln 2 \cdot M_h}{\varepsilon}\right)^{1/(1-\beta)}\right),$$

и при полном сжатии выборки ($\beta = 0$) как

$$l = \max\left(\frac{2}{\varepsilon}\ln\frac{2}{\delta}, \left(\frac{2\ln 2 \cdot M_h}{\varepsilon}\right)\right).$$

Оценка M_h может быть получена $pVCD$ методом [4, 15].

Версия *Occam's Razor* теоремы, основанной на вапниковской ёмкости $VCD(H)$ семейства концептов H, из которого извлекается концепт h, определяет следующую оценку выборки, требуемую для PAC обучаемости [20, 13, 16]:

$$\max\left(\frac{VCD(H) - 1}{32\varepsilon}, \frac{1}{\varepsilon}\ln\frac{1}{\delta}\right) < l(H, \delta, \varepsilon) \leq \frac{4}{\varepsilon}\left(VCD(h)\log\frac{12}{\varepsilon} + \log\frac{2}{\delta}\right).$$

Из приведенных оценок видно, что колмогоровская сложность $KP(h)$ выбранной гипотезы $h \in H$ и $VCD(H)$ при их использовании для оценивания результатов машинного обучения дают близкие результаты. Действительно, выбор семейства гипотез наименьшей ёмкости влечёт минимизацию колмогоровской сложности этого семейства, что согласуется с установленным в [4, 15] неравенством для колмогоровской сложности $K_l(H)$ произвольного конечного семейства гипотез H (семейства рекурсивных функций H):

$$VCD(H) < K_l(H) \leq VCD(H) \log l.$$

В случае конечного семейства гипотез H оценка длины выборки, обеспечивающей обучаемость для любого согласованного с выборкой концепта $h \in H$, имеет вид:

$$l(H, \delta, \varepsilon) \geq \frac{1}{\varepsilon} \ln \frac{|H|}{\delta}.$$

Это неравенство, как и многие другие фундаментальные результаты, связанные с обучаемостью, были получены В. Н. Вапником еще в начале 1970-х годов [1, 2].

Связь между сжатием обучающей выборки, обучаемостью и VCD была изучена в работе Флойда и Вармута [17] не основе следующих понятий. Для любого $Y \subset \mathbf{X}$ ($\mathbf{X}$ — признаковое пространство) и произвольного класса концептов C вводится обозначение $C|Y = \{c \cap Y : c \in C\}$ — ограничение концепта по области (множеству) Y.

Схема сжатия выборки размера не более k для класса концептов C описывается *функцией сжатия, функцией реконструкции* и их применением следующим образом. Используя конечную обучающую выборку, согласованную с классом концептов C, функция сжатия K отбирает из неё так называемое *множество сжатия* A, состоящее из не более k помеченных обучающих примеров. Функция реконструкции φ использует это множество сжатия для построения концепта-гипотезы $c_{\mathrm{A}} = \varphi(\mathrm{A})$ — результата обучения. При этом гипотеза c_{A}, вообще говоря, может не содержаться в классе C, но должна быть согласованной со всеми примерами исходной обучающей выборки.

Пример. Рассмотрим класс C_{L^0} однородных линейных концептов в $\mathbf{R}^n$ и согласованную выборку D длины $l > n$, состоящую из точек $\tilde{x} = x_1, ..., x_n$, удовлетворяющих уравнению $a_1 x_1 + ... + a_n x_n = 0$. Неизвестные коэффициенты $\tilde{a} = a_1, ..., a_n$ определяют один из концептов $c_{\tilde{a}} \in C_{L^0}$. Пусть множество сжатия A_L состоит из любых $k = n$ попарно различных примеров обучающей выборки. Тогда, используя эти k примеров, функция реконструкции φ, определяемая алгоритмом решения системы однородных линейных уравнений, однозначно восстанавливает $\varphi(\mathrm{A}_L) = c_{\tilde{a}}$. Заметим,

что $VCD(C_{L^0}) = n$. Если $l < n$, то функции реконструкции, обеспечивающей безошибочное нахождение неизвестного целевого концепта, для этого примера не существует, так как по $l < n$ точкам невозможно однозначное восстановление линейного концепта.

Для класса неоднородных линейных концептов C_L, соответствующих уравнениям $a_1x_1+...+a_nx_n = a_0$, параметр сжатия k должен быть не меньше $d = VCD(C_L) = n+1$.

□

Класс концептов называется *максимальным*, если добавление любого концепта к этому классу увеличивает его VCD. Класс концептов C, имеющий $VCD(C) = d$, называется *классом-максимумом*, если для каждого конечного подмножества $Y \subseteq C$, при $|Y| = m > d$, семейство $C|Y$ содержит $\Phi_d(|Y|) = \sum_{i=0}^{d} C^i_{|Y|}$ концептов.

Теорема 22. [17]. *Пусть класс концептов $C \subseteq 2^{\boldsymbol{X}}$ является классом-максимумом, $VCD(C) = d$, обучающая выборка X_l имеет длину $l \geq d$. Тогда для любого концепта $c \in C$ найдётся множество сжатия* A*, состоящее ровно из d примеров, и функция реконструкции такие, что $c_{\mathrm{A}} = c$.*

Теорема 23. *Пусть класс концептов $C \subseteq 2^{\boldsymbol{X}}$ является классом-максимумом, $VCD(C) = d$, и выборочное пространство может быть бесконечным. Тогда для класса концептов C при длине обучающей выборки l существует схема сжатия размера k, удовлетворяющего неравенству $d < k \leq d \log l$.*

Теорема 24. *Пусть $C \subseteq 2^{\boldsymbol{X}}$ класс концептов со схемой компрессии размером не более $d = VCD(C)$. Тогда для любых ε, δ таких, что $0 < \varepsilon, \delta < 1$, использование обучающего алгоритма, соответствующего этой схеме компрессии, обеспечит (ε, δ) обучаемость при длине выборки, удовлетворяющей неравенству*

$$l \geq \frac{1}{1-\beta}\left(\frac{1}{\varepsilon}\ln\frac{1}{\delta} + VCD(C) + \frac{VCD(C)}{\varepsilon}\ln\frac{1}{\beta\varepsilon}\right)$$

для любого $\beta : 0 < \beta < 1$.

Нужно подчеркнуть, что сжатие в последних теоремах характеризуется относительно длины выборки, а не длиной бинарной строки. Но, тем не менее, в указанных условиях возможно сжатие информации о семействе концептов ёмкости d до бинарной строки, длина которой не будет превышать $O(d \log l)$[4]. Теоретически колмогоровская сложность произвольного класса вычислимых функций может быть равной его ёмкости d, в силу чего, с учетом перечислимости колмогоровской сложности сверху, возможно сжатие информации о таком классе до строки длины d.

В работе [24] схема компрессии размера k уточняется следующим образом. Функция сжатия K ставит в соответствие каждой обучающей выборке X_l длины l единственную её подвыборку $V = V(X_l)$ длины k, называемую *ядром сжатия*. Функция K в схеме k-сжатия полагается зафиксированной. Функция реконструкции $\varphi = \varphi(V, \tilde{x})$ тоже зафиксирована и ставит в соответствие паре ядро-точка значение 1 или 0. Таким образом определяется решающее правило и некоторый концепт $c_V = c_\varphi(K, \varphi, \tilde{x})$. Этот концепт c_V, вообще говоря, может не принадлежать классу концептов C. Но для любого целевого концепта семейства C и для любой заданной выборки длины l функция реконструкции согласована со всеми точками этой выборки.

Ядерным размером называется минимальная мощность ядра сжатия по всем возможным схемам сжатия (варьируются функции сжатия, реконструкции и выборки длины l).

Если зафиксировать любую схему компрессии с ядерным размером k и использовать определяемую ею функцию реконструкции φ^*, то в соответствии с данными выше определениями, применение этой функции к произвольным точкам признакового пространства, вообще говоря, может давать ошибки. Нужно убедиться, что использование функции φ^* обеспечивает обучаемость.

Характеризация сжатия ядерным размером позволяет считать произвольным признаковое пространство, поскольку речь идёт о числе примеров в ядре, а не о битовой строке, кодирующей сложность.

Будем полагать, что концепты класса C и функция реконструкции измеримы по Борелю. Из этого следует измеримость множеств, определённых ниже при доказательстве теоремы, и правомочность использования теоремы Фубини.

Теорема 25. [24]. *Для любой схемы компрессии с ядерным размером k при длине выборки $l > k$, ошибка Err функции реконструкции как решающего правила, определяющего принадлежность произвольной точки x целевому концепту G, может быть оценена неравенством*

$$P(Err > \varepsilon) < C_l^k (1 - \varepsilon)^{l-k}.$$

Доказательство. Пусть $\mathbf{X}^l$ — множество любых выборок длины l; $(\tilde{x}_1, ..., \tilde{x}_l) = X_l$ — произвольная выборка длины l из $\mathbf{X}^l$; P^l — вероятностная мера на множестве $\mathbf{X}^l$ выборок длины l (по этой мере оценивается вероятность события $Err > \varepsilon$). Будем обозначать A^* — ядро сжатия произвольной схемы компрессии с ядерным размером k; $\varphi^*(\mathrm{A}^*, \tilde{x})$ — результат применения функции реконструкции φ^*, определяющий, возможно с ошибкой, принадлежность точки $\tilde{x}$ концепту c; $c(\tilde{x})$ — истинное значение

этой принадлежности. Обозначим $E = \{X_l \in \mathbf{X}_l : \mathbf{Pr}(\tilde{x} \in X_l \wedge \varphi^*(\mathrm{A}^*, \tilde{x}) \neq c(\tilde{x})) > \varepsilon\}$ — множество всевозможных выборок длины l, точки которых классифицируются функцией φ^* с вероятностью ошибки, превышающей ε. Эквивалентное определение — $E = \{X_l \in \mathbf{X}_l : \mathbf{Pr}(\tilde{x} \in X_l \wedge \varphi^*(\mathrm{A}^*, \tilde{x}) = c(\tilde{x})) < 1 - \varepsilon\}$.

Пусть T — множество всех C_l^k подпоследовательностей номеров любых k точек выборки; $\tilde{t} = (t_1, ..., t_k) \in T$. Набор $\tilde{t}$ определяет подпоследовательность выборки $\tilde{x}_{t_1}, ..., \tilde{x}_{t_k}$. Введём следующие обозначения.

$A_{\tilde{t}}$ — множество всех выборок длины l, для которых по каждой выборке $\tilde{x}_1, ..., \tilde{x}_l$ функция сжатия K выделяет ядро, состоящее из точек $\tilde{x}_{t_1}, ..., \tilde{x}_{t_k}$ этой выборки. Очевидно, $\bigcup_{\tilde{t} \in T} A_{\tilde{t}} = X_l$.

$E_{\tilde{t}} \subseteq A_{\tilde{t}}$ — такое подмножество выборок, на котором применение функции реконструкции φ^* даёт правильное решение с вероятностью, меньшей $1 - \varepsilon$. То есть, $E_{\tilde{t}}$ — это все выборки, для которых функция сжатия K выделяет ядро, состоящее из точек этих выборок с номерами $t_1, ..., t_k$, а функция реконструкции даёт правильное решение с вероятностью, меньшей $1 - \varepsilon$.

По определению соответствующих подмножеств, $E_{\tilde{t}} = E \cap A_{\tilde{t}}$, откуда с учётом равенства $\bigcup_{\tilde{t} \in T} A_{\tilde{t}} = X_l$ следует $E = \bigcup_{\tilde{t} \in T} E_{\tilde{t}}$.

Обозначим далее:

$U_{\tilde{t}}$ — множество всех выборок длины l, для которых вероятность правильной классификации при помощи функции реконструкции φ^* с выделяемым функцией компрессии K ядром $\{x_{t_1}, ..., x_{t_k}\}$ ограничена величиной $1 - \varepsilon$. Тогда $E_{\tilde{t}} = U_{\tilde{t}} \cap A_{\tilde{t}}$.

$B_{\tilde{t}}$ — множество всех выборок длины l таких, что входящие в них точки с номерами вне множества $\{t_1, ..., t_k\}$ правильно классифицируются функцией реконструкции.

Если выборка принадлежит множеству $A_{\tilde{t}}$, то функция сжатия K выделяет из выборок этого множества ядро, состоящее из точек $x_{t_1}, ..., x_{t_k}$этой выборки. По определению схемы компрессии, все остальные точки этой выборки с номерами вне множества $\{t_1, ..., t_k\}$ должны классифицироваться правильно. Поэтому $A_{\tilde{t}} \subset B_{\tilde{t}}$. Вместе с равенством $E_{\tilde{t}} = U_{\tilde{t}} \cap A_{\tilde{t}}$ это даёт

$$P^l(E_{\tilde{t}}) = P^l(A_{\tilde{t}} \cap U_{\tilde{t}}) \leq P^l(B_{\tilde{t}} \cap U_{\tilde{t}}).$$

Пусть $\pi_{\tilde{t}}$ такая перестановка координат точек выборки $\tilde{x}_1, ..., \tilde{x}_i, ..., \tilde{x}_l$, что $t_i \mapsto i, \quad i = 1, ..., k$; $\pi_{\tilde{t}} : X_l \to X_l$. Тогда $\pi_{\tilde{t}}(U_{\tilde{t}})$ – множество всех выборок длины l, для которых вероятность правильной классификации входящих в них точек при помощи функции реконструкции φ^* с ядром $\{x_1, ..., x_k\}$ ограничена величиной $1 - \varepsilon$. Перестановка вводится для удобства дальнейших рассуждений: без потери общности

применяется замена (переименование) $\{x_{t_1}, ..., x_{t_k}\} \mapsto \{x_1, ..., x_k\}$.

$$P^l(E_{\tilde{t}}) \leq P^l(B_{\tilde{t}} \cap U_{\tilde{t}}) = P^l(\pi_{\tilde{t}}(B_{\tilde{t}}) \cap \pi_{\tilde{t}}(U_{\tilde{t}})).$$

$$P^l(\pi_{\tilde{t}}(B_{\tilde{t}}) \cap \pi_{\tilde{t}}(U_{\tilde{t}})) = \int\limits_{\pi_{\tilde{t}}(U_{\tilde{t}})} I(\pi_{\tilde{t}}(B_{\tilde{t}}))dP^l,$$

где $I(\pi_{\tilde{t}}(B_{\tilde{t}}))$ — характеристическая функция множества $\pi_{\tilde{t}}(B_{\tilde{t}})$, которая выделяет из всех выборок длины l такие выборки, что входящие в них точки с номерами вне множества $\{t_1, ..., t_k\}$ правильно классифицируются функцией реконструкции, т. е. правильно классифицируются $l - k$ точек.

Интегрирование производится по множеству $\pi_{\tilde{t}}(U_{\tilde{t}})$ выборок таких, что вероятность правильной классификации входящих в них точек при помощи функции реконструкции φ^* с ядром $\{x_1, .., x_k\}$ ограничена величиной $1 - \varepsilon$.

Ядра компрессии извлекаются из выборок, поэтому существует некоторое множество $V_{\tilde{t}}$ ядер размера k такое, что $\pi_{\tilde{t}}(U_{\tilde{t}}) = V_{\tilde{t}} \times X^{l-k}$.

По теореме Фубини

$$\int\limits_{\pi_{\tilde{t}}(U_{\tilde{t}})} I(\pi_{\tilde{t}}(B_{\tilde{t}}))dP^l = \int\limits_{V_{\tilde{t}}} dP^k \int\limits_{X^{l-k}} I(\pi_{\tilde{t}}(B_{\tilde{t}}))dP^{l-k}.$$

Обозначим $W_{x_1,...,x_k}$ — множество точек выборки X_l, правильно классифицируемое функцией реконструкции φ^* с ядром $x_1, ..., x_k$. Тогда

$$(x_1, ..., x_k) \times X^{l-k} \cap \pi_{\tilde{t}}(B_{\tilde{t}}) = (x_1, ..., x_k) \times W^{l-k}_{x_1,...,x_k}.$$

$$P^l(E_{\tilde{t}}) < \int\limits_{X^{l-k}} I(\pi_{\tilde{t}}(B_{\tilde{t}}))dP^{l-k} = \int\limits_{W^{l-k}_{x_1,...,x_k}} dP^{l-k} \qquad < (1-\varepsilon)^{l-k}.$$

$$P^l(E_{\tilde{t}}) \qquad < (1-\varepsilon)^{l-k}.$$

Число различных подпоследовательностей длины k последовательности $\tilde{x}_1, ..., \tilde{x}_l$ равно C^k_l. Поэтому

$$P^l(E) = P^l\left(\bigcup_{\tilde{t}\in T} E_{\tilde{t}}\right) < \sum_{\tilde{t}\in T} P^l(E_{\tilde{t}}) = |T|(1-\varepsilon)^{l-k} = C^k_l(1-\varepsilon)^{l-k}.$$

□

Теорема 26. [24]. *Для любой схемы компрессии, имеющей ядерный размер k, (ε, δ)-обучаемость имеет место при длине выборки l, определяемой неравенством*

$$l \geq \max\left\{\frac{2}{\varepsilon}\ln\frac{1}{\delta}, \quad \frac{2k}{\varepsilon}\ln\frac{4k}{\varepsilon} + 2k\right\}.$$

Доказательство. Преобразуем неравенство $C_l^k(1-\varepsilon)^{l-k} < \delta$ (см. предыдущую теорему) в эквивалентное неравенство

$$l \geq \frac{\ln\frac{1}{\delta} + \ln C_l^k}{-\ln(1-\varepsilon)} + k,$$

которое выполняется при условии $l \geq \frac{1}{\varepsilon}(ln\frac{1}{\delta} + k\ln l) + k = \frac{1}{\varepsilon}ln\frac{1}{\delta} + k(\frac{1}{\varepsilon}\ln l + 1)$, поскольку $l^k > C_l^k$, и для малых ε выполняется: $-\ln(1-\varepsilon) > \varepsilon$. В оценку входят два слагаемых. Поэтому неравенство будет иметь место, если одновременно каждое слагаемое будет не больше величины $l/2$, что приводит к системе из двух неравенств:

$$\begin{cases} \frac{l}{2} \geq \frac{1}{\varepsilon}\ln\frac{1}{\delta}, \\ \frac{l}{2} \geq k\left(\frac{1}{\varepsilon}\ln l + 1\right). \end{cases} \tag{2}$$

Второе из этих двух неравенств путём подстановки в правую часть оценки для l можно преобразовать следующим образом:

$$l \geq 2k\left(\frac{1}{\varepsilon}\ln(2k(\frac{1}{\varepsilon}\ln l + 1)) + 1\right); \quad l \geq 2k\left(\frac{1}{\varepsilon}\ln\frac{4k}{\varepsilon} + 1\right); \quad l \geq 2k\frac{1}{\varepsilon}\ln\frac{4k}{\varepsilon} + 2k.$$

Из полученной системы неравенств

$$\begin{cases} l \geq \frac{2}{\varepsilon}\ln\frac{1}{\delta}, \\ l \geq \frac{2k}{\varepsilon}\ln\frac{4k}{\varepsilon} + 2k. \end{cases} \tag{3}$$

получается оценка

$$l \geq \max\left\{\frac{2}{\varepsilon}\ln\frac{1}{\delta}, \quad \frac{2k}{\varepsilon}\ln\frac{4k}{\varepsilon} + 2k\right\}.$$

□

Сравнивая эту оценку длины выборки, требуемой для обучаемости с параметром размера сжатия k, с аналогичной оценкой обучаемости Блумера и Литтлстоуна [14], полученной на основе размерности Вапника-Червоненкиса $d = VCD(\mathfrak{F})$ класса функций $\mathfrak{F}$, используемого для обучения, –

$$l \geq \max\left\{\frac{4}{\varepsilon}\ln\frac{2}{\delta}, \quad \frac{8d}{\varepsilon}\ln\frac{8d}{\varepsilon}\right\},$$

можно заметить, что эти оценки достаточно близки.

Литтлстоном и Вармутом [24] также получены аналогичные результаты для схемы сжатия размера k с дополнительной информацией, обозначаемой Q — некоторым множеством, добавляемым отображением сжатия к ядру выборки. Это отображение ставит в соответствие любой выборке пару: множество Q и ядро размера k. Так что сжатие оценивается числом элементов в Q и размером ядра k.

Теорема 27. [24]. *Для любой схемы компрессии с ядерным размером k и дополнительной информацией Q при длине выборки $l > k$, ошибка Err функции реконструкции как решающего правила, определяющего принадлежность произвольной точки $\tilde{x}$ целевому концепту G, может быть оценена неравенством*

$$P(Err > \varepsilon) < |Q|C_l^k(1-\varepsilon)^{l-k}.$$

Если схему компрессии ослабить так, что классификация выборки, по которой найдено ядро, при помощи функции реконструкции допускает ошибку в $s < l - k$ её точках, то будет иметь место следующий результат:

Теорема 28. [24]. *Для любой схемы компрессии с ядерным размером k, допускающей не более s ошибок при длине выборки $l > k$, ошибка Err функции реконструкции как решающего правила, определяющего принадлежность произвольной точки целевому концепту G, может быть оценена неравенством*

$$P(Err > \varepsilon) < C_{k+s}^k C_l^k(1-\varepsilon)^{l-k}.$$

5. Использование универсального распределения для аппроксимации неизвестного распределения

Для понимания следующей теоремы нужно учесть, что полиномиальная (ε, δ) обучаемость является частным случаем (ε, δ) обучаемости и влечёт последнюю [5].

Теорема 29. [21, 22] *Полиномиальная обучаемость над универсальным распределением $\boldsymbol{m}$ имеет место тогда и только тогда, когда имеет место обучаемость над любым простым распределением P при условии, что выборка извлекается в соответствии с распределением $\boldsymbol{m}$.*

Доказательство. Пусть P — любое простое распределение: найдется константа $c_P > 0$ такая, что $c_P \mathbf{m}(x) \geq P(x)$. Предположим, что имеет место обучаемость над распределением $\mathbf{m}$ с ошибкой ε / c_P, и имеется соответствующий определению полиномиальной обучаемости алгоритм A полиномиальной сложности. Зафиксируем его. Пусть Err — множество объектов, на которых обученный концепт даёт ошибку. Тогда с вероятностью не меньшей $1 - \delta$ $\sum_{x \in Err} \mathbf{m}(x) \leq \varepsilon / c_P$ и $\sum_{x \in Err} P(x) \leq c_P \sum_{x \in Err} \mathbf{m}(x) \leq \varepsilon$.

Поскольку алгоритм A извлекает обучающую выборку *всегда в соответствии с распределением* $\mathbf{m}$, то его точное выполнение в условиях распределения P должно давать в качестве результата тот же самый концепт, определяющий множество Err. Следовательно, из полиномиальной обучаемости над универсальным распределением $\mathbf{m}$ следует полиномиальная обучаемость над любым простым распределением P.

Пусть теперь имеет место полиномиальная обучаемость над любым простым распределением P, вероятность ошибки не больше ε, и обученный алгоритм даёт ошибку только на множестве Err. Но по условию теоремы, извлечение выборки происходит в соответствии с распределением $\mathbf{m}$, поэтому $\sum_{x\in Err}\mathbf{m}(x)\leq\varepsilon$, что доказывает полиномиальную обучаемость над $\mathbf{m}$. □

Замечания, касающиеся теоремы. Параметр обучаемости ε/c_P требует знания константы $c_P = KP(P)$ – префиксной сложности неизвестного простого распределения P (см. ниже теорему об этом равенстве). При решении задач обучения приходится иметь дело с некоторыми подмножествами признакового пространства, и для таких подмножеств D использовать условные распределения $\mathbf{m}(\cdot|D)$. В связи с этим Ли и Витаньи получили более тонкий критерий обучаемости, который будет приведен ниже без доказательства.

Определение 15. [22]. Вероятностное распределение $P : S \to \mathbf{R}$, где $s = \mathbf{N}\cup u$, $\sum_{x\in\mathbf{N}}P(x)\leq 1,\quad \sum_{x\in S}P(x)=1,\ u$ — некоторый неопределенный элемент, называется *перечислимым*, если множество точек $\{(x,y) : x\in\mathbf{N},\ y\in\mathbf{Q},\ P(x)>y\}$ рекурсивно перечислимо.

Определение 16. [22] Распределение P называется *простым*, если существует перечислимое распределение Q такое, что $\exists c : \forall x\,(cQ(x)\geq P(x))$, где $c\leq 2^{KP(Q)+O(1)}$ — константа. Говорят, что P доминируется перечислимым распределением Q.

Теорема 30. [21] *Если распределение $P(x|y)$ перечислимо, то для всех допустимых x, y имеет место неравенство*

$$2^{KP(P)}\boldsymbol{m}(x|y)\geq P(x|y).$$

Теорема 31. [22] *Пусть H — класс концептов, $D\subseteq\boldsymbol{N}$ – выборочное пространство, $\mu=\min\{l(s(g)) : g\in G\}$ и c — константа. Класс H полиномиально обучаем над универсальным распределением $\boldsymbol{m}$ тогда и только тогда, когда он полиномиально обучаем над любым простым распределением P таким, что существует перечислимое распределение Q, доминирующее P, которое удовлетворяет условию $KP(Q)\leq c\log\mu+O(1)$, и кроме этого, выполняется одно из следующих условий:*

(i) выборка извлекается согласно условному распределению $\boldsymbol{m}(\cdot|D)$;

(ii) $KP(D)\leq c\log\mu+O(1)$ и выборка формируется так, что полиномиальное число примеров извлекаются в соответствии с безусловным распределением $\boldsymbol{m}(\cdot)$, причем степень полинома не превышает константы c.

6. Байесовский подход к обучению и MDL

Правило Байеса определяет наиболее вероятную гипотезу h при заданном обучающем множестве D согласно соотношению

$$\mathbf{Pr}\{h|D\} = \frac{\mathbf{Pr}\{D|h\}\mathbf{Pr}\{h\}}{\mathbf{Pr}\{D\}},$$

которое может быть представлено в эквивалентной форме:

$$-\log \mathbf{Pr}\{h|D\} = -\log \mathbf{Pr}\{D|h\} - \log \mathbf{Pr}\{h\} + \log \mathbf{Pr}\{D\}.$$

Наиболее вероятная гипотеза h при заданном обучающем множестве D должна минимизировать $-\log \mathbf{Pr}\{h|D\}$. Поскольку $\mathbf{Pr}\{D\}$ не меняется при выборе гипотез, байесовское правило выбора гипотезы из семейства может быть представлено в виде:

$$h^* = \arg\min_{h\in H}(-\log \mathbf{Pr}\{D|h\} - \log \mathbf{Pr}\{h\}).$$

Использование универсального распределения приводит к соотношению

$$\hat{h}^* = \arg\min_{h\in H}(-\log \mathbf{m}\{D|h\} - \log \mathbf{m}\{h\})$$

и далее, с учетом соотношения $-\log \mathbf{m}(x) = KP(x) + O(1)$, к правилу

$$\hat{\hat{h}}^* = \arg\min_{h\in H}(KP(D|h) + KP(h))).$$

Последнее соотношение является выражением принципа MDL (*Minimum Description Length*), который является одной из формализаций «бритвы Оккама»: *наилучшая гипотеза для данного набора данных та, которая минимизирует сумму длины описания кода гипотезы* (*также называемой моделью*) *и длины описания множества данных относительно этой гипотезы* [25].

Базирующаяся на строгом математическом обосновании, применении колмогоровской сложности и универсальной меры $\mathbf{m}$, уточнённая версия MDL называется *идеальным* MDL. Применение и обоснование идеального MDL проиллюстрировано выше на байесовской схеме выбора гипотезы [25]. Имеет место фундаментальное неравенство [26]:

$$KP(D|h) + KP(h) - \alpha(P,h) \le -\log \mathbf{Pr}\{D|h\} - \log \mathbf{Pr}(h) \le KP(D|h) + KP(h),$$

где $\alpha(P,h) = KP(P(D|h)) + KP(h)$. При малом значении $\alpha(P,h)$ левая и правая оценки становятся приблизительно равными друг другу, и тогда $KP(D|h) + KP(h) \approx -\log \mathbf{Pr}\{D|h\} - \log \mathbf{Pr}(h)$. Это рассуждение лежит в основе доказательства следующего утверждения.

Теорема 32. [26], *с.359. Байесовское правило и идеальный MDL при извлечении решения из допустимого класса гипотез H выбирают одну и ту же гипотезу: $h^* = \hat{\hat{h}}^*$ при условии, что величина $\alpha(P,h)$ является достаточно малой.*

Таким образом, минимизация суммы $KP(D|h) + KP(h)$ обеспечивает выбор гипотезы $\hat{\hat{h}}^*$ в соответствии с правилом Байеса, которое, как известно, является оптимальным: обеспечивает минимум среднего риска.

Казалось бы, если правило Байеса является оптимальным, то его и нужно применять, не изобретая новых способов выбора решений. Но непосредственное использование байесовского правила требует знания априорных распределений вероятностей, а они, как правило, неизвестны: в задачах машинного обучения в качестве начальной информации представляется обучающая выборка, по которой приходится аппроксимировать неизвестные распределения. Идеальный MDL позволяет *обойтись без информации об истинных априорных распределениях*. Но возникают другие трудности: и префиксная сложность $KP(x)$, и универсальное распределение $\mathbf{m}(x) = 2^{-KP(x)-O(1)}$ не являются вычислимыми. Поэтому нужно рассчитывать на *использование вычислимых оценок колмогоровской сложности*.

Рассмотрим условную сложность $KP(D|h)$, входящую в минимизируемую сумму $KP(D|h) + KP(h)$. По определению префиксной колмогоровской сложности, $KP(D|h) = \min\{l(p) : U(p,h) = D\}$ для некоторого оптимального декомпрессора U. Здесь декомпрессор U – префиксная машина Тьюринга, которая принимает вход в виде пары строк (p – сжатого описания и h – применяемой гипотезы) и в результате выдает описание обучающей информации в виде строки D. Если $KP(D|h) = 0$, то $U(\lambda, h) = D$, где λ – пустое слово. В таком случае будем говорить, что гипотеза h полностью описывает данные D. Действительно, декомпрессор U точно восстанавливает данные D, используя при этом в качестве входа только описание гипотезы h. В противном случае будем использовать запись $KP(D|h) = KP(D \setminus \hat{D})$, где $\hat{D}$ – часть обучающих данных, которые правильно описываются гипотезой h. Обозначим $D \setminus \hat{D} = \bar{D}(h)$ – выделенную подпоследовательность последовательности-строки D и будем говорить, что $\bar{D}(h)$ – неописанный гипотезой h остаток данных. Тогда принцип MDL принимает эквивалентный вид

$$\hat{\hat{h}}^* = \arg\min_{h \in H}(KP\{\bar{D}(h)\} + KP\{h\})$$

и формулируется так: *наилучшая гипотеза для данного набора данных та, которая минимизирует сумму длины описания кода гипотезы* (*также называемой моделью*) *и длины описания множества данных, не описываемых* (*не объясняемых*) *этой гипотезой.*

Для согласованных с данными D гипотез это правило будет выглядеть так:

$$\hat{\hat{h}}^* = \arg \min_{h \in H_c(D)} KP\{h\},$$

где $H_c(D)$ – класс гипотез, согласованных с данными D.

Обучающая выборка D является совокупностью примеров $(\tilde{x}_j, \alpha_j)_{j=1}^l$. Будем использовать для отдельного примера выборки обозначение $ex_j = (\tilde{x}_j, \alpha_j)$. Тогда $D = \bigcup_{j=1}^l ex_j$. Будем считать независимыми вероятности $\mathbf{Pr}\{ex_j|h\}$ и полагать, что

$$Pr\{D|h\} = \mathbf{Pr}\{ex_1|h\} \ldots \mathbf{Pr}\{ex_j|h\} \ldots \mathbf{Pr}\{ex_l|h\}$$

Тогда по правилу Байеса наиболее вероятная гипотеза h *при заданном обучающем множестве* D определяется соотношениями

$$\mathbf{Pr}\{h|D\} = \frac{\mathbf{Pr}\{h\} \prod_{j=1}^l \mathbf{Pr}\{ex_j|h\}}{\mathbf{Pr}\{D\}},$$

$$-\log \mathbf{Pr}\{h|D\} = -\log \mathbf{Pr}\{h\} - \sum_{j=1}^{l} \log \mathbf{Pr}\{ex_j|h\} + \log \mathbf{Pr}\{D\}.$$

Байесовское правило выбора гипотезы может быть представлено в виде:

$$h^* = \arg \min_{h \in H} (-\log \mathbf{Pr}\{h\} - \sum_{j=1}^{l} \log \mathbf{Pr}\{ex_j|h\}).$$

Переходя к универсальному распределению и далее – к префиксной сложности, получаем

$$\hat{h}^* = \arg \min_{h \in H} (-\log \mathbf{m}\{h\} - \sum_{j=1}^{l} \log \mathbf{m}\{ex_j|h\}),$$

$$\hat{\hat{h}}^* = \arg \min_{h \in H} (KP(h) + \sum_{j=1}^{l} KP(ex_j|h)).$$

Напомним, что правило Байеса предполагает заданным обучающее множество D, которое можно представить как таблицу, содержащую l строк и $n+1$ столбцов, где n — размерность признакового пространства.

Префиксная сложность гипотезы $KP(h)$ – это *кратчайшее описание гипотезы* h: кратчайшее двоичное слово p, позволяющее получить $\tilde{h} = U(p)$ при помощи некоторого декомпрессора U. По этому слову p при помощи алгоритма-декомпрессора U можно корректно восстановить гипотезу h по её описанию $\tilde{h}$ с учетом того, что $Dom(h) \subseteq \mathbf{X}$, и h является функцией из множества $\mathbf{X}$ в $\{0, 1\}$.

Оценим условную префиксную сложность $KP(ex_j|h)$ одного примера обучающей выборки. $KP(ex_j|h)$ — это наименьшая длина такой бинарной строки p, что найдётся декомпрессор U, который по двоичному слову p и представленной корректным описанием гипотезе h определяет строку, «описывающую» пример ex_j в следующем смысле. Слово p определяет соответствие между признаковым описанием примера $\tilde{x}_j$ и значением целевой гипотезы α_j в обучающей таблице D при условии использования гипотезы h. Если $h(\tilde{x}_j) = \alpha_j$, то гипотеза h полностью описывает пример. В этом случае никакое слово p не требуется, и оно полагается пустым, т. е. $KP(ex_j|h) = 0$. В противном случае гипотеза h не описывает пример, но его описание может быть получено из заданной для обучения таблицы D. Для этого требуется указать декомпрессору в слове p номер нужной строки в таблице. Поскольку в таблице D имеется ровно l строк, для указания на нужную строку потребуется $]\log l[$ бит. Извлечение целевого значения α_j потребует задания размерности n, чтобы отделить это значение от описания признаков примера. Для этого потребуется $]\log n[$ бит. Применяя стандартный самоограничивающий код, окончательно получаем: $KP(ex_j|h) = 0$, если гипотеза правильно описывает пример ex_j, и $KP(ex_j|h) \leq 2]\log\log n[+]\log n[+]\log l[$, если пример ex_j противоречит гипотезе h. Объединяя эти случаи, получаем

$$KP(D|h) = \sum_{j=1}^{l} KP(ex_j|h) \quad \leq \mathbf{k}\,(2]\log\log n[+]\log n[+]\log l[)\,,$$

где $\mathbf{k}$ – число примеров из l, неправильно классифицируемых гипотезой h, или

$$KP(D|h) \leq \nu_{emp} \cdot l \cdot (2]\log\log n[+]\log n[+]\log l[)\,,$$

где ν_{emp} – эмпирическая частота ошибок гипотезы h на выборке D длины l. Учитывая, что согласно теореме Соломонова-Колмогорова сложность определяется лишь с точностью до аддитивной константы, условную префиксную сложность *данной обучающей выборки D при данной гипотезе h* можно приближенно оценить следующим образом:

$$KP(D|h) \approx \nu_{emp} l(2\log\log n + \log(nl)).$$

Правило Байеса и оптимальная остановка при обучении

Обучение отличается от настройки на обучающую выборку или её прямой аппроксимации тем, что предполагает организацию последовательного процесса усложнения решающего правила (гипотезы) с целью достижения его способности к эмпирическому обобщению. По отношению к самой выборке, способность к обобщению проявляется в том, что часть её примеров, не использованных на некотором

этапе обучения, правильно классифицируется сформированным на этом этапе решающим правилом. В этом смысле показательна обучающая процедура линейной коррекции Розенблатта-Новикова, в которой вектор коэффициентов решающего правила — линейного отделителя — корректируется только при ошибочной классификации очередного обучающего примера. Коррекция происходит путём использования этого примера — добавления его с регулирующим скорость сходимости коэффициентом к вектору линейного отделителя.

Процесс обучения можно представить как последовательный подбор решающего правила, при котором его сложность постепенно увеличивается, а обобщающая способность оценивается на каждом шаге t. Обозначая решающее правило, полученное на шаге t, как h_t, получаем последовательность $h_0, h_1, ..., h_t, ..., h_s$, где s — номер шага остановки. При этом сложность пошагово синтезируемого решающего правила не убывает:

$$KP(h_0) \leq KP(h_1) \leq ... \leq KP(h_t) \leq ... \leq KP(h_s).$$

По мере обучения все большее число примеров классифицируется правильно, поэтому условная сложность $KP(D|h_t)$ не возрастает:

$$KP(D|h_0) \geq KP(D|h_1) \geq ... \geq KP(D|h_t) \geq ... \geq KP(D|h_s).$$

В соответствии с байесовским подходом, необходимо рассматривать последовательность суммарных сложностей $KP(D|h_t) + KP(h_t)$; и следует остановиться на том шаге t_{opt}, когда указанная суммарная сложность в процессе обучения перестанет убывать. Учитывая, что

$$KP(D|h_t) - KP(D|h_{t-1}) \leq 0, \text{ а } KP(h_t) - KP(h_{t-1}) \geq 0,$$

условие остановки можно определить следующим образом:

$$t_{opt} = t : KP(h_t) - KP(h_{t-1}) - (KP(D|h_{t-1}) - KP(D|h_t)) \geq 0.$$

Проиллюстрируем этот подход на примере последовательного обучения для случая, когда решающее правило отыскивается в классе бинарных решающих деревьев (БРД). Процесс коррекции представляет собой увеличение числа внутренних вершин бинарного дерева на единицу, что влечёт увеличение числа решающих вершин-листьев μ также на единицу: $\mu_t = \mu_{t-1}+1$. Используя $pVCD$ метод [4, 6, 7, 15], можно получить следующую оценку сложности БРД с μ листьями:

$$KS(h_\mu) < (\mu - 1)\,(]\log(n+1)[+]\log\mu[+1)\,.$$

Программирование слова p для декомпрессии любого БРД с μ листьями с целью получения оценки сложности $KS(h_\mu)$ основано на представлении каждой из $\mu-1$ вершин ветвления словом-атомом, состоящим из двух частей (конкатенации 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 признаков $1, 2, \ldots, n$. Окончание атома может иметь μ значений: 0 и 1 резервируются 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 $\mu - 2$ внутренних вершин. Всего получается μ значений для окончания атома.

Использование стандартного самоограничивающего кода позволяет получить оценку

$$KP(h_\mu) < 2\,(\lceil \log\log n \rceil + \lceil \log\log \mu \rceil) + (\mu - 1)(\lceil \log(n+1) \rceil + \lceil \log \mu \rceil)\,,$$

$$KP(h_\mu) \approx 2\,(\log\log n + \log\log \mu) + (\mu - 1)(\log(n+1) + \log \mu)\,.$$

Усложнение БРД при добавлении ровно одной условной вершины приводит к увеличению сложности $KP(h_\mu)$ на длину одного атома, приблизительно равную

$$\log(n+1) + \log \mu.$$

Если при этом число ошибочно классифицируемых примеров выборки уменьшится на единицу, то сложность $KP(h_\mu)$ уменьшится на величину $\log l$. Оптимальная остановка ветвления (синтеза БРД) определяется неравенством

$$\log(n+1) + \log \mu > \log l,$$

позволяющим определить оптимальное число листьев синтезируемого дерева. При больших n для оценки наибольшего числа листьев μ, после достижения которого должна следовать остановка синтеза БРД, можно применять неравенство

$$\log(n\mu) > \log l.$$

Тогда условие остановки синтеза определяется соотношением $\mu > l/n$.

Так, если в обучающей выборке содержится $l = 300$ примеров, а число признаков $n = 20$, то увеличивать сложность БРД ради правильной классификации ещё только

одного примера не следует при $\mu > 15$. Но нужно учесть, что при увеличении эмпирической точности классификации (на одном шаге усложнения БРД) на два и более примера, это ограничение снимается.

Заключение

Применение математического аппарата теории колмогоровской алгоритмической сложности и случайности в машинном обучении позволило получить следующие важные результаты.

1. Удалось строго обосновать подход к синтезу решающих правил (гипотез), основанный на принципе их кратчайшего представления (описания).
2. Реализовать на основе теории универсального распределения байесовский подход к синтезу решающих правил, не требующий традиционной оценки вероятностных распределений.
3. Обосновать критерий остановки процедуры синтеза решающего правила на основе оценивания минимума суммарной сложности (данных и самого правила).
4. Построить ряд моделей сжатия информации и получить для них оценки обучаемости, в частности, в PAC модели [5] и в (ε, δ) модели обучения.
5. Обнаружить связь между размерностью Вапника-Червоненкиса $VCD(\mathbf{Im}(A))$ подкласса гипотез $\mathbf{Im}(A)$, из которого извлекается решение алгоритмом обучения (алгоритмическим отображением) A, и минимальным возможным сжатием описания любой гипотезы из этого класса: $VCD(\mathbf{Im}(A))$ является оценкой сложности выбранной гипотезы снизу.

Направление дальнейших исследований связано с повышением точности выводов на основе префиксной колмогоровской сложности за счет улучшения качества её оценок.

Список литературы

1. Вапник В. Н. Восстановление зависимостей по эмпирическим данным / В. Н. Вапник. — М.: Наука, 1979. — 448 с.
2. Вапник В. Н. Теория распознавания образов / В. Н. Вапник, А Я Червоненкис. — М.: Наука, 1974. — 416 с.
3. Вьюгин В. В. Колмогоровская сложность и алгоритмическая случайность / В. В. Вьюгин. — М.: МФТИ, 2012. — 131 с.
4. Донской В. И. Колмогоровская сложность классов общерекурсивных функций с ограниченной ёмкостью / В. И. Донской // Таврический вестник математики и информатики, 2005. — № 1. — С. 25–34.

5. Донской В. И. Машинное обучение и обучаемость: сравнительный обзор / В. И. Донской // Intellectual Archive. — 2012. — № 933 — 19 с.
http://www.sciteclibrary.ru/texsts/rus/stat/st4820.pdf
6. Донской В. И. Оценки ёмкости основных классов алгоритмов эмпирического обобщения, полученные pVCD методом / В. И. Донской // Ученые записки ТНУ им. В. И. Вернадского. Серия „Физико-математические науки", 2010. — Т. 23 (62). — № 2. — С. 56–65.
7. 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.
8. Донской В. И. Эмпирическое обобщение и распознавание: классы задач, классы математических моделей и применимость теорий. Часть I; Часть II / В. И. Донской // Таврический вестник информатики и математики, 2011. — № 1. — С. 15–26; — № 2. — С. 31–42.
9. Звонкин А. К. 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.
10. Колмогоров А. Н. Теория информации и теория алгоритмов // А. Н. Колмогоров. — М.: Наука, 1987. — 304 с.
11. Мучник А. А. Гиперпростые множества, возникающие при вычислимой аппроксимации сверху префиксной сложности / А. А. Мучник, А. Л. Семенов. — М.: ВЦ РАН, Отделение кибернетики, 2002. — 9 с.
http://alexander.shen.free.fr/muchnik/publications/hh-simple.pdf
12. Успенский В. А. Колмогоровская сложность и алгоритмическая случайность / В. А. Успенский, Н. К. Верещагин, А. Шень. — М.: МЦНМО, 2010. — 556 с.
13. *Blumer A.* Occam's Razor / A. Blumer, A. Ehrenfeucht, D. Haussler, M. Warmuth // Information Processing Letters, 1987. — Vol. 24 (6). — P. 377–380.
14. Blumer A. Learning faster than promise by the Vapnik-Chervonenkis dimension / Anselm Blumer, Nick Littlestone // Discrete Applied Mathematics, 1989. — Vol. 24. — Iss. 1–3. — P. 47–63.
15. Donskoy V. I. The Estimations Based on the Kolmogorov Complexity and Machine Learning from Examples / V. I. Donskoy // Proceedings of the Fifth International Conference "Neural Networks and Artificial Intelligence"(ICNNAI'2008). — Minsk: INNS. — 2008. — P. 292–297.
16. Ehrenfeucht A. A general lower bound on the number of examples needed for learning / A. Ehrenfeucht, D. Haussler, M. Kearns, L. Valiant // Inform. Computations, 1989. — 82. — P. 247 — 261.
17. Floyd S. Sample Compression, learnability, and the Vapnik-Chervonenkis dimension / Sally Floyd, Manfred Warmuth // J. Machine Learning, 1995. — Vol. 21. — Iss. 3. — P. 269–304.
18. Gammerman A. Kolmogorov complexity: Sources, theory and applications / Alexander Gammerman, Vladimir Vovk // Computer Journal, 1999. — Vol. 42. — No. 4. — p. 252–255.
19. Hutter M. Algoritmic complexity // Scholarpedia. — 2008. — 3 (1):2573.
http://www.scholarpedia.org/article/Algorithmic complexity.Prefix Turing machine
20. Li M. Sharpening Occam's Razor / Ming Li, John Tromb, Paul M. B. Vitanyi. — Research Rep. CT-94-03. — Amsterdam: ILLC, 1994. — 13 p.
http://www.illc.uva.nl/Research/Reports/CT-1994-03.text.pdf

21. Li M. An introduction to Kolmogorov complexity and its applications / Ming Li, Paul M. B. Vitanyi. — New York: Springer-Verlag, 1997. — 637 p.
22. Li M. Learning Simple Concepts under Simple Distributions / Ming Li, Paul M. B. Vitanyi // SIAM J. Comput. — Vol. 20. — Iss. 5. — P. 911–935.
23. Li M. Theories of Learning / Ming Li, Paul M. B. Vitanyi // In Proc. Int. Conf. Of Young Computer Scientists. — Beijing, China. — 1993. — 8 P.
24. Littlestone L. Relaring Data Compression and Learnability / Nick Littlestone, Manfred K. Warmuth. — Technical Report. — Santa-Cruz: University of California, 1986. — 13 p. http://users.soe.ucsc.edu/ manfred/pubs/T1.pdf
25. Vitanyi P. Ideal MDL and Its Relation to Bayesianism Bayesianism / Paul M. B. Vitanyi, Ming Li // In Proc. ISIS: Information, Statistic and Induction in Science. — Singapore: World Scientific, 1996. — P. 282–291.
26. Vitanyi P. Minimum description length induction, Bayesianism, and Kolmogorov complexity / Paul M. B. Vitanyi, Ming Li // IEEE Transactions on Information Theory, 2000 — Vol. 46. — N2. — P. 446–464.

Статья поступила в редакцию 24.11.2012