Ученые записки Таврического национального университета
им. В. И. Вернадского

Серия «Физико-математические науки»
Том 23 (62) № 2 (2010), с. 56–65.

УДК 519.95

В. И. Донской

ОЦЕНКИ ЕМКОСТИ ОСНОВНЫХ КЛАССОВ АЛГОРИТМОВ ЭМПИРИЧЕСКОГО ОБОБЩЕНИЯ, ПОЛУЧЕННЫЕ pVCD МЕТОДОМ

В работе представлены оценки емкости (VC-Dimension) основных классов алгоритмов эмпирического обобщения, используемых в задачах распознавания образов. Оценки получены новым $pVCD$ методом, который основан на колмогоровском подходе к определению сложности и сводится к построению сжатого описания информации об алгоритмах каждого рассматриваемого класса в виде битовой строки.

Ключевые слова: емкость Вапника-Червоненкиса, машинное обучение.

Введение

Используемый в статье $pVCD$ метод предполагает сужение семейства моделей эмпирического обобщения до классов частично-рекурсивных функций и даже уже – до классов вычислимых функций, реализуемых на компьютерах. Целесообразность такого подхода объясняется реализацией алгоритмов обучения и распознавания, как правило, именно на компьютерах. Поэтому исходное математическое описание семейств используемых моделей и правил распознавания в более широких классах, в частности, использование семейств непрерывных функций, влечет завышение оценок их сложности, которые к тому же очень трудно получать. Традиционное завышение оценок сложности семейств моделей обучения и распознавания объясняется тем, что математическое мышление создателей моделей чаще всего (и особенно на начальном творческом уровне) не рекурсивно по своей природе. Но при переходе к компьютерной реализации исходные модели и семейства решаюших правил автоматически сужаются до конечных рекурсивных схем. Именно последние используются для получения решений, и поэтому именно они и должны оцениваться.

Далее используются следующие обозначения. S — произвольное семейство частично-рекурсивных функций (алгоритмов), состоящее из функций вида A :

$X^n \to \{0,1\}$; $X^n = \{X = (x_1, ...x_n) : x_i \in \{0, 1, \ldots, 2^M - 1\}\}$. Выборка, состоящая из l произвольных элементов множества X^n, обозначается $\widetilde{X}_l = X_1, \ldots, X_l$ и представляет собой упорядоченный набор $n \times l$ ограниченных чисел из расширенного натурального ряда. Теоретически и практически допустимо считать все рассматриваемые числа представленными в виде бинарных строк. Множество всех выборок обозначается X^l.

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

Семейство S, внутри которого отыскивается решение, определяется условиями, которым должна удовлетворять искомая функция, и выбором модели обучения (и соответствующего класса алгоритмов распознавания), например, вычисления оценок, нейронных сетей, деревьев решений или алгебраических корректирующих моделей над перечисленными и/или другими эвристическими алгоритмами [2, 4, 6, 7, 9]. В частности, отыскивается такая функция $A^* \in S$, для которой эмпирический риск $\nu^l(A) = l^{-1} \sum_{j=1}^{l} |A(X_j) - \alpha_j|$ минимален.

Сложность семейств алгоритмов, применяемых в указанных задачах, имеет большое значение для обоснования выбора решений. Впервые важнейшее значение сложности семейств решающих правил в задачах эмпирического обобщения показали В. Н. Вапник и А. Я. Червоненкис [1]. Предложенная ими мера сложности, вообще говоря, произвольных вещественнозначных функций – так называемая ёмкость или VC-размерность, – возможно, является одним из самых ярких и полезных понятий для развития теории индуктивной математики (и не только). В данной статье приводится алгоритмическое определение сложности семейств алгоритмов, основанное на идеях А. Н. Колмогорова [5], и рассматривается применение введенной меры алгоритмической сложности для оценивания VC-размерности. Статья развивает новое направление в алгоритмической теории обучения, элементы которого впервые появились в работе автора [3].

Множество $\{0,1\}^*$ всех строк из нулей и единиц любой длины обычным образом (лексикографически и по длине)представляет целые неотрицательные числа $0, 1, 2, \ldots$. Длина слова $p \in \{0,1\}^*$ обозначается $len(p)$. Класс частично-рекурсивных функций обозначается $P_{p.r.}$. Логарифмы полагаются по основанию 2.

Определение сложности по Комогорову и $pVCD$ метод

Определение 1. Пусть U — такая частично-рекурсивная функция, что для каждого алгоритма $A \in S$ и для любой выборки $\widetilde{X}_l$ найдется двоичное слово p, которое обеспечивает выполнение равенства $U(p, \widetilde{X}_l) = \widetilde{y}$, где $\widetilde{y} = A(X_1), ..., A(X_l)$ — двоичное слово (строка) длины l. При этом каждый алгоритм $A \in S$ 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
$$K_U(A|\widetilde{X}_l) = \min len(p) : U(p, \widetilde{X}_l) = \widetilde{y}.$$

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

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

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

В приведенном определении сложность семейства алгоритмов S на множестве всех возможных выборок X^l длины l - это наименьшая длина двоичного слова p, по которому можно восстановить самый сложных (и любой) алгоритм $A \in S$. 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 S в слово p длины $K_l(S)$. Мажоранту сложности $K_l(S)$ 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 1. [10] *Пусть система частично-рекурсивных функций S вида $A : X^n \longrightarrow \{0,1\}$ имеет ограниченную емкость h_S и колмогоровскую сложность $K_l(S)$. Тогда при конечных значениях $h_S \geq 2$ и $l > h_S$ имеет место двойное неравенство $h_S \leq K_l(S) < h_S \log l$.*

Свойства сложности $K_l(S)$.

(1) Колмогоровская сложность семейства алгоритмов равна наименьшему целому, большему или равному логарифму функции роста этого семейства: $K_l(S) = \lceil \log m^S(l) \rceil$.
(2) $0 \leq K_l(S) \leq l$.
(3) Если $K_l(S) = o(l)$ при $l \to \infty$, то имеет место равномерная сходимость эмпирических частот ошибок к их вероятностям по всему классу S.
(4) Для любого алгоритма $U \in P_{p.r.}$ выполняется неравенство $K_l(S) \leq K_{U,X^l}(S)$.

Будем обозначать $h_S = VCD(S)$ емкость класса S. Подход к оцениванию VCD на основе неравенства $VCD(S) \leq len(p) : U(p, \widetilde{X}_l) = \widetilde{y} = (A(X_1), \ldots, A(X_l))$ называется методом программирования оценки VCD, сокращенно — $pVCD$ [3]. Метод $pVCD$ предполагает конструирование сжатого описания (слова) одной и той же структуры для любого $A \in S$ и указания алгоритма U, обрабатывающего вход $(p, \widetilde{X}_l)$. Длина такого слова p обозначается $pVCD(S) = len(p)$. Оценка $pVCD(S)$ может быть получена не единственным способом, и ее качество определяется найденным алгоритмом сжатия U. Иначе говоря, $pVCD(S)$ - это длина любого слова p такого, что его структура позволяет расшифровать при помощи некоторой программы U любой алгоритм из семейства S. При этом, хотя $pVCD(S)$ определяется неоднозначно, имеет место оценка $VCD(S) \leq pVCD(S)$. Построение слова p как можно меньшей длины и указание расшифровывающей его программы требует искусства программирования и изобретательности, всегда необходимой для получения новых математических результатов. В большинстве случаев точное определение структуры слова p делает очевидным алгоритм его расшифровки, что исключает необходимость подробного выписывания этого алгоритма.

Лемма 1. *(Об аддитивности $pVCD$ оценки композиции алгоритмов).*

Пусть $S_0^r = \{f_0 = f_1 \circ \cdots \circ f_r : f_1 \in S_1, \ldots, f_r \in S_r\}$ — класс композиций алгоритмов зафиксированной структуры $f_0(f_1, \ldots, f_r)$, принадлежащих семействам $S_1, \ldots, S_r$, для которых известны оценки $pVCD(S_1) = L_1, \ldots, pVCD(S_r) = L_r$. Тогда справедлива оценка $pVCD(S_0^r) = \sum_{j=1}^{r} L_j$.

Доказательство. Поскольку структура композиции неизменна, любой входящий в нее алгоритм определяется совокупностью слов $p_1, ..., p_j, ..., p_r$, имеющих длины $L_1, ..., L_j, ..., L_r$. Для обработки этих слов, согласно $pVCD$ методу программирования оценок и соотношению $U_j(p_j, \tilde{X}_l) = \tilde{y} = f_j(\tilde{X}_l)$, указаны алгоритмы

$U_j, j = 1, ..., r$, каждый из которых по слову p_j восстанавливает алгоритм f_j. Поэтому легко указать алгоритм (программу) U_{S_0}, обрабатывающую конкатенацию $p_0 = p_1p_2...p_r$ и соответствующую композиции $f_0 = f_1 \circ \cdots \circ f_r$. Такая программа будет содержать подпрограммы $U_j, j = 1, ..., r$, которые восстанавливают все алгоритмы $f_1, ..., f_r$, и переходы между этими подпрограммами, предопределенные зафиксированной структурой композиции и известными длинами $L_1, ..., L_j, ..., L_r$ подслов, входящих в конкатенацию $p_0 = p_1p_2...p_r$.

1. Оценивание VCD класса $DNF_{m,\mu,n}$ дизъюнктивных нормальных форм, содержащих не более чем μ конъюнкций над n переменными и не более m литералов

Дизъюнктивной нормальной формой (ДНФ) представления булевых функций называется выражение вида $\bigvee\limits_{j=1}^{\mu}(x_{j,1}^{\sigma_{j,1}} \wedge x_{j,2}^{\sigma_{j,2}} \wedge ... \wedge x_{j,k_j}^{\sigma_{j,k_j}})$, где $x^\sigma = x$ при $\sigma = 1$ (положительный литерал) и $x^\sigma = \bar{x}$ при $\sigma = 0$ (отрицательный литерал); μ – число конъюнкций в ДНФ; $m = \sum\limits_{j=1}^{\mu} k_j$ – суммарное число литералов, входящих в ДНФ.

Обозначим $DNF_{m,\mu,n}$ — семейство булевых функций, представимых в виде ДНФ, содержащих не более чем μ конъюнкций над n переменными и не более m литералов. Покажем, что для этого семейства функций справедлива оценка

$$pVCD(DNF_{m,\mu,n}) = m + (\mu - 1 + m)\lceil \log(n + 1)\rceil.$$

Действительно, слово p_f, позволяющее закодировать информацию о любой ДНФ, состоящей из μ конъюнкций над n переменными, можно представить конкатенацией двоичных слов сформированных из таких блоков, как показано в таблице 1.

Номер переменной x_j, $j \in 1, ..., n$, входящей в конъюнкцию, или ноль - разделитель блоков	Двоичная цифра 1, если x_j входит в конъюнкцию с инверсией, или 0 - в противном случае

Таблица 1. Фрагмент слова, кодирующего литерал

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

$$(\mu - 1)\lceil \log(n + 1)\rceil + \sum\nolimits_{j=1}^{\mu} k_j(\lceil \log(n + 1)\rceil + 1) = m + (\mu - 1 + m)\lceil \log(n + 1)\rceil.$$

Если ДНФ содержит $\nu < \mu$ конъюнкций, то последние $\mu - \nu$ блоков слова p_f заполняются нулями.

Пусть дана ДНФ $x_3\bar{x_5} \vee \bar{x_2}x_4$ из класса $DNF_{10,2,5}$ – не более чем с 10 литералами и не более чем с двумя конъюнкциями. Число переменных $n = 5$. Десятичная (для облегчения восприятия) структура слова p_f: |3|1|5|0|0|2|0|4|1|0|. Алгоритм расшифровки U поясняется следующей таблицей 2.

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

Таблица 2. Расшифровка ДНФ по слову p_f

2. Оценивание VCD нейронной сети с единственным скрытым слоем, содержащим k элементов (класс $NN_{k,1}$)

В работе [11] для нейронной сети с единственным скрытым слоем, содержащим k элементов, и зафиксированной непараметрической активационной функцией σ представлена оценка

$$VCD(NN_{k,1}) = (2kn + 4k + 2) \times \log(e(kn + 2k + 1)).$$

Используя pVCD метод, легко получить оценку

$$pVCD(NN_{k,1}) = M(nk + 2k + 1),$$

где M — число бит памяти, выделяемых для записи одного параметра; n — размерность входа.

Действительно, нейронные сети рассматриваемого класса полностью определяются $nk+2k+1$ параметрами: nk параметров соответствуют коэффициентам связи каждой из k внутренних вершин с каждым из n входов; k параметров определяют пороги суммирования для внутренних вершин и один параметр соответствует порогу выходной вершины сети. Если для каждого параметра используется M бит памяти, то каждую сеть рассматриваемого класса можно задать словом p_f длины $M(nk+2k+1)$. Алгоритм расшифровки этого слова состоит в последовательном считывании параметров (по M бит) согласно единому зафиксированному их порядку по всему классу. Считанные параметры подставляются в зафиксированные участки памяти алгоритма расшифровки.

Оценка, полученная pVCD методом, будет лучше известной [11] при $M < 2log(e(kn+2k+1))$, и ее выигрыш растет с ростом размерности задачи n.

3. VCD класса $N_{k,m}$ нейронных сетей с k элементами в каждом из m скрытых слоев

Для класса $N_{k,m}$ нейронных сетей с k элементами в каждом из m скрытых слоев аналогичным образом получена оценка

$$pVCD(N_{k,m}) = M(nk + 2mk^2).$$

4. VCD суперпозиции $f(F_1,\dots,F_k)$ с фиксированным логическим корректором $f \in P_2(k)$

Пусть $F_1,\dots,F_k$ — семейства алгоритмов вида $A : X^n \longrightarrow \{0,1\}$, имеющие VCD $VCD(F_1),\dots,VCD(F_k)$, и f — зафиксированная булева функция. Обозначим $f(F_1,\dots,F_k) = \{f(f_1,\dots,f_k) : f_i \in F_i, i=\overline{1,k}\}$. В работе [11] получена оценка

$$VCD(f(F_1,\dots,F_k)) \le 2k\log(ek) \max_{i=1,\dots,k}\{VCD(F_i)\}$$

Использование pVCD метода (см. лемму 1) позволяет получить

$$pVCD(f(F_1,\dots,F_k)) = \sum_{i=1}^{k} pVCD(F_i) \le k \max_{i=1,\dots,k} pVCD(F_i).$$

5. VCD класса бинарных решающих деревьев с μ листьями

pVCD метод позволяет получить оценку

$$pVCD(BFT_{n,\mu}) = (\mu-1)(\lceil\log n\rceil + \lceil\log(\mu+3)\rceil)$$

для класса $BFT_{n,\mu}$ бинарных решающих деревьев с μ листьями; n — число булевых переменных. Логико-комбинаторным методом автору ранее удалось получить оценку $VCD(BFT_{n,\mu}) < \mu \log(n\mu)$, которая в результате применения pVCD метода оказалась улучшенной.

Для класса $BSP_{n,\mu}$ [9] композиций бинарных решающих деревьев не более чем с μ листьями и линейными предикатами во внутренних вершинах, зависящими от n числовых переменных, занимающих по M бит памяти каждая, pVCD оценка имеет вид (см. лемму 1)

$$pVCD(BSP_{n,\mu}) = (\mu - 1)(\lceil \log n \rceil + \lceil \log(\mu + 3) \rceil + nM).$$

6. VCD структурной композиции линейного алгебраического корректора k эвристических моделей $F_1, \ldots, F_k$ (класс $L(F_1, \ldots, F_k)$)

Для совокупности эвристических алгоритмов с произвольным линейным корректором pVCD метод позволяет получить оценку

$$pVCD(L(F_1, \ldots, F_k)) = Mk + \sum_{i=1}^{k} pVCD(F_i)$$

7. Оценивание VCD интервальных множественных автоматов (IMA)

Класс решающих функций $\mathfrak{F}_{IMA}$, порождаемый IMA, описывается на основе следующих двух определений.

Определение 2. [8] Множественным автоматом (MA) называется пятерка

$$< Q, \Sigma, \delta, q_0, F >,$$

где Q — конечное множество состояний; Σ — конечный алфавит; $\delta : Q \times \Sigma \longrightarrow 2^Q$ — множественная функция переходов; $q_0 \in Q$ — начальное состояние; $F \subset Q$ — множество финальных состояний. Последовательность $p_0, p_1, \ldots, p_n$ называется принимаемым путем для входа $\omega_1, \ldots, \omega_n$, если $p_0 = q_0$, $p_i = \delta(p_{i-1}, \omega_i)$ для любого $i = 1, \ldots, n$ и $p_n \in F$. Автомат MA вычисляет функцию $f_{MA} : \Sigma^* \longrightarrow \{0, 1\}$, где $f_{MA}(\omega) = 1$, если число принимаемых путей для $\omega = (\omega_1, \omega_2, \ldots, \omega_n)$ является нечетным, и $f_{MA}(\omega) = 0$, если это число — четное.

Определение 3. [8] Интервальным множественным автоматом (IMA) называется пара $< A, C >$, где A — множественный автомат с алфавитом $\Sigma = \{0, 1, \ldots, \mu - 1\}$, C — множество, состоящее из μ вещественных чисел: $C = \{c_0, c_1, \ldots, c_{\mu-1}\}$; $c_0 = -\infty$; $c_0 < c_1, \cdots < c_{\mu-1}$. Индексом числа a, обозначаемым $ind_C(a)$, называется $\max\{i : c_i \leq a\}$.

Функция $f_{<A,C>}$, вычисляемая IMA $< A, C >$, ставит в соответствие вещественной числовой последовательности $(x_1, \ldots, x_n) \in R^n$ значение
$f_A(ind_C(x_1), \ldots, ind_C(x_n)))$.

В работе [8] получена оценка

$$VCD(\mathfrak{F}_{IMA}) = O(\mu(\log \mu + r^2)),$$

где $\mu = |\Sigma|$, $r = |Q|$. Сначала авторы [8] оценили сверху число способов обработки IMA входной последовательности как $(VCD(\mathfrak{F}_{IMA}) \cdot n+2)^{\mu} \cdot 2^{O(\mu r^2)}$, затем получили окончательный результат.

Применение pVCD метода дает существенно лучшую оценку

$$pVCD(\mathfrak{F}_{IMA}) = \mu(M + r^2) + r.$$

Выводы

Приведенные примеры оценивания VCD классов алгоритмов эмпирического обобщения при помощи pVCD метода и сравнение полученных оценок с известными ранее оценками позволяют заключить, что pVCD метод дает не худшие, а иногда даже лучшие результаты, чем логико-комбинаторные методы, применяемые для оценивания емкости. При этом оценки, получаемые pVCD методом, согласуются по своей структуре с известными из литературы оценками. Метод pVCD особенно эффективен для оценивания VCD классов сложных композиций алгоритмов эмпирического обобщения.

В дальнейшем целесообразно получить $pVCD$-оценки для классов моделей эмпирического обобщения, не рассмотренных в данной статье, в частности, АВО, потенциальных функций и др. [1, 4, 9]

Список литературы

[1] Вапник В. Н. *Восстановление зависимостей по эмпирическим данным.* – М: Наука, 1979. – 447 с.

[2] Донськой В. Й. *Бінарні вирішуючи дерева у задачах інтелектуального аналізу інформації* // Наукові вісті Національного технічного університету "Київський політехнічний Інститут". – 2001. – Вип. 5. –С.12–18.

[3] Донской В. И. *Колмогоровская сложность классов общерекурсивных функций с ограниченной емкостью* // Таврический вестник информатики и математики. – 2005. – №1. – С. 25–34.

[4] Журавлев Ю. И. *Об алгебраическом подходе к решению задач распознавания* // Проблемы кибернетики. – 1978. – Вып. 33. – С. 5–68.

[5] Колмогоров А. Н. *Теория информации и теория алгоритмов.* – М: Наука, 1987. – 304 с.

[6] Матросов В. А. *Корректные алгебры ограниченной емкости над множествами некорректных алгоритмов* // ДАН СССР. – 1980. – Т.253, №1. – С. 25–30.

[7] Хайкин С. *Нейронные сети: полный курс* – М.:Из-во "Вильямс 2006. – 995 с.

[8] Beimel A., Kushilevitz E. *Learning Unions of High Dimentional Boxes over the Reals* // Inf. Proc. Letters. – 2000. – vol.73, Issue 5–6. – P. 213–220.

[9] Devroye L. A., Gyorfi L., Lugosi G. *Probabilistic Theory of Pattern Recognition* – NY: Springer-Verlag, 1996. – 636 p.

[10] Donskoy V. I. *The estimations based on the Kolmogorov Complexity and Machine Learning from Examples* // Proc. of the 5-th Int. Conf. "Neural Networks and Artificial Intelligence"(ICNNAI'2008). – Minsk:INNS. – 2008. – P. 292–297.

[11] Sontag E.D. *VC dimension of Neural Networks* // In Neural Networks and Machine Learning. – Berlin: Springer, 1998. – P. 69–95.

Оцінки місткості основних класів алгоритмів емпіричного узагальнення, одержані pVCD методом

У роботі представлені оцінки місткості (VC-Dimension) основних класів алгоритмів емпіричного узагальнення, використованих в задачах розпізнавання образів. Оцінки одержані новим pVCD методом, який заснований на колмогоровському підході до визначення складності і зводиться до побудови стислого опису інформації про алгоритми кожного даного класу у вигляді бітового рядка.

Ключові слова: Місткість Вапника-Червоненкиса, машинне навчання.

VC-Dimension Estimations of the Basic Algorithms of Empirical Generalization for Pattern Recognition Problems obtained by the pVCD method

In this paper, VC-Dimension estimations for Decision Trees, Neural Networks, DNF, Function Compositions obtained by pVCD method are presented. The pVCD method is based on the Kolmogorov' approach to complexity definition.

Keywords: VC-Dimension, Machine Learning