УДК 519.7

М-МОДЕЛИ АЛГОРИТМОВ. ЁМКОСТЬ И КОЛМОГОРОВСКАЯ СЛОЖНОСТЬ КЛАССА М-ПОЛИНОМОВ

© Анафиев А.С.

Таврический национальный университет им. В.И. Вернадского
факультет математики и информатики
пр-т Вернадского, 4, г. Симферополь, 95007, Украина
e-mail: *anafiyev@gmail.com*

Abstract. The problems with elements bounded by a bit array are extracted in a special class of learning by precedents problems. A notion of M-models of learning algorithm is introduced. The Kolmogorov complexity and the VCD of M-polynomials and M-polynomials of Zhegalkin with k-component are estimated. The notions of complexity and a degree of compression by algorithms of M-models for training samples are introduced.

Введение

Рассмотрим постановку *задачи обучения по прецедентам*. Пусть заданы множество объектов X, множество ответов Y и существует некоторая целевая зависимость f^* между объектами и ответами, о которой известны лишь значения $y_i = f^*(x_i)$ на конечном наборе точек $\{x_1, \ldots, x_\ell\} \subset X$. Пары (объект – x_i, ответ – y_i) называют *прецедентами*, а совокупность таких пар $X^\ell = \{(x_i, y_i)\}_{i=1}^{\ell}$ – *обучающей выборкой*.

Необходимо построить алгоритм[1] $a : X \to Y$ восстанавливающий неизвестную целевую зависимость f^* на всем множестве объектов X, т.е. обладающий обобщающей способностью. Очевидно, что возникает вопрос, как оценивать качество восстановления алгоритмом a целевой зависимости f^*. Учитывая, что f^* задана частично – обучающей выборкой X^ℓ, – то и оценивание обобщающей способности алгоритма происходит по выборке X^ℓ относительно некоторого функционала качества $\Phi(a, X^\ell)$.

Обычно, при решении такого рода задач алгоритм a выбирается из некоторого параметрического семейства отображений

$$A = \{\varphi(x, \gamma) \mid \gamma \in \Gamma\}$$

называемого *моделью* [1] *алгоритма* a, где $\varphi : X \times \Gamma \to Y$ – некоторая фиксированная функция, Γ – множество допустимых значений параметра γ, называемое пространством параметров или пространством поиска. А одним из методом решения задачи обучения по прецедентам является выбор алгоритма $a \in A$, доставляющего максимум функционалу качества Φ:

$$a(x) = \arg\max_{a \in A} \Phi(a, X^\ell).$$

Таким образом, задачу обучения по прецедентам можно представить как пятерку $\langle X, Y, X^\ell, A, \Phi \rangle$. В зависимости от априорной информации и различного рода предположений относительно каждого элемента данной пятерки, возникают разного рода

[1]Под алгоритмом понимается вычислимая (частично-рекурсивная) функция.

задачи обучения и методы их решения. *Основными проблемами при решении задач обучения являются адекватность выбранной модели A модели неизвестной целевой зависимости f^* и оценка качества обобщающей способности построенного в результате обучения алгоритма $a \in A$.*

Одними из важнейших характеристик обучающей способности класса решающих правил служат ёмкость класса [2] и колмогоровская сложность [3]. Связь между этими двумя важными характеристиками классов показана в [4]. Развитие колмогоровского подхода к поиску закономерностей в данных привело к принципу минимума длины описания (*minimum description length – MDL*) [5, 6, 7], где под обучением понимается сжатие исходных данных, в нашем случае, сжатие обучающей выборки.

Обозначим через $\mathbb{N}_M = [0, 2^M - 1]$ – расширенный натуральный ряд ограниченный разрядной сеткой длины M. *Выделим следующий класс задач обучения по прецедентам: множество объектов* $X \subseteq \underbrace{\mathbb{N}_M \times \mathbb{N}_M \times \ldots \times \mathbb{N}_M}_{n \text{ раз}}$, *множество ответов* $Y \subseteq \mathbb{N}_M$, *а в модели алгоритмов A будем требовать, чтобы φ была вычислимой (частично-рекурсивной) функцией и* $\Gamma \subseteq \underbrace{\mathbb{N}_M \times \mathbb{N}_M \times \ldots \times \mathbb{N}_M}_{k \text{ раз}}$. Такого рода модели будем называть *M-моделями.*

Замечание. В дальнейшем будем рассматривать только такого рода задачи.

Для М-модели A произвольный алгоритм $a \in A$ можно запрограммировать (закодировать) последовательностью длины[2]

$$\ell(a) = \ell(\varphi) + kM,$$

где $\ell(\varphi)$ – длина программы, которая по конечному слову $x \in X$ и параметру $\gamma \in \Gamma$ восстанавливает слово $y = \varphi(x, \gamma)$, исходную выборку X^ℓ длины ℓ – последовательностью длины $\ell(n+1)M$ без учета разделительных битов, а обучающую выборку X^ℓ с помощью произвольного алгоритма $a \in A$ – словом длины $\ell(a) + n\ell M$, т.е. вначале кодируется алгоритм, а затем объекты выборки.

1. Полиномиальные модели

Рассмотрим модель алгоритмов $A_{n,k,r}$ М-полиномов[3] от n переменных с k слагаемыми степени не выше r:

$$A_{n,k,r} = \left\{ \sum_{j=1}^{k} \gamma_j \cdot x_{i_1^j} \cdot \ldots \cdot x_{i_{r_j}^j},\ r_j \leqslant r \right\}.$$

Вычислим длину $\ell(a)$ двоичной последовательности, с помощью которой можно восстановить любой полином $a \in A_{n,k,r}$. Для того, чтобы закодировать полином, необходимо закодировать все его k отличных от нуля слагаемых. Каждое слагаемое s

[2]Длина вычисляется без учета разделительных битов, если таковые требуются.

[3]Будем говорить М-полиномы, так как они определяются М-моделями.

полностью представляется двоичным словом[4]:

$$p^1(s) = p_1^1 \,.\, p_2^1 \,.\, \ldots \,.\, p_r^1,$$

длины $r\log(n+1)$, или

$$p^2(s) = p_1^2 \,.\, p_2^2 \,.\, \ldots \,.\, p_n^2,$$

длины $n\log(r+1)$, где p_i^1 – либо номер (в двоичной системе счисления) переменной, которая стоит на i месте в слагаемом s, либо нуль, если переменная на i-ом месте отсутствует, $i = \overline{1,r}$, а p_j^2 – число вхождений (в двоичной системе счисления) переменной x_j в слагаемое s, $j = \overline{1,n}$.

Например, для $n = 5$ и $r = 4$ слагаемое $s_1 = x_1x_4x_5$ полностью определяется словом $p^1(s_1) = 001.100.101.000$ или словом $p^2(s_1) = 001.000.000.001.001$, а слагаемое $s_2 = x_5^4 = x_5x_5x_5x_5$ – словом $p^1(s_2) = 101.101.101.101$ или словом $p^2(s_2) = 000.000.000.000.100$.

Тогда весь полином $a = \gamma_1 s_1 + \ldots + \gamma_k s_k$ можно представить словом

$$p^1 = \gamma_1 \,.\, p^1(s_1) \,.\, \gamma_2 \,.\, p^1(s_2) \,.\, \ldots \,.\, \gamma_k \,.\, p^1(s_k)$$

длины $k(r\log(n+1) + M)$, или словом

$$p^2 = \gamma_1 \,.\, p^2(s_1) \,.\, \gamma_2 \,.\, p^2(s_2) \,.\, \ldots \,.\, \gamma_k \,.\, p^2(s_k)$$

длины $k(n\log(r+1) + M)$, или словом

$$p_3 = \eta_1 \,.\, \eta_2 \,.\, \ldots \,.\, \eta_{P(n,r)},$$

длины $P(n,r)M$, где $P(n,r) = \dfrac{(n+r)!}{n!\,r!}$ – число всех возможных слагаемых полинома от n переменных степени не выше r, при этом $\eta_i = 0$, если слагаемое s_i не входит в полином, и $\eta_i = \gamma_j$, где γ_j – коэффициент при слагаемом s_i.

Теорема 1. *Ёмкость h класса $A_{n,k,r}$ полиномов от n переменных с k слагаемыми степени не выше r, удовлетворяет неравенству*

$$h(A_{n,k,r}) \leqslant \min\left(k(r\log(n+1) + M),\, k(n\log(r+1) + M),\, \frac{(n+r)!}{n!\,r!}M\right).$$

Доказательство. Доказательство теоремы проведем согласно принципу программирования оценки VCD, сокращенно – pVCD, предложенного в работе [8], основными этапами которого являются:

1. Изучение семейства решающих правил A и определение минимальной совокупности свойств (параметров, структурных особенностей), конкретное указание которых позволяет для некоторого алгоритма U сформировать двоичное слово p_a такое, что для любого входа x выполняется $U(p_a, x) = a(x)$, где $a \in A$ и U — некоторый алгоритм (машина), который для аргумента x по слову (программе) p вычисляет правильный ответ y.
2. Определение длины $\ell(p_a)$ слова p_a как искомой оценки сверху.

[4]Точкой . будем обозначать конкатенацию строк.

Рассмотрим произвольный полином $a \in A_{n,k,r}$, который, как показано выше, однозначно определяется словом длины

$$\ell(a) = \min\left(k(r\log(n+1)+M),\ k(n\log(r+1)+M),\ \frac{(n+r)!}{n!\,r!}M\right).$$

Пусть $K_\ell(A)$ – сложность семейства решающих правил A по Колмогорову: длина самого короткого двоичного слова, содержащее всю информацию, необходимую для восстановления произвольно решающего правила $a \in A$ при помощи какого-нибудь фиксированного способа (алгоритма) декодирования [3]. В нашем случае, учитывая, что произвольный полином $a \in A$ можно закодировать последовательностью длиной $\ell(a)$, то $K_\ell(A) = \ell(a)$. Тогда, в силу неравенства [8]

$$h(A) \leqslant K_\ell(A) \leqslant h(A)\log\ell,$$

справедливо утверждение теоремы. □

Теорема 2. *Ёмкость h класса $A^2_{n,k,r}$ полиномов Жегалкина от n переменных с k слагаемыми степени не выше r, удовлетворяет неравенству*

$$h(A^2_{n,k,r}) \leqslant \min\left(k(r\log(n+1)+M),\ k(n\log(r+1)+M),\ \sum_{i=0}^{r} C_n^i\right).$$

Доказательство. Доказательство проводится аналогично доказательству Теоремы 1. □

Зависимости оценки ёмкостей классов $A_{100,k,1}$ и $A_{100,k,2}$ от числа слагаемых k при фиксированных значениях числа переменных $n = 100$ и степени $r = 1$ и $r = 2$, т.е., соответственно, класса линейных и квадратичных отделителей, приводятся в табл. 1 и табл. 2, соответственно.

Таблица 1. Зависимость оценки $\hat{h}$ ёмкости $A_{100,k,1}$ от числа слагаемых k

k	$kr\log(n+1)$	$kn\log(r+1)$	$\frac{(n+r)!}{n!r!}$	$\hat{h}(P_{k,\,1}(100))$
1	6,658	100	101	6,658
2	13,316	200	101	13,316
3	19,975	300	101	19,975
4	26,633	400	101	26,633
5	33,291	500	101	33,291
6	39,949	600	101	39,949
7	46,607	700	101	46,607
8	53,266	800	101	53,266
9	59,924	900	101	59,924
10	66,582	1000	101	66,582
...	...	...	...	...
14	93,215	1400	101	93,215
15	99,873	1500	101	99,873
16	106,531	1600	101	101

Таблица 2. Зависимость оценки $\hat{h}$ ёмкости $A_{100,k,2}$ от числа слагаемых k

k	$kr\log(n+1)$	$kn\log(r+1)$	$\frac{(n+r)!}{n!r!}$	$\hat{h}(P_{k,2}(100))$
1	13,316	158,496	5151	13,316
2	26,633	316,993	5151	26,633
3	39,949	475,489	5151	39,949
4	53,266	633,985	5151	53,266
5	66,582	792,481	5151	66,582
6	79,899	950,978	5151	79,899
7	93,215	1109,474	5151	93,215
8	106,531	1267,970	5151	106,531
9	119,848	1426,466	5151	119,848
10	113,164	1584,963	5151	113,164

Как видно из таблиц 1 и 2, число слагаемых полиномов является важной характеристикой класса решающих правил $A_{n,k,r}$, и его учет при оценивании ёмкости классов $A_{n,k,r}$ позволяет существенно улучшить оценки.

Более того, оценку из теоремы 1 можно улучшить.

Теорема 3. *Ёмкость h класса $A_{n,k,r}$ полиномов от n переменных с k слагаемыми степени не выше r, удовлетворяет неравенству*

$$h(A_{n,k,r}) \leqslant kM \left\rceil \log\left(\frac{(n+r)!}{n!\,r!}\right)\right\lceil .$$

Доказательство. Доказательство проведем аналогично доказательству теоремы 1. Так как любое слагаемое полинома можно пронумеровать от 0 до $2^{P(n,r)-1}$, где $P(n,r) = \dfrac{(n+r)!}{n!\,r!}$ – число слагаемых полинома от n переменных степени не выше r, то, чтобы закодировать все слагаемые, нам потребуется $L_2 = \left\rceil\log(P(n,r))\right\lceil$ двоичных разрядов. Тогда произвольный полином $a \in A$, $a = \gamma_1\, s_1 + \ldots + \gamma_k\, s_k$, с k отличными от нуля коэффициентами можно закодировать словом

$$p = \underbrace{\underbrace{\gamma_1}_{M\text{ бит}} \cdot \underbrace{s_1}_{L_2\text{ бит}} \cdot \ldots \cdot \underbrace{\gamma_k}_{M\text{ бит}} \cdot \underbrace{s_k}_{L_2\text{ бит}}}_{k\text{ раз}},$$

откуда, согласно принципу pVCD, следует утверждение теоремы. □

На следующей таблице, на примере класса $A_{10,k,2}$, приведено сравнение оценок из теоремы 1 и теоремы 3.

Пример. Пусть дана выборка длины $\ell = 10$ от 10 переменных и $M = 64$. Как уже отмечалось, чтобы закодировать такую выборку нам необходимо $\ell M = 640$ бит. Тогда по таблице 3 видно, что для сжатия выборки (обучения по выборке) полиномами от 10 переменных степени не выше 2, нам необходимо построить полином хотя бы

Таблица 3. Сравнение оценок ёмкости класса полиномов $A_{10,k,2}$ при $M = 64$

k	$kr(\log(n+1)+M)$	$kM\,]\log(P(n,r))[$
1	130,10	71
2	260,20	142
3	390,30	213
4	520,39	284
5	650,49	355
6	780,59	426
7	910,69	497
8	1040,79	568
9	1170,89	639
10	1300,98	710

с 9 ($639 < 640$) отличными от нуля коэффициентами. Но для надежности (согласно MDL) желательно, чтобы число таких слагаемых было как можно меньше.

Определение 1. *Сложностью выборки* X^ℓ будем называть величину

$$C(X^\ell, A) = \min_{a \in A} \ell(a, X^\ell),$$

где $\ell(a, X^\ell)$ длина последовательности, по которой с помощью алгоритма a можно восстановить выборку X^ℓ.

Определение 2. Степенью сжатия выборки относительно семейства решающих правил A будем называть величину

$$R(X^\ell, A) = \begin{cases} \ell M - C(X^\ell, A), \text{ если } C(X^\ell, A) < \ell M \\ \infty, \text{ иначе.} \end{cases}$$

Становятся интересными следующие вопросы: способны ли алгоритмы заданной модели сжимать исходную выборку, если да, то насколько хорошо; как оценивать на основании оценки сложности и степени сжатия выборки относительно данной модели качество восстановления, желательно, иметь численную оценку надежности построенного в качестве решения алгоритма; вычислить введенные характеристики C и R класса решающих правил для известных моделей алгоритмов.

Заключение

Выделены особые классы задач обучения по прецедентам и M-моделей алгоритмов обучения, элементы которых ограничены разрядной сеткой. Сделаны оценки колмогоровской сложности и VCD классов M-полиномов и M-полиномов Жегалкина. Введены понятия сложности и степени сжатия обучающей выборки алгоритмами семейства M-моделей, влияющих на качество обучения. Выделяется ряд интересных и важных проблем, которые планируется решать в дальнейшем в рамках данной темы.

Список литературы

1. *Воронцов К.В.* Вычислительные методы обучения по прецедентам. Курс лекций по машинному обучению. – 2009 г. – 42 с.
   http://www.machinelearning.ru/wiki/images/8/8d/Voron-ML-Intro.pdf
2. *Вапник В.Н.* Восстановление зависимостей по эмпирическим данным. – М.: Наука, 1979. – 448 с.
3. *Колмогоров А.Н.* Теория информации и теория алгоритмов. –М.: Наука, 1987. – 304 с.
4. *Donskoy V. I.* The Estimations Based on the Kolmogorov Complexity and Machine Learning from Examples // Proceedings of the Fifth International Conference «Neural Networks and Artificial Intelligence» (ICNNAI'2008), Minsk, 2008, - p.p. 292-297.
5. *Li M., Vitanyi.* An Introduction to Kolmogorov Comlexity and Its Applications (revised and expanded second ed.). New York: Springed-Verlag, 1997.
6. *Rissanen J.* Modeling by shortest data description. Automatica, 14:465-471, 1978.
7. *Rissanen J., Tabus I.* Kolmogorov's structure function in MDL theory and lossy data compression. In P.D. Grunwald, I.J. Myung, and M.A. Pitt (Eds.) // Advances in Minimum Description Length: Theory and Applications. – MIT Press. – 2004.
8. *Донской В.И.* Колмогоровская сложность классов общерекурсивных функций с ограниченной ёмкостью // Таврический вестник информатики и матем. – 2005. – №1. – С. 25-34.
9. *Донской В.И.* Использование колмогоровской сложности для обоснования значимости эмпирических закономерностей // Интеллектуализация обработки информации – 2006. Тезисы докладов. – Симферополь. – 2006. – С. 63-67.

Статья поступила в редакцию 10.06.2010