ОБОЗРЕНИЕ
ПРИКЛАДНОЙ И ПРОМЫШЛЕННОЙ
Том 29 МАТЕМАТИКИ Выпуск 3
2022

А. А. Васильева, **Н. Р. Токарев** (Москва, МИЭМ НИУ ВШЭ)**. Построение и исследование теоретико-вероятностных моделей процедур построения префиксного кода.**

УДК 519.212.2+004.032.2 DOI https://doi.org/10.52513/08698325_2022_29_3_1

Резюме: Исследуется возможность практического применения вероятностного алгоритма построения префиксных кодов. Вычислены информационные характеристики кода, построенного с помощью указанного алгоритма. Проведен сравнительный анализ с другими алгоритмами построения префиксных кодов.

Ключевые слова: бинарное дерево, вероятностный алгоритм, префиксный код, равномерный код, средняя длина кодового слова.

Пусть задан алфавит источника сообщений $A = \{a_1, \ldots, a_N\}$, $N \in \mathbb{N}$, и распределение на его элементах $\overline{p} = (p(a_1), \ldots, p(a_N))$.

Пусть, далее, задан алфавит кодера $B = \{b_1, \ldots, b_D\}$, $D \leqslant N$, и реализовано кодирование φ, сопоставляющее буквам алфавита A слова в алфавите B. Одной из важных характеристик, описывающих эффективность передачи информации по каналу связи на основе кодирования φ, является *средняя длина* кодового слова [1]:

$$L = \sum_{i=1}^{N} p(a_i) l_i,$$

где l_i — длина кодового слова, соответствующего букве a_i, $i = 1, \ldots, N$.

Сравнительный анализ построенных ниже кодов будем проводить именно по указанной характеристике.

Итак, пусть требуется построить код мощности $N \in \mathbb{N}$. В качестве исходного множества для формирования кода будем использовать множество B_c, состоящее из всех двоичных слов с длинами из множества $\{T_{\min}, ..., T_{\max}\}$. Максимальная длина кодового слова $T_{\max}$ определяется соотношением

$$T_{\max} = \min\{i : 2^i \geqslant N\},$$

а минимальная длина $T_{\min}$:

$$T_{\min} = 1 + \max\{i : 2^{T_{\max}} - 2^{T_{\max} - i} + 1 < N\}.$$

Через M обозначим количество элементов во множестве B_c. Очевидно, что

$$M = 2^{T_{\min}} + 2^{T_{\min}+1} + \cdots + 2^{T_{\max}}.$$

Рассмотрим вероятностный алгоритм, который предполагает последовательный равновероятный выбор без возвращения N элементов множества B_c. Через ξ обозначим случайную величину, равную длине извлеченного кодового слова для соответствующей схемы извлечения. Распределение данной случайной величины ξ имеет вид:

$$\xi \sim \begin{pmatrix} T_{\min} & T_{\min}+1 & \cdots & T_{\max} \\ \dfrac{2^{T_{\min}}}{M} & \dfrac{2^{T_{\min}+1}}{M} & \cdots & \dfrac{2^{T_{\max}}}{M} \end{pmatrix}.$$

При каждом очередном извлечении по формуле Байеса вероятностное распределение на множестве M будет пересчитываться. Причем пересчет будет зависеть от элемента множества B_c, который был извлечен при отдельной итерации алгоритма.

Для реализации вероятностного алгоритма использовано представление кодов в виде бинарных деревьев [1]. Кроме того, на языке Python 3.9 реализован алгоритм вычисления надежности формирования префиксного кода с использованием вероятностного алгоритма [2].

Алгоритм. Вероятность P выбора префиксного кода

function Recursion $(B_c, N, T_{\min}, T_{\max}, s)$ ▷ Подается s=|B $_c$|
  if длина (B_c без неподходящих значений) $< N$ **then**
    return 0
  end if
  if $N = 0$ **then**
    return 1
  end if
  $l \leftarrow 0$
  $B_c^* \leftarrow B_c$
  $P \leftarrow 0$
  $f \leftarrow 1$
  for from $T_i \leftarrow T_{\min}$ **to** $T_i < T_{\max}+1$ **do** ▷ Цикл по длине слова
    $r \leftarrow l + (2^{T_i}) - 1$
    Для каждого слова длины T_i вычисляется количество неподходящих элементов-предков для выявления идентичных случаев
    for from $j \leftarrow l$ **to** $j < r+1$ **do** ▷ Цикл по словам длины T $_i$
      if j — неподходящее или учтенное слово **then**
        Перейти на следующий шаг цикла *for*
      end if
      $f \leftarrow$ количество идентичных кодовых слов элементу j
      Убираются из B_c^* все родительские и дочерние элементы для j
      $P^* \leftarrow$ RECURSION$(B_c^*, N-1, T_{\min}, T_{\max}, s-1)$ ▷ Вызов рекурсии
      $P = P + (P^* \cdot f)/s$
      $f \leftarrow 1$
      $B_c^* \leftarrow B_c$
    end for
    $l \leftarrow r + 1$
  end for
  return P
end function

В результате для различных N вычислена надежность P формирования префиксного кода, а также средняя длина кодового слова L. Полученные результаты представлены в табл. 1 и на рисунках 1 и 2.

Таблица 1. P и L при заданных N

N	$T_{\min}$	$T_{\max}$	P	L_r	L_{res}	N	$T_{\min}$	$T_{\max}$	P	L_r	L_{res}
1	1	1	1.000	1.00	1	9	1	4	0.005692	**3.80**	4
2	1	2	1.000	1.00	1	10	2	4	0.003768	**3.84**	4
3	1	2	0.300	**1.89**	2	11	2	4	0.001035	**3.87**	4
4	2	2	1.000	2.00	2	12	2	4	0.000239	**3.90**	4
5	1	3	0.072	**2.83**	3	13	2	4	0.000045	**3.93**	4
6	2	3	0.062	**2.90**	3	14	3	4	0.000133	**3.95**	4
7	2	3	0.015	**2.95**	3	15	3	4	0.000018	**3.98**	4
8	3	3	1.000	3.00	3	16	4	4	1.000000	4.00	4

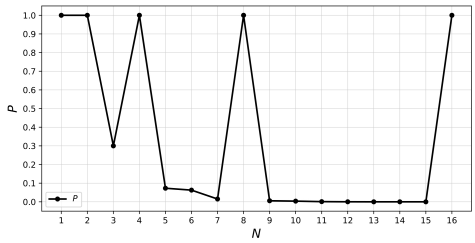

Рис. 1: $P(N)$

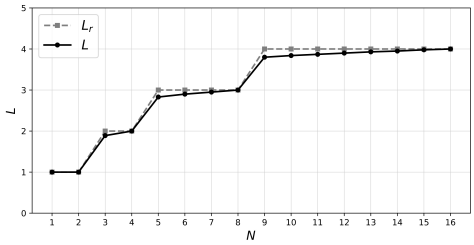

Рис. 2: $L(N)$

Здесь L_r — длина кодового слова в равномерном коде, позволяющем однозначно закодировать элементы исходного множества мощности N.

Таким образом, в ряде случаев вероятностный алгоритм является эффективнее с точки зрения минимизации средней длины кодового слова по сравнению с указанным равномерным кодом (выделенные значения в табл. 1). Вместе с тем, вероятность выполнения свойства префиксности P в общем случае меньше L/L_r.

Реализуем эксперимент, состоящий в среднем из $1/P > L_r/L$ независимых реализаций вероятностного алгоритма и соответствующего кодирования исходного сообщения. Результаты кодирования конкатенируем и направляем в канал связи приемнику-декодеру. В этом случае декодер однозначно восстановит исходное сообщение, однако средняя длина кодового слова составит величину L_{res}, большую или равную L_r (см. табл. 2).

Таблица 2. L_r и L_{res} при заданных N

N	$T_{\min}$	$T_{\max}$	L_r	L_{res}	N	$T_{\min}$	$T_{\max}$	L_r	L_{res}
1	1	1	1	1.00	9	1	4	4	667.60
2	1	1	1	1.00	10	2	4	4	1018.02
3	1	2	2	6.30	11	2	4	4	3737.02
4	2	2	2	2.00	12	2	4	4	16293.01
5	1	3	3	38.81	13	2	4	4	87335.62
6	2	3	3	46.15	14	3	4	4	29825.02
7	2	3	3	194.86	15	3	4	4	216706.68
8	3	3	3	3.00	16	4	4	4	4.00

Проведенные экспериментальные исследования показали, что в общем случае вероятностный алгоритм дает преимущества перед равномерным алгоритмом кодирования, использующим слова фиксированной длины, достаточной для описания всего множества из N элементов.

СПИСОК ЛИТЕРАТУРЫ

1. *Духин А. А.* Теория информации: Учебное пособие, М: Гелиос АРВ, 2007, 248 с.// *Dukhin A. A.* Information Theory: A textbook, Moscow: Gelios ARV, 2007, 248 p. (In Russian.)
2. https://github.com/aavasileva/prefix_codes

Поступила в редакцию
1.XII.2022

UDC 519.212.2+004.032.2 DOI https://doi.org/10.52513/08698325_2022_29_3_1

Vasil'yeva A. A., Tokarev N. R. (Moscow, HSE Tikhonov Moscow Institute of Electronics and Mathematics, National Research University Higher School of Economics)**. Stochastic models for an algorithm to generate prefix codes: Design and study.**

Abstract: The possibility of practical application of a probabilistic algorithm to generate prefix codes is investigated. The information characteristics of the code constructed by making use of the specified algorithm are calculated. A comparative analysis with other algorithms generating prefix codes is carried out.

Keywords: average codeword length, binary tree, fixed-length code, prefix code, probabilistic algorithm.