

БЫСТРОЕ КОДИРОВАНИЕ МАРКОВСКИХ ИСТОЧНИКОВ С МАЛОЙ ЭНТРОПИЕЙ*)

М. П. Шарова

Рассматривается задача кодирования марковских источников информации с малой энтропией. Начиная с кода «длин серий», предложенного К. Шенноном в [5], было известно, что для кодирования таких источников существуют значительно более простые методы, чем для произвольных источников. Однако известные методы кодирования источников с малой энтропией не позволяют строить коды с наперед заданной избыточностью. В работах [4, 10] был предложен новый метод кодирования бинарных бернуллиевских источников с малой энтропией, позволяющий строить коды с любой наперед заданной фиксированной избыточностью. Память кодера и декодера этого метода по порядку равна памяти общих методов, а его скорость кодирования и декодирования существенно выше. В данной работе обобщается метод кодирования, описанный в [4, 10], на марковские источники с малой энтропией (с двоичным алфавитом), а также на источники с недвоичным алфавитом.

Введение

Рассматривается задача кодирования источников информации, энтропия которых имеет малое значение. Простейшим примером таких источников является бернуллиевский источник, порождающий последовательность нулей и единиц с вероятностями q и p соответственно, когда p мало, а вероятность порождения очередной буквы не зависит от предшествующих букв. В общем случае сообщение, порождаемое бернуллиевским источником с малой энтропией, состоит в основном из длинных серий высоковероятной буквы.

Задача кодирования источников с малой энтропией вызывала и вызывает интерес многих исследователей (см., например, [5, 6]), так как известно, что для кодирования таких источников существуют более простые методы, чем для произвольных источников. Одной из наиболее

*) Работа выполнена при финансовой поддержке Российского фонда фундаментальных исследований (код проекта 96-01-00052).

известных схем сжатия информации, порожденной бернуллиевским источником с малой энтропией, является кодирование «длин серий» [5]. В этом методе последовательность букв, порождаемая источником, разбивается на серии по числу нулей между двумя последовательными единицами: 1, 01, 001 и т. д., а затем длины серий кодируются двоичными словами. К. Шеннон предложил использовать специальное кодовое слово для маловероятной единицы, играющей роль запятой. При этом длины серий кодируются двоичными словами с исключением специального кодового слова.

Еще один метод кодирования длин серий был предложен П. Элайесом [6]. Он описал три универсальных представления целых чисел и на их основе построил универсальные кодовые множества. Отметим, что конструкция лучшего кода из [6] аналогична известному представлению чисел, предложенному ранее В. И. Левенштейном [3].

Как известно, одной из важных характеристик эффективности кода является его избыточность, определяемая разностью между средней длиной кодового слова и энтропией источника. В методе, предложенном К. Шенноном, при росте длины кода для маловероятной буквы и $p \rightarrow 0$ избыточность такого кодирования стремится к нулю. Кроме того, можно показать, что она не меньше $C_1 p \log(1/p)$, где $C_1 \leq 1$ — константа, зависящая от длины кода маловероятной буквы. Избыточность лучшего кода, построенного методом П. Элайеса, не меньше $C_2 p \log \log(1/p)$, где C_2 — константа.

Однако методы К. Шеннона и П. Элайеса не позволяют строить коды с наперед заданной избыточностью. В работах [4, 10] описан новый метод кодирования информации, порожденной бернуллиевским источником с малой энтропией, который в отличие от ранее предложенных методов [5, 6] позволяет строить коды с любой наперед заданной избыточностью. В данной работе метод из [4, 10] переносится на случай марковских источников (или источников с памятью) с малой энтропией, когда вероятность порождения каждой очередной буквы зависит от уже порожденных букв. Кроме того, конструкция кода, описанная в [4, 10] для случая бинарных бернуллиевских источников с малой энтропией, обобщается на случай алфавита $A = \{a_1, a_2, \dots, a_n\}$, когда $n > 2$.

Эффективность кодов мы будем оценивать объемом памяти кодера и декодера (в битах), который требуется при реализации метода на компьютере со свободным доступом к памяти (это модель «обычного» компьютера; см. определение в [1]), и средним временем кодирования и декодирования одной буквы источника информации, измеряемым числом операций над однобитовыми словами. Память кодера и декодера предложенного в [4, 10] метода кодирования по порядку равна памяти общих

методов, а его скорость кодирования и декодирования существенно выше. Так, например, среднее время кодирования и декодирования «быстрого» алгоритма из [9] равно $O(\log^3(1/r) \log \log(1/r))$, а время предложенного в [4, 10] метода не превосходит $C\sqrt{p} \log^3(1/(rp)) \log \log(1/(rp))$, где r — избыточность кода, C — константа. Отметим, что метод кодирования мы рассматриваем только для случая, когда известны вероятности порождения букв источником.

В п. 1 рассматривается задача кодирования бинарных марковских источников с малой энтропией, в п. 2 — бернуллиевских источников с малой энтропией в случае недвоичного алфавита.

1. Кодирование марковских источников с малой энтропией

Пусть ω — марковский источник с малой энтропией, порождающий буквы из алфавита $A = \{0, 1\}$ с известными условными вероятностями. Напомним, что источник ω , порождающий последовательность $x_1 x_2 \dots x_t \dots$, называется *марковским источником* (или *источником с памятью*) порядка n ($n > 0$), если при любом $k \geq n$ и каждом i

$$P_\omega(x_i | x_{i-k} \dots x_{i-1}) = P_\omega(x_i | x_{i-n} \dots x_{i-1}),$$

т. е. вероятность порождения каждой буквы зависит не более чем от n предшествующих букв (вероятность первоначальной буквы задана). Пусть $r > 0$ — произвольное число. Наша задача — указать метод кодирования источника, который позволяет строить код с избыточностью, не превосходящей величины r .

Приведем сначала необходимые определения. Как известно, марковский источник можно задать множеством состояний и множеством букв, порождаемых источником. Случайная последовательность состояний, когда вероятность перехода в каждое состояние зависит только от предыдущего состояния, является конечной однородной цепью Маркова ([2]). Отметим, что i -е состояние марковской цепи можно рассматривать как i -й бернуллиевский источник. Поэтому при кодировании исходный марковский источник порядка n можно представить в виде $|A|^n$ бернуллиевских источников. Энтропией $H(\omega)$ марковского источника ω называется величина

$$H(\omega) = \sum_i \pi_i H_i, \quad (1)$$

где H_i — энтропия i -го состояния, π_i — предельные вероятности марковской цепи, т. е.

$$\pi_i = \lim_{t \rightarrow \infty} P_{ji}(t),$$

где $P_{ji}(t)$ — вероятность перехода цепи из j -го состояния в i -е за t шагов ($t \geq 1$), причем при каждом i предел не зависит от j . (В дальнейшем рассматриваются только цепи, для которых предельные вероятности и тем самым энтропия существуют.) Избыточностью $R(\omega)$ марковского источника ω называется величина

$$R(\omega) = \sum_i \pi_i l_i - H(\omega),$$

где l_i — средняя длина кода i -го состояния (т. е. i -го бернуллиевского источника), $H(\omega)$ — энтропия марковского источника, определяемая формулой (1).

Теперь опишем алгоритм кодирования, который является эффективным для марковского источника с малой энтропией, порождающего последовательность нулей и единиц с условными вероятностями $p(0 | x)$ и $p(1 | x)$, где $x \in A^n$, $A = \{0, 1\}$, n — порядок марковского источника, A^n — множество всех слов длины n в алфавите A . Важно отметить, что в отличие от бернуллиевского источника с малой энтропией, порождающего последовательность нулей и единиц с вероятностями $p(1) = \varepsilon$ и $p(0) = 1 - \varepsilon$, а $\varepsilon \rightarrow 0$ (т. е. сообщение содержит одну высоковероятную букву), в случае марковского источника с малой энтропией условие $p(1 | x) < p(0 | x)$ может выполняться не для всех $x \in A^n$.

В предлагаемом методе кодирование осуществляется в два этапа. На первом этапе порождаемое источником сообщение сжимается простым методом, а полученная последовательность кодируется на втором этапе более сложным методом. Так как источник имеет малую энтропию, то после первого этапа кодирования длина входной последовательности существенно сокращается, а применение «быстрого» алгоритма на втором этапе обеспечивает небольшое суммарное время кодирования и декодирования в пересчете на букву исходного сообщения. На втором этапе кодирования могут быть использованы многие универсальные коды, например арифметический код [7, 8] или код из [9]. Мы будем использовать код из [9], так как для него известны среднее время и память. Следует отметить, что использование некоторых вариантов универсального арифметического кода дает тот же результат.

В работе [9] предложен «быстрый» метод кодирования источников информации, обладающий свойством произвольно малой избыточности (кодеры, реализующие этот метод, ниже обозначены через K_0 , K_i , \hat{K}). Приведем необходимые в дальнейшем характеристики кода из [9]. Известно, что в случае бернуллиевского источника для этого кода зависимость объема памяти V и среднего времени T кодирования и декодирования одной буквы от избыточности r' при $r' \rightarrow 0$ удовлетворяет

соотношениям

$$V = O\left(\frac{1}{r'} \log \frac{1}{r'}\right), \quad T = O\left(\log^3\left(\frac{1}{r'}\right) \log \log \frac{1}{r'}\right). \quad (2)$$

Предлагаемый метод кодирования опишем более подробно. Начнем со случая, когда $A = \{0, 1\}$, $p(1 | x) < p(0 | x)$ для всех $x \in A^n$. Пусть $x_1 x_2 \dots x_i \dots$ — последовательность, порожденная марковским источником порядка n , $x_i \in A$. Рассмотрим первый этап кодирования. Положим

$$\tilde{p} = \max_{x_1 \dots x_n} p(1 | x_1 \dots x_n) \quad (3)$$

и разобьем последовательность на блоки длины $l = \left\lceil \frac{1}{\sqrt{\tilde{p}}} \right\rceil$. Кодирование полученных блоков будем осуществлять следующим образом. Если блок состоит только из нулей, то кодом этого блока является 0. В остальных случаях длина кодового слова равна $l + 1$: первой буквой этого слова является 1, за которой следует тот же блок длины l .

Пусть теперь $y_1 y_2 \dots y_s$ — последовательность, полученная после первого этапа кодирования, $y_i \in A$. Рассмотрим второй этап кодирования, осуществляемый «быстрым» алгоритмом из [9]. Так как полученная последовательность уже не может рассматриваться как порожденная обычным марковским источником, то для ее кодирования на втором этапе предлагается новый метод. Для удобства изложения метода последовательность $y_1 \dots y_s$ сначала представим в виде

$$\underline{0} \dots \underline{0} \underline{1} \underbrace{y_1 \dots y_l}_l \underline{0} \dots \underline{0} \underline{1} \underbrace{y_1 \dots y_l}_l \dots$$

В этой последовательности выделены блоки длины l , следующие за каждой выделенной единицей, и особые буквы 0 и 1, не входящие в блоки (в последовательности особые буквы подчеркнуты).

Кодирование различных y_i осуществляется «быстрым» алгоритмом из [9] с помощью различных кодеров K_0, K_i, \hat{K} , настроенных на различные вероятности появления нулей и единиц. Это кодирование осуществляется следующим способом.

Сначала опишем, как кодируются особые буквы $\underline{0}$ и $\underline{1}$. Каждому блоку исходной последовательности $\underbrace{0 \dots 0}_l$ соответствует особая буква

$\underline{0}$. Пусть $x_1 \dots x_n$ — последовательность букв перед соответствующим блоком $\underbrace{0 \dots 0}_l$. Тогда особые буквы $\underline{0}$ и $\underline{1}$ кодируются с помощью кодера

K_0 с вероятностями δ и $1 - \delta$ соответственно, где

$$\delta = p(0 | x_1 \dots x_n) \cdot p(0 | x_2 \dots x_n 0) \cdot \dots \cdot p(0 | x_l \dots x_n \underbrace{0 \dots 0}_{l-1}). \quad (4)$$

Опишем кодирование букв, находящихся внутри блока $y_1 \dots y_l$ ($y_1 \dots y_l \neq \underbrace{0 \dots 0}_l$). Пусть $y_1 \dots y_{i-1} = \underbrace{0 \dots 0}_{i-1}$ ($i = 1, \dots, l$). Определим вероятность буквы y_i , находящейся после $i - 1$ нулей. Каждой букве исходной последовательности $x_{k+n+1} \dots x_{k+n+i}$ ($k \in N$), порожденной марковским источником, соответствует определенная буква последовательности $y_1 \dots y_l$, полученной после первого этапа кодирования (например, букве x_{k+n+1} исходной последовательности соответствует буква y_1). Имеем

$$\begin{aligned} \tau_i &= P(y_i = 1 \mid y_1 \dots y_{i-1} = 0 \dots 0; y_1 \dots y_l \neq 0 \dots 0) \\ &= \frac{P(y_i = 1; y_1 \dots y_{i-1} = 0 \dots 0 \mid y_1 \dots y_l \neq 0 \dots 0)}{P(y_1 \dots y_{i-1} = 0 \dots 0 \mid y_1 \dots y_l \neq 0 \dots 0)}. \end{aligned}$$

Так как

$$\begin{aligned} &P(y_i = 1; y_1 \dots y_{i-1} = 0 \dots 0 \mid y_1 \dots y_l \neq 0 \dots 0) \\ &= \frac{P(y_i = 1; y_1 \dots y_{i-1} = 0 \dots 0; y_1 \dots y_l \neq 0 \dots 0)}{P(y_1 \dots y_l \neq 0 \dots 0)} \\ &= \frac{p(1 \mid x_{k+i} \dots x_{k+n} \underbrace{0 \dots 0}_{i-1}) [p(0 \mid x_{k+1} \dots x_{k+n}) p(0 \mid x_{k+2} \dots x_{k+n} 0) \dots]}{1 - p(0 \mid x_{k+1} \dots x_{k+n}) p(0 \mid x_{k+2} \dots x_{k+n} 0) \dots} \\ &\quad \frac{\dots p(0 \mid x_{k+i-1} \dots x_{k+n} \underbrace{0 \dots 0}_{i-2})}{\dots p(0 \mid x_{k+l} \dots x_{k+n} \underbrace{0 \dots 0}_{l-1})} \end{aligned}$$

и

$$\begin{aligned} &P(y_1 \dots y_{i-1} = 0 \dots 0 \mid y_1 \dots y_l \neq 0 \dots 0) \\ &= \frac{P(y_1 \dots y_{i-1} = 0 \dots 0; y_1 \dots y_l \neq 0 \dots 0)}{P(y_1 \dots y_l \neq 0 \dots 0)} \\ &= \frac{[p(0 \mid x_{k+1} \dots x_{k+n}) \dots p(0 \mid x_{k+i-1} \dots x_{k+n} \underbrace{0 \dots 0}_{i-2})]}{1 - p(0 \mid x_{k+1} \dots x_{k+n}) p(0 \mid x_{k+2} \dots x_{k+n} 0) \dots} \\ &\quad \times \frac{[1 - p(0 \mid x_{k+i} \dots x_{k+n} \underbrace{0 \dots 0}_{i-1}) \dots p(0 \mid x_{k+l} \dots x_{k+n} \underbrace{0 \dots 0}_{l-1})]}{\dots p(0 \mid x_{k+l} \dots x_{k+n} \underbrace{0 \dots 0}_{l-1})}, \end{aligned}$$

то

$$\tau_i = \frac{p(1 \mid x_{k+i} \dots x_{k+n} \underbrace{0 \dots 0}_{i-1})}{1 - p(0 \mid x_{k+i} \dots x_{k+n} \underbrace{0 \dots 0}_{i-1}) \dots p(0 \mid x_{k+l} \dots x_{k+n} \underbrace{0 \dots 0}_{l-1})}. \quad (5)$$

Кроме того, $P(y_i = 0 \mid y_1 \dots y_{i-1} = 0 \dots 0; y_1 \dots y_l \neq 0 \dots 0) = 1 - \tau_i$.

Следовательно, буква y_i , находящаяся после $i - 1$ нулей, кодируется с помощью кодера K_i с вероятностью τ_i и $1 - \tau_i$ для 1 и 0 соответственно.

Наконец, буквы из блока $y_1 \dots y_l$, расположенные после появления в этом блоке 1, кодируются с помощью кодера \hat{K} с исходными условными вероятностями $p(0 \mid x)$ и $p(1 \mid x)$ для 0 и 1 соответственно, где $x \in \{0, 1\}^n$.

Итак, кодирование букв в блоке $y_1 \dots y_l$ осуществляется по следующей схеме.

Шаг 1. Вычисляется τ_i и буква y_i кодируется с вероятностью τ_i и $1 - \tau_i$ для 1 и 0 соответственно.

Шаг 2. Если $y_i = 1$, то все буквы, расположенные после y_i , кодируются с исходными условными вероятностями $p(0 \mid x)$ и $p(1 \mid x)$ для 0 и 1 соответственно. Иначе осуществляется переход к следующей букве и возврат к шагу 1.

Рассмотрим пример, поясняющий описанную выше схему кодирования. Пусть марковский источник первого порядка с вероятностями $p(0 \mid 0) = 17/21$; $p(1 \mid 0) = 4/21$; $p(0 \mid 1) = 4/5$; $p(1 \mid 1) = 1/5$ порождает последовательность 000000001000110000101000000. Из (3) следует, что $\tilde{p} = 1/5$, $l = 3$ и после первого этапа кодирования исходная последовательность преобразуется в последовательность 0 0 1001 0 1110 0 1101 0 0 (пробелы поставлены для удобства чтения). На втором этапе кодирования полученную последовательность представим в виде

$$\begin{array}{cccccccccccccccccccc} \underline{0} & \underline{0} & \underline{1} & \underline{0} & \underline{0} & \underline{1} & \underline{0} & \underline{1} & \underline{1} & \underline{1} & \underline{0} & \underline{0} & \underline{1} & \underline{1} & \underline{0} & \underline{1} & \underline{0} & \underline{0} \\ y_1 & y_2 & y_3 & y_4 & y_5 & y_6 & y_7 & y_8 & y_9 & y_{10} & y_{11} & y_{12} & y_{13} & y_{14} & y_{15} & y_{16} & y_{17} & y_{18} \\ K_0 & K_0 & K_0 & K_1 & K_2 & K_3 & K_0 & K_0 & K_1 & \hat{K} & \hat{K} & K_0 & K_0 & K_1 & \hat{K} & \hat{K} & K_0 & K_0. \end{array}$$

Из (4) определяем вероятности особых букв 0 и 1. Так, например, буква $y_2 = 0$ кодируется с вероятностью $p(y_2) = p(0 \mid 0)^3 = (17/21)^3$; буква $y_7 = 0$ — с вероятностью $p(y_7) = p(0 \mid 1) \cdot p(0 \mid 0)^2 = 4/5 \cdot (17/21)^2$. Из (5) следует, что буква y_4 ($i = 1$) первого блока $y_4 y_5 y_6$ кодируется с помощью кодера K_1 с вероятностью $1 - \tau_4 = 1 - \frac{p(1|0)}{1 - p(0|0)^3} = \frac{4/21}{1 - (17/21)^3}$. Так как $y_4 = 0$, то осуществляется переход к букве y_5 ($i = 2$). Из (5) следует, что $\tau_5 = \frac{p(1|0)}{1 - p(0|0)^2} = \frac{4/21}{1 - (17/21)^2}$, и буква $y_5 = 0$ кодируется кодером K_2 с вероятностью $1 - \tau_5$. Так как $y_5 = 0$, то буква $y_6 = 1$ ($i = 3$) кодируется с помощью кодера K_3 с вероятностью $\tau_6 = 1$ (т. е. буква y_6 известна до кодирования). Во втором блоке $y_9 y_{10} y_{11}$ буква y_9 ($i = 1$) кодируется с помощью кодера K_1 с вероятностью $\tau_9 = \frac{p(1|0)}{1 - p(0|0)^3} = \frac{4/21}{1 - (17/21)^3}$. Так как $y_9 = 1$, то оставшиеся буквы y_{10}, y_{11} кодируются с помощью кодера \hat{K} с вероятностями $p(y_{10}) = p(1 \mid 1) = 1/5$; $p(y_{11}) = p(0 \mid 1) = 4/5$. Аналогично кодируются буквы третьего блока $y_{14} y_{15} y_{16}$.

Рассмотрим теперь случай, когда условие $p(1 | x) < p(0 | x)$ выполняется не для всех $x \in \{0, 1\}^n$. Исходную последовательность, порожденную марковским источником, преобразуем в новую по следующему правилу. Если $p(0 | x) = \varepsilon_1$ ($\varepsilon_1 < 1/2$), то все нули, расположенные после данного $x \in \{0, 1\}^n$, будем кодировать единицей, в противном случае — нулем. Аналогично, если $p(1 | x) = \varepsilon_2$ ($\varepsilon_2 < 1/2$), то все единицы, расположенные после данного $x \in \{0, 1\}^n$, будем кодировать единицей, в противном случае — нулем. Заметим, что новая последовательность может быть легко преобразована (декодирована) в исходную. Для этого достаточно знать лишь первые n букв полученной последовательности. Кроме того, в новой последовательности ноль является высоковероятным. Например, пусть марковский источник второго порядка с вероятностями $p(0 | 00) = p(0 | 01) = 1/5$, $p(0 | 10) = p(0 | 11) = 3/4$, $p(1 | 00) = p(1 | 01) = 4/5$, $p(1 | 10) = p(1 | 11) = 1/4$ порождает последовательность 001100110001011100110. Тогда после данного преобразования новая последовательность имеет вид 000000000010110100000.

Заметим, что на втором этапе кодирования в памяти необходимо хранить вероятности τ_i . На первом этапе в памяти кодера и декодера хранятся счетчики нулей объема $O(\log(1/\tilde{p}))$. Общий объем памяти кодера и декодера предложенного метода кодирования марковских источников равен памяти метода для случая бернуллиевских источников, умноженной на общее число кодеров, равное 2^n , где n — порядок марковского источника.

Скорость кодирования и декодирования предложенного метода характеризует

Теорема 1. Пусть даны марковский источник порядка n ($n > 0$) с малой энтропией, порождающий последовательность нулей и единиц с известными условными вероятностями $p(0 | x)$, $p(1 | x)$ ($x \in \{0, 1\}^n$), и некоторое r , $r > 0$. Пусть для кодирования данного источника используется описанный выше код с $l = \left\lceil \frac{1}{\sqrt{\tilde{p}}} \right\rceil$, где \tilde{p} определяется (3). Тогда общая избыточность кода не превосходит r , а среднее время T кодирования и декодирования одной буквы удовлетворяет неравенству

$$T < C_1 \sqrt{\tilde{p}} \log^3\left(\frac{1}{r}\right) \log \log \frac{1}{r} + C_2,$$

где C_1, C_2 — константы.

Доказательство. Согласно определению избыточность

$$R = \sum_i \pi_i l_i - H,$$

где H — энтропия марковского источника, определяемая (1). Учитывая, что согласно [4, 10] в случае бернуллиевского источника избыточность

$l_i - H_i$ не превосходит r , имеем

$$R = \sum_i \pi_i l_i - \sum_i \pi_i H_i = \sum_i \pi_i (l_i - H_i) \leq r,$$

т. е. общая избыточность кода в случае марковского источника также не превосходит r .

Вычисление среднего времени T основано на подсчете числа бинарных операций, затрачиваемых на кодирование и декодирование. На первом этапе время кодирования равно $O(1)$. Чтобы определить время кодирования на втором этапе, сначала оценим полученную после первого этапа среднюю длину \tilde{l} кода на букву исходного слова. Имеем

$$\begin{aligned} \tilde{l} &= \frac{1}{l} \left(\sum_{x_1 \dots x_n} p(x_1 \dots x_n) p(\underbrace{0 \dots 0}_l | x_1 \dots x_n) \right. \\ &\quad \left. + (l+1) \sum_{x_1 \dots x_n} p(x_1 \dots x_n) \sum_{z_1 \dots z_l \neq 0 \dots 0} p(z_1 \dots z_l | x_1 \dots x_n) \right) \\ &= \frac{1}{l} \left(1 + l \sum_{x_1 \dots x_n} p(x_1 \dots x_n) \sum_{z_1 \dots z_l \neq 0 \dots 0} p(z_1 \dots z_l | x_1 \dots x_n) \right) \\ &= 1 + \frac{1}{l} - \sum_{x_1 \dots x_n} p(x_1 \dots x_n) p(\underbrace{0 \dots 0}_l | x_1 \dots x_n). \end{aligned}$$

Так как согласно (3)

$$\begin{aligned} &p(\underbrace{0 \dots 0}_l | x_1 \dots x_n) \\ &= p(0 | \underbrace{0 \dots 0}_{l-1} x_1 \dots x_n) p(0 | \underbrace{0 \dots 0}_{l-2} x_1 \dots x_n) \dots p(0 | x_1 \dots x_n) \\ &= (1 - p(1 | \underbrace{0 \dots 0}_{l-1} x_1 \dots x_n)) (1 - p(1 | \underbrace{0 \dots 0}_{l-2} x_1 \dots x_n)) \\ &\dots (1 - p(1 | x_1 \dots x_n)) > (1 - \tilde{p})^l, \end{aligned}$$

то

$$\begin{aligned} \tilde{l} &= 1 + \frac{1}{l} - \sum_{x_1 \dots x_n} p(x_1 \dots x_n) p(\underbrace{0 \dots 0}_l | x_1 \dots x_n) < 1 + \frac{1}{l} - (1 - \tilde{p})^l \\ &\leq l\tilde{p} + \frac{1}{l} < \tilde{p} \left(\frac{1}{\sqrt{\tilde{p}}} + 1 \right) + \sqrt{\tilde{p}} = 2\sqrt{\tilde{p}} + \tilde{p} < 3\sqrt{\tilde{p}}. \end{aligned}$$

(Здесь мы воспользовались известным неравенством $(1+x)^n \geq 1+nx$ ($x > -1$).)

В этой последовательности a_1 является высоковероятной буквой \mathcal{M}_0 , а a_2, a_3 — маловероятными буквами $\mathcal{M}_1, \mathcal{M}_2$. Из (6) следует, что $\hat{p} = 1/7$, $l = 3$ и закодированная последовательность имеет вид

$$a_1 \ a_1 \ a_1 \ a_2 a_1 a_1 a_2 \ a_1 \ a_3 a_2 a_1 a_3 \ a_1.$$

Пусть теперь $z_1 z_2 \dots z_s$ — последовательность, полученная после первого этапа кодирования, $z_i \in A, A = \{a_1, a_2, \dots, a_n\}, n > 2$. Рассмотрим второй этап кодирования полученной последовательности, осуществляемый «быстрым» алгоритмом из [9]. Так как последовательность $z_1 z_2 \dots z_s$ не может рассматриваться как бернуллиевская, то для ее кодирования на втором этапе мы применим алгоритм, близкий к алгоритму из п. 1. В последовательности $z_1 z_2 \dots z_s$ выделим блоки длины l , которые следуют после появления какой-либо редкой буквы \mathcal{M}_k ($k = 1, \dots, m$), и особые буквы, не входящие в блоки. Иначе говоря, последовательность $z_1 z_2 \dots z_s$ представим в виде

$$\underline{\mathcal{M}_0} \dots \underline{\mathcal{M}_0} \underline{\mathcal{M}_k} \underbrace{z_1 \dots z_l}_l \underline{\mathcal{M}_0} \dots \underline{\mathcal{M}_0} \underline{\mathcal{M}_k} \underbrace{z_1 \dots z_l}_l$$

(в последовательности особые буквы подчеркнуты). Кодирование осуществляется следующим образом.

Особые буквы \mathcal{M}_0 и \mathcal{M}_k ($k = 1, \dots, m$) кодируются с помощью кодера K_0 с вероятностями \hat{q}^l и $1 - \hat{q}^l$ для \mathcal{M}_0 и \mathcal{M}_k соответственно.

Опишем кодирование букв, находящихся внутри блока $z_1 \dots z_l$ длины l . Пусть $z_1 \dots z_{i-1} = \underbrace{\mathcal{M}_0 \dots \mathcal{M}_0}_{i-1}$ ($i = 1, 2, \dots, l$). Вычислим вероятность буквы z_i , находящейся в i -й позиции после $i - 1$ появлений букв \mathcal{M}_0 . Имеем

$$\begin{aligned} \tau_i^k &= P\{z_i = \mathcal{M}_k \mid z_1 \dots z_{i-1} = \mathcal{M}_0 \dots \mathcal{M}_0; z_1 \dots z_l \neq \mathcal{M}_0 \dots \mathcal{M}_0\} \\ &= \frac{P\{z_i = \mathcal{M}_k; z_1 \dots z_{i-1} = \mathcal{M}_0 \dots \mathcal{M}_0 \mid z_1 \dots z_l \neq \mathcal{M}_0 \dots \mathcal{M}_0\}}{P\{z_1 \dots z_{i-1} = \mathcal{M}_0 \dots \mathcal{M}_0 \mid z_1 \dots z_l \neq \mathcal{M}_0 \dots \mathcal{M}_0\}} \\ &= \frac{p(\mathcal{M}_k)}{1 - \hat{q}^{l-i+1}}, \end{aligned}$$

так как

$$\begin{aligned} &P\{z_i = \mathcal{M}_k; z_1 \dots z_{i-1} = \mathcal{M}_0 \dots \mathcal{M}_0 \mid z_1 \dots z_l \neq \mathcal{M}_0 \dots \mathcal{M}_0\} \\ &= \frac{P\{z_i = \mathcal{M}_k; z_1 \dots z_{i-1} = \mathcal{M}_0 \dots \mathcal{M}_0; z_1 \dots z_l \neq \mathcal{M}_0 \dots \mathcal{M}_0\}}{P\{z_1 \dots z_l \neq \mathcal{M}_0 \dots \mathcal{M}_0\}} \\ &= \frac{p(\mathcal{M}_k) \hat{q}^{i-1}}{1 - \hat{q}^l} \end{aligned}$$

и

$$\begin{aligned}
 P\{z_1 \dots z_{i-1} = \mathcal{M}_0 \dots \mathcal{M}_0 \mid z_1 \dots z_l \neq \mathcal{M}_0 \dots \mathcal{M}_0\} \\
 = \frac{P\{z_1 \dots z_{i-1} = \mathcal{M}_0 \dots \mathcal{M}_0; z_1 \dots z_l \neq \mathcal{M}_0 \dots \mathcal{M}_0\}}{P\{z_1 \dots z_l \neq \mathcal{M}_0 \dots \mathcal{M}_0\}} \\
 = \frac{\hat{q}^{i-1} (1 - \hat{q}^{l-i+1})}{1 - \hat{q}^l}.
 \end{aligned}$$

Кроме того,

$$P\{z_i = \mathcal{M}_0 \mid z_1 \dots z_{i-1} = \mathcal{M}_0 \dots \mathcal{M}_0; z_1 \dots z_l \neq \mathcal{M}_0 \dots \mathcal{M}_0\} = 1 - \sum_{k=1}^m \tau_i^k.$$

Следовательно, буква z_i , находящаяся в i -й позиции после $i-1$ появлений букв \mathcal{M}_0 в блоке $z_1 \dots z_l$, кодируется с помощью кодера K_i с вероятностями τ_i^k для букв \mathcal{M}_k и $1 - \sum_{k=1}^m \tau_i^k$ для буквы \mathcal{M}_0 .

Наконец, буквы из блока $z_1 \dots z_l$, расположенные после появления в этом блоке какой-либо буквы \mathcal{M}_k , кодируются с помощью кодера \hat{K} с исходными вероятностями (т. е. с вероятностями \hat{q} для букв \mathcal{M}_0 и $p(\mathcal{M}_k)$ для букв \mathcal{M}_k).

Важно отметить, что вероятности τ_i^k не хранятся в памяти кодера и декодера (это потребовало бы слишком много памяти), а вычисляются рекуррентно (см. формулу (7)). Сначала кодируется z_1 с вероятностью τ_1^k и $1 - \sum_{k=1}^m \tau_1^k$ для букв \mathcal{M}_k и \mathcal{M}_0 соответственно. Если $z_1 = \mathcal{M}_k$, то все буквы, следующие после \mathcal{M}_k , кодируются с помощью кодера \hat{K} с исходными вероятностями $p(\mathcal{M}_k)$ и \hat{q} для \mathcal{M}_k и \mathcal{M}_0 соответственно. В противном случае вычисляется τ_2^k и буква z_2 кодируется с вероятностями τ_2^k и $1 - \sum_{k=1}^m \tau_2^k$ для \mathcal{M}_k и \mathcal{M}_0 соответственно. Иначе говоря, кодирование букв в блоке $z_1 \dots z_l$ осуществляется по следующей схеме.

Шаг 1. Вычисляется τ_i^k и буква z_i кодируется с вероятностью τ_i^k и $1 - \sum_{k=1}^m \tau_i^k$ для \mathcal{M}_k и \mathcal{M}_0 соответственно.

Шаг 2. Если $z_i = \mathcal{M}_k$, то все буквы, следующие после z_i , кодируются с вероятностями \hat{q} и $p(\mathcal{M}_k)$ для \mathcal{M}_0 и \mathcal{M}_k соответственно. Иначе осуществляется переход к следующей букве и возврат к шагу 1.

Для вычисления τ_i^k используем следующую рекуррентную формулу:

$$\frac{1}{\tau_{i+1}^k} = \frac{1}{\tau_i^k} - \frac{\hat{p}\hat{q}^{l-i}}{p(\mathcal{M}_k)},$$

где \hat{p} и \hat{q} определяются формулой (6). Значит, вычисление вероятностей τ_i^k можно организовать по схеме

$$\sigma := \sigma/q; \quad \hat{\tau}^{-1} := \hat{\tau}^{-1} - \sigma \quad (7)$$

с начальными данными

$$\sigma := \frac{\hat{q}^l \hat{p}}{p(\mathcal{M}_k)}, \quad \hat{\tau}^{-1} := \frac{1 - \hat{q}^l}{p(\mathcal{M}_k)}.$$

Таким же образом кодируются буквы следующего блока $z_1 \dots z_l$, причем перед каждым новым блоком начальные данные обновляются.

Отметим, что время вычисления τ_i^k не превышает $C \left(\log^2 \frac{1}{r} + \log^2 \frac{1}{p} \right)$, где C — константа (мы используем одно вычитание, одно умножение и два деления чисел длины $\log \frac{1}{r}$ бит).

Рассмотрим описанную выше схему кодирования на примере предыдущей последовательности, полученной после первого этапа. Пусть $p(a_1) = p(\mathcal{M}_0) = \hat{q} = 6/7$, $p(a_2) = p(\mathcal{M}_1) = 2/21$, $p(a_3) = p(\mathcal{M}_2) = 1/21$, $\hat{p} = 1/7$, $l = 3$ и кодируется последовательность

$$a_1 \ a_1 \ a_1 \ a_2 a_1 a_1 a_2 \ a_1 \ a_3 a_2 a_1 a_3 \ a_1.$$

Данную последовательность запишем в виде

$$\begin{array}{cccccccccccccc} \mathcal{M}_0 & \mathcal{M}_0 & \mathcal{M}_0 & \mathcal{M}_1 & \underbrace{\mathcal{M}_0 \ \mathcal{M}_0 \ \mathcal{M}_1} & \mathcal{M}_0 & \mathcal{M}_2 & \underbrace{\mathcal{M}_1 \ \mathcal{M}_0 \ \mathcal{M}_2} & \mathcal{M}_0 \\ z_1 & z_2 & z_3 & z_4 & z_5 & z_6 & z_7 & z_8 & z_9 & z_{10} & z_{11} & z_{12} & z_{13} \\ K_0 & K_0 & K_0 & K_0 & K_1 & K_2 & K_3 & K_0 & K_0 & K_1 & K & K & K_0. \end{array}$$

Тогда особые буквы кодируются с помощью кодера K_0 с вероятностями

$$\begin{aligned} p(z_1) = p(z_2) = p(z_3) = p(z_8) = p(z_{13}) &= (6/7)^3 = 216/343; \\ p(z_4) = p(z_9) &= 1 - (6/7)^3 = 127/343. \end{aligned}$$

Кодируется первый блок $z_5 z_6 z_7$. Из (7) следует, что

$$(\tau_5^1)^{-1} = \frac{1 - (6/7)^3}{2/21} = \frac{381}{98}, \quad (\tau_5^2)^{-1} = \frac{1 - (6/7)^3}{1/21} = \frac{381}{49}.$$

Значит, буква $z_5 = \mathcal{M}_0$ кодируется с помощью кодера K_1 с вероятностью $1 - \tau_5^1 - \tau_5^2 = 78/127$. Переходим к букве z_6 . Из (7) следует, что

$$(\tau_6^1)^{-1} = \frac{381}{98} - \frac{(6/7)^2 \cdot 1/7}{2/21} = \frac{39}{14}, \quad (\tau_6^2)^{-1} = \frac{381}{49} - \frac{(6/7)^2 \cdot 1/7}{1/21} = \frac{39}{7}.$$

Значит, $z_6 = \mathcal{M}_0$ кодируется с помощью кодера K_2 с вероятностью $1 - \tau_6^1 - \tau_6^2 = 6/13$.

Переходим к букве z_7 . Из (7) следует, что $(\tau_7^1)^{-1} = \frac{39}{14} - \frac{6 \cdot 7 \cdot 1/7}{2/21} = \frac{3}{2}$. Поэтому $z_7 = \mathcal{M}_1$ кодируется кодером K_3 с вероятностью $\tau_7^1 = 2/3$. Аналогично кодируются буквы второго блока $z_{10}z_{11}z_{12}$. Так как $(\tau_9^1)^{-1} = 381/98$, то буква $z_{10} = \mathcal{M}_1$ кодируется с помощью кодера K_1 с вероятностью $\tau_9^1 = 98/381$. Оставшиеся буквы z_{11} и z_{12} кодируются с помощью кодера \hat{K} с вероятностями $p(z_{11}) = p(\mathcal{M}_0) = 6/7$; $p(z_{12}) = p(\mathcal{M}_2) = 1/21$.

Таким образом, в памяти хранятся вероятности только трех кодеров K_0, K_i, \hat{K} , один из которых переменный, и (как и в известных методах [5, 6]) счетчики нулей объема $O\left(\log \frac{1}{\hat{p}}\right)$. Поэтому память кодера и декодера предложенного метода по порядку равна памяти общих методов. Под общими методами мы понимаем методы, предназначенные для кодирования любых источников (не только источников с малой энтропией). К ним, например, относятся методы, основанные на арифметическом коде [7, 8], коде из [9] и др.

Скорость кодирования и декодирования предложенного метода характеризуется

Теорема 2. Пусть даны бернуллиевский источник, порождающий буквы из алфавита $A = \{a_1, a_2, \dots, a_n\}$, $n > 2$, с известными вероятностями $p(a_1), p(a_2), \dots, p(a_n)$, и некоторое r , $r > 0$. Пусть для кодирования данного источника применяется описанный выше метод, в котором на первом этапе используется код с длиной блока $l = \left\lceil \frac{1}{\sqrt{\hat{p}}} \right\rceil$ ($\hat{p} < 1/2$), где \hat{p} определяется (6), а на втором этапе используется код с избыточностью $\bar{r} = r/2$. Тогда общая избыточность кода не превосходит r , а среднее время T кодирования и декодирования одной буквы удовлетворяет неравенству

$$T < C_1 \sqrt{\hat{p}} \log^3 \left(\frac{1}{r\hat{p}} \right) \log \log \left(\frac{1}{r\hat{p}} \right) + C_2,$$

где C_1 и C_2 — константы.

Доказательство. Кодирование осуществляется по схеме

$$S \longrightarrow S' \longrightarrow S'',$$

где S — слово длины l , а S', S'' — слова, полученные после первого и второго этапов кодирования соответственно. Пусть l_1, l_2 — средние длины кодовых слов S' и S'' ; H_0 — первоначальная энтропия, H_1 — энтропия после первого этапа кодирования. После проведения первого этапа полученную последовательность можно представить в виде нескольких бернуллиевских источников. В силу взаимной однозначности преобразования имеем

$$H_0 l = H_1 l_1.$$

Следовательно,

$$H_1 = H_0 \frac{l}{l_1}. \quad (8)$$

По определению избыточность \bar{r} , возникающая на втором этапе кодирования, определяется по формуле

$$\bar{r} = \frac{l_2}{l_1} - H_1, \quad (9)$$

а общая избыточность

$$R = \frac{l_2}{l} - H_0. \quad (10)$$

Используя (8)–(10), имеем

$$R = \frac{l_2}{l} - H_0 = \frac{l_2}{l_1} \frac{l_1}{l} - H_0 \frac{l}{l_1} \frac{l_1}{l} = \frac{l_1}{l} \left(\frac{l_2}{l_1} - H_0 \frac{l}{l_1} \right) = \frac{l_1}{l} \left(\frac{l_2}{l_1} - H_1 \right) = \frac{l_1}{l} \bar{r} = l' \bar{r},$$

где $l' = \frac{l_1}{l}$ — средняя длина кода (на букву исходного слова), полученного после первого этапа кодирования. Покажем, что

$$l' < 2\sqrt{\hat{p}} + \hat{p}. \quad (11)$$

Действительно, вероятность появления блока, состоящего только из букв \mathcal{M}_0 , равна $\hat{q}^l = (1 - \hat{p})^l$. Следовательно,

$$\begin{aligned} l' &= \frac{1}{l}(\hat{q}^l + (l+1)(1 - \hat{q}^l)) = 1 - \hat{q}^l + \frac{1}{l} \\ &= 1 - (1 - \hat{p})^l + \frac{1}{l} \leq l\hat{p} + \frac{1}{l} < \hat{p} \left(\frac{1}{\sqrt{\hat{p}}} + 1 \right) + \sqrt{\hat{p}} = 2\sqrt{\hat{p}} + \hat{p}, \end{aligned}$$

что совпадает с (11). При $\hat{p} < 1/2$ имеем

$$R = l' \bar{r} < \frac{r}{2} (2\sqrt{\hat{p}} + \hat{p}) < r,$$

т. е. общая избыточность не превосходит r .

Оценим среднее время кодирования и декодирования данного метода. Согласно (2) время кодирования, затрачиваемое на букву в «сжатой» на первом этапе последовательности, равно $O\left(\log^3\left(\frac{1}{R}\right) \log \log \frac{1}{R}\right)$. Кроме того, как отмечено выше, время вычисления величин τ_i^k , используемых на втором этапе, не превышает $C\left(\log^2 \frac{1}{r} + \log^2 \frac{1}{\hat{p}}\right)$. Умножив общее время второго этапа кодирования на среднюю длину кода l' и учитывая время первого этапа, равное $O(1)$, получим, что время T кодирования одной буквы предложенного метода удовлетворяет неравенству $T < C_1 \sqrt{\hat{p}} \log^3\left(\frac{1}{r\hat{p}}\right) \log \log \left(\frac{1}{r\hat{p}}\right) + C_2$, где C_1, C_2 — константы. Теорема 2 доказана.

ЛИТЕРАТУРА

1. Ахо А., Хопкрофт Дж., Ульман Дж. Построение и анализ вычислительных алгоритмов. М.: Мир, 1979.
2. Галлагер Р. Теория информации и надежная связь. М.: Наука, 1969.
3. Левенштейн В. И. Об избыточности и замедлении разделимого кодирования натуральных чисел // Проблемы кибернетики. М.: Наука, 1968. Вып. 20. С. 173–179.
4. Шарова М. П. Алгоритм быстрого кодирования низкоэнтропийных источников // Распределенная обработка информации: Тр. 6-го Междунар. семинара. Новосибирск: Ин-т физики полупроводников СО РАН, 1998. С. 421–424.
5. Шеннон К. Работы по теории информации и кибернетике. М.: Изд-во иностр. лит., 1963.
6. Elias P. Universal codeword sets and representation of the integers // IEEE Trans. Inform. Theory. 1975. V. 21, N 2. P. 195–203.
7. Rissanen J. Arithmetic coding as number representations // Acta Polytechnica. 1979. V. 31. P. 44–51.
8. Rissanen J., Langdon G. G. Universal modeling and coding // IEEE Trans. Inform. Theory. 1981. V. 27, N 1. P. 12–23.
9. Ryabko B. Ya. Fast and effective coding of information sources // IEEE Trans. Inform. Theory. 1994. V. 40, N 1. P. 96–99.
10. Ryabko B. Ya., Sharova M. P. Fast coding of low-entropy sources // IEEE Intern. Symp. on Inform. Theory (MIT, Cambridge, August 16–21, 1998). 1998. P. 44.

Адрес автора:

Институт математики
им. С. Л. Соболева СО РАН,
пр. Академика Коптюга, 4,
630090 Новосибирск, Россия.
E-mail: ir@net.ict.nsc.ru

Статья поступила

17 апреля 1998 г.,
переработанный вариант —
17 сентября 1998 г.