

Fast Enumerative Source Coding

Boris Ryabko

Novosibirsk Telecommunication Institute, Kirov st. 86, 630102, Russia

Abstract — The problem of enumerative coding was considered in [1] for the first time. By coding words of a length n the method from [1] has an encoding and decoding speed which equals to $0(n)$ when $n \rightarrow \infty$. We propose a code which has the high speed: $0(\log^2 n \log \log n)$, $n \rightarrow \infty$. This code is close to author's method from [2].

I. Introduction and the Main Idea

The problem of enumerative coding is well known in Information Theory and widely applied to retrieval problems and combinatorial analysis [1]. The suggested fast code uses the method from [2]. The simplest but important example of enumerative coding is the problem of translation numbers from one number system to another. We use this example for the description of the main idea of the proposed method. Let we have to translate the number $x_1 \dots x_n$ from the m -system ($m > 2$) to the binary system. A "common" method is based on well-known Horner scheme:

$$(1) \quad code(x_1 x_2 \dots x_n) = (\dots(x_1 m + x_2)m + x_3) \dots m + x_n$$

When we calculate in the binary - system, we obtain the value $x_1 \dots x_n$ in the binary - system. We shall assess the calculation time by the number of operations on single-bit words. We use the Schönhager- Strassen method of multiplication and division of numbers. For this method the time of multiplication of two numbers with L digits each, is equal to $O(L \log L \log \log L)$, $L \rightarrow \infty$. [3]. It is easy to see that the time for calculation by (1) is not less than cn^2 , $c > 0$, $n \rightarrow \infty$. Hence, the speed is not less than cn . We suggest computing by the scheme

$$(2) \quad code(x_1 \dots x_n) = ((\dots((x_1 m + x_2)(mm) + (x_3 m + x_4)) \\ ((mm)(mm)) + (x_5 m + x_6)(mm) + (x_7 m + x_8) \dots)$$

In this case the main part of multiplications will be implemented on comparatively small numbers and when (2) is used, the time for computing is equal to $O(n \log^2 n \log \log n)$, $n \rightarrow \infty$ and the encoding speed is equal to $O(\log^2 n \log \log n)$... We can see that "proper" arrangement of brackets allows to decrease the calculation time essentially. It is worthy of noting that the described method is known as "divide and conquer" principle [3].

II. Main Result

We use definitions from [1]. Let $A = \{0, 1, \dots, m-1\}$ be an alphabet of m letters, $m \geq 2$, A^n be the set of all

words of length n over the alphabet A . Let an arbitrary $S \subset A^n$ be a source. Let's give the lexicographic order to words S , and for the integer $1 \leq k \leq n$ and for the word $x_1 \dots x_k \in A_k$, denote by $N_s(x_1 \dots x_k)$ the quantity of words produced by S and having the prefix $x_1 \dots x_k$. In [1] the code by formula

$$(3) \quad code(x_1 \dots x_n) = \sum_{i=1}^n \sum_{a < x_i} N_s(x_1 \dots x_{i-1} a)$$

was proposed. Let's define for $x_1 \dots x_n \in S$.

$$(4) \quad P(x_1) = N_s(x_1)/|S|, \quad P(x_k/x_1 \dots x_{k-1}) = \\ N_s(x_1 \dots x_k)/N_s(x_1 \dots x_{k-1}), k = 2, \dots, n$$

$$(5) \quad q(x_k/x_1 \dots x_{k-1}) = \sum_{a < x_k} P(a/x_1 \dots x_{k-1}), k = 1, \dots, n$$

From (3), (4), (5) it is easy to obtain

$$(6) \quad code(x_1 \dots x_n) = |S|(q(x_1) + q(x_2/x_1)P(x_1) + \\ q(x_3/x_1 x_2)P(x_1)P(x_2/x_1) + \dots)$$

The scheme of the proposed method is following: Each $P(x_k/x_1 \dots x_{k-1})$, $q(x_k/x_1 \dots x_{k-1})$, $x_1 \dots x_k \in A^k$ can be written in the form of a word with $2 \log n + 0(1)$ digits. Then (6) resulted in the form

$$code(x_1 \dots x_n) = |S|(((q(x_1) + q(x_2/x_1)P(x_1)) + \\ (P(x_1)P(x_2/x_1))(q(x_3/x_1 x_2) + q(x_4/\dots)P(x_3/\dots)) + \\ ((P(x_1)P(x_2/x_1))(P(x_3/\dots)P(x_4/\dots)))(q(x_5/\dots) + \\ q(x_6/\dots)P(x_5/\dots) \dots)$$

(Here we used "proper" arrangement of brackets, as if we go over to (2) from (1)). Decoding is constructed similarly, by using division. It is easy to calculate that the encoding and decoding speed is equal to $O(\log^3 n \log \log n)$ when $n \rightarrow \infty$.

References

- [1] T.M.Cover, "Enumerative source encoding", *IEEE Trans. Inf. Theory*, vol. 19, n 1, pp. 73-77, 1973.
- [2] B.Ya.Ryabko, "Fast and effective coding of information sources", *IEEE Trans. Inf. Theory*, vol. 30, n 1, pp.96-99, 1994.
- [3] A.V.Aho, J.E.Hopcroft, J.D.Ullman, "The design and analysis of computer algorithms", *Addison-Wesley*, 1976.