

REPETITION TIMES AND UNIVERSAL DATA COMPRESSION

Frans M.J. Willems*

A new universal data compression algorithm is described. This algorithm encodes L source symbols at a time. The code alphabet is binary. For the class of binary stationary sources, the efficiency is shown to be not more than $[H(u_0, u_1, \dots, u_{L-1}) + \text{ceil}(\log(L+1))]/L$ bits per source symbol.

In the analysis of our algorithm a result on repetition times turns out to be crucial.

The algorithm can be generalized to arbitrary source and arbitrary code alphabet sizes.

I. INTRODUCTION

In a data compression situation an encoder observes the output stream of an information source and transforms it into a code stream. This code stream is sent to a decoder, which is capable of reconstructing the source stream by interpreting the code stream. The efficiency (the rate) of such a system is defined as the expected number of code symbols per source symbol.

If the source statistics are known, we can design encoder-decoder pairs with the efficiencies arbitrarily close to the entropy of the source. In addition it can be shown that efficiencies smaller than source entropy can not be achieved.

A data compression algorithm is called universal if the corresponding encoder and decoder are designed without knowing the source statistics. Such a universal compression technique is optimal if there exists encoder-decoder pairs, designed accordingly, with efficiencies arbitrarily close to source entropy, no matter what the actual source is.

Here we present an optimal universal data compression method for binary stationary sources and a binary code alphabet.

*Frans M.J. Willems is with the Department of Electrical Engineering, Eindhoven University of Technology, Den Dolech 2, P.O.Box 513, 5600 MB Eindhoven, The Netherlands.

II. STATEMENT OF RESULT

Consider a binary source that produces an output every time unit. Hence it generates $\{u_t\}_{t=-\infty}^{\infty}$, i.e. a sequence of source outputs with values in the alphabet $\{0,1\}$. The index t takes on integer values and can be identified with time. Throughout this report we assume that the source is stationary (for definitions we refer to Gallager [1], par. 3.5.).

The encoder chops the output stream into pieces $u_k^L \triangleq (u_{(k-1)L}, \dots, u_{(k-1)L+1}, \dots, u_{kL-1})$ of length L , with k integer. At time $t=kL$ the source word u_k^L is transformed into a codeword c_k^* of length $\lg(c_k^*)$ code symbols. These code symbols take on values also in the (binary) alphabet $\{0,1\}$. The encoder uses the M most recent source symbols. They are assumed to be stored in a buffer. More precisely let this buffer at $t=kL$ contain $\phi_k \triangleq (u_{(k-1)L-M}, u_{(k-1)L-M+1}, \dots, u_{(k-1)L-1})$ then the encoder forms

$$c_k^* = E(u_k^L, \phi_k). \quad (1)$$

The decoder uses an identical buffer and it is assumed that at $t=-\infty$ the contents of this buffer and the one used in the decoder are equal. At $t=kL$ the decoder receives c_k^* and reconstructs

$$u_k^L = D(c_k^*, \phi_k). \quad (2)$$

Subsequently, using ϕ_k and u_k^L the decoder forms ϕ_{k+1} .

To keep the coding delay small, we require that the set of codewords produced by the encoder satisfies the prefix condition (see Gallager [1], par. 3.2). In (1) and (2) it is implicitly stated that for all source words u^L and buffer contents ϕ

$$u^L = D(E(u^L, \phi), \phi) \quad (3)$$

should hold. We say that the code used by the encoder and the decoder has to be uniquely decodable.

The efficiency R of the described encoder-decoder pair is defined as

$$R \triangleq E(\lg(c_k^*)) / L, \quad (4)$$

where the integer k is arbitrarily. The expectation in (4) is evaluated using the statistics of the source being compressed.

In section IV we describe and analyse an encoder-decoder pair and we are able to prove that for each binary stationary source

$$RL \leq H(U_0, U_1, \dots, U_{L-1}) + \text{ceil}(\log(L+1)). \quad (5)$$

All logarithms in this manuscript unless stated otherwise are assumed to have base 2. For the size of the buffers we find that it suffices to take

$$M = 2^L - 1. \quad (6)$$

Note that essentially our coding strategy is of the fixed-to-variable type.

It is well known (see Gallager [1], par. 3.3) that in our situation when the source is stationary

$$RL \geq H(U_0, U_1, \dots, U_{L-1} | U_{-M}, U_{1-M}, \dots, U_{-1}). \quad (7)$$

Also it is clear that (see again Gallager [1], par. 3.5), because of (5) and (7),

$$\lim_{L \rightarrow \infty} R = H_{\infty}(U), \quad (8)$$

where $H_{\infty}(U)$ is the entropy of the (stationary) source. It is (8) that makes our universal method optimal.

A crucial point in our argumentation is a result on repetition times. The next section is devoted to this subject.

III. REPETITION TIMES

A source generates $\dots, x_{-2}, x_{-1}, x_0, x^1, x^2, \dots$ with $x_t \in A_x$, a finite alphabet. We assume that this source is stationary.

Let A_x^+ be the subset of A_x that contains all x with $P(X_0=x) > 0$. Now for $m = 1, 2, 3, \dots$ and $x \in A_x^+$ we define

$$Q_m(x) \triangleq P(X_{-m}=x, X_{1-m} \neq x, X_{2-m} \neq x, \dots, X_{-1} \neq x | X_0=x). \quad (9)$$

and

$$T(x) = \sum_{m=1, \infty} m Q_m(x), \quad (10)$$

where it is understood that

$$\sum_{n=1, \infty} a_n \triangleq \lim_{N \rightarrow \infty} \sum_{n=1, N} a_n. \quad (11)$$

In this section we state the following theorem.

THEOREM: For a discrete stationary source, for $x \in A_x^+$

$$\sum_{m=1, \infty} Q_m(x) = 1, \text{ and} \quad (a)$$

$$P(X_0=x)T(x) = 1 - \lim_{N \rightarrow \infty} P(X_0 \neq x, X_1 \neq x, \dots, X_N \neq x). \quad (b)$$

PROOF: The proof of this theorem is not given here.

IV. THE ALGORITHM

We start the description of our algorithm by introducing the following concept.

The L-th ordered derived source of the source $\{u_t\}_{t=-\infty}^{\infty}$ is defined as the source that generates $\{v_t\}_{t=-\infty}^{\infty}$ with $v_t \triangleq (u_{t-L}, u_{t-L+1}, \dots, u_{t-1})$. Without proof we give the following lemma.

LEMMA: The L-th order derived source of a stationary source is stationary. (end)

It is important to note that this lemma implies that the theorem in section III holds for the L-th order derived source. We will now describe the encoding process of our universal algorithm.

Let $t=kL$, hence $v_t = u_k^L = (u_{t-L}, u_{t-L+1}, \dots, u_{t-1})$ is being encoded. The buffer now contains $\phi_k = (u_{t-L-M}, u_{t-L-M+1}, \dots, u_{t-L-1})$ with $M = 2^L - 1$.

Note that using ϕ_k and u_k^L the encoder can form (has access to) v_{t-m} with $1 \leq m \leq M$. With these L-vectors the encoder determines the integer m_k . This m_k is set equal to the smallest m , $1 \leq m \leq 2^L - 1$, for which

$$v_{t-m} = v_t. \quad (12)$$

If such an m_k can not be found set $m_k = M+1 = 2^L$. From the above it follows that $m_k \in S \triangleq \{1, 2, \dots, 2^L\}$.

We now assume that S is partitioned in $L+1$ subsets. These subsets S_p , $p=0, 1, 2, \dots, L$, are defined as follows

$$\begin{aligned} S_p &\triangleq \{2^p, 2^{p+1}, \dots, 2^{p+1}-1\}, \text{ for } p=0, 1, 2, \dots, L-1 \text{ and} \\ S_L &\triangleq \{2^L\}. \end{aligned} \quad (13)$$

Note that S_p for $p=0, 1, 2, \dots, L-1$ contains 2^p elements.

Next suppose that $m_k \neq 2^L$. Then, using the subsets of S , it is possible to assign to each m_k a subset number p which indicates that $m_k \in S_p$, and a member index q which is defined as

$$q \triangleq m - 2^p. \quad (14)$$

After having determined m_k , the encoder constructs a codeword $c_k^*(m_k)$.

If $m_k \neq 2^L$ the codeword c_k^* is obtained by concatenating the subset number p and the member index q of m_k , both in radix-2 notation. For the subset number $\text{ceil}(\log(L+1))$ binary digits are needed, for the member index p binary digits. Hence if $m_k \neq 2^L$ (this means that u_k^L appears somewhere in the buffer),

$$\lg(c_k^*) = \text{bot}(\log(m_k)) + \text{ceil}(\log(L+1)). \quad (15)$$

If $m_k = 2^L$ (this corresponds to the situation where no match for u_k^L is found in the buffer), the codeword c_k^* is obtained by concatenating the subset number L and the source word u_k^L , the subset number in radix-2 notation. Now for the subset number again $\text{ceil}(\log(L+1))$ binary digits are needed and for the source word L digits. Hence for $m_k = 2^L$,

$$\lg(c_k^*) = L + \text{ceil}(\log(L+1)). \quad (16)$$

One easily verifies that the decoder after having received c_k^* can reconstruct u_k^L . Also note that the codewords emitted by the encoder satisfy the prefix condition.

We will now analyse the described algorithm. Suppose that $v_t (= u_k^L) = v$ is the codeword being encoded at $t=kL$. Now what is the average length $L(v)$ of the codeword assigned to it? We extrapolate the notation of (9) somewhat and obtain

$$\begin{aligned} L(v) &= \sum_{m=1, 2^{L-1}} Q_m(v) [\text{bot}(\log(m)) + \text{ceil}(\log(L+1))] \\ &\quad + \sum_{m=2^L, \infty} Q_m(v) [L + \text{ceil}(\log(L+1))] \\ &\leq \sum_{m=1, \infty} Q_m(v) [\log(m) + \text{ceil}(\log(L+1))] \end{aligned}$$

$$\begin{aligned}
& \text{(a)} \\
& = \sum_{m=1, \infty} Q_m(v) \log(m) + \text{ceil}(\log(L+1)) \\
& \text{(b)} \\
& \leq \log \left[\sum_{m=1, \infty} m Q_m(v) \right] + \text{ceil}(\log(L+1)) \\
& = \log(T(v)) + \text{ceil}(\log(L+1)) \\
& \text{(c)} \\
& \leq -\log(P(v_t=v)) + \text{ceil}(\log(L+1)). \tag{17}
\end{aligned}$$

Here (b) follows from the (a)-part of the theorem in section III, (b) from the convexity of the log function and (c) from the (b)-part of this theorem. Note that throughout the derivation (17) we have used the fact that $P(v_t) > 0$. Fortunately only those v appear in the source output stream as v_t ($=u_k^L$).

Using (17) we can now upperbound the efficiency of our system:

$$\begin{aligned}
RL &= \sum_{v: P(v_t=v) > 0} P(v_t=v) L(v) \\
&\leq \sum_{v: P(v_t=v) > 0} P(v_t=v) [-\log(P(v_t=v)) + \text{ceil}(\log(L+1))] \\
&= H(V) + \text{ceil}(\log(L+1)) \\
&= H(U_0, U_1, \dots, U_{L-1}) + \text{ceil}(\log(L+1)), \tag{18}
\end{aligned}$$

where we have obeyed the convention that $0 \log(0) = 0$. This concludes the proof of the result announced in section II.

V. CONCLUSION AND REMARKS

We conclude that our algorithm is easy to implement and that its minimax redundancy with respect to $H(U_0, U_1, \dots, U_{L-1})$ instead of $LH_\infty(U)$ is acceptable for stationary sources.

The algorithm can be generalized to arbitrary source and code alphabet sizes.

The author was motivated by a number of very interesting papers in the field of universal source coding. These papers are well known and need not be referred to here.

REFERENCE

- [1] R.G.GALLAGER, *Information Theory and Reliable Communication*, New York: Wiley, 1968.