HOMOPHONIC SEQUENCE SUBSTITUCION

Valdemar C. da Rocha Jr.* and Hélio M. de Oliveira.
Communications Research Group - CODEC
Department of Electronics and Systems, P.O. Box 7800
Federal University of Pernambuco
50711-970 Recife PE

Resumo - Substituição homofônica de seqüências é o nome dado neste trabalho para a técnica que consiste em substituir um-a-um uma dada seqüência de símbolos, finita ou semi-infinita, por outra seqüência, respectivamente finita ou semi-infinita, sobre o mesmo alfabeto, porém com uma taxa de entropia mais elevada. A seqüência de saída de uma dada fonte discreta, estacionária e ergódica, é codificada com um código de fonte binário sem perdas C. Uma concatenação de palavras código de C é então convenientemente segmentada e recodificada com um código de fonte binário sem perdas Iterando um certo número de vezes o último passo descrito acima, prova-se que a taxa de entropia da seqüência na saída do último codificador aproxima-se do valor 1, assintoticamente, e portanto realizando a substituição homofônica ótima. A redundância remanescente, após k codificações consecutivas, é $1-H_k(S)$ bits por dígito binário, onde $H_k(S)$ denota a taxa de entropia da seqüência resultante após a k-ésima codificação. Um modelo de fonte de Markov é apresentado para descrever as seqüências binárias codificadas e para computar as respectivas taxas de entropia.

Abstract - Homophonic sequence substitution is the name given in this paper to the technique which consists of substituting one-to-one a given finite (or semi-infinite) sequence of symbols by another finite (or semi-infinite) sequence over the same alphabet but having a higher entropy rate. The output sequence of a given discrete stationary and ergodic source is encoded with a binary lossless source code C. A concatenation of codewords of C is then conveniently parsed and reencoded with a binary lossless source code. By iterating the latter step a number of times, it is proved that the entropy rate of the binary sequence at the output of the last encoder approaches the value 1 asymptotically, therefore performing optimum homophonic sequence substitution. The remaining redundancy, after k consecutive encodings, is $1-H_k(S)$ bits per binary digit, where $H_k(S)$ is the entropy rate of the binary sequence resulting after the k^{th} encoding. A Markov source model is presented to describe the binary encoded sequences and to compute their entropy rate.

Keywords:source coding, homophonic substitution, Markov sources, Huffman coding.

1. INTRODUCTION

Source coding is a technique whose aim is to represent the output of an information source with as few code digits per source symbol as possible. In this paper we will consider only *lossless source coding* in which case it is possible to reconstruct *exactly* the source output from its encoded representation. We will concentrate our attention on binary coding both for its practical importance and because the generalizations to higher order alphabets are immediate. We will consider the problem of removing redundancy of a message sequence with an alternative, and perhaps complementary, approach to that in [1]. The distinguishing feature of our approach is that we neither resort to intentional plaintext expansion, as in conventional (symbol) homophonic substitution [1], nor to coding extensions of the original source, as suggested by Shannon's lossless source coding theorem [2, p.69]. In Section 2 we present basic notions of source coding and briefly review the main properties of uniquely decodable codes. In Section 3 we define the Markov source associated with a rooted tree with probabilities [3] and consider encoding the output of such a source with a Huffman code. In Section 4 we introduce *alternate* Huffman codes and give an example. Following [1] we will call a sequence of D-ary random variables *completely random* if each of its digits is statistically independent of the preceding digits and is equally likely to take on any of the D possible values. Finally, in Section 5 we show how to perform *homophonic sequence substitution* and prove that a cascade consisting of Markov sources encoded by lossless source codes produces in the limit a completely random sequence. The decoding operation is simple and consists of applying the received encoded binary sequence through a cascade of k look-up tables (corresponding to the number k of iterations used for encoding), where the i^{th}, $1 \leq i \leq k$, look-up table in the cascade is a decoder for the $(k+1-i)^{th}$ code. Contrasting with coding extensions of a source, where there is no control at all on the implementation complexity, our approach gives more flexibility in controlling both the redundancy and the implementation complexity. Furthermore, contrasting with conventional homophonic substitution, in our approach there is no cleartext expansion caused by the iterations. Of course the binary sequence after the k^{th} encoding will still have some redundancy (measured in bits) which is equal to $1-H_k(S)$ bits per binary digit, where $H_k(S)$ is the entropy rate of the binary sequence after the k^{th} encoding.

2. SOURCE CODING FUNDAMENTALS

Let $U_1, U_2, \ldots$, denote the output sequence of symbols of a discrete information source. This source is said to be *stationary* if, for every positive integer L and every se-

*The research of this author was supported in part by the Brazilian National Council for Scientific and Technological Development (CNPq) under Grant No. 304214/77-9.

quence $u_1, u_2, \ldots, u_L$ of letters from the source alphabet we have $P(U_1, U_2, \ldots, U_L = u_1, u_2, \ldots, u_L) = P(U_{i+1}, U_{i+2}, \ldots, U_{i+L} = u_1, u_2, \ldots, u_L)$, for all $i \geq 0$. A stationary source is said to be *ergodic* if the number of times that the sequence $u_1, u_2, \ldots, u_L$ occurs within the source output sequence $U_1, U_2, \ldots, U_{N+L-1}$ of length $N+L-1$, when divided by N, equals $P(U_1, U_2, \ldots, U_L = u_1, u_2, \ldots, u_L)$ with probability 1 as $N \rightarrow \infty$ [4]. In the sequel we will considerer only *discrete stationary and ergodic sources* (DSES) since they are general enough to model any real information source. The source codes employed in lossless source coding are called *uniquely decodable codes* [5, p.48]. A sufficient condition for the unique decodability of a concatenation of codewords is that the encoding be *prefix-free*, i.e., that no codeword be the first part (prefix) of another codeword. This prefix-free condition is equivalent to the condition that a decoder be able to immediately recognize the end of a codeword without need to read the beginning of the next codeword. Codes with this property are called *instantaneous codes* [5, p.50]. A uniquely decodable code is further said to be a *compact code* [5, p.66] whenever its average codeword length is equal to or less than the average codeword length of all other uniquely decodable codes for the same source and the same code alphabet.

Shannon's lossless source coding theorem [2, p.69] implicitly suggests that the way for reducing redundancy in a message to be transmitted or stored is by performing data compression. As the cryptographic community very well knows that is not necessarily the case however, as exemplified by homophonic substitution. *Homophonic substitution* is a cryptographic technique for reducing the redundancy of a message to be enciphered at the cost of plaintext expansion. This definition concerns homophonic *symbol* substitution [1], however iterative source coding can be seen as a form of homophonic substitution (*homophonic sequence substitution*) and not necessarily leads to cleartext expansion.

3. ROOTED TREES AND MARKOV SOURCES

Very often we are interested in determining the probability of single binary digits, or pairs of binary digits, etc., produced by a source code driven by a source. It turns out that the computation of these probabilities, directly from the code rooted tree with probabilities [3], is possible but becomes very complicated as the order of the statistics considered increases. We found a neater way for calculating these probabilities by defining a representation of the code rooted tree with probabilities by a Markov source. We define the Markov source whose *states* correspond one-to-one to the nodes of the code tree, whose *branches* are labeled with the same binary numbers as those in the corresponding branches of the code tree and each state *transition probability* is given by the conditional probability of emitting a 0 (or a 1) given the current state (or node in the code tree). A return to state σ_I occurs always after the last digit of a codeword is generated by the encoder. Let $|\mathcal{S}|$ denote the number of states in a given Markov source. The probability P_{σ_i}, $i = 1, 2, \ldots, |\mathcal{S}|$, of

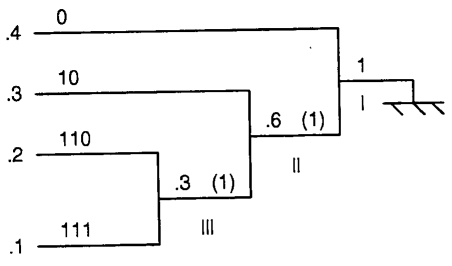

Figure 1: Huffman tree.

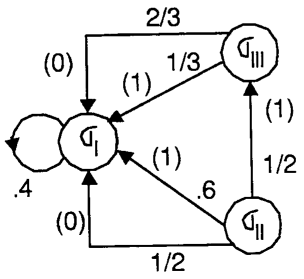

Figure 2: Markov source model.

every state σ_i is equal to the probability of the corresponding tree node P_i divided by the average codeword length [6]. We give next an example to clarify the above description of a Markov source, where the code employed is a Huffman code.

Example 1
Let S denote a discrete source with a four symbol alphabet whose probabilities are .4, .3, .2 and .1, respectively. We show in Figure 1 the Huffman tree and in Figure 2 the corresponding Markov source for the given discrete source.

3.1. Probability computation

In order to simplify the representation of the operations to be performed to compute probabilities in a Markov source, we will use matrices as follows. We will denote by $\mathbf{P}(l)$, $l \in \{0, 1\}$, the $|\mathcal{S}| \times |\mathcal{S}|$ matrix whose $(i,j)^{th}$ entry, denoted as $P_{ij}(l)$, is the branch probability of going from state σ_i to state σ_j.

Example 2
Continuing with *Example 1*, we have the following matrix representation for the transition probabilities.

$$\mathbf{P}(0) = \begin{bmatrix} 0.4 & 0 & 0 \\ 0.5 & 0 & 0 \\ 2/3 & 0 & 0 \end{bmatrix} \qquad \mathbf{P}(1) = \begin{bmatrix} 0 & 0.6 & 0 \\ 0 & 0 & 0.5 \\ 1/3 & 0 & 0 \end{bmatrix}$$

The code average codeword length is 1.9 and thus the states have the following probabilities: $P(\sigma_1) = 1/1.9$, $P(\sigma_2) = .6/1.9$ and $P(\sigma_3) = .3/1.9$. As an example we consider next the computation of $P(01)$, i.e., the probability of a zero ocurring, followed by a one.

$$\mathbf{P}(01) = [P_{\sigma_1} P_{\sigma_2} P_{\sigma_3}]\mathbf{P}(0)\mathbf{P}(1)[111]^T$$

$$\mathbf{P}(0)\mathbf{P}(1) = \begin{bmatrix} 0.4 & 0 & 0 \\ 0.5 & 0 & 0 \\ 2/3 & 0 & 0 \end{bmatrix} \begin{bmatrix} 0 & 0.6 & 0 \\ 0 & 0 & 0.5 \\ 1/3 & 0 & 0 \end{bmatrix}$$

$$= \begin{bmatrix} 0 & 0.24 & 0 \\ 0 & 0.3 & 0 \\ 0 & 0.4 & 0 \end{bmatrix}$$

Thus,

$$\mathbf{P}(01) = \begin{bmatrix} P_{\sigma_1} & P_{\sigma_2} & P_{\sigma_3} \end{bmatrix} \begin{bmatrix} 0 & 0.24 & 0 \\ 0 & 0.3 & 0 \\ 0 & 0.4 & 0 \end{bmatrix} \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}$$

$$= \begin{bmatrix} .284 \end{bmatrix}$$

The probability $P(a_1, a_2, \ldots, a_n)$ of the binary n-tuple $a_1, a_2, \ldots, a_n$ occurring is computed (in this example) from the following expression.

$$\mathbf{P}(a_1, a_2, .., a_n) = [P_{\sigma_1} P_{\sigma_2} P_{\sigma_3}]\mathbf{P}(a_1)\mathbf{P}(a_2)..\mathbf{P}(a_n)[111]^T$$

The extension of the above expansion for the general case is immediate.

4. ALTERNATE BINARY HUFFMAN CODES

As far as source specific codes for source coding are concerned Huffman codes are *compact* in the sense that a Huffman code for a specific DSES has an average codeword length equal to or less than the average codeword length among all instantaneous codes for that source [5, p.77] with the same code alphabet. We notice the well known fact that for a given DSES, in general, we can construct more than one Huffman code, but that all such codes have the same average codeword length.

In the construction of a binary Huffman code, or equivalently, a binary Huffman tree, whenever two subtrees stem out from a node a decision has to be made as to which subtree we should label with a 0 and to which subtree we should label with a 1. Whenever that decision is arbitrary the resulting Huffman code is called an *arbitrary* Huffman code [8]. Whenever the subtree of higher total probability is always labeled with a 0, the resulting code is called a *0-heavy* Huffman code. We introduce next a third case of interest that we call *alternate* Huffman coding. Starting from the root, whenever two subtrees stemming out from the same node have identical probabilities we arbitrarily label one of them with a 0 and the other with a 1. At the first node whose two subtrees stemming out have different probabilities, we label with a 0 the subtree of higher probability and keep a record of that fact. At the next node whose two subtrees stemming out have different probabilities we label with a 1 the subtree of higher probability. This procedure is applied over and over until the tree is traversed. Summarizing, this subtree labeling rule keeps a record of which label was given to the subtree of higher probability at the last node visited whose associated subtrees had different probabilities and *alternates* that labeling for the next node whose associated subtrees have different probabilities. We illustrate with a simple example the usefulness of alternate Huffman coding.

Example 3

Consider the source of *Example 1*. We present in *Table 1* the alternate code and the 0-heavy code for this source.

Probability	Alternate code	0-heavy code
.4	0	1
.3	10	00
.2	110	010
.1	111	011

Table 1: Alternate and 0-heavy codes for the source of *Example 1*.

The entropy per binary digit of the associated Markov source model is identical for both codes and its value is 1.8565. In *Table 2* we present first order and second order statistics for both codes. By computing the absolute value of the difference between each one of the statistics in the table and the corresponding value for a completely random source, and then adding the results we see that the alternate code produces a smaller sum and thus its digits are more *random looking* than those produced by the 0-heavy code. The *divergence* [7] could also be used as a convenient measure of the distance between a given probability distribution and that of a completely random source. Again the results favor alternate codes versus 0-heavy codes.

	Alternate code	0-heavy code
P(0)	.474	.579
P(00)	.189	.315
P(01)	.284	.263
P(10)	.284	.263
P(11)	.242	.159

Table 2: First order and second order statistics.

Definition: A uniquely decodable code is *optimum* if it is both compact and its symbol statistics is the closest to that of a completely random sequence among all compact codes for that source.

5. ITERATIVE PROCEDURE

Let S denote a DSES encoded using a compact binary prefix-free code. We chose to use an alternate Huffman code

C_1 with average codeword length L_1 for that purpose. The iterative procedure for performing homophonic sequence substitution consists of parsing a concatenation of codewords of C_1 in blocks of r digits forming a source S_1 with 2^r symbols. S_1 is then encoded with an alternate binary Huffman code C_2 with average codeword length L_2. A concatenation of codewords of C_2 is then parsed in blocks of r digits forming a source S_2 with 2^r symbols. This procedure is continued in a manner that at the i^{th} step, a concatenation of codewords of C_i is parsed into blocks of r digits forming a source S_i with 2^r symbols. As we prove in *Theorem (5..1)*, at each new step the entropy of the resulting binary sequence is increased, if not, the block size in the parsing is increased to $r+1$ and the procedure is repeated. A stopping rule will specify for a given application that, starting with $r=2$, the number of steps k is given by the smallest k for which $1 - H_k(S) \leq \epsilon$, where $\epsilon \ll 1$ is a small positive quantity. Our proof is more general then needed for the iterative procedure for it employs a general lossless code at no extra increase in difficulty.

Theorem 5..1 *Let S denote an entropy $H(S)$ DSES whose output is encoded by a binary lossless code C_1 with average codeword length L_1. Let us parse a concatenation of codewords of C_1 in blocks of r digits forming a source S_1 with 2^r symbols. We encode S_1 with a lossless code C_2, etc., and proceed as described above. The entropy rate $H_k(S)$ of the coded sequence at step k is greater than or equal to the entropy rate $H_{k-1}(S)$ of the coded sequence at step $k-1$, $k=2,3,\ldots$.*

Proof. Let $H_k(S)$ denote the entropy rate of the binary sequence generated by a concatenation of codewords of C_k, $k=1,2,\ldots$ (starting with C_1 driven by S). It is well known that $H_1(S) = H(S)/L_1$, Reference [11], and that $H(S_1) = rH_1(S)$. It follows that

$$H_2(S) = H(S_1)/L_2 = rH_1(S)/L_2 \geq H_1(S),$$

where the inequality follows from the observation that $L_2 \leq r$ is an upperbound for the average codeword length of a binary lossless code for a source with 2^r symbols. Proceeding with the iterations we obtain at the k^{th} step that $H_k(S) \geq H_{k-1}(S)$, $k=2,3,\ldots$. As we proceed with the iterations a step will be reached where the lossless code specified for the source with r symbols will have all codewords with the same length. That source is no longer an ergodic Markov source but instead it is a *periodic* Markov source [2, p.65]. Whenever such a situation happens we may consider repeating that step, however parsing then with a block size of at least $r+1$ symbols and proceed in the same manner or simply to stop. Since the property of increasing entropy of coded sequences is valid for source extensions of any order, it follows that the entropy rate $H_k(S)$ of the k^{th} coded sequence tends in the limit to 1 as the number k of iterations grows. ■

ACKNOWLEDGEMENT

We are grateful to our colleague Dr. Cecílio Pimentel for helpful discussions.

REFERENCES

[1] H.N. Jendal, Y.J.B. Kuhn and J.L. Massey, "An information-theoretic treatment of homophonic substitution", *Advances in Cryptology - Eurocrypt'89* (Eds. J.-J. Quisquater and J. Vandewalle), Lecture Notes in Computer Science, No. 434, Heidelberg and New York: Springer, 1990, pp.382-394.

[2] R.G. Gallager, *Information Theory and Reliable Communication*, John Wiley and Sons., Inc., Ney York, 1968.

[3] J.L. Massey, "Applied Digital Information Theory", *Fach Nr. 35-417 G, 7. Semester*, Class notes at the ETH Zurich, Chapter2, Wintersemester 1988-1989.

[4] J.L. Massey, "Some applications of source coding in cryptography", *Proc. 3rd Symp. on State and Progress of Research in Cryptography*, Rome, 1993, pp. 143-160.

[5] N. Abramson, *Information Theory and Coding*, McGraw Hill, New York, 1963.

[6] V.C. da Rocha Jr. and H.M. de Oliveira, "The entropy of a code with probabilities", Int. Telecom. Symposium, 9-13 August 1998, São Paulo.

[7] R.E. Blahut, *Principles and Practice of Information Theory*, Addison-Wesley Publishing Company, Reading, Massachusetts, 1987.

[8] D.W. Gillman, M. Mohtashemi and R.L. Rivest, "On breaking a Huffman code", *IEEE Trans. on Inform. Theory*, vol.42, no.3, May 1996, pp.972-976.

[9] C.E. Shannon, "Communication Theory of Secrecy Systems", *Bell System Tech. J.*, vol.28, pp.656-715, Oct., 1949.

[10] B.L. Montgomery and B.V.K.V. Kumar, "On the average codeword length of optimal binary codes for extended sources", *IEEE Trans. on Inform. Theory*, vol.33, no.2, March 1987, pp.293-296.

[11] B.L. Montgomery, H. Diamond and B.V.K.V. Kumar, "Bit probabilities of optimal binary source codes", *IEEE Trans. on Inform. Theory*, vol.36, no.6, November 1990, pp.1446-1450.

[12] P. Sen, "On noiseless source coding with specified encoder output symbol frequencies", *IEEE Trans. on Inform. Theory*, vol.30, no.5, September 1984, pp.752-754.

[13] G. Longo and G. Galasso, "An application of informational divergence to Huffman codes", *IEEE Trans. on Inform. Theory*, vol.28, no.1, January 1982, pp.752-754.

[14] S. Vembu and S. Verdú, "Generating random bits from an arbitrary source: fundamental limits", *IEEE Trans. on Inform. Theory*, vol.41, no.5, September 1995, pp.1322-1395.

[15] K. Visweswariah and S. Verdú, "Source codes as random number generators", *IEEE Trans. on Inform. Theory*, vol.44, no.2, March 1998, pp.462-471.

Valdemar C. da Rocha, Jr. (M'77) was born in Jaboatão, Pernambuco, Brazil, on August 27, 1947. He received the B.Sc. degree in Electrical Engineering from the Escola Politécnica, Recife, Brazil, in 1970, and the Ph.D. degree in Electronics from the University of Kent at Canterbury, England, in 1976. In 1976 he joined the faculty of the Federal University of Pernambuco (UFPE), Recife, Brazil, as an Associate Professor in Electrical Engineering. During 1990-1992, he was a Guest Professor at the Swiss Federal Institute of Technology-Zurich. In 1993, he assumed his present position as Professor of Telecommunications.

He was Head of Department at the Department of Electronics and Systems at UFPE in the period 1993-1996. From 1993-1995 he was the Chairman of the Electrical Engineering Committee in the Brazilian National Council for Scientific and Technological Development (CNPq), Brasília.

He was the Chairman of the Technical Program Committee of the 1988 Brazilian Telecommunication Symposium, Campina Grande, organized the Coding session of the 1990 International Telecommnunication Symposium, Rio de Janeiro, and was a co-organizer of the Cryptography session of the 1992 IEEE Info. Theory Workshop in Salvador, Bahia, Brazil. He was the General Chairman and Chairman of the Technical Program Committee for the 1997 Brazilian Telecommunication Symposium, Recife.

He is a member of the IEEE Communication Society (1977), the IEEE Information Theory Society (1981), the Brazilian Telecommunication Society (Founding Member,1983), the Brazilian Society for the Progress of Science, the Brazilian Society of Applied and Computational Mathematics, and the Institute of Mathematics and its Applications (Fellow, 1992, England). For the past four years he has served as a member of the Board of Directors of the Brazilian Telecommunication Society. His area of research interest is applied digital information theory, including error-correcting codes and cryptography.