Homophonic Sequence Substitution

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, BRAZIL

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. It is proved that by sequentially encoding the output of a discrete stationary and ergodic source with binary lossless codes makes the entropy rate of the resulting encoded binary sequence asymptotically approach the value 1, therefore performing optimum homophonic sequence substitution. The cleartext 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.

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 the redundancy of a message sequence with an alternative, and perhaps complementary, approach to that in [3]. The distinguishing feature of our approach is that we neither resort to intentional plaintext expansion, as in conventional (symbol) homophonic substitution, nor to coding extensions of the original source, as suggested by Shannon's lossless source coding theorem [1, 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 [5] 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 [3] 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 codes produces in the limit a completely random sequence. The decoding operation is very simple and consists of applying the encoded binary sequence through a cascade of k look-up tables (corresponding to the number k of iterations), 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. The nice aspect in this approach is that the resulting implementation complexity remains small, because it grows linearly with the number of encodings, and there is no cleartext expansion caused by the iterations. Of course the binary sequence after the k^{th} encoding will still have a 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.

*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.

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 sequence $u_1, u_2, \ldots, u_L$ 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 $N \to \infty$ [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* [2, 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* [2, p.50]. A uniquely decodable code is further said to be a *compact code* [2, 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 [1, p.69] implicitly suggests that the only 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,

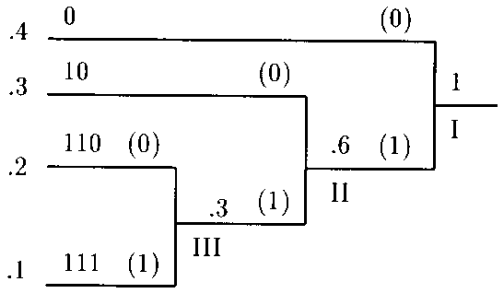

Figure 1: Huffman tree.

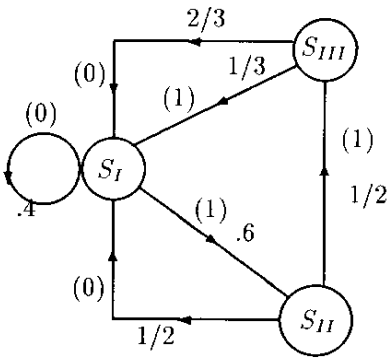

Figure 2: Markov source model.

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 [3], 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 compution of these probabilities, directly from the code rooted tree with probabilities [5], 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 S_I occurs whenever the last digit transmitted is the last digit of a codeword. The example below is provided to illustrate the above description of a Markov source and employs 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.

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 [2, 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 [6]. 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 2

Consider a discrete memoryless source whose alphabet has three symbols occurring with probabilities .4, .35 and .25, respectively.

Probability	Alternate code	0-heavy code
.4	0	1
.35	10	00
.25	11	01

The entropy per binary digit of the associated Markov source model is identical for both codes and its value is .974. The table below presents first order and second order statistics for both codes. By computing for each code the 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. This seems to be a property true in general for alternate codes.

	Alternate code	0-heavy code
P(0)	.4678	.59375
P(00)	.1875	.35
P(01)	.28125	.24375
P(10)	.28125	.24375
P(11)	.25	.1625

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. S_2 is then encoded ...etc. As we prove in *Theorem (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 continues. 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 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 $H_1(S) = H(S)/L_1$ 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. The resulting binary coded sequence is no longer ergodic and whenever this situation happens we may consider repeating that step however parsing with a block size of at least $r+1$ symbols and proceed from then on 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. □

We notice that whenever this equal codeword length phenomena happens in the iterative procedure the entropy of the coded output sequence is usually quite close to 1.

Acknowledgement

We are grateful to our student Guilherme Nunes Melo for helping with the computer simulation.

References

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

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

[3] 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.

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

[6] 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.

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