HOMOPHONIC SEQUENCE SUBSTITUTION

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 sequências é o nome dado neste trabalho para a técnica que consiste em substituir um a um uma dada sequência de símbolos, finita ou semi-infinita, por outra sequência, respectivamente finita ou semi-infinita, sobre o mesmo alfabeto, porém com uma taxa de entropia mais elevada. A sequência 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 sequência 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 sequência resultante após a k-ésima codificação. Um modelo de fonte de Markov é apresentado para descrever as sequências 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 last 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 root 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 $i^{th}$ 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 $U_1$, $U_2$, ..., 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-
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 \( \sigma_i \) occurs always after the last digit of a codeword is generated by the encoder. Let \(|S|\) denote the number of states in a given Markov source. The probability \( P_{\sigma_i} \), \( i = 1, 2, \ldots, |S| \), of every state \( \sigma_i \) is equal to the probability of the corresponding tree node \( P_i \) divided by the average codeword length \( \frac{|S|}{L} \).

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 0.4, 0.3, 0.2, and 0.1, 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 \( P(i) \), \( i \in \{0, 1\} \), the \(|S| \times |S|\) matrix whose \((i, j)^{th}\) entry, denoted as \( P_{ij}(i) \), is the branch probability of going from state \( \sigma_i \) to state \( \sigma_j \).

**Example 2**

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

\[
P(0) = \begin{bmatrix} 0.4 & 0 & 0 \\ 0.5 & 0 & 0 \\ 2/3 & 0 & 0 \end{bmatrix} \quad P(1) = \begin{bmatrix} 0 & 0.6 & 0 \\ 0 & 0.5 & 0 \\ 1/3 & 0 & 0 \end{bmatrix}
\]
Valdemar C. da Rocha Jr. and Hélio M. de Oliveira
Homophonic Sequence Substitution

The code average codeword length is 1.9 and thus the states have the following probabilities: \(P(\sigma_2) = 1/1.9\), \(P(\sigma_2) = 0.6/1.9\) 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 occurring, followed by a one.

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

Thus,

\[
P(0|P(1) = \begin{bmatrix} 0.4 & 0 & 0 \\ 0.5 & 0 & 0 \\ 2/3 & 0 & 0 \end{bmatrix} \begin{bmatrix} 0 & 0.6 & 0 \\ 0 & 0 & 0.5 \\ 1/3 & 0 & 0 \end{bmatrix}
\]

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

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.

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

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.

<table>
<thead>
<tr>
<th>Probability</th>
<th>Alternate code</th>
<th>0-heavy code</th>
</tr>
</thead>
<tbody>
<tr>
<td>4</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>.5</td>
<td>10</td>
<td>00</td>
</tr>
<tr>
<td>.2</td>
<td>110</td>
<td>010</td>
</tr>
<tr>
<td>.1</td>
<td>111</td>
<td>011</td>
</tr>
</tbody>
</table>

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.

<table>
<thead>
<tr>
<th></th>
<th>Alternate code</th>
<th>0-heavy code</th>
</tr>
</thead>
<tbody>
<tr>
<td>P(0)</td>
<td>.474</td>
<td>.579</td>
</tr>
<tr>
<td>P(00)</td>
<td>.189</td>
<td>.315</td>
</tr>
<tr>
<td>P(01)</td>
<td>.284</td>
<td>.263</td>
</tr>
<tr>
<td>P(10)</td>
<td>.284</td>
<td>.263</td>
</tr>
<tr>
<td>P(11)</td>
<td>.242</td>
<td>.159</td>
</tr>
</tbody>
</table>

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
REFERENCES


ACKNOWLEDGEMENT

We are grateful to our colleague Dr. Cecílio Pimentel for helpful discussions.
Valdemar C. da Rocha Jr. and Hélio M. de Oliveira
Homophonic Sequence Substitution


Valdemar C. da Rocha, Jr. (M'77) was born in Jabotião, 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 Telecommunication 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.