THE ENTROPY OF A CODE WITH PROBABILITIES

Valdemar C. da Rocha Jr.[1] 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
e-mail: 99vcr@npd.ufpe.br

Abstract – *The entropy of a code with probabilities is defined and as a consequence the concept of conservation of entropy in lossless coding emerges in a natural manner. For any given probability distribution* $(p_1, p_2, \ldots, p_T)$ *all the distinct decompositions of the associated entropy function* $h(p_1, p_2, \ldots, p_T)$*, as a function of entropies of lower order, are obtained from the terminal uncertainty of the distinct rooted trees with leaf probabilities* $(p_1, p_2, \ldots, p_T)$

1. Introduction

In this paper we begin by reviewing some basic theory of rooted trees with probabilities and then define the entropy of a code with probabilities. As a consequence the concept of conservation of entropy in lossless coding emerges in a natural manner and we remark that, although implicitly a known result, the fact that entropy is preserved in lossless coding apparently has not been previously stated explicitly in the literature. We apply the concepts introduced to source coding as a matter of preference and not due to scope limitations.

2. Basics of Source Coding

Source coding is a well known technique used to represent the symbols at the output an information source with as few code digits per source symbol as possible. We will consider only *lossless source coding* in which case it is possible to reconstruct *exactly* the source output from its encoded representation. The source codes employed in lossless source coding are called *uniquely decodable codes* [1, 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. We will refer to codes with this property as *prefix-free* codes. For simplicity, we consider here only coding the output sequence $U_1, U_2, U_3, \ldots$, of an L-ary discrete *memoryless* source (DMS) [8, p.32], but the theory presented can be applied to sources with memory simply by replacing the probability distribution for U_i with the conditional probability distribution for U_i given the observed values $U_1, U_2, \ldots, U_{i-1}$. For the L-ary discrete memoryless source, the coding problem reduces to the problem of such coding for a single L-ary random variable U_i. We concentrate our attention on binary coding both for its practical importance and because the generalizations to higher order alphabets are immediate.

3. Rooted Trees Revisited

Definition 1 *A* **rooted tree with probabilities** *[3] is a finite rooted tree with probabilities assigned to each node such that*

(a) The root node is assigned probability 1.

(b) The probability of every intermediate node (including the root node) is the sum of the probabilities of the nodes at depth 1 in the subtree stemming from this intermediate node.

An immediate result of **Definition 1**, known as the *Path Length Lemma*, was proved in [3] and goes as follows.

Proposition 1 *In a rooted tree with probabilities, the average depth of the terminal nodes equals the sum of the probabilities of the intermediate nodes (including the root node).*

Various uncertainties can naturally be defined in a rooted tree with probabilities. In [4] the probability of each node was defined as the *probability that we would reach that node in a random journey* through the tree starting at the root node and ending on some terminal node. Then, given that one is at a specified intermediate node, the conditional probability (the branching probability) of choosing each outgoing branch as the next leg of the journey is just the probability on the node at the end of that branch divided by the probability of the node at its beginning, i.e., the probability

0-7803-5030-8/98/$10.00 ©1998 IEEE

of this intermediate node itself. The *branching uncertainty* was defined [3, 4] at each intermediate node so that it equals the uncertainty of a random variable that specifies the branch followed out of that node, given that we had reached that node on our random journey.

Definition 2 *Suppose that* $q_{i1}, q_{i2}, \ldots, q_{iL_i}$ *are the probabilities of the nodes (some of which may be intermediate nodes and some terminal nodes) at the ends of the* L_i *branches stemming outward from the intermediate node whose probability is* P_i. *Then the* **branching uncertainty** *[3],* H_i, *at this node is defined as*

$$H_i = -\sum_{j:q_{ij}\neq 0} \frac{q_{ij}}{P_i} \log \frac{q_{ij}}{P_i},$$

because q_{ij}/P_i *is the conditional probability of choosing the* j^{th} *of these branches as the next leg on our journey given that we are at this node.*

The average depth L of a rooted tree with T terminal nodes is given by the sum of T terms such that each term is the product of a terminal node probability by its depth from the root node.

After representing the output of a DMS by means of a prefix-free code we are often interested in computing the probability of single digits, or pairs of digits, etc., produced by the prefix-free code driven by the source. It turns out that the computation of such probabilities, directly from the associated rooted tree, is possible but becomes very complicated as longer blocks of code digits are considered, i.e., as the order of the statistics increases. We introduce next the *code uncertainty* and the *code state diagram with probabilities*. The latter provides a simpler way for computing code digit statistics.

Suppose that the code C has T codewords whose lengths are $l_1, l_2, \ldots, l_T$ and whose associated probabilities are $p_1, p_2, \ldots, p_T$.

4. Decomposition of Entropies

In what follows we use of the definition of terminal uncertainty of a rooted tree with probabilities combined with *Proposition 2* to establish all the distinct decomposition of an entropy function.

Definition 3 *The* **terminal uncertainty** H_T *[3] of the rooted tree is defined as the quantity*

$$H_T = -\sum_{i=1}^{T} p_i \log p_i.$$

Proposition 2 *Let* $p_1, p_2, \ldots, p_T$ *denote a probability distribution and let* $h(p_1, p_2, \ldots, p_T)$ *denote the entropy in bits of this probability distribution. The number of distinct decompositions of* $h(p_1, p_2, \ldots, p_T)$ *as a function of entropies involving* $p_1, p_2, \ldots, p_T$ *is equal to the number of distinct rooted trees having* T *terminal nodes whose probabilities are* $p_1, p_2, \ldots, p_T$, *respectively.*

Proof: The proof follows directly from *Theorem 1*, [4], which states that

$$H_T = \sum_{i=1}^{N} P_i H_i,$$

where P_i is the node probability and H_i is the corresponding branching uncertainty. □

Example 1

Suppose that $T = 4$. Let $h(x)$ denote the binary entropy, i.e., $h(x) = -x \log x - (1-x)\log(1-x)$. There are essentially four distinct decompositions of $h(p_1, p_2, p_3, p_4)$ as a function of entropies involving p_1, p_2, p_3 and p_4, namely

$$\begin{aligned}
h(p_1,p_2,p_3,p_4) &= h(p_1,p_2,p_3,p_4), \\
&\quad \text{(identity)} \\
h(p_1,p_2,p_3,p_4) &= h(p_1) + \\
&\quad \lambda_1 h(p_2/\lambda_1, p_3/\lambda_1, p_4/\lambda_1), \\
h(p_1,p_2,p_3,p_4) &= h(p_1) + \\
&\quad \lambda_1 h(p_2/\lambda_1) + \lambda_2 h(p_3/\lambda_2), \\
h(p_1,p_2,p_3,p_4) &= h(1-\lambda_2) + \\
&\quad (1-\lambda_2) h(p_1/(1-\lambda_2)) + \\
&\quad \lambda_2 h(p_3/\lambda_2),
\end{aligned}$$

where $\lambda_1 = 1 - p_1$ and $\lambda_2 = 1 - p_1 - p_2$.

We would like to mention here that *Proposition 2* is a generalization of Property 3 of the entropy function $h(.,.,\ldots)$ in Section 6 of [7] as exemplified above by the fourth decomposition of $h(p_1, p_2, p_3, p_4)$.

5. The Entropy of a Code with Probabilities

After representing the output of a DMS by means of a prefix-free code we are often interested in computing the probability of single digits, pairs of

digits, etc., produced by the prefix-free code driven by the source. It turns out that the computation of such probabilities, directly from the associated rooted tree, is possible but becomes very complicated as longer blocks of code digits are considered, i.e., as the order of the statistics increases. We introduce next the *code uncertainty* and the *code state diagram with probabilities*. The latter provides a systematic way for computing code digit statistics.

Definition 4 *A* **code state diagram with probabilities** *is that state diagram obtained by modification of the code rooted tree as follows*

(a) The tree terminal nodes are connected to the root node.

(b) The tree nodes are called states of the code state diagram with probabilities and the root node is called state 1.

(c) The probability P_{σ_i}, $i = 1, 2, \ldots, N$, of every state σ_i is equal to the probability of the corresponding tree node P_i divided by the tree average depth L.

For prefix-free codes **Definition 4** implies that every codeword must start and end at state 1.

Definition 5 *The* **code uncertainty**, H_C, *of a code state diagram with probabilities is defined as the quantity*

$$H_C = \frac{-\sum_{i=1}^{T} p_i \log p_i}{\sum_{i=1}^{T} p_i l_i},$$

i.e., as the ratio between the associated rooted tree terminal uncertainty by the tree average depth.

We notice that our definition of code uncertainty is very general and is not limited to prefix-free codes. Actually, it is not even limited to unique decodability of the code considered.

Example 2

Suppose that $T = 4$ binary codewords have been numbered such that $l_1 = 1$, $l_2 = 2$, $l_3 = 3$ and $l_4 = 3$, and that $p_1 = .4$, $p_2 = .3$, $p_3 = .2$ and $p_4 = .1$, respectively. Then

$$H_C = \frac{-\sum_{i=1}^{4} p_i \log p_i}{\sum_{i=1}^{4} p_i l_i} = \frac{1.5437}{1.9} = .812.$$

Theorem 1 *The code uncertainty of a code state diagram with probabilities equals the sum over all states of the branching uncertainty at that state weighted by the state probability, i.e.,*

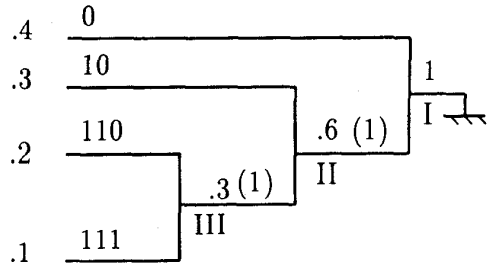

Figure 1: Code tree.

$$H_C = \sum_{i=1}^{N} P_{\sigma_i} H_i. \qquad (1)$$

Proof: Each term $P_{\sigma_i} H_i$ in the sum in (1) can be written as

$$\begin{aligned} P_{\sigma_i} H_i &= -\frac{P_i}{L} \sum_{j:q_{ij}\neq 0} \frac{q_{ij}}{P_i} \log \frac{q_{ij}}{P_i} \\ &= -\frac{1}{L} \sum_{j:q_{ij}\neq 0} q_{ij} \log q_{ij} + \frac{1}{L} P_i \log P_i. \end{aligned}$$

We notice that the term $\frac{1}{L} P_i \log P_i$ for every state σ_i, $i \neq 1$, is cancelled in the sum (1) by the term $-\frac{1}{L} P_i \log P_i$ originated by the state from which state σ_i is reached. In this manner only terms of the form $-\frac{1}{L} \sum_{j:q_{i1}\neq 0} q_{i1} \log q_{i1}$, i.e., terms originating from those states connected to state σ_1 by a single branch, are not cancelled in the sum (1). We notice that the q_{i1}'s, $1 \leq i \leq T$, are precisely the probabilities of the rooted tree terminal nodes. Hence we have proved that

$$H_C = \sum_{i=1}^{N} P_{\sigma_i} H_i.$$

□

Example 3

Let us consider the code of **Example 2**. Figure 1 shows the corresponding code rooted tree and Figure 2 shows the associated code state diagram.

6. Conservation of Entropy

We now introduce the concept of *Conservation of Entropy in Lossless Coding*. Although implicitly a known result, the fact that entropy is preserved in lossless coding apparently has not been previously stated explicitly in the literature. Whenever a uniquely decodable code with an alphabet of r

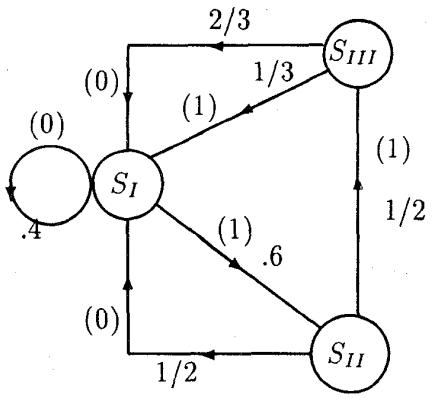

Figure 2: Code state diagram.

symbols is used for the lossless coding of a DMS source S with entropy $H(S)$, usually the ratio $\eta = H(S)/L \log r$ is the parameter of interest to measure the coding efficiency, where L denotes the average codeword length. By our *Definition 5* we observe that $H(S)/L$ is precisely the entropy of the source formed by the cascade of the original source and the encoder. The entropy of the encoded sequence plays an important role both in information theory and in cryptography.

Proposition 3 *Let $H(S)$ denote the entropy of a discrete memoryless source S. Consider that the output of S is fed to an encoder E_1 which implements a lossless code C_1 with an r_1-ary alphabet and average codeword length L_1. The output of E_1 is then fed to the input of a second encoder E_2 which implements a lossless code C_2 with an r_2-ary alphabet and average codeword length L_2, etc.. The following string of equalities hold true.*

$$\begin{aligned} H(S) &= L_1 \log r_1 H_1(S) = L_2 \log r_2 H_2(S) \\ &= \ldots = L_k \log r_k H_k(S), \end{aligned}$$

where $H_i(S)$ and L_i, $1 \leq i \leq k$, denote respectively the entropy and the average codeword length for the i^{th} code in the cascade.

The proof of Shannon's noiseless source coding theorem [8, p.69] implicitly suggests that the only way to reduce the redundancy of a message to be transmitted or stored is by encoding extensions of the original source. In [9] we introduced an iterative procedure to reduce redundancy, as an alternative to coding the extensions of a source. We remark that source coding is an efficient way to remove redundancy by performing data compression, in contrast to homophonic substitution [5, 6] which removes redundancy at the cost of data expansion.

References

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

[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, "The Entropy of a Rooted Tree with Probabilities", in *Abstracts of Papers, IEEE Int. Symp. on Info. Th.*, 1983, p.127.

[5] H.N. Jendal, Y.J.B. Kuhn and J.L. Massey, "An Information-Theoretic Approach to Homophonic Substitution", pp. 382-394 in *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.

[6] V.C. da Rocha and J.L. Massey, "On the Entropy Bound for Optimum Homophonic Substitution", in *Abstracts of Papers, IEEE Int. Symp. on Info. Th.*, 1996, p.93.

[7] C. E. Shannon, "A Mathematical Theory of Communications," *Bell System Tech. J.*, vol. 27, pp. 379-423 and 623-656, July and Oct., 1948.

[8] R.G. Gallager, *Discrete Stochastic Processes*, Kluwer Academic Publishers, Boston, 1996.

[9] V.C. da Rocha Jr. and H.M. de Oliveira, "Homophonic Sequence Substitution", XV Brazilian Telecommunication Symposium, 08-11 September 1997, Recife, Brazil, pp.162-164.