

The Rényi Redundancy of Generalized Huffman Codes

ANSELM C. BLUMER, MEMBER, IEEE, AND ROBERT J. MCELIECE, FELLOW, IEEE

Abstract—If optimality is measured by average codeword length, Huffman's algorithm gives optimal codes, and the redundancy can be measured as the difference between the average codeword length and Shannon's entropy. If the objective function is replaced by an exponentially weighted average, then a simple modification of Huffman's algorithm gives optimal codes. The redundancy can now be measured as the difference between this new average and Rényi's generalization of Shannon's entropy. By decreasing some of the codeword lengths in a Shannon code, the upper bound on the redundancy given in the standard proof of the noiseless source coding theorem is improved. The lower bound is improved by randomizing between codeword lengths, allowing linear programming techniques to be used on an integer programming problem. These bounds are shown to be asymptotically equal, providing a new proof of Kricevski's results on the redundancy of Huffman codes. These results are generalized to the Rényi case and are related to Gallager's bound on the redundancy of Huffman codes.

PREVIOUS WORK

In 1961, Rényi [12] proposed that the Shannon entropy could be generalized to

$$H_s(\bar{p}) = \frac{s+1}{s} \log \left(\sum_{i=1}^m p_i^{1/(s+1)} \right), \quad s > 0,$$

which approaches the Shannon entropy as $s \rightarrow 0^+$. In 1965, Campbell [1] showed that just as the Shannon entropy is a lower bound on the average codeword length of a uniquely decodable code, the Rényi entropy is a lower bound on the exponentially weighted average codeword length

$$\frac{1}{s} \log \left(\sum_{i=1}^m p_i 2^{sl_i} \right), \quad s > 0.$$

Also,

$$\lim_{s \rightarrow 0^+} \frac{1}{s} \log \left(\sum_{i=1}^m p_i 2^{sl_i} \right) = \sum_{i=1}^m p_i l_i.$$

Manuscript received March 18, 1986. This work was supported in part by the Joint Services Electronics Program under Contract N00014-79-C-0424 with the University of Illinois, Urbana-Champaign, and in part by the National Science Foundation under Grant IST-8317918 to the University of Denver, CO. This work was partially presented at the IEEE International Symposia on Information Theory, Santa Monica, CA, January 1982, and Les Arcs, France, June 1982. It also formed part of a dissertation submitted to the Department of Mathematics, University of Illinois, Urbana-Champaign, in partial fulfillment of the requirements for the Ph.D. degree.

A. C. Blumer is with the Department of Computer Science, Tufts University, Medford, MA 02155.

R. J. McEliece is with the Department of Electrical Engineering, 116-81 California Institute of Technology, Pasadena, CA 91125.

IEEE Log Number 8824499.

We define the Rényi redundancy of a code as

$$R_s(\bar{p}, \bar{l}) = \frac{1}{s} \log \left(\sum_{i=1}^m p_i 2^{sl_i} \right) - H_s(\bar{p}).$$

(Note: It will be assumed that the code alphabet is binary, though generalization is not difficult. As a consequence, "log" will always mean the base 2 logarithm; the natural logarithm is denoted by "ln.") Hu [5], Humblet [6], and Parker [11] have observed that a simple generalization of Huffman's algorithm solves the problem of finding a uniquely decodable code which minimizes $R_s(\bar{p}, \bar{l})$. In Huffman's algorithm, each new node is assigned the weight $p_i + p_j$, where p_i and p_j are the lowest weights on available nodes. In the generalized algorithm, the new node is assigned the weight $2^s(p_i + p_j)$. Note that if $s > 0$ this differs from the usual Huffman algorithm in that the root will not have weight 1.

To summarize, the generalized Huffman algorithm finds the optimal solution of the following nonlinear integer programming problem. Given $\bar{p} = (p_1, p_2, \dots, p_m)$ with $p_i > 0$, $\sum_{i=1}^m p_i = 1$, and $s \geq 0$, find $\bar{l} = (l_1, l_2, \dots, l_m)$ with positive integer components to minimize $R_s(\bar{p}, \bar{l})$ subject to

$$\sum_{i=1}^m 2^{-l_i} \leq 1. \quad (\text{KM})$$

Call the value of this optimal solution $R_s(\bar{p})$.

The constraint (KM), known as the Kraft-McMillan inequality, is a necessary and sufficient condition for the existence of a uniquely decodable code with codeword lengths l_i . Equality holds if setting $l_i = -\log p_i$ gives integral lengths. In any case, the inequality is satisfied by letting

$$l_i = \lceil -\log p_i \rceil.$$

A code with these codeword lengths is known as a Shannon code. For $s = 0$ the existence of such a code shows [10] that the redundancy is in $[0, 1]$. In [1], Campbell generalized this by choosing

$$l_i = \left\lceil \log \left(\sum_{j=1}^m p_j^{1/(s+1)} \right) - \frac{1}{s+1} \log p_i \right\rceil$$

which gives the following.

Theorem 1: $0 \leq R_s(\bar{p}) < 1$ for $s \geq 0$.

Although Theorem 1 shows that Shannon coding is always within 1 bit of the optimum on the average, individual codewords can be much longer than necessary. For example, when coding a two-letter alphabet with two codewords, each codeword should be 1 bit. If one of the letters has arbitrarily small probability, the Shannon codeword for that letter is arbitrarily long.

The remainder of this paper is devoted to showing how to get a better upper bound on the redundancy of optimal (Huffman) codes by shortening these long codewords, and how to get a better lower bound by using the idea of randomizing codewords. Randomizing will not result in codes which can be used in practice, but it will enable us to obtain much better bounds on the redundancy.

APPLICATIONS

In 1968, Jelinek [7] showed that coding with respect to the Rényi redundancy is useful when source symbols are produced at a fixed rate and code symbols are transmitted at a high fixed rate. In this case, the instantaneous rate at which code symbols are produced depends on the length of the current codeword. For long codewords, this rate will be higher than the average rate at which code symbols are produced. Excess code symbols must be stored temporarily in a finite buffer. This buffer may still overflow if an unusually long sequence of low probability source symbols is encoded. This problem can be reduced by shortening the lengths of the long codewords. Minimizing the lengths of the longest codewords results in a code with uniform lengths (corresponding to $s = \infty$), which in most cases will not have a good average rate. Jelinek [7] shows how to pick s to solve this problem and gives bounds on the probability of buffer overflow based on s .

Similar considerations apply to the construction of optimal search trees [8]. Each internal node in such a tree corresponds to a decision made during the search. An item is found when a leaf is reached. The items correspond to the source symbols in Huffman coding, and Huffman's algorithm constructs the tree which minimizes the average search time. The search time for an item is proportional to the path length from the root to the leaf corresponding to that item, which is equivalent to the codeword length above. If there is a requirement that searches be completed within a certain time after they are requested, then this time limit corresponds to the buffer length above. Jelinek's analysis then shows that the generalized Huffman algorithm can be used to reduce the probability that this limit is exceeded, and to obtain a bound on this probability.

Campbell [2] has shown that the lengths given by the generalized Huffman algorithm arise in a natural way from geometric considerations when interpolating between the distribution \bar{p} and the uniform distribution along curves in a Riemannian geometry which correspond to straight lines in Euclidean geometry. Parker [11] provides a list of other possible applications and references.

LOWER BOUND

The 0 lower bound on $R_s(\bar{p})$ was obtained by relaxing the integer restriction on the l_i and letting

$$l_i = \hat{l}_i = \log \left(\sum_{j=1}^m p_j^{1/(s+1)} \right) - \frac{1}{s+1} \log p_i.$$

Setting $l_i = \lceil \hat{l}_i \rceil$ gives a solution to (KM) and leads to the upper bound $R_s(\bar{p}) < 1$.

Given the probability distribution \bar{p} and the parameter s , it will be useful to define a new probability distribution \hat{p} based on the optimal (but not necessarily integral) lengths \hat{l}_i :

$$\hat{p}_i = 2^{-\hat{l}_i} = \frac{p_i^{1/(s+1)}}{\sum_{j=1}^m p_j^{1/(s+1)}}. \quad (1)$$

Let t_i denote the fractional part of $-\hat{l}_i$

$$t_i = \{-\hat{l}_i\} = \lceil \hat{l}_i \rceil - \hat{l}_i, \quad (2)$$

and let n_i denote the difference

$$n_i = l_i - \lceil \hat{l}_i \rceil = l_i - \hat{l}_i - t_i.$$

Thus $n_i + t_i$ gives the discrepancy between a solution l_i and the optimal solution \hat{l}_i .

Using the fact that $\hat{p}_i 2^{\hat{l}_i} = 1$, (KM) may be rewritten as an average with respect to the probability distribution \hat{p} :

$$1 - \sum_{i=1}^m 2^{-l_i} = 1 - \sum_{i=1}^m \hat{p}_i 2^{\hat{l}_i - l_i} = 1 - \sum_{i=1}^m \hat{p}_i 2^{-n_i - t_i} \geq 0.$$

If $s > 0$, the objective function may also be rewritten as

$$\begin{aligned} R_s(\bar{p}, \bar{l}) &= \frac{1}{s} \log \left(\sum_{i=1}^m p_i 2^{sl_i} \right) - \frac{s+1}{s} \log \left(\sum_{i=1}^m p_i^{1/(s+1)} \right) \\ &= \frac{1}{s} \log \left(\sum_{i=1}^m p_i^{1/(s+1)} p_i^{s/(s+1)} 2^{s\hat{l}_i} 2^{s(l_i - \hat{l}_i)} \right) \\ &\quad - \frac{1}{s} \log \left[\left(\sum_{i=1}^m p_i^{1/(s+1)} \right) \left(\sum_{i=1}^m p_i^{1/(s+1)} \right)^s \right] \\ &= \frac{1}{s} \log \left(\sum_{i=1}^m \hat{p}_i \hat{p}_i^{s/(s+1)} 2^{s(n_i + t_i)} \right) \\ &= \frac{1}{s} \log \left(\sum_{i=1}^m \hat{p}_i 2^{s(n_i + t_i)} \right). \end{aligned}$$

Since $(1/s) \log x$ is a strictly increasing function of x , this objective function may be replaced by

$$V_s(\bar{p}, \bar{l}, \bar{n}) = \sum_{i=1}^m \hat{p}_i 2^{s(n_i + t_i)}. \quad (3a)$$

If $s = 0$, we define

$$\begin{aligned} V_0(\bar{p}, \bar{l}, \bar{n}) &= R_0(\bar{p}, \bar{l}) = \sum_{i=1}^m p_i l_i - \sum_{i=1}^m p_i \log \frac{1}{p_i} \\ &= \sum_{i=1}^m p_i (l_i - \hat{l}_i) = \sum_{i=1}^m \hat{p}_i (n_i + t_i), \quad (3b) \end{aligned}$$

since $p_i = \hat{p}_i$ in this case. We have transformed the original integer programming problem into the following.

Problem 1: Given a probability vector \bar{p} and $s \geq 0$, compute \hat{p} and \bar{t} by (1) and (2). Then find integers n_i satisfying (KM) and minimizing $V_s(\bar{p}, \bar{t}, \bar{n})$ given by (3).

The optimal value of this program is $2^{sR_s(\bar{p})}$ if $s > 0$, and $R_0(\bar{p})$ if $s = 0$. Inequality (KM) may be further rewritten as $E^{\hat{p}}[1 - 2^{-N-T}] \geq 0$, where $E^{\hat{p}}$ denotes expectation with respect to the probability distribution \hat{p} and T and N are random variables. T is defined so that

$$P(T=t) = \sum_{i: t_i=t} \hat{p}_i$$

and N is defined similarly. We can also rewrite

$$V_s(\bar{p}, \bar{t}, \bar{n}) = \begin{cases} E^{\hat{p}}[2^{s(N+T)}], & \text{for } s > 0 \\ E^{\hat{p}}[N+T], & \text{for } s = 0 \end{cases}$$

This notation suggests the following modification to the above problem.

Problem 2: Given $s \geq 0$ and a random variable T with values in $[0, 1)$ and discrete probability distribution \hat{p} , find an integer-valued random variable N satisfying

$$E^{\hat{p}}[1 - 2^{-N-T}] \geq 0$$

and minimizing $E^{\hat{p}}[2^{s(N+T)}]$ or, if $s = 0$, minimizing $E^{\hat{p}}[N+T]$.

Let $L_s(\bar{p})$ be defined so that $2^{sL_s(\bar{p})}$ is the value of the minimal solution to the above problem. If $s = 0$, let this value be $L_0(\bar{p})$.

Theorem 2: $0 \leq L_s(\bar{p}) \leq R_s(\bar{p})$ for $s \geq 0$.

Proof: Every feasible solution to Problem 1 corresponds to a feasible solution to Problem 2, since T can be defined as above, and N can be defined by

$$P(N=n, T=t) = \sum_{i: t_i=t, n_i=n} \hat{p}_i.$$

On the other hand, not all feasible solutions to Problem 2 correspond to feasible solutions to Problem 1. For example, any N with $0 < P(N=n) < \min \hat{p}_i$ for some n cannot correspond to a feasible solution to Problem 1. Where there are corresponding feasible solutions, they have the same value. It follows that $2^{sL_s(\bar{p})} \leq 2^{sR_s(\bar{p})}$ and $L_0(\bar{p}) \leq R_0(\bar{p})$, proving the right-hand inequality.

Any feasible solution to Problem 2 satisfies $E^{\hat{p}}[2^{-N-T}] \leq 1$, so by Jensen's inequality

$$2^{E^{\hat{p}}[-N-T]} \leq E^{\hat{p}}[2^{-N-T}] \leq 1.$$

Taking logarithms gives

$$E^{\hat{p}}[N+T] \geq 0$$

which proves the $s = 0$ case. Applying Jensen's inequality again and using this last inequality,

$$1 = 2^0 \leq 2^{E^{\hat{p}}[s(N+T)]} \leq E^{\hat{p}}[2^{s(N+T)}].$$

Since this holds for any feasible solution to Problem 2, taking $1/s$ times the logarithm of both sides gives the left-hand inequality in the $s > 0$ case.

UPPER BOUND

The previous section showed that computing $L_s(\bar{p})$ provides a lower bound to $R_s(\bar{p})$. Since the value of any feasible solution to Problem 1 provides an upper bound to $R_s(\bar{p})$, and Shannon coding provides a feasible solution with value exceeding the lower bound by less than 1, we had lower and upper bounds showing that the redundancy is in $[0, 1)$. The feasible solution to Problem 1 corresponding to Shannon coding is $\bar{n} = 0$. The following algorithm improves this solution by changing some components of \bar{n} to -1 . The value $U_s(\bar{p})$ of this solution will provide an improved upper bound to the redundancy. As with the bound derived from Shannon coding, the difference between the upper and lower bounds will be estimated.

Algorithm 1

1) Given \bar{p}_s and $s \geq 0$, compute \hat{p}, \bar{t} using (1) and (2) and

$$C = 1 - \sum_{i=1}^m \hat{p}_i 2^{-t_i}.$$

2) Set $\bar{n} = 0$.

3) Repeat the following pair of steps, in order of decreasing t_i , until step 3a causes C to become negative (step 3b is skipped when this happens);

- a) decrease C by $\hat{p}_i 2^{-t_i}$.
- b) replace the corresponding n_i by -1 .

4) Compute

$$U_s(\bar{p}) = \begin{cases} \frac{1}{s} \log \left(\sum_{i=1}^m \hat{p}_i 2^{s(n_i+t_i)} \right), & \text{if } s > 0 \\ \sum_{i=1}^m \hat{p}_i (n_i + t_i), & \text{if } s = 0. \end{cases}$$

After Step 3b, C is the value of the left side of (KM) for the current solution. The algorithm will stop in m or fewer steps, since $t_i < 1$ for all i , and so

$$1 - 2 \sum_{i=1}^m \hat{p}_i 2^{-t_i} = 1 - \sum_{i=1}^m \hat{p}_i 2^{1-t_i} < 0.$$

The reason that the algorithm proceeds in order of decreasing t_i is the following. The change in (KM) resulting from replacing n_i by $n_i - 1$ is $-\hat{p}_i 2^{-n_i-t_i}$, while the change in the objective function, $V_s(\bar{p}, \bar{t}, \bar{n})$ is $-\hat{p}_i (1 - 2^{-s}) 2^{s(n_i+t_i)}$ (or if $s = 0$, $-\hat{p}_i$). Thus the component of \bar{n} which gives the greatest decrease in the objective function per decrease in (KM) is the component with the largest value of $n_i + t_i$.

The following example, with $s = 2$, shows that this algorithm will not always yield optimal solutions to Problem 1.

i	p_i	\hat{p}_i	\hat{l}_i	$ \hat{l}_i $	Optimal l_i
1	169	13	1.25...	2	1
	299	31			
2	64	8	1.95...	2	2
	299	31			
3	64	8	1.95...	2	3
	299	31			
4	1	1	4.95...	5	4
	299	31			
5	1	1	4.95...	5	4.
	299	31			

Note that $l_3 = 3$ in the optimal solution, which corresponds to $n_3 = 1$, so any algorithm which can reach an optimal solution must be able to consider positive values for n_i . The following algorithm finds $L_s(\bar{p})$, as shown by Theorem 3 (to follow).

Algorithm 2

- 1) Given $s \geq 0$ and \bar{p} , compute \hat{p} and \hat{t} using (1) and (2).
- 2) Find the smallest $t \in [0, 1)$ satisfying the constraint

$$\sum_{i: t_i \leq t} \hat{p}_i 2^{-t_i} + \sum_{i: t_i > t} \hat{p}_i 2^{1-t_i} \leq 1.$$

- 3) Find the largest $b \in [0, 1)$ satisfying the constraint

$$\begin{aligned} \sum_{i: t_i < t} \hat{p}_i 2^{-t_i} + (1-b) \sum_{i: t_i = t} \hat{p}_i 2^{-t_i} \\ + b \sum_{i: t_i = t} \hat{p}_i 2^{1-t_i} + \sum_{i: t_i > t} \hat{p}_i 2^{1-t_i} \leq 1. \end{aligned}$$

- 4) If $s = 0$, let $z(x) = x$; otherwise, let $z(x) = 2^{sx}$. Let

$$\begin{aligned} W_s(\bar{p}) = \sum_{i: t_i < t} \hat{p}_i z(t_i) + (1-b) \sum_{i: t_i = t} \hat{p}_i z(t_i) \\ + b \sum_{i: t_i = t} \hat{p}_i z(t_i - 1) + \sum_{i: t_i > t} \hat{p}_i z(t_i - 1). \end{aligned}$$

- 5) Let

$$L_s(\bar{p}) = \begin{cases} W_s(\bar{p}), & \text{if } s = 0 \\ \frac{1}{s} \log(W_s(\bar{p})), & \text{if } s > 0. \end{cases}$$

The only difference between this algorithm and the previous one is the extra randomization allowed when $t_i = t$.

Theorem 3: $W_s(\bar{p})$ is the minimum value of Problem 2.

Proof: Let $p_{in} = P(T = t_i, N = n)$. Problem 2 can be viewed as a linear programming problem with the p_{in} as variables, as follows:

Problem 2 (Restated): Given $s \geq 0$ and \bar{p} , find \hat{p} and \hat{t} using (1) and (2). If $s > 0$, find p_{in} to minimize $\sum_{i=1}^m \sum_{n \in Z} p_{in} 2^{s(n+t_i)}$ or, if $s = 0$, to minimize

$\sum_{i=1}^m \sum_{n \in Z} p_{in} (n + t_i)$ subject to the constraints

$$\sum_{i=1}^m \sum_{n \in Z} p_{in} 2^{-n-t_i} = 1$$

and

$$\sum_{n \in Z} p_{in} = \hat{p}_i.$$

The first constraint is (KM) with equality holding. Inequality cannot hold in an optimal solution, since some p_{in} could then be reduced by some $\epsilon > 0$ while $p_{i,n-1}$ was increased by ϵ . This results in a reduction of $\epsilon(1 - 2^{-s})2^{s(n+t_i)}$ in the objective function for the $s > 0$ case, and a reduction of ϵ for the $s = 0$ case. ϵ must be chosen so that the increase $\epsilon 2^{-n-t_i}$ in (KM) will not violate that inequality.

The proof of the theorem now proceeds by constructing the dual program and finding solutions to both the original program and the dual program with the same value. The dual program is as follows.

Find q_0, q_1, \dots, q_m to maximize $q_0 + \sum_{i=1}^m q_i \hat{p}_i$, subject to the constraints

$$q_0 2^{-n-t_i} + q_i \leq 2^{s(n+t_i)}, \quad \text{if } s > 0$$

or

$$q_0 2^{-n-t_i} + q_i \leq n + t_i, \quad \text{if } s = 0.$$

For any feasible solutions to these linear programs, the following inequalities show that the value of the dual program is at most the value of the original program:

$$\begin{aligned} q_0 + \sum_{i=1}^m q_i \hat{p}_i &= q_0 \sum_{i=1}^m \sum_{n \in Z} p_{in} 2^{-n-t_i} + \sum_{i=1}^m q_i \sum_{n \in Z} p_{in} \\ &= \sum_{i=1}^m \sum_{n \in Z} p_{in} (q_0 2^{-n-t_i} + q_i) \\ &\leq \sum_{i=1}^m \sum_{n \in Z} p_{in} 2^{s(n+t_i)}, \quad \text{if } s > 0 \\ &\leq \sum_{i=1}^m \sum_{n \in Z} p_{in} (n + t_i), \quad \text{if } s = 0. \end{aligned}$$

Thus, if feasible solutions with the same value can be found for both the original and the dual programs, this common value must be the optimal value.

Let t and b be chosen by Algorithm 2, and let

$$p_{in} = \begin{cases} \hat{p}_i, & \text{if } t_i < t, n = 0, \text{ or } t_i > t, n = -1 \\ b\hat{p}_i, & \text{if } t_i = t, n = -1 \\ (1-b)\hat{p}_i, & \text{if } t_i = t, n = 0 \\ 0, & \text{in all other cases.} \end{cases}$$

By construction of t and b , this is clearly a feasible solution to the original program with the appropriate value. If $s > 0$, let

$$q_0 = -(1 - 2^{-s})2^{(s+1)t}$$

$$q_i = \begin{cases} 2^{st_i} - q_0 2^{-t_i}, & \text{for } t_i \leq t \\ 2^{s(t_i-1)} - q_0 2^{1-t_i}, & \text{for } t_i \geq t. \end{cases}$$

Similarly, if $s = 0$, let

$$q_0 = -2^t$$

$$q_i = \begin{cases} t_i - q_0 2^{-t_i}, & \text{for } t_i \leq t \\ t_i - q_0 2^{1-t_i}, & \text{for } t_i \geq t. \end{cases}$$

Note that, in both cases, the alternate expressions for q_i give the same value when $t_i = t$.

It remains to be shown that these formulas give admissible solutions to the dual program and that these solutions have the appropriate values. For $s > 0$, the value of the dual program is

$$\begin{aligned} q_0 + \sum_{t_i < t} \hat{p}_i q_i + \sum_{t_i = t} \hat{p}_i q_i + \sum_{t_i > t} \hat{p}_i q_i \\ = \sum_{t_i < t} \hat{p}_i 2^{st_i} + (1-b) \sum_{t_i = t} \hat{p}_i 2^{st_i} \\ + b \sum_{t_i = t} \hat{p}_i 2^{s(t_i-1)} + \sum_{t_i > t} \hat{p}_i 2^{s(t_i-1)} \\ + q_0 \left[1 - \sum_{t_i < t} \hat{p}_i 2^{-t_i} - (1-b) \sum_{t_i = t} \hat{p}_i 2^{-t_i} \right. \\ \left. - b \sum_{t_i = t} \hat{p}_i 2^{1-t_i} - \sum_{t_i > t} \hat{p}_i 2^{1-t_i} \right]. \end{aligned}$$

The factor multiplying q_0 above is zero by choice of t and b , leaving only the first four terms, which are equal to $W_s(\bar{p})$, as desired. Similarly, for $s = 0$, the value of the dual program is

$$\begin{aligned} q_0 + \sum_{t_i < t} \hat{p}_i q_i + \sum_{t_i = t} \hat{p}_i q_i + \sum_{t_i > t} \hat{p}_i q_i \\ = \sum_{t_i < t} \hat{p}_i t_i + (1-b) \sum_{t_i = t} \hat{p}_i t_i \\ + b \sum_{t_i = t} \hat{p}_i (t_i-1) + \sum_{t_i > t} \hat{p}_i (t_i-1) \\ + q_0 \left[1 - \sum_{t_i < t} \hat{p}_i 2^{-t_i} - (1-b) \sum_{t_i = t} \hat{p}_i 2^{-t_i} \right. \\ \left. - b \sum_{t_i = t} \hat{p}_i 2^{1-t_i} - \sum_{t_i > t} \hat{p}_i 2^{1-t_i} \right]. \end{aligned}$$

All that remains is to show that these solutions are admissible. If $s > 0$, this requires showing that

$$q_i \leq 2^{s(n+t_i)} - q_0 2^{-n-t_i} = 2^{s(n+t_i)} + (1-2^{-s}) 2^{st_i} 2^{-(n+t_i-t)}$$

for $i = 1, 2, \dots, m$ and all integers n . Call this last quantity $g_{is}(n)$, and let

$$\begin{aligned} \Delta g_{is}(n) &= g_{is}(n+1) - g_{is}(n) \\ &= (2^s - 1) 2^{s(n+t_i)} - (1-2^{-s}) 2^{st_i} 2^{-(n+1+t_i-t)} \\ &= 2^{st_i} (1-2^{-s}) [2^{s(n+1+t_i-t)} - 2^{-(n+1+t_i-t)}]. \end{aligned}$$

Similarly, for $s = 0$ it is necessary to show that

$$q_i \leq n + t_i - q_0 2^{-n-t_i} = n + t_i + 2^{-(n+t_i-t)}$$

for $i = 1, 2, \dots, m$ and all integers n . Call this last quantity

$g_{i0}(n)$, and let

$$\Delta g_{i0}(n) = g_{i0}(n+1) - g_{i0}(n) = 1 - 2^{-(n+1+t_i-t)}.$$

For $t_i < t$, $\Delta g_{is}(n)$ is negative for $n \leq -1$ and positive for $n \geq 0$, so the minimum value of $g_{is}(n)$ is $g_{is}(0) = q_i$. For $t_i \geq t$, $\Delta g_{is}(n)$ is negative for $n \leq -2$ and nonnegative for $n \geq -1$, so in this case the minimum value of $g_{is}(n)$ is $g_{is}(-1) = q_i$ again, as desired.

Theorem 4: $0 \leq L_s(\bar{p}) \leq R_s(\bar{p}) \leq U_s(\bar{p}) < 1$.

Proof. The only part that remains to be proved is the last inequality, which follows from the fact that the upper bound $U_s(\bar{p})$ is an improvement over that obtained from Shannon coding.

The following theorem bounds the difference $U_s(\bar{p}) - L_s(\bar{p})$.

Theorem 5:

$$\begin{aligned} 0 \leq U_s(\bar{p}) - L_s(\bar{p}) &< \max_{1 \leq i \leq m} \hat{p}_i, & \text{for } s = 0 \\ &< \frac{(1-2^{-s})}{s \ln 2} \max_{1 \leq i \leq m} \hat{p}_i 2^{st_i}, & \text{for } s > 0. \end{aligned}$$

Proof: In case 1 ($s = 0$), the difference between $U_0(\bar{p})$, which is the result of Algorithm 1, and $L_0(\bar{p})$, which is the result of Algorithm 2, is less than \hat{p}_i for the component which caused C to become negative in Algorithm 1. In case 2 ($s > 0$), the difference between $2^{sU_s(\bar{p})}$ and $W_s(\bar{p}) = 2^{sL_s(\bar{p})}$ is less than $(1-2^{-s})\hat{p}_i 2^{st_i}$, for the same reason as in case 1. Therefore,

$$\begin{aligned} U_s(\bar{p}) - L_s(\bar{p}) &= \frac{1}{s} \log \frac{2^{sU_s(\bar{p})}}{W_s(\bar{p})} \\ &< \frac{1}{s} \log \frac{W_s(\bar{p}) + (1-2^{-s})\hat{p}_i 2^{st_i}}{W_s(\bar{p})} \\ &\leq \frac{1}{s} \log (1 + (1-2^{-s})\hat{p}_i 2^{st_i}) \end{aligned}$$

since $W_s(\bar{p}) = 2^{sL_s(\bar{p})} \geq 2^0 = 1$. The conclusion now follows from the fact that $y = \log(1+x)$ is convex, and therefore lies below its tangent line at $(0,0)$.

APPLICATION TO MEMORYLESS SOURCES

One case where the preceding theorem is particularly useful is fixed-to-variable-length (FV) coding of a memoryless source (also known as block-to-variable-length or BV coding). Suppose that a memoryless source has an output alphabet of size m , with probabilities p_1, p_2, \dots, p_m with all $p_i > 0$ and $\sum_{i=1}^m p_i = 1$. If b_i denotes the number of times that the i th letter occurs in a block B from this source, then the probability of this block is

$$P(B) = \prod_{i=1}^m p_i^{b_i} \leq \left(\max_{1 \leq i \leq m} p_i \right)^n < 1$$

and so the probabilities of the block approach zero uniformly as $n \rightarrow \infty$. The $s = 0$ case of the above theorem now applies, to show that $U_s(\bar{p}^n) - L_s(\bar{p}^n) \rightarrow 0$ as $n \rightarrow \infty$, where

\bar{p}^n is the probability distribution on blocks of length n from this source.

The $s > 0$ case can be illustrated by examining the binary memoryless source with $p_1 = p$ and $p_2 = 1 - p$ for $0 < p < 1$. In this case

$$P(B) = P_k = p^k(1-p)^{n-k} = (1-p)^n \left(\frac{p}{1-p} \right)^k$$

if the first letter occurs k times in the block. There are $\binom{n}{k}$ blocks having this probability, so

$$\begin{aligned} \hat{P}_k &= \frac{(1-p)^{n/(s+1)} \left(\frac{p}{1-p} \right)^{k/(s+1)}}{\sum_{i=0}^n \binom{n}{i} p^{i/(s+1)} (1-p)^{(n-i)/(s+1)}} \\ &= \frac{(1-p)^{n/(s+1)} \left(\frac{p}{1-p} \right)^{k/(s+1)}}{\left(p^{1/(s+1)} + (1-p)^{1/(s+1)} \right)^n}. \end{aligned}$$

To simplify this, let

$$p' = \frac{p^{1/(s+1)}}{p^{1/(s+1)} + (1-p)^{1/(s+1)}}.$$

Thus

$$\hat{P}_k = (1-p')^n \left(\frac{p'}{1-p'} \right)^k = (p')^k (1-p')^{n-k},$$

which again approaches zero uniformly as $n \rightarrow \infty$. For a source with more output letters, the only difference is that the multinomial theorem must be used instead of the binomial theorem to simplify the denominator. Thus we have the following.

Theorem 6: $L_s(\bar{p}^n) - U_s(\bar{p}^n) \rightarrow 0$ as $n \rightarrow \infty$ for any memoryless source.

It is possible to compute an asymptotic formula for the minimum redundancy $R_s(\bar{p}^n)$ of a binary memoryless source by computing $L_s(\bar{p}^n)$. By Theorem 6, this must approach the minimum redundancy of the source. We will need the $n+1$ fractional parts

$$t_k = \{\log \hat{P}_k\} = \left\{ n \log(1-p') + k \log \left(\frac{p'}{1-p'} \right) \right\}.$$

Suppose that $\log(p'/(1-p'))$ is rational with denominator r when written in lowest terms, so

$$t_k = t_{k'} \quad \text{if and only if } k \equiv k' \pmod{r}$$

(the irrational case will not be treated here). In this case, steps 2–5 of Algorithm 2 depend only on

$$P_k^* = \sum_{k: t_{k'} = t_k} \hat{P}_{k'} = \sum_{k' \equiv k(r)} \binom{n}{k'} (p')^{k'} (1-p')^{n-k'}$$

and not on the individual \hat{P}_k . The following lemma by Ramus [8], shows that $P_k^* \rightarrow 1/r$ as $n \rightarrow \infty$.

Lemma : $P_k^* = (1/r) + O(\rho^n)$ for some $\rho \in (0, 1)$.

Proof: Let $S = e^{2\pi i/r}$, then $S, S^2, S^3, \dots, S^{r-1}$ are the roots of $f(x) = x^{r-1} + x^{r-2} + \dots + x + 1$, so

$$f(S^j) = \begin{cases} r, & \text{if } j \equiv 0 \pmod{r} \\ 0, & \text{otherwise} \end{cases}.$$

P_k^* can be rewritten as

$$\begin{aligned} P_k^* &= \sum_{l=0}^n \frac{1}{r} f(S^{l-k}) \binom{n}{l} (p')^l (1-p')^{n-l} \\ &= \frac{1}{r} \sum_{l=0}^n \sum_{j=0}^{r-1} S^{(l-k)j} \binom{n}{l} (p')^l (1-p')^{n-l} \\ &= \frac{1}{r} \sum_{j=0}^{r-1} S^{-kj} \sum_{l=0}^n \binom{n}{l} (p' S^j)^l (1-p')^{n-l} \\ &= \frac{1}{r} \sum_{j=0}^{r-1} S^{-kj} (p' S^j + 1 - p')^n. \end{aligned}$$

Let $\rho = \max_{1 \leq j \leq r-1} |p' S^j + 1 - p'|$, and the result follows.

Let $t_1^* < t_2^* < \dots < t_m^*$ be the fractional parts with duplications omitted. The values t and b computed by Algorithm 2 can be combined into a single number $t^* \in (0, m^*]$ by choosing t^* so that it rounds up to the index for which $t_i^* = t$, and so that $\{-t^*\} = b$. Let P_i^* be the total probability mass at t_i^* , i.e.,

$$P_i^* = \sum_{j: t_j = t_i^*} \hat{P}_j.$$

If g represents the left half of the constraint in step 3 of Algorithm 2, expressed in terms of P_i^* , t_i^* , and t^* , then g is a continuous function of these variables, and a strictly increasing function of t^* . For example, the following chart shows what happens as t^* goes from $5 - \epsilon$ to $5 + \epsilon$. The terms in g which change are those multiplying P_5^* and P_6^* . Let c_1, c_2, c_3 , and c_4 be the coefficients of $P_5^* 2^{-t_5^*}$, $P_5^* 2^{1-t_5^*}$, $P_6^* 2^{-t_6^*}$, and $P_6^* 2^{1-t_6^*}$, respectively:

t^*	t	b	c_1	c_2	c_3	c_4
$5 - \epsilon$	t_5^*	ϵ	$1 - \epsilon$	ϵ	0	1
5	t_5^*	0	1	0	0	1
$5 + \epsilon$	t_6^*	$1 - \epsilon$	1	0	ϵ	$1 - \epsilon$

Similarly, the objective function in step 4 is a continuous function of P_i^* , t_i^* , and t^* . It follows that $L_s(\bar{p}^n)$ is a continuous function of the P_i^* , and it follows from the above lemma that using $P_k^* = 1/r$ will give a result which is asymptotically true.

Now suppose that $\eta = t_1^*$. In other words,

$$\begin{aligned}\eta &= \frac{1}{r} \left\{ rm \log(1-p') + rk \log\left(\frac{p'}{1-p'}\right) \right\} \\ &= \frac{1}{r} \{ rm \log(1-p') \}.\end{aligned}$$

Then $t_i^* = \eta + ((i-1)/r)$ for $i=1, 2, \dots, r$. If $t = t_a^* = \eta + ((a-1)/r)$, the constraint in step 3 of Algorithm 2 is (using $P_i^* = 1/r$)

$$\begin{aligned}&\sum_{k=0}^{a-1} \frac{1}{r} 2^{-\eta-(k/r)} + \sum_{k=a}^{r-1} \frac{1}{r} 2^{1-\eta-(k/r)} \\ &= \frac{1}{r} \sum_{k=0}^{r-1} 2^{-\eta-(k/r)} + \frac{1}{r} \sum_{k=a}^{r-1} 2^{-\eta-(k/r)} \\ &= \frac{2^{-\eta}}{r} \left(\frac{1-2^{-1}}{1-2^{-1/r}} \right) + \frac{2^{-\eta}}{r} \left(\frac{2^{-a/r}-2^{-1}}{1-2^{-1/r}} \right) \\ &= \frac{2^{-\eta-(a/r)}}{r(1-2^{-1/r})} = \frac{2^{((1-a)/r)-\eta}}{r(2^{1/r}-1)} \leq 1.\end{aligned}$$

This is equivalent to

$$a \geq 1 - r\eta - r \log(r(2^{1/r}-1)),$$

so

$$a = [1 - r\eta - r \log(r(2^{1/r}-1))]$$

leads to the best choice for t . Let

$$\delta = \frac{1}{r} \{ -1 + r\eta + r \log(r(2^{1/r}-1)) \}$$

so that

$$a - r\delta = 1 - r\eta - r \log(r(2^{1/r}-1)).$$

The next step is to find b , as in step 4 of Algorithm 2:

$$\begin{aligned}&\sum_{k=0}^{a-2} \frac{1}{r} 2^{-\eta-(k/r)} + \frac{1-b}{r} 2^{-\eta-((a-1)/r)} \\ &+ \frac{b}{r} 2^{1-\eta-((a-1)/r)} + \sum_{k=a}^{r-1} \frac{1}{r} 2^{1-\eta-(k/r)} \\ &= \frac{1}{r} \sum_{k=0}^{r-1} 2^{-\eta-(k/r)} + \frac{b}{r} 2^{-\eta-((a-1)/r)} \\ &+ \frac{1}{r} \sum_{k=a}^{r-1} 2^{-\eta-(k/r)} \\ &= \frac{2^{((1-a)/r)-\eta}}{r(2^{1/r}-1)} + \frac{b}{r} 2^{-\eta-((a-1)/r)} \leq 1.\end{aligned}$$

If equality holds, then

$$b = r 2^{\eta+((a-1)/r)} - \frac{1}{2^{1/r}-1} = \frac{2^\delta-1}{2^{1/r}-1}.$$

Now, if $s = 0$, $L_s(\bar{p}^n)$ can be evaluated directly. Let

$$\begin{aligned}L_0(\bar{p}^n) &= \sum_{k=0}^{a-2} \frac{1}{r} \left(\eta + \frac{k}{r} \right) + \frac{1-b}{r} \left(\eta + \frac{a-1}{r} \right) \\ &+ \frac{b}{r} \left(\eta + \frac{a-1}{r} - 1 \right) + \sum_{k=a}^{r-1} \frac{1}{r} \left(\eta + \frac{k}{r} - 1 \right) \\ &= \eta + \frac{1}{r^2} \frac{(r-1)r}{2} - \frac{b}{r} - \frac{r-a}{r} \\ &= \eta + \frac{1}{2} - \frac{1}{2r} - \frac{1}{r} \frac{2^\delta-1}{2^{1/r}-1} \\ &- 1 + \frac{1}{r} [1 + r\delta - r\eta - r \log(r(2^{1/r}-1))] \\ &= \frac{1-r}{2r} - \log(r(2^{1/r}-1)) + \frac{1}{r(2^{1/r}-1)} \\ &+ \delta - \frac{2^\delta}{r(2^{1/r}-1)},\end{aligned}$$

which agrees with [9, eq. (6)].

If $s > 0$,

$$\begin{aligned}W_s(\bar{p}^n) &= \sum_{k=0}^{a-2} \frac{1}{r} 2^{s(\eta+(k/r))} + \frac{1-b}{r} 2^{s[\eta+((a-1)/r)]} \\ &+ \frac{b}{r} 2^{s[\eta+(a-1)/r]-1} + \sum_{k=a}^{r-1} \frac{1}{r} 2^{s(\eta+(k/r)-1)} \\ &= (1-2^{-s}) \sum_{k=0}^{a-2} \frac{1}{r} 2^{s(\eta+(k/r))} \\ &+ (1-2^{-s}) \frac{1-b}{r} 2^{s[\eta+((a-1)/r)]} \\ &+ \sum_{k=0}^{r-1} \frac{1}{r} 2^{s(\eta+(k/r)-1)} \\ &= (1-2^{-s}) \frac{2^{s\eta} 2^{s(a-1)/r}-1}{r 2^{s/r}-1} \\ &+ (1-2^{-s}) \frac{2^{s\eta}}{r} (1-b) 2^{s(a-1)/r} \\ &+ \frac{2^{s\eta-s}}{r} \frac{2^s-1}{2^{s/r}-1} \\ &= \frac{(1-2^{-s})}{r} 2^{s[\eta+((a-1)/r)]} \left[\frac{1}{2^{s/r}-1} + 1-b \right] \\ &= \frac{(1-2^{-s})}{r} \left(\frac{2^{1/r}-2^\delta}{2^{1/r}-1} + \frac{1}{2^{s/r}-1} \right) \\ &\cdot \frac{2^{s\delta}}{r^s (2^{1/r}-1)^s},\end{aligned}$$

where the last equality follows from the fact that

$$\delta = \log(r(2^{1/r}-1)) + \eta + \frac{a-1}{r}.$$

Thus

$$\begin{aligned} L_s(\bar{p}^n) &= \frac{1}{s} \log W_s(\bar{p}^n) \\ &= \frac{1}{s} \log(1 - 2^{-s}) + \frac{1}{s} \log \left(\frac{2^{1/r} - 2^\delta}{2^{1/r} - 1} + \frac{1}{2^{s/r} - 1} \right) \\ &\quad + \delta - \log(r(2^{1/r} - 1)). \end{aligned}$$

An interesting application of the preceding results is in the case of a binary memoryless source with probabilities such that $p' = 1/(2^m + 1)$ for some integer $m > 0$. In this case, $\log(p'/(1 - p')) = -m$ is an integer, so $r = 1$ and all of the fractional parts t_k are identical. Thus the lower bounds can be calculated exactly, rather than approximately as in the analysis using Raman's lemma. Plugging $r = 1$ into the formulas for $L_0(\bar{p}^n)$ and $L_s(\bar{p}^n)$ obtained above gives

$$L_0(\bar{p}^n) = 1 + \delta - 2^\delta$$

and

$$L_s(\bar{p}^n) = \delta + \frac{1}{s} \log(1 + (1 - 2^{-s})(1 - 2^\delta)), \quad \text{for } s > 0.$$

Also,

$$\eta = \{n \log(1 - p')\} = \left\{ n \log \left(\frac{2^m}{2^m + 1} \right) \right\}$$

and

$$\delta = \{\eta - 1\} = \eta.$$

Since $\log(2^m/(2^m + 1))$ is irrational, Kronecker's theorem [4] says that the set of values taken on by δ as n increases is dense in $(0, 1)$, and

$$\max_{\delta \in (0, 1)} L_0(\bar{p}^n) = 1 + \log \log e - \log e = \sigma$$

a constant which appears in the following theorem due to Gallager [3].

Theorem 7: Let p be the probability of the most likely letter from a finite discrete source. The redundancy of the Huffman code for this source is at most $p + \sigma$.

The above remarks provide a sequence of examples showing that σ is the smallest possible value for this theorem.

CONCLUSION

The standard proof of the noiseless source coding theorem shows that the redundancy is between 0 and 1. Campbell [1] has generalized this to the Rényi redundancy. The upper bound was obtained by the feasible solution known as Shannon coding. This paper has improved this bound by shortening some of the codewords in the Shannon code while still retaining feasibility. The lower bound was improved by randomizing between codeword lengths. This transformed the integer programming problem of minimizing the (exponentially weighted) average codeword length subject to the Kraft-McMillan inequality into a linear programming problem. The optimal feasible solution to this problem provided a lower bound but did not result in an implementable code. In the case of memoryless sources, these bounds were shown to approach each other asymptotically, providing a new proof and generalization of Kricevski's results [9]. In the case of a binary memoryless source with $s = 0$, this provided a sequence of examples showing that the constant in Gallager's theorem [3] is the best possible.

REFERENCES

- [1] L. L. Campbell, "A coding theorem and Rényi's entropy," *Inform. Contr.*, vol. 8, pp. 423-429, 1965.
- [2] ———, "The relation between information theory and the differential geometry approach to statistics," *Inform. Sci.*, vol. 35, pp. 199-210, 1985.
- [3] R. G. Gallager, "Variations on a theme by Huffman," *IEEE Trans. Inform. Theory*, vol. IT-24, no. 6, pp. 668-674, Nov. 1978.
- [4] G. H. Hardy and E. M. Wright, *An Introduction to the Theory of Numbers*, 4th ed. Oxford, England: Clarendon, 1971.
- [5] T. C. Hu, D. J. Kleitman, and J. T. Tamaki, "Binary trees optimal under various criteria," *SIAM J. Appl. Math.*, vol. 37, no. 2, pp. 246-256, Oct. 1979.
- [6] P. A. Humblet, "Generalization of Huffman coding to minimize the probability of buffer overflow," *IEEE Trans. Inform. Theory*, vol. IT-27, no. 2, pp. 230-232, Mar. 1981.
- [7] F. Jelinek, "Buffer overflow in variable length coding of fixed rate sources," *IEEE Trans. Inform. Theory*, vol. IT-14, pp. 490-501, May 1968.
- [8] D. E. Knuth, "The art of computer programming," in *Fundamental Algorithms*, vol. I, 2nd ed. Reading, MA: Addison-Wesley, 1973.
- [9] R. E. Kricevski, "The block length necessary to obtain a given redundancy," *Sov. Math. Dokl.*, vol. 7, no. 6, pp. 1416-1419, 1966.
- [10] R. J. McEliece, *The Theory of Information and Coding*, 2nd printing with revisions. Reading, MA: Addison-Wesley, 1979.
- [11] D. S. Parker, Jr., "Conditions for the optimality of the Huffman algorithm," *SIAM J. Comput.*, vol. 9, no. 3, pp. 470-489, Aug. 1980.
- [12] A. Rényi, "On measures of entropy and information," in *Proc. 4th Berk. Symp. Math. Stat. Prob.*, vol. 1, 1961, pp. 547-561.