Tight Bounds on Minimum Maximum Pointwise Redundancy

Michael B. Baer
vLnks
Mountain View, CA 94041-2803, USA
Email: calbear@ieee.org

***Abstract*— This paper presents new lower and upper bounds for the optimal compression of binary prefix codes in terms of the most probable input symbol, where compression efficiency is determined by the nonlinear codeword length objective of minimizing maximum pointwise redundancy. This objective relates to both universal modeling and Shannon coding, and these bounds are tight throughout the interval. The upper bounds also apply to a related objective, that of d^{th} exponential redundancy.**

I. Introduction

A lossless binary prefix coding problem takes a probability mass function $p(i)$, defined for all i in the input alphabet $\mathcal{X}$, and finds a binary code for $\mathcal{X}$. Without loss of generality, we consider an n-item source emitting symbols drawn from the alphabet $\mathcal{X} = \{1, 2, \ldots, n\}$ where $\{p(i)\}$ is the sequence of probabilities for possible symbols ($p(i) > 0$ for $i \in \mathcal{X}$ and $\sum_{i\in\mathcal{X}} p(i) = 1$) in monotonically nonincreasing order ($p(i) \geq p(j)$ for $i < j$). The source symbols are coded into binary codewords. The codeword $c(i) \in \{0, 1\}^*$ in code c, corresponding to input symbol i, has length $l(i)$, defining length vector $\boldsymbol{l}$.

The goal of the traditional coding problem is to find a prefix code minimizing expected codeword length $\sum_{i\in\mathcal{X}} p(i)l(i)$, or, equivalently, minimizing average redundancy

$$\bar{R}(\boldsymbol{l}, p) \triangleq \sum_{i\in\mathcal{X}} p(i)l(i) - H(p) = \sum_{i\in\mathcal{X}} p(i)(l(i) + \lg p(i))$$

where H is $-\sum_{i\in\mathcal{X}} p(i) \lg p(i)$, Shannon entropy, and $\lg \triangleq \log_2$. A prefix code is a code for which no codeword begins with a sequence that also comprises the whole of a second codeword. This problem is equivalent to finding a minimum-weight external path

$$\sum_{i\in\mathcal{X}} w(i)l(i)$$

among all rooted binary trees, due to the fact that every prefix code can be represented as a binary tree. In this tree representation, each edge from a parent node to a child node is labeled 0 (left) or 1 (right), with at most one of each type of edge per parent node. A leaf is a node without children; this corresponds to a codeword, and the codeword is determined by the path from the root to the leaf. Thus, for example, a leaf that is the right-edge (1) child of a left-edge (0) child of a left-edge (0) child of the root will correspond to codeword 001. Leaf depth (distance from the root) is thus codeword length. The weights are the probabilities (i.e., $w(i) = p(i)$), and, in fact, we will refer to the problem inputs as $\{w(i)\}$ for certain generalizations in which their sum, $\sum_{i\in\mathcal{X}} w(i)$, need not be 1.

If formulated in terms of $\boldsymbol{l}$, the constraints on the minimization are the integer constraint (i.e., that codes must be of integer length) and the Kraft inequality [1]; that is, the set of allowable codeword length vectors is

$$\mathcal{L}_n \triangleq \left\{ \boldsymbol{l} \in \mathbb{Z}_+^n \text{ such that } \sum_{i=1}^{n} 2^{-l(i)} \leq 1 \right\}.$$

Drmota and Szpankowski [2] investigated a problem which, instead of minimizing average redundancy $\bar{R}(\boldsymbol{l}, p) \triangleq \sum_{i\in\mathcal{X}} p(i)(l(i) + \lg p(i))$, minimizes maximum pointwise redundancy

$$R^*(\boldsymbol{l}, p) \triangleq \max_{i\in\mathcal{X}} (l(i) + \lg p(i)).$$

Related to a universal modeling problem [3, p. 176], the idea here is that, given a symbol to be compressed, we wish the length of the compressed data ($l(i)$) to exceed self-information ($-\lg p(i)$) by as little as possible, and thus consider the worst case in this regard. This naturally relates to Shannon coding, as a code with lengths $\lceil -\lg p(i) \rceil$ would never exceed self-information by more than 1 bit. Any solution, then, would necessarily have no codeword longer than its Shannon code counterpart. Indeed, Drmota and Szpankowski used a generalization of Shannon coding to solve the problem, which satisfies

$$0 \leq R^*(\boldsymbol{l}^{\text{opt}}, p) < 1.$$

We will improve the bounds, given $p(1)$, for minimum maximum pointwise redundancy and discuss the related issue of the length of the most likely codeword in these coding problems. These bounds are the first of their kind for this objective, analogous to those for traditional Huffman coding [4]–[9] and other nonlinear codes [10]–[12].

The bounds are derived using an alternative solution to this problem, a variation of Huffman coding [13] derived from that in [14]. In order to explain this variation, we first review the Huffman algorithm and some of the ways in which it can be modified.

It is well known that the Huffman algorithm [15] finds a code minimizing average redundancy. The Huffman algorithm is a greedy algorithm built on the observation that the two least likely symbols will have the same length and can thus

be considered siblings in the coding tree. A reduction can thus be made in which the two symbols with weights $w(i)$ and $w(j)$ can be considered as one with combined weight $w(i)+w(j)$, and the codeword of the combined item determines all but the last bit of each of the items combined, which are differentiated by this last bit. This reduction continues until there is one item left, and, assigning this item the null string, a code is defined for all input symbols. In the corresponding optimal code tree, the i^{th} leaf corresponds to the codeword of the i^{th} input item, and thus has weight $w(i)$, whereas the weight of parent nodes are determined by the combined weight of the corresponding merged item. Van Leeuwen gave an implementation of the Huffman algorithm that can be accomplished in linear time given sorted probabilities [16]. Shannon [17] had previously shown that an optimal $\boldsymbol{l}^{\text{opt}}$ must satisfy

$$H(p) \le \sum_{i\in\mathcal{X}} p(i) l^{\text{opt}}(i) < H(p)+1, \text{ i.e., } 0 \le \bar{R}(\boldsymbol{l}^{\text{opt}}, p) < 1.$$

Simple changes to the Huffman algorithm solve several related coding problems which optimize for different objectives. Generalized versions of the Huffman algorithm have been considered by many authors [18]–[21]. These generalizations change the combining rule; instead of replacing items i and j with an item of weight $w(i)+w(j)$, the generalized algorithm replaces them with an item of weight $f(w(i),w(j))$ for some function f. Thus the weight of a combined item (a node) no longer need be equal to the sum of the probabilities of the items merged to create it (the sum of the leaves of the corresponding subtree). This has the result that the sum of weights in a reduced problem need not be 1, unlike in the original Huffman algorithm. In particular, the weight of the root, w_{root}, need not be 1. However, we continue to assume that the sum of $p(\cdot)$, the inputs before reduction, will always be 1.

One such variation of the Huffman algorithm was used in Humblet's dissertation [22] for a queueing application (and further discussed in [18], [19], [23]). The problem this variation solves is as follows: Given probability mass function p and $a>1$, find a code minimizing

$$L_a(p,\boldsymbol{l}) \triangleq \log_a \sum_{i\in\mathcal{X}} p(i) a^{l(i)}. \tag{1}$$

This growing exponential average problem is solved by using combining rule

$$f(w(i),w(j)) = aw(i)+aw(j). \tag{2}$$

This problem was proposed (without solution) by Campbell [24], who later noted that this formulation can be extended to decaying exponential base $a\in(0,1)$ [25]; Humblet noted that the Huffman combining method (2) finds the optimal code for (1) with $a\in(0,1)$ as well [23].

Another variation, proposed in [26] and solved for in [19], can be called d^{th} exponential redundancy [13], and is the minimization of the following:

$$R^d(\boldsymbol{l},p) \triangleq \frac{1}{d}\lg \sum_{i\in\mathcal{X}} p(i)^{1+d} 2^{dl(i)}.$$

Here we assume that $d>0$, although $d\in(-1,0)$ is also a valid problem. Clearly, this can be solved via reduction to (1) by assigning $a=\lg d$ and using input weights $w(i)=p(i)^{1+d}$.

Minimizing maximum redundancy is equivalent to minimizing d^{th} exponential redundancy for $d\to\infty$. This observation leads to a Huffman-like solution with the combination rule

$$f(w(i),w(j)) = 2\max(w(i),w(j)) \tag{3}$$

as in [13].

In the next section, we find tight exhaustive bounds for the values of optimal $R^*(\boldsymbol{l},p)$ and corresponding $l(1)$ in terms of $p(1)$, then find how we can extend these to exhaustive — but not tight — bounds for optimal $R^d(\boldsymbol{l},p)$.

II. Bounds on the Redundancy Problems

It is useful to come up with bounds on the performance of an optimal code, often in terms of the most probable symbol, $p(1)$. In minimizing average redundancy, such bounds are often referred to as "redundancy bounds" because they are in terms of this average redundancy, $\bar{R}(\boldsymbol{l},p)=\sum_{i\in\mathcal{X}} p(i)l(i) - H(p)$. The simplest bounds for the optimal solution to the minimum maximum pointwise redundancy problem

$$R^*_{\text{opt}}(p) \triangleq \min_{\boldsymbol{l}\in\mathcal{L}_n} \max_{i\in\mathcal{X}} (l(i)+\lg p(i))$$

can be combined with those for the average redundancy problem:

$$0 \le \bar{R}_{\text{opt}}(p) \le R^*_{\text{opt}}(p) < 1 \tag{4}$$

where $\bar{R}_{\text{opt}}(p)$ is the average redundancy of the average redundancy-optimal code. The average redundancy case is a lower bound because the maximum ($R^*(\boldsymbol{l},p)$) of the values ($l(i)+\lg p(i)$) that average to a quantity ($\bar{R}(\boldsymbol{l},p)$) can be no less than the average (a fact that holds for all $\boldsymbol{l}$ and p). The upper bound is found similarly to the average redundancy case; we can note that Shannon code $l^0_p(i) \triangleq \lceil -\lg p(i)\rceil$ results in $R^*_{\text{opt}}(p) \le R^*(\boldsymbol{l}^0_p,p) = \max_{i\in\mathcal{X}} (\lceil -\lg p(i)\rceil + \lg p(i)) < 1$.

A few observations can be used to find a series of improved lower and upper bounds on optimum maximum pointwise redundancy based on (4):

Lemma 1: Suppose we apply (3) to find a Huffman-like code tree in order to minimize maximum pointwise redundancy. Then the following holds:

1) Items are always merged by nondecreasing weight.
2) The weight of the root w_{root} of the coding tree determines the maximum pointwise redundancy, $R^*(\boldsymbol{l},p) = \lg w_{\text{root}}$.
3) The total probability of any subtree is no greater than the total weight of the subtree.
4) If $p(1) \le 2p(n-1)$, then a minimum maximum pointwise redundancy code can be represented by a complete tree, that is, a tree with leaves at depth $\lfloor \lg n\rfloor$ and $\lceil \lg n\rceil$ only (with $\sum_{i\in\mathcal{X}} 2^{-l(i)} = 1$).

Proof: We use an inductive proof in which base cases of sizes 1 and 2 are trivial, and we use weights w, instead of probabilities p, to emphasize that the sums of weights need not necessarily add up to 1. Assume first that all properties

here are true for trees of size $n-1$ and smaller. We wish to show that they are true for trees of size n.

The first property is true because $f(w(i), w(j)) = 2\max(w(i), w(j)) > w(i)$ for any i and j; that is, a compound item always has greater weight than either of the items combined to form it. Thus, after the first two weights are combined, all remaining weights, including the compound weight, are no less than either of the two original weights.

Consider the second property; after merging the two least weighted of n (possibly merged) items, the property holds for the resulting $n-1$ items. For the $n-2$ untouched items, $l(i) + \lg w(i)$ remains the same. For the two merged items, let $l(n-1)$ and $w(n-1)$ denote the maximum depth/weight pair for item $n-1$ and $l(n)$ and $w(n)$ the pair for n. If l' and w' denote the depth/weight pair of the combined item, then $l'+\lg w' = l(n)-1+\lg(2\max(w(n-1), w(n))) = \max(l(n-1)+\lg w(n-1), l(n)+\lg w(n))$, so the two trees have identical maximum redundancy, which is equal to $\lg w_{\text{root}}$ since the root node is of depth 0. Consider, for example, $p = (0.5, 0.3, 0.2)$, which has optimal codewords with lengths $\boldsymbol{l} = (1, 2, 2)$. The first combined pair has $l' + \lg w' = 1 + \lg 0.6 = \max(2 + \lg 0.3, 2 + \lg 0.2) = \max(l(2) + \lg p(2), l(3) + \lg p(3))$. This value is identical to that of the maximum redundancy, $\lg 1.2 = \lg w_{\text{root}}$.

For the third property, the first combined pair yields a weight that is no less than the combined probabilities. Thus, via induction, the total probability of any (sub)tree is no greater than the weight of the (sub)tree.

In order to show the final property, first note that $\sum_{i\in\mathcal{X}} 2^{-l(i)} = 1$ for any tree created using the Huffman-like procedure, since all internal nodes have two children. Now think of the procedure as starting with a queue of input items, ordered by nondecreasing weight from head to tail. After merging two items, obtained from the head of the queue, into one compound item, that item is placed back into the queue as one item, but not necessarily at the tail; an item is placed such that its weight is no smaller than any item ahead of it and is smaller than any item behind it. In keeping items ordered, this results in an optimal coding tree. A variant of this method can be used for linear-time coding [13].

In this case, we show not only that an optimal complete tree exists, but that, given an n-item tree, all items that finish at level $\lceil \lg n \rceil$ appear closer to the head of the queue than any item at level $\lceil \lg n \rceil - 1$ (if any), using a similar approach to the proof of Lemma 2 in [27]. Suppose this is true for every case with $n-1$ items for $n > 2$, that is, that all nodes are at levels $\lfloor \lg(n-1) \rfloor$ or $\lceil \lg(n-1) \rceil$, with the latter items closer to the head of the queue than the former. Consider now a case with n nodes. The first step of coding is to merge two nodes, resulting in a combined item that is placed at the end of the combined-item queue, as we have asserted that $p(1) \le 2p(n-1) = 2\max(p(n-1), p(n))$. Because it is at the end of the queue in the $n-1$ case, this combined node is at level $\lfloor \lg(n-1) \rfloor$ in the final tree, and its children are at level $1 + \lfloor \lg(n-1) \rfloor = \lceil \lg n \rceil$. If n is a power of two, the remaining items end up on level $\lg n = \lceil \lg(n-1) \rceil$, satisfying this lemma. If $n-1$ is a power of two, they end up on level $\lg(n-1) = \lfloor \lg n \rfloor$, also satisfying the lemma. Otherwise, there is at least one item ending up at level $\lceil \lg n \rceil = \lceil \lg(n-1) \rceil$ near the head of the queue, followed by the remaining items, which end up at level $\lfloor \lg n \rfloor = \lfloor \lg(n-1) \rfloor$. In any case, all properties of the lemma are satisfied for n items, and thus for any number of items. ■

We can now present the improved redundancy bounds.

Theorem 1: For any distribution in which $p(1) \ge 2/3$, $R^*_{\text{opt}}(p) = 1 + \lg p(1)$. If $p(1) \in [0.5, 2/3)$, then $R^*_{\text{opt}}(p) \in [1+\lg p(1), 2+\lg(1-p(1)))$ and these bounds are tight. Define $\lambda \triangleq \lceil -\lg p(1) \rceil$, which, for $p(1) \in (0, 0.5)$, is greater than 1. For this range the following bounds for $R^*_{\text{opt}}(p)$ are tight:

$p(1)$	$R^*_{\text{opt}}(p)$
$\left[\frac{1}{2^\lambda}, \frac{1}{2^{\lambda-1}}\right)$	$\left[\lambda + \lg p(1), 1 + \lg \frac{1-p(1)}{1-2^{-\lambda}}\right)$
$\left[\frac{1}{2^{\lambda-1}}, \frac{2}{2^{\lambda+1}}\right)$	$\left[\lg \frac{1-p(1)}{1-2^{-\lambda+1}}, 1 + \lg \frac{1-p(1)}{1-2^{-\lambda}}\right)$
$\left[\frac{2}{2^{\lambda+1}}, \frac{1}{2^{\lambda-1}}\right)$	$\left[\lg \frac{1-p(1)}{1-2^{-\lambda+1}}, \lambda + \lg p(1)\right]$

Proof: The key here is generalizing the simple bounds of (4).

Upper bound: Let us define what we call a *first-order Shannon code*:

$$l^1_p(i) = \begin{cases} \lambda \triangleq \lceil -\lg p(1) \rceil, & i = 1 \\ \left\lceil -\lg\left(p(i)\left(\frac{1-2^{-\lambda}}{1-p(1)}\right)\right)\right\rceil, & i \in \{2, 3, \ldots, n\} \end{cases}$$

This code, previously presented in the context of finding *average* redundancy bounds given *any* probability [28], improves upon the original "zero-order" Shannon code l^0_p by taking the length of the first codeword into account when designing the rest of the code. The code satisfies the Kraft inequality, and thus, as a valid code, its redundancy is an upper bound on the redundancy of an optimal code. Note that

$$\begin{aligned} &\max_{i>1}(l^1_p(i) + \lg p(i)) \\ &\quad = \max_{i>1}\left(\left\lceil \lg \tfrac{1-p(1)}{p(i)(1-2^{-\lambda})} \right\rceil + \lg p(i)\right) \\ &\quad < 1 + \lg \tfrac{1-p(1)}{1-2^{-\lambda}}. \end{aligned}$$

If $p(1) \in [2/(2^\lambda + 1), 1/2^{\lambda-1})$, the maximum pointwise redundancy of the first item is no less than $1+\lg((1-p(1))/(1-2^{-\lambda}))$, and thus $R^*_{\text{opt}}(p) \le R^*(\boldsymbol{l}^1_p, p) = \lambda + \lg p(1)$. Otherwise, $R^*_{\text{opt}}(p) \le R^*(\boldsymbol{l}^1_p, p) < 1 + \lg((1-p(1))/(1-2^{-\lambda}))$.

The tightness of the upper bound in $[0.5, 1)$ is shown via

$$p = (p(1), 1 - p(1) - \epsilon, \epsilon)$$

for which the bound is achieved in $[2/3, 1)$ for any $\epsilon \in (0, (1-p(1))/2]$ and approached in $[0.5, 2/3)$ as $\epsilon \downarrow 0$. If $\lambda > 1$ and $p(1) \in [2/(2^\lambda + 1), 1/2^{\lambda-1})$, use probability mass function

$$p = \left(p(1), \underbrace{\frac{1-p(1)-\epsilon}{2^\lambda - 2}, \ldots, \frac{1-p(1)-\epsilon}{2^\lambda - 2}}_{2^\lambda - 2}, \epsilon\right)$$

where

$$\epsilon \in (0, 1 - p(1)2^{\lambda-1}).$$

Because $p(1) \geq 2/(2^\lambda + 1)$, $1 - p(1)2^{\lambda-1} \leq (1 - p(1) - \epsilon)/(2^\lambda - 2)$, and $p(n-1) \geq p(n)$. Similarly, $p(1) < 1/2^{\lambda-1}$ assures that $p(1) \geq p(2)$, so the probability mass function is monotonic. Since $2p(n-1) > p(1)$, by Lemma 1, an optimal code for this probability mass function is $l(i) = \lambda$ for all i, achieving $R^*(\boldsymbol{l}, p) = \lambda + \lg p(1)$, with item 1 having the maximum pointwise redundancy.

This leaves only $p(1) \in [1/2^\lambda, 2/(2^\lambda + 1))$, for which we consider

$$p = \left(p(1), \underbrace{\frac{1 - p(1) - \epsilon}{2^\lambda - 1}, \ldots, \frac{1 - p(1) - \epsilon}{2^\lambda - 1}}_{2^\lambda - 1}, \epsilon \right)$$

where $\epsilon \downarrow 0$. This is a monotonic probability mass function for sufficiently small ϵ, for which we also have $p(1) < 2p(n-1)$, so (again from Lemma 1) this results in optimal code where $l(i) = \lambda$ for $i \in \{1, 2, \ldots, n-2\}$ and $l(n-1) = l(n) = \lambda + 1$, and thus the bound is approached with item $n-1$ having the maximum pointwise redundancy.

Lower bound: Consider all optimal codes with $l(1) = \mu$ for some fixed $\mu \in \{1, 2, \ldots\}$. If $p(1) \geq 2^{-\mu}$, $R^*(\boldsymbol{l}, p) \geq l(1) + \lg p(1) = \mu + \lg p(1)$. If $p(1) < 2^{-\mu}$, consider the weights at level μ (i.e., μ edges below the root). One of these weights is $p(1)$, while the rest are known to sum to a number no less than $1 - p(1)$. Thus at least one weight must be at least $(1-p(1))/(2^\mu - 1)$ and $R^*(\boldsymbol{l}, p) \geq \mu + \lg((1-p(1))/(2^\mu - 1))$. Thus,

$$R^*_{\text{opt}}(p) \geq \mu + \lg \max \left(p(1), \frac{1 - p(1)}{2^\mu - 1} \right)$$

for $l(1) = \mu$, and, since μ can be any positive integer,

$$R^*_{\text{opt}}(p) \geq \min_{\mu \in \{1,2,3,\ldots\}} \left(\mu + \lg \max \left(p(1), \frac{1 - p(1)}{2^\mu - 1} \right) \right)$$

which is equivalent to the bounds provided.

For $p(1) \in [1/(2^{\mu+1} - 1), 1/2^\mu)$ for some μ, consider

$$\left(p(1), \underbrace{\frac{1 - p(1)}{2^{\mu+1} - 2}, \ldots, \frac{1 - p(1)}{2^{\mu+1} - 2}}_{2^{\mu+1} - 2} \right).$$

By Lemma 1, this will have a complete coding tree and thus achieve the lower bound for this range ($\lambda = \mu + 1$). Similarly

$$\left(p(1), \underbrace{2^{-\mu-1}, \ldots, 2^{-\mu-1}}_{2^{\mu+1} - 2}, 2^{-\mu} - p(1) \right)$$

has a fixed-length optimal coding tree for $p(1) \in [1/2^\mu, 1/(2^\mu - 1))$, achieving the lower bound for this range ($\lambda = \mu$). ■

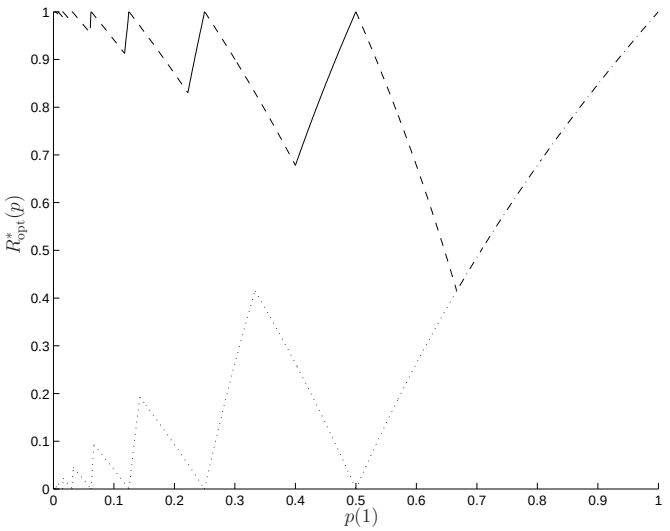

Fig. 1. Tight bounds on minimum maximum pointwise redundancy, including achievable upper bounds (solid), approachable upper bounds (dashed), achievable lower bounds (dotted), and fully determined values for $p(1) \geq 2/3$ (dot-dashed).

Note that the bounds of (4) are identical to the tight bounds at powers of two. In addition, the tight bounds clearly approach 0 and 1 as $p(1) \downarrow 0$. This behavior is in stark contrast with average redundancy, for which bounds get closer, not further apart, due to Gallager's redundancy bound [4] — $\bar{R}_{\text{opt}}(p) \leq p(1) + 0.086$ — which cannot be significantly improved for small $p(1)$ [9]. Moreover, approaching 1, the upper and lower bounds on minimum average redundancy coding converge but never merge, whereas the minimum maximum redundancy bounds are identical for $p(1) \geq 2/3$.

In addition to finding redundancy bounds in terms of $p(1)$, it is also often useful to find bounds on the behavior of $l(1)$ in terms of $p(1)$, as was done for optimal average redundancy in [29].

Theorem 2: Any optimal code for probability mass function p, where $p(1) \geq 2^{-\nu}$, must have $l(1) \leq \nu$. This bound is tight, in the sense that, for $p(1) < 2^{-\nu}$, one can always find a probability mass function with $l(1) > \nu$. Conversely, if $p(1) \leq 1/(2^\nu - 1)$, there is an optimal code with $l(1) \geq \nu$, and this bound is also tight.

Proof: Suppose $p(1) \geq 2^{-\nu}$ and $l(1) \geq 1 + \nu$. Then $R^*_{\text{opt}}(p) = R^*(\boldsymbol{l}, p) \geq l(1) + \lg p(1) \geq 1$, contradicting the simple bounds of (4). Thus $l(1) \leq \nu$.

For tightness of the bound, suppose $p(1) \in (2^{-\nu-1}, 2^{-\nu})$ and consider $n = 2^{\nu+1}$ and

$$p = \left(p(1), \underbrace{2^{-\nu-1}, \ldots, 2^{-\nu-1}}_{n-2}, 2^{-\nu} - p(1) \right).$$

If $l(1) \leq \nu$, then, by the Kraft inequality, one of $l(2)$ through $l(n-1)$ must exceed ν. However, this contradicts the simple bounds of (4). For $p(1) = 2^{-\nu-1}$, a uniform distribution results in $l(1) = \nu + 1$. Thus, since these two results hold

for any ν, this extends to all $p(1) < 2^{-\nu-1}$, and this bound is tight.

Suppose $p(1) \le 1/(2^\nu - 1)$ and consider an optimal length distribution with $l(1) < \nu$. Consider the weights of the nodes of the corresponding code tree at level $l(1)$. One of these weights is $p(1)$, while the rest are known to sum to a number no less than $1-p(1)$. Thus there is one node of at least weight

$$\frac{1-p(1)}{2^{l(1)}-1} \ge \frac{1-p(1)}{2^{l(1)}-2^{l(1)+1-\nu}}$$

and thus, taking the logarithm and adding $l(1)$ to the right-hand side,

$$R^*(\boldsymbol{l}, p) \ge \nu - 1 + \lg \frac{1-p(1)}{2^{\nu-1}-1}.$$

Note that $l(1) + 1 + \lg p(1) \le \nu + \lg p(1) \le \nu - 1 + \lg((1 - p(1))/(2^{\nu-1}-1))$, a direct consequence of $p(1) \le 1/(2^\nu - 1)$. Thus, if we replace this code with one for which $l(1) = \nu$, the code is still optimal. The tightness of the bound is easily seen by applying Lemma 1 to distributions of the form

$$p = \left(p(1), \underbrace{\frac{1-p(1)}{2^\nu - 2}, \ldots, \frac{1-p(1)}{2^\nu - 2}}_{2^\nu - 2} \right)$$

for $p(1) \in (1/(2^\nu - 1), 1/2^{\nu-1})$. This results in $l(1) = \nu - 1$ and thus $R^*_{\mathrm{opt}}(p) = \nu + \lg(1-p(1)) - \lg(2^\nu - 2)$, which no code with $l(1) > \nu - 1$ could achieve. ∎

In particular, if $p(1) \ge 0.5$, $l(1) = 1$, while if $l(1) \le 1/3$, there is an optimal code with $l(1) > 1$.

We now briefly address the d^{th} exponential redundancy problem. Recall that this is the minimization of

$$R^d(p, \boldsymbol{l}) \triangleq \frac{1}{d} \lg \sum_{i \in \mathcal{X}} p(i)^{1+d} 2^{dl(i)}.$$

This can be rewritten as

$$R^d(p, \boldsymbol{l}) = \frac{1}{d} \lg \sum_{i \in \mathcal{X}} p(i) 2^{d(l(i) + \lg p(i))}.$$

A straightforward application of Lyapunov's inequality for moments yields $R^c(p, \boldsymbol{l}) \le R^d(p, \boldsymbol{l})$ for $c \le d$, which, taking limits to 0 and ∞, results in

$$0 \le \bar{R}(p, \boldsymbol{l}) \le R^d(p, \boldsymbol{l}) \le R^*(p, \boldsymbol{l}) < 1$$

for any valid p, $d > 0$, and $\boldsymbol{l}$, resulting in an extension of (4),

$$0 \le \bar{R}_{\mathrm{opt}}(p) \le R^d_{\mathrm{opt}}(p) \le R^*_{\mathrm{opt}}(p) < 1$$

where $R^d_{\mathrm{opt}}(p)$ is the optimal d^{th} exponential redundancy, an improvement on the bounds found in [13]. This implies that this problem can be bounded in terms of the most likely symbol using the upper bounds of Theorem 1 and the lower bounds of average redundancy (Huffman) coding [7]:

$$\bar{R}_{\mathrm{opt}} \ge \xi - (1-p(1)) \lg(2^\xi - 1) - H(p(1), 1-p(1))$$

where

$$\xi = \left\lceil \lg \frac{1 - 2^{\frac{1}{p(1)-1}}}{1 - 2^{\frac{p(1)}{p(1)-1}}} \right\rceil$$

for $p(1) \in (0,1)$ (and, recall, $H(x) = -\sum_i x(i) \lg x(i)$).

References

[1] B. McMillan, "Two inequalities implied by unique decipherability," *IRE Trans. Inf. Theory*, vol. IT-2, no. 4, pp. 115–116, Dec. 1956.

[2] M. Drmota and W. Szpankowski, "Precise minimax redundancy and regret," *IEEE Trans. Inf. Theory*, vol. IT-50, no. 11, pp. 2686–2707, Nov. 2004.

[3] Y. M. Shtarkov, "Universal sequential coding of single messages," *Probl. Inform. Transm.*, vol. 23, no. 3, pp. 175–186, July–Sept. 1987.

[4] R. G. Gallager, "Variations on a theme by Huffman," *IEEE Trans. Inf. Theory*, vol. IT-24, no. 6, pp. 668–674, Nov. 1978.

[5] O. Johnsen, "On the redundancy of binary Huffman codes," *IEEE Trans. Inf. Theory*, vol. IT-26, no. 2, pp. 220–222, Mar. 1980.

[6] R. M. Capocelli, R. Giancarlo, and I. J. Taneja, "Bounds on the redundancy of Huffman codes," *IEEE Trans. Inf. Theory*, vol. IT-32, no. 6, pp. 854–857, Nov. 1986.

[7] B. L. Montgomery and J. Abrahams, "On the redundancy of optimal binary prefix-condition codes for finite and infinite sources," *IEEE Trans. Inf. Theory*, vol. IT-33, no. 1, pp. 156–160, Jan. 1987.

[8] R. M. Capocelli and A. De Santis, "Tight upper bounds on the redundancy of Huffman codes," *IEEE Trans. Inf. Theory*, vol. IT-35, no. 5, pp. 1084–1091, Sept. 1989.

[9] D. Manstetten, "Tight bounds on the redundancy of Huffman codes," *IEEE Trans. Inf. Theory*, vol. IT-37, no. 1, pp. 144–151, Jan. 1992.

[10] I. J. Taneja, "A short note on the redundancy of degree α," *Inf. Sci.*, vol. 39, no. 2, pp. 211–216, Sept. 1986.

[11] A. C. Blumer and R. J. McEliece, "The Rényi redundancy of generalized Huffman codes," *IEEE Trans. Inf. Theory*, vol. IT-34, no. 5, pp. 1242–1249, Sept. 1988.

[12] M. B. Baer, "Rényi to Rényi — source coding under siege," in *Proc., 2006 IEEE Int. Symp. on Information Theory*, July 9–14, 2006, pp. 1258–1262.

[13] ——, "A general framework for codes involving redundancy minimization," *IEEE Trans. Inf. Theory*, vol. IT-52, no. 1, pp. 344–349, Jan. 2006.

[14] M. C. Golumbic, "Combinatorial merging," *IEEE Trans. Comput.*, vol. C-25, no. 11, pp. 1164–1167, Nov. 1976.

[15] D. A. Huffman, "A method for the construction of minimum-redundancy codes," *Proc. IRE*, vol. 40, no. 9, pp. 1098–1101, Sept. 1952.

[16] J. van Leeuwen, "On the construction of Huffman trees," in *Proc. 3rd Int. Colloquium on Automata, Languages, and Programming*, July 1976, pp. 382–410.

[17] C. E. Shannon, "A mathematical theory of communication," *Bell Syst. Tech. J.*, vol. 27, pp. 379–423, July 1948.

[18] T. C. Hu, D. J. Kleitman, and J. K. Tamaki, "Binary trees optimum under various criteria," *SIAM J. Appl. Math.*, vol. 37, no. 2, pp. 246–256, Apr. 1979.

[19] D. S. Parker, Jr., "Conditions for optimality of the Huffman algorithm," *SIAM J. Comput.*, vol. 9, no. 3, pp. 470–489, Aug. 1980.

[20] D. E. Knuth, "Huffman's algorithm via algebra," *J. Comb. Theory, Ser. A*, vol. 32, pp. 216–224, 1982.

[21] C. Chang and J. Thomas, "Huffman algebras for independent random variables," *Disc. Event Dynamic Syst.*, vol. 4, no. 1, pp. 23–40, Feb. 1994.

[22] P. A. Humblet, "Source coding for communication concentrators," Ph.D. dissertation, Massachusetts Institute of Technology, 1978.

[23] ——, "Generalization of Huffman coding to minimize the probability of buffer overflow," *IEEE Trans. Inf. Theory*, vol. IT-27, no. 2, pp. 230–232, Mar. 1981.

[24] L. L. Campbell, "A coding problem and Rényi's entropy," *Inf. Contr.*, vol. 8, no. 4, pp. 423–429, Aug. 1965.

[25] ——, "Definition of entropy by means of a coding problem," *Z. Wahrscheinlichkeitstheorie und verwandte Gebiete*, vol. 6, pp. 113–118, 1966.

[26] P. Nath, "On a coding theorem connected with Rényi entropy," *Inf. Contr.*, vol. 29, no. 3, pp. 234–242, Nov. 1975.

[27] M. B. Baer, "Optimal prefix codes for infinite alphabets with nonlinear costs," *IEEE Trans. Inf. Theory*, vol. IT-54, no. 3, pp. 1273–1286, Mar. 2008.

[28] C. Ye and R. W. Yeung, "A simple bound of the redundancy of Huffman codes," *IEEE Trans. Inf. Theory*, vol. IT-48, no. 7, pp. 2132–2138, July 2002.

[29] R. M. Capocelli and A. De Santis, "A note on D-ary Huffman codes," *IEEE Trans. Inf. Theory*, vol. IT-37, no. 1, pp. 174–179, Jan. 1991.