On the Redundancy of Huffman Codes with Exponential Objectives

Michael B. Baer
Vista Research
4 Lower Ragsdale Drive, Suite 220
Monterey, CA, USA
Email: calbear@ieee.org

***Abstract*— We present new lower and upper bounds for the compression rate of binary prefix codes over memoryless sources optimized according to two related exponential codeword length objectives. The objectives explored here are exponential-average length and exponential-average redundancy. The first of these relates to various problems involving queueing, uncertainty, and lossless communications. It can be reduced to the second, which has properties more amenable to analysis. These bounds, some of which are tight, are in terms of a form of entropy and/or the probability of an input symbol, improving on recently discovered bounds of similar form. We also observe related properties of optimal codes over the exponential-average redundancy utility.**

I. Introduction

Among Shannon's many observations in the seminal paper on information theory [1] was that, by increasing block size, the compression rate of a block code for a memoryless source can get arbitrarily close to the source entropy rate. In particular, given a block of Shannon entropy H bits, prefix coding methods such as Huffman coding can code the block with an expected length of L bits, where $L \in [H, H+1)$. If $p_i \in (0,1)$ is the probability of the ith item, which has a codeword of length l_i, then

$$L \triangleq \sum_i p_i l_i \text{ and } H \triangleq -\sum_i p_i \lg p_i$$

where $\lg \triangleq \log_2$ and the sum is, without loss of generality, taken over the n possible items. A constant absolute difference translates into an arbitrarily close-to-entropy compression ratio as blocks grow in size without bound. The lower bound is fundamental to the definition of entropy, while the upper bound is easily seen by observing the suboptimal Shannon code: This code, that in which an event of probability p is coded into a codeword of length $\lceil -\lg p \rceil$, will always have expected length less than $H+1$, and thus the optimal $L < H+1$ as well.

This *unit-sized bound* holds even for many nonlinear optimization criteria. Such criteria are encountered in a variety of lossless compression problems in which expected length is no longer the value to minimize. In particular, consider

$$L_a = L_a(p, \boldsymbol{l}) \triangleq \log_a \sum_i p_i a^{l_i}. \tag{1}$$

Minimizing this utility — introduced in [2] — solves several problems involving avoiding buffer overflow in queueing [3], compression with uncertainty [4], one-shot communications [5], and unreliable communications [6]. It is closely related to Rényi entropy [7]:

$$H_\alpha(p) \triangleq \frac{1}{1-\alpha} \lg \sum_i p_i^\alpha \tag{2}$$

in the sense that, for $\alpha = 1/(1+\lg a)$,

$$H_\alpha(p) \le L_a^{\text{opt}} < H_\alpha(p) + 1.$$

Limits define Rényi entropy for 0, 1, and ∞, so that

$$\begin{aligned} H_0(p) &\triangleq \lim_{\alpha \downarrow 0} H_\alpha(p) = \lg \|p\| = \lg n \\ H_1(p) &\triangleq \lim_{\alpha \to 1} H_\alpha(p) = -\sum_i p_i \lg p_i \text{ (Shannon entropy)} \\ H_\infty(p) &\triangleq \lim_{\alpha \uparrow \infty} H_\alpha(p) = -\lg \max_i p_i. \end{aligned}$$

Over a constant p, entropy is nonincreasing over α [7].

L_a is also closely related to *exponential-average redundancy* or *exponential redundancy*

$$R^d(p, \boldsymbol{l}) \triangleq \frac{1}{d} \lg \sum_i p_i^{1+d} 2^{d l_i} = \frac{1}{d} \lg \sum_i p_i 2^{d(l_i + \lg p_i)}.$$

If we substitute $d = \lg a$ and

$$\hat{p}_i \triangleq \frac{p_i^\alpha}{\sum_k p_k^\alpha} = \frac{p_i^\alpha}{2^{(1-\alpha)H_\alpha(p)}}$$

we find

$$\begin{aligned} R^{\lg a}(\hat{p}, \boldsymbol{l}) &= \frac{1}{\lg a} \lg \sum_i \hat{p}_i^{1+\lg a} a^{l_i} \\ &= \log_a \sum_i p_i a^{l_i} - \log_a \left(\sum_i p_i^\alpha \right)^{\frac{1}{\alpha}} \\ &= L_a(p, \boldsymbol{l}) - H_\alpha(p). \end{aligned} \tag{3}$$

This transformation — shown in [8] — provides a reduction from L_a to R^d, allowing bounds for the former to apply — with the addition of the entropy term — to the latter.

However, for both the traditional and exponential utilities, we can improve on the unit-sized bound given the probability of one of the source events. Improving bounds is useful if Shannon's concept of increasing block size no longer applies. For example, in the case of queueing, increasing block size makes queue overflow more likely, while in the case of one-shot communications, there is only one event to encode.

There can also be more practical reasons for not using increasingly large block codes — e.g., restrictions on coding or simplicity of computation — so improved bounds have been looked into in depth in the case of the traditional linear utility, $L = L_1$. This was first done with the constraint that the given probability be the most probable of these events [9], but here, as in some subsequent work [6], [10], [11], we drop this constraint. Without loss of generality, we call the source symbols $\{1, 2, \ldots, n\} = \mathcal{X}$ (from most to least probable), and call the symbol with known probability j; that is, p_j is known, but not necessarily j itself.

In traditional linear optimization, upper and lower bounds for R^d are known such that probability distributions can be found achieving or approaching these bounds [10], [11]; i.e., they are tight. In the exponential cases, [6] took $a \uparrow \infty$ ($d \uparrow \infty$) and $a \downarrow 1$ ($d \downarrow 0$), using inequality relations to find not-necessarily-tight bounds on these problems in terms of tight bounds for the limit cases. The goal here is to improve the bounds.

We seek to find an upper bound $\omega^d(p_j)$ and lower bound $o^d(p_j)$ such that, for every probability distribution p, optimal codeword lengths $\boldsymbol{l}$ satisfy:

$$0 \leq o^d(p_j) \leq \min_{\boldsymbol{l}} R^d(p, \boldsymbol{l}) < \omega^d(p_j) \leq 1$$

for any j. For such values, (3) results in:

$$\begin{aligned} o^{\lg a}\left(p_j^{\tilde{\alpha}} 2^{(\tilde{\alpha}-1)H_{\tilde{\alpha}}(p)}\right) &\leq L_a^{\mathrm{opt}}(p) - H_{\tilde{\alpha}}(p) \\ &< \omega^{\lg a}\left(p_j^{\tilde{\alpha}} 2^{(\tilde{\alpha}-1)H_{\tilde{\alpha}}(p)}\right) \end{aligned}$$

where $\tilde{\alpha} = 1/(1 + \lg a)$ and $L_a^{\mathrm{opt}}(p)$ denotes the utility for optimal lengths given p and a. Thus we can restrict ourselves to exponential redundancy, which is more amenable to the analysis used here.

The bounds found here are given in the following as theorems, with Fig. 1 illustrating the bounds. As mentioned, many prior bounds for the traditional case further assume that the known probability is the most probable, i.e., $j = 1$. Although not assuming this results in a slightly more general problem, for $p_j > 0.5$, clearly the two are equivalent.

II. Applications

A. $d > 0$ ($a > 1$)

Most applications of the exponential length utility concern only $a > 1$ ($d > 0$ for the redundancy equivalent). The first known application, introduced in Humblet's dissertation [3], [12], is in a queueing problem originally posed by Jelinek [13]. Codewords coding a random source are temporarily stored in a finite buffer; these are chosen such that overflow probability is minimized.

Another application considers a source with uncertain probabilities, one in which we only know that the relative entropy between the actual probability mass function and p is within a known bound [4]. A third, more recent application, omitted in the interest of brevity but described in [6], is a modified case of the application in the next paragraph.

B. $d < 0$ ($a < 1$)

An application for $a < 1$ involves single-shot communications with a communication channel having a window of opportunity of geometrically-distributed length (in bits) [5]. If the distribution has parameter a, the probability of successful transmission is

$$\mathbb{P}[\text{success}] = a^{L_a(p,\boldsymbol{l})} = \sum_{i=1}^{n} p_i a^{l_i}.$$

Maximizing this is equivalent to minimizing (1). The solution is trivial for $a \leq 0.5$ ($d \leq -1$), a case not covered by Rényi entropy, and thus not applicable here.

III. Bounds

The variation of the Huffman algorithm which finds an optimal code for exponential redundancy differs as follows: While Huffman coding inductively pairs the two lowest probabilities (weights) w_x and w_y, combining them into an item *weighted* $f(w_x, w_y) \triangleq w_x + w_y$, optimizing exponential redundancy requires the combined item to be weight

$$f^d(w_x, w_y) \triangleq \left(2^d w_x^{1+d} + 2^d w_y^{1+d}\right)^{\frac{1}{1+d}}. \quad (4)$$

The optimality of this is shown in [14] and can illustrated with an exchange argument (e.g., [15, pp. 124-125] for the linear case). An exchange argument also inductively illustrates that such an algorithm, depending on how ties are broken, can achieve *any* optimal set of codeword lengths: Clearly the only optimal code is obtained for $n = 2$. Let n' be the smallest n for which there is a set of $\{l_i\}$ that is optimal but cannot be obtained via the algorithm. Since $\{l_i\}$ is optimal, consider the two smallest probabilities, $p_{n'}$ and $p_{n'-1}$. In this optimal code, two items having these probabilities (although not necessarily items $n' - 1$ and n') must have the longest codewords and must have the same codeword lengths. Otherwise, we could exchange the codeword with a longer codeword corresponding to a more probable item and improve the utility function, showing nonoptimality. Merge these two items into one with probability $f^d(p_{n'}, p_{n'-1})$, as per the algorithm. Because of the nature of f^d, this is a reduced problem, i.e., an equivalent optimization to the original problem. This means that there is a set of lengths optimal for this problem such that all non-merged items are identical to the corresponding l_i, while the merged item is simply one shorter than the longest l_i. Since we inductively assumed all optimal length sets could be produced for $n' - 1$, the assumption is verified for all n.

Related observations form the following theorem, similar to that in [6] for a non-exponential utility:

Theorem 1: Suppose we apply (4) to find a Huffman-like code tree in order to minimize exponential redundancy $R^d(p, \boldsymbol{l})$ for $d > -1$. Then the following holds for any optimal $\boldsymbol{l}$:

1) For $d > 0$, items are always merged by nondecreasing weight and the total probability of any subtree is no greater than the weight of the (root of the) subtree. For

$d < 0$, the total probability of any subtree is no less than the weight of the subtree.

2) The weight of the root of the coding tree is $w_{\text{root}} = 2^{R^d(p,\boldsymbol{l})}$.
3) If $p_1 \leq f^d(p_{n-1}, p_n)$, then an optimal 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 $\sum_i 2^{-l_i} = 1$).

Proof: Again we use induction, this time using trivial base cases of sizes 1 and 2, and assuming the propositions true for sizes $n-1$ and smaller. We assume without loss of generality that, for size n, items $n-1$ and n are the first to be merged. We use weight terminology (w) instead of probabilities (p) because reduced problems need not have weights sum to 1.

The subtree part of the first property considers subtrees of size n, not necessarily the whole coding tree. All we need to have a successful reduction to size $n-1$ is to show:

$$
\begin{aligned}
f^d(w_x, w_y) &= \left(2^d w_x^{1+d} + 2^d w_y^{1+d}\right)^{\frac{1}{1+d}} && (5)\\
&\geq w_x + w_y && (6)
\end{aligned}
$$

for $d > 0$, and

$$f^d(w_x, w_y) \leq w_x + w_y \tag{7}$$

for $d \in (-1, 0)$, with equality in either case if and only if $w_x = w_y$. The inequalities are due to the identical property of the generalized mean in [16, 3.2.4]:

$$M(t) = \left(\frac{1}{m}\sum_{k=1}^{m} a_k^t\right)^{\frac{1}{t}}$$

with, in this case, $m = 2$, $a_1 = 2w_x$, $a_2 = 2w_y$, and t as $1+d$ in (5) (left-hand side of (7)) and 1 on (6) (right-hand side of (7)).

It immediately follows in the $d > 0$ case that $f^d(w_x, w_y) > w_x$. Thus, the first two merged weights of the tree form a weight no less than either original weight, and all remaining weights are also no less than those two weights. Call the resulting lengths $\boldsymbol{l}'$.

To prove the second property, note that, after merging the aforementioned two least weighted items, we have $n-1$ weights, and thus a conforming reduced problem. Call the combined weight w'_{c}. Then

$$
\begin{aligned}
w_{\text{root}} &= 2^{R^d(p,\boldsymbol{l}')}\\
&= \left({w'_{\text{c}}}^{1+d} 2^{(l_n-1)d} + \sum_{i=1}^{n-2} p_i^{1+d} 2^{l_i d}\right)^{\frac{1}{d}}\\
&= \left(p_{n-1}^{1+d} 2^{l_{n-1} d} + p_n^{1+d} 2^{l_n d} + \sum_{i=1}^{n-2} p_i^{1+d} 2^{l_i d}\right)^{\frac{1}{d}}\\
&= 2^{R^d(p,\boldsymbol{l})}
\end{aligned}
$$

where the third equality is due to $l_{n-1} = l_n$ and (4).

The third property is shown via the operation of the algorithm from start to finish: First note that $\sum_i 2^{-l_i} = 1$ 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 priority queue of input items, ordered by nondecreasing weight from head to tail. After merging two items, obtained from the head, into one compound item, that item is placed back into the queue. Since we are using a priority queue, the merged 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, we obtain an optimal coding tree. A first derivative test shows that f^d is nondecreasing on both inputs for any d. Thus merged items are created in nondecreasing weight. If $p_1 \leq f^d(p_{n-1}, p_n)$, the first merged item can be inserted to the tail of the queue; since merged items are created in nondecreasing weight, subsequent items are as well. This is a sufficient condition for a complete tree being optimal [5, Lemma 2]. ■

Next is our main result:

Theorem 2: Suppose we know $d > -1$ ($d \neq 0$) and one p_j of probability mass function p for which we want to find the optimal code $\boldsymbol{l}$ under exponential redundancy. Consider

$$\omega^d(p_j) = \min_{\lambda \in \mathbb{Z}^+} \left(\lambda + \frac{1}{d}\lg\left(p_j^{1+d} + \frac{2^d(1-p_j)^{1+d}}{(2^\lambda - 1)^d}\right)\right) \tag{8}$$

making transitions between λ and $\lambda+1$ at

$$p_\lambda = \left(1 + \left(\left(1-2^{-d}\right)\left(\tfrac{1}{(2^\lambda-1)^d} - \tfrac{1}{(2^\lambda-0.5)^d}\right)^{-1}\right)^{\frac{1}{1+d}}\right)^{-1}$$

and

$$o^d(p_j) = \min_{\mu \in \mathbb{Z}^+} \left(\mu + \frac{1}{d}\lg\left(p_j^{1+d} + \frac{(1-p)^{1+d}}{(2^\mu - 1)^d}\right)\right)$$

with transitions between μ and $\mu+1$ at

$$p_\mu = \left(1 + \left(\left(2^d-1\right)\left(\tfrac{1}{(2^\mu-1)^d} - \tfrac{1}{(2^\mu-0.5)^d}\right)^{-1}\right)^{\frac{1}{1+d}}\right)^{-1}.$$

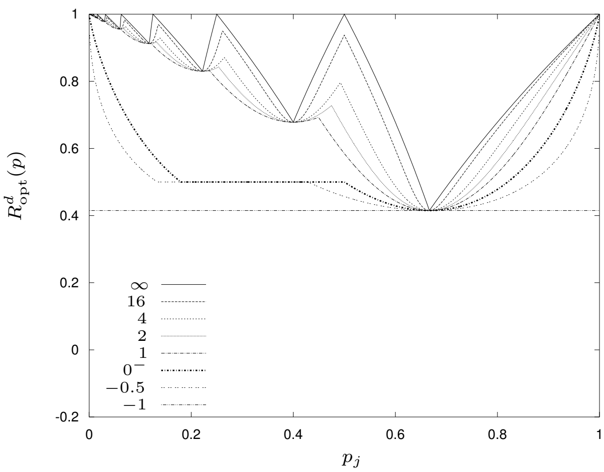
These improve bounds on the optimal code, and the upper bound is a strict inequality, in that

$$0 \leq o^d(p_j) \leq R^d(p,\boldsymbol{l}) < \omega^d(p_j) \leq 1. \tag{9}$$

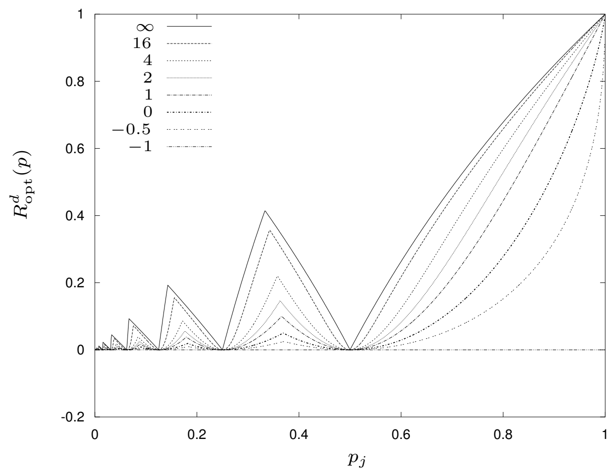
The lower bounds are achievable given p_1 and the upper bounds are approachable given $p_1 \geq 0.5$. Also, for $p_j < 0.5$ and $d < 0$, there is a secondary upper bound:

$$R^d(p,\boldsymbol{l}) < \max\left(0.5, \frac{1}{d}\lg\left(p_j^{1+d}4^d + (1-p_j)^{1+d}2^d\right)\right) \tag{10}$$

Proof:



(a) Upper bounds



(b) Lower bounds

Fig. 1. Bounds on optimal $R^d_{\rm opt}(p)$ given p_j over various d (see legends). The thick (dash-dotted) lines correspond to the usual linear redundancy utility ($d \to 0$), while the uppermost (solid) lines are minimum maximum pointwise redundancy ($d \to \infty$). Lower bounds are tight over all $d > -1$, while upper bounds are only tight for minimum maximum pointwise redundancy, for $p_j \geq 0.5$ if $d \in (-1, \infty)$, and for $(0, \pi_0^d)$ if $d \in (-1, 0)$, where π_0^d as the first root of the equality of the two terms in the maximization at (10). The tight upper bounds for $d < \infty$ are approached by $p = (p_j, 1-p_j-\epsilon, \epsilon)$. This also shows the tightness of the $d \to -1$ bounds of $(0, 2-\lg 3) \approx (0, 0.415)$, which, although calculated using (9), are not dependent on p_j.

1) Lower bound: The lower bound calculation is:

$$\begin{aligned} R^d(p, \boldsymbol{l}) &\overset{\text{(a)}}{=} \frac{1}{d} \lg \Bigg(p_j^{1+d} 2^{dl_j} + (1-p_j)^{1+d} 2^{dl_n} \\ &\qquad \cdot \sum_{i \in \mathcal{X} \setminus \{j\}} \sum_{k=1}^{2^{l_n - l_i}} \left(\frac{p_i 2^{l_i - l_n}}{1-p_j} \right)^{1+d} \Bigg) \\ &\overset{\text{(b)}}{\geq} \frac{1}{d} \lg \Bigg(p_j^{1+d} 2^{dl_j} \\ &\qquad + (1-p_j)^{1+d} 2^{dl_n} \left(2^{l_n} - 2^{l_n - l_j} \right)^{-d} \Bigg) \\ &= l_j + \frac{1}{d} \lg \left(p_j^{1+d} + (1-p_j)^{1+d} \left(2^{l_j} - 1 \right)^{-d} \right). \end{aligned}$$

The final equality follows from algebra. The summation following (a) is a sum of the $(1+d)$ power of $2^{l_n} - 2^{l_n - l_j}$ positive terms which sum to 1. Consider these values, which include $2^{l_n - l_i}$ repetitions of each $p_i 2^{l_i - l_n}/(1-p_j)$ for $i \neq j$, as a probability distribution called q. Then the summation is related to the $(1+d)$-Rényi entropy of q; substituting using its definition (2) leads to (11) below. Furthermore, because $H_0(q) = \lg \|q\|$ and H_α is nonincreasing with α:

$$\begin{aligned} \left(\sum_{m=1}^{2^{l_n} - 2^{l_n - l_j}} q_m^{1+d} \right)^{\frac{1}{d}} &= 2^{-H_{1+d}(q)} \qquad (11) \\ &\geq 2^{-\lg \|q\|} = (2^{l_n} - 2^{l_n - l_j})^{-1}. \end{aligned}$$

This results in inequality (b), completing the lower bound by substituting minimizing μ for l_j. The transitions follow from algebraically finding where there are two minimizing values.

A code achieving this lower bound, for $p_1 = p_j \in [1/(2^{\mu+1}-1), 1/2^\mu)$ for some μ, is

$$\Bigg(p_1, \underbrace{\frac{1-p_1}{2^{\mu+1}-2}, \ldots, \frac{1-p_1}{2^{\mu+1}-2}}_{2^{\mu+1}-2} \Bigg).$$

By Theorem 1, this has a complete coding tree — recall $f^d(w_x, w_x) = 2w_x$ — in this case with l_1 one bit shorter than the other lengths. This is easily calculated as achieving the lower bound.

2) Upper bounds: Given λ, define, as in [10]:

$$l_i^j(p) = \begin{cases} \lambda & \text{if } i = j \\ \left\lceil -\lg \left(p_i \left(\frac{1 - 2^{-\lambda}}{1-p_j} \right) \right) \right\rceil & \text{if } i \neq j. \end{cases}$$

Satisfying the Kraft inequality, this possibly suboptimal code has a utility upper-bounding that of the optimal code:

$$\begin{aligned} R^d(p, \boldsymbol{l}) &\leq \frac{1}{d} \lg \Bigg(p_j^{1+d} 2^{d\lambda} \\ &\qquad + \sum_{i \in \mathcal{X} \setminus \{j\}} p_i^{1+d} 2^{d \left\lceil -\lg\left(p_i (1-2^{-\lambda})/(1-p_j) \right) \right\rceil} \Bigg) \\ &< \frac{1}{d} \lg \Bigg(p_j^{1+d} 2^{d\lambda} \\ &\qquad + \sum_{i \in \mathcal{X} \setminus \{j\}} p_i^{1+d} \left(\frac{p_i}{2} \cdot \frac{1-2^{-\lambda}}{1-p_j} \right)^{-d} \Bigg) \\ &= \frac{1}{d} \lg \left(p_j^{1+d} 2^{d\lambda} + (1-p_j)^{1+d} \left(\frac{2}{1-2^{-\lambda}} \right)^d \right) \end{aligned}$$

Since λ is arbitrary, the bound is obtained by choosing the value offering the strictest bound. This upper bound is approached for any $d > -1$ over $p_1 = p_j \in (0.5, 1)$ for $p = (p_j, 1-p_j-\epsilon, \epsilon)$ (i.e., $j = 1$ and $\lambda = 1$).

Now consider $d < 0$ and $p_j < 0.5$. As noted in [6], an application of Lyapunov's inequality for moments [17, p. 27] yields $R^{d'}(p, \boldsymbol{l}) \leq R^d(p, \boldsymbol{l})$ for $d' \leq d$, and, in particular, $R^d(p, \boldsymbol{l}) \leq R^0(p, \boldsymbol{l})$ in this case, where

$$R^0(p, \boldsymbol{l}) = \sum_{i \in \mathcal{X}} p_i l_i - H_1(p)$$

via limits. Since this is true for all values, it is true over the minimization, and bounds for the usual linear case apply here. In particular, as found in [18] and noted in [11], if we define

$$f(p_1) = \begin{cases} 3 - 5p_1 - H_1(2p_1) & \text{if } \pi_1 \leq p_1 < 0.5 \\ 2 - \lg 3 & \text{if } 0 < p_1 < \pi_1 \end{cases} \tag{12}$$

where $\pi_1 \approx 0.491$ is the root of the equality of the two terms, then this serves as an upper bound (given most probable p_1) on optimal redundancy (linear, and thus also $d < 0$) in $(0, 0.5)$.

Since this never exceeds the bound we seek here, we can now consider only $p_j < p_1$. Consider first those cases in which (10) is greater than 0.5. In these cases, we use the fact that $p_1 \in [p_j, 1-p_j]$ to note that the maximum upper bound over this range — using (8) and (12) — is $\omega^d(p_1)$ at $p_1 = 1 - p_j$, thus supplying the upper bound for the range $(0, \pi_0^d)$, where π_0^d is the first root of the equality of the two terms in the maximization at (10).

Over $p_j \in (\pi_0^d, 0.5)$, we first note that 0.5 is an upper bound via similar logic: If $p_1 \leq 0.5$, we already know that this is an upper bound. Otherwise $p_1 \in (0.5, 1-\pi_0^d)$, and (8) using $j = 1$ provides an upper bound not exceeding 0.5. ■

Fig. 1 illustrates these bounds at a handful of d values, and at limits -1, 0, and ∞. For $d \to 0$, l'Hôpital's rule reveals the lower bound to be the optimal one of Theorem 2 of [19] for $j = 1$ and Theorem 4 of [11] for arbitrary j. If one replaces optimal λ with (possibly suboptimal) $\lceil -\lg p_j \rceil$, the upper bound becomes the suboptimal one of Lemma 1 of [10]. Taking $d \to \infty$ using, for any positive x, y, a, b,

$$\lim_{d \to \infty} \frac{1}{d} \lg(xa^d + yb^d) = \lg \max(a, b)$$

yields the minimum maximum pointwise redundancy bounds of [6], which are both tight.

The upper bound is clearly not optimal here, since it is not optimal for $d \to 0$ from either direction. In particular, the flat portion for $d < 0$ — the 0.5 term in secondary bound (10) — is not flat in the optimum bounds for traditional Huffman coding. However, the following might be of help in improving this in future work:

Theorem 3: If $d < 0$ and $p_1 \geq 0.4$, an optimal code exists with $l_1 = 1$.

Proof: The approach here is similar to [20]. Consider the coding step at which item 1 gets combined with other items; we wish to prove that this is the last step. At the beginning of this step the (possibly merged) items left to combine are $\{1\}, S_2^k, S_3^k, \ldots, S_k^k$, where we use S_j^k to denote the set of (individual) items combined into a (possibly) compound item, and $w(S_j^k)$ to denote its weight. At this step, p_1 is smaller than all but possibly one of S_j^k, so $(k-1)p_1 \geq (k-1)0.4$ is less than the sum of weights, which in turn is less than or equal to 1. Thus k is at most three.

Consider items $\{1\}$, S_2^3, and S_3^3. Assume without loss of generality that $w(S_2^3) \geq w(S_3^3)$. If $w(S_2^3)$ is not compound, $\{1\}$ has the greatest weight and we are finished. If it is compound, call its two subtrees S_3^4 and S_4^4, in order of nonincreasing weight. Clearly $w(S_3^4) \leq w(S_3^3)$ due to the combination order, so $w(S_2^3) \leq 2w(S_3^3)$. Thus $1.5w(S_2^3) \leq w(S_2^3) + w(S_3^3) \leq 0.6$, so $w(S_3^3) \leq w(S_2^3) \leq 0.4$, and we can combine these two items to achieve the optimal code. This is tight in the sense that $(p_1, (1-p_1)/3, (1-p_1)/3, (1-p_1)/3)$ has $l_1 = 2$ for $p_1 \in (0.25, 0.4)$. ■

References

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

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

[3] P. A. Humblet, "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.

[4] F. Rezaei and C. D. Charalambous, "Robust coding for uncertain sources: A minimax approach," in *Proc., 2005 IEEE Int. Symp. on Information Theory*, Sept. 4–9, 2005, pp. 1539–1543.

[5] 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.

[6] ——, "Redundancy-related bounds for generalized Huffman codes," *IEEE Trans. Inf. Theory*, vol. IT-57, no. 4, pp. 2278–2290, Apr. 2011.

[7] A. Rényi, "Some fundamental questions of information theory," *Magyar Tudományos Akadémia III. Osztalyanak Közlemenei*, vol. 10, no. 1, pp. 251–282, 1960.

[8] 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.

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

[10] 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.

[11] S. Mohajer, S. Pakzad, and A. Kakhbod, "Tight bounds on the redundancy of Huffman codes," in *Proc., IEEE Information Theory Workshop*, Mar. 13–17, 2006, pp. 131–135.

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

[13] F. Jelinek, "Buffer overflow in variable length coding of fixed rate sources," *IEEE Trans. Inf. Theory*, vol. IT-14, no. 3, pp. 490–501, May 1968.

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

[15] T. Cover and J. Thomas, *Elements of Information Theory*, 2nd ed. New York, NY: Wiley-Interscience, 2006.

[16] M. Abramowitz and I. A. Stegun, *Handbook of Mathematical Functions with Formulas, Graphs, and Mathematical Tables*. Mineola, NY: Dover Publications, 1964.

[17] G. H. Hardy, J. E. Littlewood, and G. Pólya, *Inequalities*. Cambridge, UK: Cambridge Univ. Press, 1934.

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

[19] 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.

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