Redundancy-Related Bounds for Generalized Huffman Codes

Michael B. Baer, *Member, IEEE*

***Abstract*— This paper presents new lower and upper bounds for the compression rate of binary prefix codes optimized over memoryless sources according to various nonlinear codeword length objectives. Like the most well-known redundancy bounds for minimum average redundancy coding — Huffman coding — these are in terms of a form of entropy and/or the probability of an input symbol, often the most probable one. The bounds here, some of which are tight, improve on known bounds of the form $L \in [H, H+1)$, where H is some form of entropy in bits (or, in the case of redundancy objectives, 0) and L is the length objective, also in bits. The objectives explored here include exponential-average length, maximum pointwise redundancy, and exponential-average pointwise redundancy (also called d^{th} exponential redundancy). The first of these relates to various problems involving queueing, uncertainty, and lossless communications; the second relates to problems involving Shannon coding and universal modeling. For these two objectives we also explore the related problem of the necessary and sufficient conditions for the shortest codeword of a code being a specific length.**

***Index Terms*— Huffman codes, optimal prefix code, queueing, Rényi entropy, Shannon codes, worst case minimax redundancy.**

I. Introduction

Since Shannon introduced information theory, we have had entropy bounds for the expected codeword length of optimal lossless fixed-to-variable-length binary codes. The lower bound is entropy, while the upper bound is one bit greater — corresponding to a maximum average redundancy of one bit for an optimal code — thus yielding *unit-sized* bounds. The upper bound follows from the suboptimal Shannon code, a code for which the codeword length of an input of probability p is $\lceil -\log_2 p \rceil$ [1].

Material in this paper was presented at the 2006 International Symposium on Information Theory, Seattle, Washington, USA and the 2008 International Symposium on Information Theory, Toronto, Ontario, Canada.

The author is with Vista Research, 4 Lower Ragsdale Drive, Suite 220, Monterey, California 93940 USA (email: calbear@ieee.org).

Huffman found a method of producing an optimal code by building a tree in which the two nodes with lowest weight (probability) are merged to produce a node with their combined weight summed [2]. On the occasion of the twenty-fifth anniversary of the Huffman algorithm, Gallager introduced bounds in terms of the most probable symbol which improved on the unit-sized redundancy bound [3]. Since then, improvements in both upper and lower bounds given this most probable symbol [4]–[8] have yielded bounds that are tight when this symbol's probability is at least $1/127$ (and close-to-tight when it has lower probability). Tight bounds also exist for upper and lower bounds given the less-specific information of an *arbitrary* symbol's probability [9], [10]. Such bounds are useful for quickly bounding the performance of an optimal code without running the algorithm that would produce the code. The bounds are for a fixed-sized input alphabet; *asymptotic* treatment of redundancy for block codes of growing size, based on *binary* memoryless sources, is given in [11].

Others have given consideration to objectives other than expected codeword length [12, §2.6]. Many of these nonlinear objectives, which have a variety of applications, also have unit-sized bounds but have heretofore lacked tighter closed-form bounds achieved using a symbol probability and, if necessary, some form of entropy. We address such problems here, finding upper and lower bounds for the optimal codes of given probability mass functions for nonlinear objectives. "Optimal" in this paper refers to optimality over the objective in question, not necessarily over the linear objective of expectation.

A lossless binary prefix coding problem takes a probability mass function $\boldsymbol{p} = \{p_i\}$, 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 $i < j$). Thus the most probable symbol is p_1. The source symbols are coded into binary codewords. The codeword $c_i \in \{0,1\}^*$ in code $\boldsymbol{c}$, 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

$$\bar{R}(\boldsymbol{l},\boldsymbol{p}) \triangleq \sum_{i\in\mathcal{X}} p_i l_i - H(\boldsymbol{p}) = \sum_{i\in\mathcal{X}} p_i(l_i + \lg p_i) \qquad (1)$$

where H is $-\sum_{i\in\mathcal{X}} p_i \lg p_i$ (Shannon entropy) and $\lg \triangleq \log_2$. A *prefix code* — also referred to as a *comma-free code*, a *prefix-free code*, or an *instantaneous 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 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. If we represent external path weight as $\sum_{i\in\mathcal{X}} w(i) l_i$, the weights are the probabilities (i.e., $w(i) = p_i$), and, in fact, we 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 [13]; that is, the set of allowable codeword length vectors is

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

Because Huffman's algorithm [2] finds codes minimizing average redundancy (1), the *minimum-average redundancy problem* itself is often referred to as the "*Huffman problem*," even though the problem did not originate with Huffman himself. Huffman's algorithm is a greedy algorithm built on the observation that the two least likely items will have the same length and can thus be considered siblings in the coding tree. A reduction is thus made in which the two items of weights $w(i)$ and $w(j)$ are considered as one with combined weight $w(i) + w(j)$. 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 items. 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.

We began by stating that an optimal $\boldsymbol{l}^{\text{opt}}$ must satisfy

$$H(\boldsymbol{p}) \leq \sum_{i\in\mathcal{X}} p_i l_i^{\text{opt}} < H(\boldsymbol{p}) + 1$$

or, equivalently,

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

Less well known is that simple changes to the Huffman algorithm solve several related coding problems which optimize for different objectives. We discuss three such problems, all three of which have been previously shown to satisfy redundancy bounds for optimal $\tilde{\boldsymbol{l}}$ of the form

$$\tilde{H}(\boldsymbol{p}) \leq \tilde{L}(\boldsymbol{p}, \tilde{\boldsymbol{l}}) < \tilde{H}(\boldsymbol{p}) + 1$$

or

$$0 \leq \tilde{R}(\tilde{\boldsymbol{l}}, \boldsymbol{p}) < 1$$

for some entropy measure $\tilde{H}$ and cost measure $\tilde{L}$, or for some redundancy measure $\tilde{R}$.

In this paper, we improve these bounds in a similar manner to improvements made to the Huffman problem: Given p_1, the probability of the most likely item, the Huffman problem improvements find functions $\bar{o}(p_1)$ and/or $\bar{\omega}(p_1)$ such that

$$0 \leq \bar{o}(p_1) \leq \bar{R}(\boldsymbol{l}^{\text{opt}}, \boldsymbol{p}) \leq \bar{\omega}(p_1) < 1.$$

The smallest $\bar{\omega}$, tight over most p_1, is given in [8], while a tight $\bar{o}$ is given in [6]. Tight bounds given any value p_j in $\boldsymbol{p}$, would yield alternative functions $\bar{o}'(p_j)$ and $\bar{\omega}'(p_j)$ such that

$$0 \leq \bar{o}'(p_j) \leq \bar{R}(\boldsymbol{l}^{\text{opt}}, \boldsymbol{p}) \leq \bar{\omega}'(p_j) < 1.$$

In this case, tight bounds are given by [10], which also addresses lower bounds given the *least* probable symbol, which we do not consider here.

In the following, we wish to find functions $\tilde{o}$, $\tilde{\omega}$, $\tilde{o}'$, and/or $\tilde{\omega}'$ such that

$$0 \leq \tilde{o}(p_1) \leq \tilde{R}(\tilde{\boldsymbol{l}}, \boldsymbol{p}) \leq \tilde{\omega}(p_1) \leq 1$$

and/or

$$0 \leq \tilde{o}'(p_j) \leq \tilde{R}(\tilde{\boldsymbol{l}}, \boldsymbol{p}) \leq \tilde{\omega}'(p_j) \leq 1$$

in the case of redundancy objectives, and to find $\tilde{\tilde{o}}$, $\tilde{\tilde{\omega}}$, $\tilde{\tilde{o}}'$, and/or $\tilde{\tilde{\omega}}'$ such that

$$0 \leq \tilde{\tilde{o}}(\tilde{H}(\boldsymbol{p}), p_1) \leq \tilde{L}(\boldsymbol{p}, \tilde{\boldsymbol{l}}) \leq \tilde{\tilde{\omega}}(\tilde{H}(\boldsymbol{p}), p_1) \leq 1$$

and/or

$$0 \leq \tilde{\tilde{o}}'(\tilde{H}(\boldsymbol{p}), p_j) \leq \tilde{L}(\boldsymbol{p}, \tilde{\boldsymbol{l}}) \leq \tilde{\tilde{\omega}}'(\tilde{H}(\boldsymbol{p}), p_j) \leq 1$$

in the case of other length objectives.

All of the nonlinear objectives we consider have been shown to be solved by generalized versions of the Huffman algorithm [14]–[18]. 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. The weight of a combined item (a node) therefore need not be equal to the sum of the probabilities of the items merged to create it (the sum of the leaves of the corresponding subtree). Thus 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 *inputs* to the coding problems will be 1 (with the exception of reductions among problems).

The next section introduces the objectives of interest, along with their motivations and our main contributions. These contributions, indicated by (☏), are bounds on performance of optimal codes according to their optimizing objectives, as well as related properties. We defer the formal presentation of these contributions, along with proofs, until later sections, where they are presented as theorems and corollaries, along with the remarks immediately following them and associated figures. These begin in Section III, where we find tight exhaustive bounds for the values of minimized maximum pointwise redundancy (2) and corresponding l_j in terms of p_j. Pointwise redundancy for a symbol i is $l_i + \lg p_i$. In Section IV, we then extend these to exhaustive — but not tight — bounds for minimized d^{th} exponential redundancy (4), a measure which takes a β-exponential average [19] of pointwise redundancy (where, in this case, parameter β is d). In Section V, we investigate the behavior of codes with minimized exponential average (6), including bounds and optimizing l_1 in terms of p_1.

II. Objectives, Motivations, and Main Results

A. Maximum pointwise redundancy (R^*)

The most recently proposed problem objective we consider is that formulated by Drmota and Szpankowski [20]. Instead of minimizing average redundancy $\bar{R}(\boldsymbol{l}, \boldsymbol{p}) \triangleq \sum_{i \in \mathcal{X}} p_i(l_i + \lg p_i)$, here we minimize maximum pointwise redundancy

$$R^*(\boldsymbol{l}, \boldsymbol{p}) \triangleq \max_{i \in \mathcal{X}} (l_i + \lg p_i). \tag{2}$$

An extension of Shannon coding introduced by Blumer and McEliece [21, p. 1244] to upper-bound the problem considered in Section II-C was later rediscovered and efficiently implemented by Drmota and Szpankowski as a solution to this maximum pointwise redundancy problem. A subsequent solution to the problem is a variation of Huffman coding [18] derived from that in [22], one using combining rule

$$f^*(w(i), w(j)) \triangleq 2 \max(w(i), w(j)). \tag{3}$$

Applications in prior literature: The solution of this worst-case pointwise redundancy problem is relevant to optimizing maximal (worst-case) minimax redundancy, a universal modeling problem (as in [23, p. 176]) for which the set $\mathcal{P}$ of possible probability distributions results in a normalized "maximum likelihood distribution." [20] More recently Gawrychowski and Gagie proposed a second worst-case redundancy problem which also finds its solution in minimizing maximum pointwise redundancy [24]. For this problem, normalization is not relevant and one allows any probability distribution that is consistent with an empirical distribution based on sampling.

Prior and current results: The first proposed algorithm for the maximum pointwise redundancy problem is a codeword-wise improvement on the Shannon code in the sense that each codeword is the same length as or one bit shorter than the corresponding codeword in the Shannon code. This method is called "generalized Shannon coding." (With proper tie-breaking techniques, the Huffman-like solution guarantees that each codeword, in turn, is no longer than the generalized Shannon codeword. As both methods guarantee optimality, this makes no difference in the maximum pointwise redundancy.) Notice a property true of Shannon codes — generalized or not — but not minimum average redundancy (Huffman) codes: Because any given codeword has a length l_i not exceeding $\lceil -\lg p_i \rceil$, this length is within one bit of the associated input symbol's self-information, $-\lg p_i$. This results in bounds of $R^*_{\text{opt}}(\boldsymbol{p}) \in [0, 1)$, which are improved upon in Section III. The bound can also be considered a degenerative case from a result of Shtarkov [23, p. 176], that for which the probabilities are fully known.

The aforementioned papers contain further analysis of this problem, but no improved closed-form bounds of the type introduced here. Here results are given as strict upper and lower bounds in Theorem 1 in Section III-A. Specifically, whether considering known p_j in general or known p_1 in particular, this problem has upper bound

$$\omega'^*(p_j) = \max\left(1 + \lg \frac{1 - p_j}{1 - 2^{-\lambda_j}}, \lambda_j + \lg p_j\right) \quad (☏)$$

where this and $\omega^*(p_1) = \omega'^*(p_1)$ are tight bounds. Also, it has lower bound

$$o'^*(p_j) = \min\left(\lambda_j + \lg p_j, \lg \frac{1 - p_j}{1 - 2^{-\lambda_j + 1}}\right) \quad (☏)$$

for $p_j < 0.5$, and $1 + \lg p_j$ otherwise, and, again, this and $o'^*(p_j) = o^*(p_j)$ are tight. Here $\lambda_j \triangleq \lceil -\lg p_j \rceil$, the results are illustrated in Fig. 1, and the values for which the maximum and minimum apply are given in the theorem (i.e., the bounds are tight).

Further results here include those regarding codeword length; Theorem 2 states that any optimal code will have $l_j \leq \nu$ if $p_j \geq 2^{-\nu}$ and that any probability distribution with $p_j \leq 1/(2^\nu - 1)$ will be optimized by at least one code with $l_j \geq \nu$.

B. d^{th} exponential redundancy (R^d)

A spectrum of problems bridges the objective of Huffman coding with the objective optimized by generalized Shannon coding using an objective proposed in [25] and solved for in [15]. In this particular context, the range of problems, parameterized by a variable d, can be called d^{th} exponential redundancy [18]. Such problems involve the minimization of the following:

$$R^d(\boldsymbol{p}, \boldsymbol{l}) \triangleq \frac{1}{d} \lg \sum_{i \in \mathcal{X}} p_i^{1+d} 2^{dl_i} = \frac{1}{d} \lg \sum_{i \in \mathcal{X}} p_i 2^{d(l_i + \lg p_i)}. \tag{4}$$

Although positive parameter d is the case we consider most often here, $d \in (-1, 0)$ is also a valid minimization problem. If we let $d \to 0$, we approach the average redundancy (Huffman's objective), while $d \to \infty$ is maximum pointwise redundancy [18]. The combining rule, introduced in [15, p. 486], is

$$f^d(w(i), w(j)) \triangleq \left(2^d w(i)^{1+d} + 2^d w(j)^{1+d}\right)^{\frac{1}{1+d}}. \tag{5}$$

Prior and current results and applications: This redundancy objective is less analyzed than the others mentioned here, likely because there are no direct applications in the published literature. However, it is closely related not only to average redundancy and to maximum pointwise redundancy, but also to the more applicable objective considered in Section II-C. Solution properties of these objectives — including redundancy bounds — can therefore be related via d^{th} exponential redundancy. In particular, as we show in Section IV, the upper bound for maximum pointwise redundancy also improves upon the already-known bound for d^{th} exponential redundancy,

$$R^d_{\text{opt}}(\boldsymbol{p}) \triangleq \min_{\boldsymbol{l} \in \mathcal{L}_n} R^d_{\text{opt}}(\boldsymbol{l}, \boldsymbol{p}) \in [0, 1).$$

Given $d > 0$, we show in Corollary 1 that any upper bound on minimax pointwise redundancy and any lower bound on minimum average redundancy serve to bound d^{th} exponential redundancy.

Specifically, consider ω^* given above (identical to ω'^*) and lower bound on minimum average redundancy $\bar{o}$ given in the literature [6] (identical to $\bar{o}'$ [10]). For any probability p_j in input distribution $\boldsymbol{p}$ and any $d > 0$,

$$\bar{o}(p_j) \leq R^d_{\text{opt}}(\boldsymbol{p}) \leq \omega^*(p_j) \tag{☜}$$

as illustrated in Fig. 2 and detailed in the corollary. Furthermore, any upper bound on minimum average redundancy — e.g., $\bar{\omega}(p_1)$ [8] or $\bar{\omega}'(p_j)$ [9], [10] — similarly bounds d^{th} exponential redundancy with $d \in (-1, 0)$.

C. Exponential average (L_a)

A related problem is one proposed by Campbell [26], [27]. This exponential problem, given probability mass function $\boldsymbol{p}$ and $a \in (0, \infty) \backslash 1$, is to find a code minimizing

$$L_a(\boldsymbol{p}, \boldsymbol{l}) \triangleq \log_a \sum_{i \in \mathcal{X}} p_i a^{l_i}. \tag{6}$$

In this case our parameter a is the base, rather than the exponential scaling factor, although much prior work does express this problem in the equivalent alternative form,

$$L_a(\boldsymbol{p}, \boldsymbol{l}) = \frac{1}{(\lg a)} \lg \sum_{i \in \mathcal{X}} p_i 2^{(\lg a) l_i}.$$

The solution to this [14], [15], [28] uses combining rule

$$f_a(w(i), w(j)) \triangleq aw(i) + aw(j). \tag{7}$$

A change of variables transforms the d^{th} exponential redundancy problem into (6) by assigning $a = \lg d$ and using input weights $w(i)$ proportional to p_i^{1+d}, which yields (5). We illustrate this precisely in Section IV in (18), which we use in Section V to find initial improved entropy bounds. These are supplemented by additional bounds for problems with $a \in (0.5, 1)$ and $p_1 \geq 2a/(2a + 3)$ (as illustrated in Fig. 3 at the end of Section V).

Applications ***($a < 1$):*** It is important to note here that $a > 1$ is an average of growing exponentials, while $a < 1$ is an average of decaying exponentials. These two subproblems have different properties and have often been considered separately in the literature. An application for the decaying exponential variant involving single-shot communications has a communication channel with a window of opportunity of a total duration (in bits) distributed geometrically with parameter a [29]. The probability of successful transmission is

$$\mathbb{P}[\text{success}] = a^{L_a(\boldsymbol{p}, \boldsymbol{l})} = \sum_{i=1}^{n} p_i a^{l_i}. \tag{8}$$

For $a > 0.5$, the unit-sized bound we improve upon is in terms of Rényi entropy, as in (16); the solution is trivial for $a \leq 0.5$, as we note at the start of Section V.

Applications *($a > 1$):* We add an additional observation on a modified version of this problem: Suppose there are a sequence of windows of opportunities rather than only one. The probability that a window stays open long enough to send a message of length l_i is a^{l_i}, since each additional bit has independent probability a of getting through. Thus, given l_i, the expected number of windows needed to send a message — assuming it is necessary to resend communication for each window — is the multiplicative inverse of this. Overall expectation is therefore

$$\mathbb{E}[N] = \sum_{i=1}^{n} p_i a^{-l_i} = a^{-L_{a^{-1}}(\boldsymbol{p},\boldsymbol{l})}.$$

Although such a resending of communications is usually not needed for a constant message, this problem is a notable dual to the first problem. In this dual problem, we seek to minimize the expectation of a growing exponential of lengths rather than maximize the expectation of a decaying exponential.

Originally, the $a > 1$ variation of (6) was used in Humblet's dissertation [30] for a queueing application originally proposed by Jelinek [31] and expounded upon in [21]. This problem is one in which overflow probability should be minimized, where the source produces symbols randomly and the codewords are temporarily stored in a finite buffer. In this problem, there exists an $a > 1$ such that optimizing $L_a(\boldsymbol{p}, \boldsymbol{l})$ optimizes this problem; the correct a is found through iteration. The Huffman-like coding method was simultaneously published in [14], [15], [28]; in the last of these, Humblet noted that the Huffman combining method (7) finds the optimal code with $a \in (0, 1)$ as well.

More recently, the $a > 1$ variation was shown to have a third application [32]. In this problem, the true probability of the source is not known; it is only known that the relative entropy between the true probability and $\boldsymbol{p}$ is within some known bound. As in Humblet's queueing problem, there is an $a > 1$, found via iteration, such that optimizing $L_a(\boldsymbol{p}, \boldsymbol{l})$ solves the problem.

Prior and current results*:* Note that $a \to 1$ corresponds to the usual linear expectation objective. Problems for a near 1 are of special interest, since $a \downarrow 1$ corresponds to the minimum-variance solution if the problem has multiple solutions — as noted in [15], among others — while $a \uparrow 1$ corresponds to the maximum-variance solution.

Most of the aforementioned improved bounds are based on a given highest symbol probability, p_1. We thus give this case special attention and also discuss the related property of the length of the most likely codeword in these coding problems. The bounds in this paper are the first of their kind for nontraditional Huffman codes, bounds which are, for L_a, functions of both entropy and p_1, as in the traditional case. However, they are not the first improved bounds for such codes. More sophisticated bounds on the optimal solution for the exponential-average objective are given in [21] for $a > 1$; these appear as solutions to related problems rather than in closed form, however, and these problems require no less time or space to solve than the original problem. They are mainly useful for analysis. Bounds given elsewhere for a closely related objective having a one-to-one correspondence with (6) are demonstrated under the assumption that $p_1 \geq 0.4$ always implies l_1 can be 1 for the optimal code [33]. We show that this is not necessarily the case due to the difference between the exponential-average objective and the usual objective of an arithmetic average.

Specifically, Theorem 3 states that, for $a \in (0.5, 1]$, a code with shortest codeword of length 1 is optimal if $p_1 \geq 2a/(2a+3)$. Furthermore, for $a > 1$, no value of $p_1 \in (0, 1)$ guarantees $l_1 = 1$, and, for $a \leq 0.5$, there is always an optimal code with $l_1 = 1$, regardless of the input distribution. This results in the improved bounds of Corollary 3; when $a \in (0.5, 1)$ and $p_1 \geq 2a/(2a+3)$, optimal $\boldsymbol{l}$ satisfies

$$\begin{aligned} a^2 \left[a^{\alpha H_\alpha(\boldsymbol{p})} - p_1^\alpha\right]^{\frac{1}{\alpha}} + ap_1 < \left(\sum_{i=1}^{n} p_i a^{l_i}\right) \\ \leq a \left[a^{\alpha H_\alpha(\boldsymbol{p})} - p_1^\alpha\right]^{\frac{1}{\alpha}} + ap_1 \end{aligned} \qquad (☜)$$

where $\alpha = 1/(1 + \lg a)$, and Rényi entropy

$$H_\alpha(\boldsymbol{p}) \triangleq \frac{1}{1-\alpha} \lg \sum_{i=1}^{n} p_i^\alpha.$$

This is an improvement on the unit-sized bounds,

$$H_\alpha(\boldsymbol{p}) \leq \log_a \sum_{i \in \mathcal{X}} p_i a^{l_i} < H_\alpha(\boldsymbol{p}) + 1.$$

In addition, we show in Corollary 2 that a reduction from this problem to d^{th} exponential redundancy extends the nontrivial bounds for the redundancy utility to nontrivial bounds for any $a > 0.5$, resulting in

$$0 \leq L_a^{\text{opt}}(\boldsymbol{p}) - H_\alpha(\boldsymbol{p}) \leq \bar{\omega}\left(p_1^\alpha 2^{(\alpha-1)H_\alpha(\boldsymbol{p})}\right) \qquad (☜)$$

and

$$0 \leq L_a^{\text{opt}}(\boldsymbol{p}) - H_\alpha(\boldsymbol{p}) \leq \bar{\omega}'\left(p_j^\alpha 2^{(\alpha-1)H_\alpha(\boldsymbol{p})}\right) \qquad (☜)$$

for $a \in (0.5, 1)$ and

$$\begin{aligned} \bar{o}\left(p_j^\alpha 2^{(\alpha-1)H_\alpha(\boldsymbol{p})}\right) &\leq L_a^{\text{opt}}(\boldsymbol{p}) - H_\alpha(\boldsymbol{p}) \\ &\leq \omega^*\left(p_j^\alpha 2^{(\alpha-1)H_\alpha(\boldsymbol{p})}\right) \end{aligned} \qquad (☜)$$

for $a > 1$.

III. Maximum Pointwise Redundancy

A. Maximum pointwise redundancy bounds

Shannon found redundancy bounds for $\bar{R}_{\rm opt}(\boldsymbol{p})$, the average redundancy $\bar{R}(\boldsymbol{l},\boldsymbol{p}) = \sum_{i\in\mathcal{X}} p_i l_i - H(\boldsymbol{p})$ of the average redundancy-optimal $\boldsymbol{l}$. The simplest bounds for minimized maximum pointwise redundancy

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

are quite similar to and can be combined with Shannon's bounds as follows:

$$0 \le \bar{R}_{\rm opt}(\boldsymbol{p}) \le R^*_{\rm opt}(\boldsymbol{p}) < 1 \tag{9}$$

The average redundancy case is a lower bound because the maximum ($R^*(\boldsymbol{l},\boldsymbol{p})$) of the values ($l_i + \lg p_i$) that average to a quantity ($\bar{R}(\boldsymbol{l},\boldsymbol{p})$) can be no less than the average (a fact that holds for all $\boldsymbol{l}$ and $\boldsymbol{p}$). The upper bound is due to Shannon code $l^0_i(\boldsymbol{p}) \triangleq \lceil -\lg p_i \rceil$ resulting in

$$R^*_{\rm opt}(\boldsymbol{p}) \le R^*(\boldsymbol{l}^0(\boldsymbol{p}),\boldsymbol{p}) = \max_{i\in\mathcal{X}} (\lceil -\lg p_i \rceil + \lg p_i) < 1.$$

A few observations can be used to find a series of improved lower and upper bounds on optimum maximum pointwise redundancy based on (9):

***Properties, Maximum Pointwise Redundancy**:*

Lemma 1: Suppose we apply (3) to find a Huffman-like code tree in order to minimize maximum pointwise redundancy ($\bar{R}(\boldsymbol{l},\boldsymbol{p})$ given $\boldsymbol{p}$). Then the following holds:

1) Items are always merged by nondecreasing weight.
2) The weight of the root $w_{\rm root}$ of the coding tree determines the maximum pointwise redundancy, $R^*(\boldsymbol{l},\boldsymbol{p}) = \lg w_{\rm root}$.
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 $\sum_{i\in\mathcal{X}} 2^{-l_i} = 1$). (This property is similar to the property noted in [34] for optimal-expected-length codes of sources termed *quasi-uniform* in [35].)

Proof: We use an inductive proof in which base cases of sizes 1 and 2 are trivial, and we use weight function w instead of probability mass function $\boldsymbol{p}$ 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

$$\begin{aligned} l' + \lg w' &= l_n - 1 + \lg(2\max(w(n-1), w(n))) \\ &= \max(l_{n-1} + \lg w(n-1), l_n + \lg w(n)). \end{aligned}$$

Thus the two trees have identical maximum redundancy, which is equal to $\lg w_{\rm root}$ 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

$$\begin{aligned} l' + \lg w' = 1 + \lg 0.6 &= \max(2 + \lg 0.3, 2 + \lg 0.2) \\ &= \max(l_2 + \lg p_2, l_3 + \lg p_3). \end{aligned}$$

This value is identical to that of the maximum redundancy, $\lg 1.2 = \lg w_{\rm root}$.

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 (priority) 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 [18].

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 [29]. 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.

Bounds, Maximum Pointwise Redundancy:

Theorem 1: For any distribution in which there exists a $p_j \ge 2/3$, $R^*_{\rm opt}(\boldsymbol{p}) = 1 + \lg p_j$. If $p_j \in [0.5, 2/3)$, then $R^*_{\rm opt}(\boldsymbol{p}) \in [1 + \lg p_j, 2 + \lg(1-p_j))$ and these bounds are tight not only for general p_j, but for p_1, in the sense that we can find probability mass functions with the given $p_1 = p_j$ achieving the lower bound and approaching the upper bound. Define $\lambda_j \triangleq \lceil -\lg p_j \rceil$. Thus λ_j satisfies $p_j \in [2^{-\lambda_j}, 2^{-\lambda_j+1})$, and $\lambda_j > 1$ for $p_j \in (0, 0.5)$; in this range, the following bounds for $R^*_{\rm opt}(\boldsymbol{p})$ are tight for general p_j and p_1 in particular:

p_j	$R^*_{\rm opt}(\boldsymbol{p})$
$\left[\frac{1}{2^{\lambda_j}}, \frac{1}{2^{\lambda_j}-1}\right)$	$\left[\lambda_j + \lg p_j, 1 + \lg \frac{1-p_j}{1-2^{-\lambda_j}}\right)$
$\left[\frac{1}{2^{\lambda_j}-1}, \frac{2}{2^{\lambda_j}+1}\right)$	$\left[\lg \frac{1-p_j}{1-2^{-\lambda_j+1}}, 1 + \lg \frac{1-p_j}{1-2^{-\lambda_j}}\right)$
$\left[\frac{2}{2^{\lambda_j}+1}, \frac{1}{2^{\lambda_j-1}}\right)$	$\left[\lg \frac{1-p_j}{1-2^{-\lambda_j+1}}, \lambda_j + \lg p_j\right]$

Proof: The key here is generalizing the unit-sized bounds of (9).

1) Upper bound: Before we prove the upper bound, note that, once proven, the tightness of the upper bound in $[0.5, 1)$ is shown via

$$\boldsymbol{p} = (p_j, 1 - p_j - \epsilon, \epsilon)$$

for which the bound is achieved in $[2/3, 1)$ for any $\epsilon \in (0, (1-p_j)/2]$ and approached in $[0.5, 2/3)$ as $\epsilon \downarrow 0$.

Let us define what we call a *j-Shannon code*:

$$l^j_i(\boldsymbol{p}) = \begin{cases} \lambda_j \triangleq \lceil -\lg p_j \rceil, & i = j \\ \left\lceil -\lg\left(p_i\left(\frac{1-2^{-\lambda_j}}{1-p_j}\right)\right)\right\rceil, & i \ne j \end{cases}$$

This code was previously presented in the context of finding *average* redundancy bounds given any probability [9]. Here it improves upon the original Shannon code $\boldsymbol{l}^0(\boldsymbol{p})$ by making the length l^j_j of the $j^{\rm th}$ known codeword λ_j, and taking this length 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

$$\begin{aligned} &\max_{i \ne j}(l^j_i(\boldsymbol{p}) + \lg p_i) \\ &\quad = \max_{i \ne j}\left(\left\lceil \lg \frac{1-p_j}{p_i(1-2^{-\lambda_j})}\right\rceil + \lg p_i\right) \qquad (10) \\ &\quad < 1 + \lg \frac{1-p_j}{1-2^{-\lambda_j}}. \end{aligned}$$

There are two cases:

a) $p_j \in [2/(2^{\lambda_j}+1), 1/2^{\lambda_j-1})$: In this case, the maximum pointwise redundancy of the item j in code $l^j(\boldsymbol{p})$ is no less than $1 + \lg((1-p_j)/(1-2^{-\lambda_j}))$. Thus, due to (11),

$$R^*_{\rm opt}(\boldsymbol{p}) \le R^*(\boldsymbol{l}^j(\boldsymbol{p}), \boldsymbol{p}) = \lambda_j + \lg p_j.$$

If $\lambda_j > 1$ and $p_j \in [2/(2^{\lambda_j}+1), 1/2^{\lambda_j-1})$, consider $j = 1$ and probability mass function

$$\boldsymbol{p} = \left(p_1, \underbrace{\frac{1-p_1-\epsilon}{2^{\lambda_1}-2}, \ldots, \frac{1-p_1-\epsilon}{2^{\lambda_1}-2}}_{2^{\lambda_1}-2}, \epsilon\right)$$

where $\epsilon \in (0, 1 - p_1 2^{\lambda_1 - 1})$. Because $p_1 \ge 2/(2^{\lambda_1}+1)$,

$$1 - p_1 2^{\lambda_1 - 1} \le (1 - p_1 - \epsilon)/(2^{\lambda_1} - 2)$$

and $p_{n-1} \ge p_n$. Similarly, $p_1 < 1/2^{\lambda_1 - 1}$ assures that $p_1 \ge p_2$, 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 $l_i = \lambda_1$ for all i, achieving $R^*(\boldsymbol{l}, \boldsymbol{p}) = \lambda_1 + \lg p_1$. Since $j = 1$ has the maximum pointwise redundancy, this upper bound is tight whether considering p_1 or general p_j.

b) $p_j \in [1/2^{\lambda_j}, 2/(2^{\lambda_j}+1))$: In this case, (11) immediately results in

$$R^*_{\rm opt}(\boldsymbol{p}) \le R^*(\boldsymbol{l}^j(\boldsymbol{p}), \boldsymbol{p}) < 1 + \lg((1-p_j)/(1-2^{-\lambda_j})).$$

Again considering $j = 1$, the probability mass function

$$\boldsymbol{p} = \left(p_1, \underbrace{\frac{1-p_1-\epsilon}{2^{\lambda_1}-1}, \ldots, \frac{1-p_1-\epsilon}{2^{\lambda_1}-1}}_{2^{\lambda_1}-1}, \epsilon\right)$$

illustrates the tightness of this bound for $\epsilon \downarrow 0$. For sufficiently small ϵ, this probability mass function is monotonic and $p_1 < 2p_{n-1}$. Lemma 1 then indicates that an optimal code has $l_i = \lambda_1$ for $i \in \{1, 2, \ldots, n-2\}$ and $l_{n-1} = l_n = \lambda_1 + 1$. Thus the bound is approached with item $n-1$ having the maximum pointwise redundancy.

2) Lower bound: Here we first address the lower bound given p_1. Consider all optimal codes with $l_1 = \mu$ for some fixed $\mu \in \{1, 2, \ldots\}$. If $p_1 \geq 2^{-\mu}$, $R^*(\boldsymbol{l}, \boldsymbol{p}) \geq l_1 + \lg p_1 = \mu + \lg p_1$. 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 $R^*(\boldsymbol{l}, \boldsymbol{p}) \geq \mu + \lg((1 - p_1)/(2^\mu - 1))$. Thus,

$$R^*_{\rm opt}(\boldsymbol{p}) \geq \mu + \lg \max\left(p_1, \frac{1 - p_1}{2^\mu - 1}\right)$$

for $l_1 = \mu$, and since this can be any positive integer,

$$R^*_{\rm opt}(\boldsymbol{p}) \geq \min_{\mu \in \{1,2,3,\ldots\}} \left(\mu + \lg \max\left(p_1, \frac{1 - p_1}{2^\mu - 1}\right)\right)$$

which is equivalent to the bounds provided.

For arbitrary p_j, the approach is similar, but a modification is required to the above when $p_j < 2^{-\mu}$; we are no longer guaranteed to have 2^μ nodes on level μ (where $\mu = l_j$). Instead, consider the set of leaves $\mathcal{A}$ above level μ and the set of nodes $\mathcal{N}$ on level μ. Let $\mathcal{A}'$ be a set of nodes (not actually in the optimal tree) such that, for each leaf i in $\mathcal{A}$, there are $2^{\mu - l_i}$ nodes in $\mathcal{A}'$, each one having weight $p_i 2^{l_i - \mu}$. Thus the combined probability of $\mathcal{A}'$ remains the same and the combined weight of $\mathcal{A}'$ and $\mathcal{N}$ is no less than 1. The cardinality of the combined sets — which can be considered as the level of an extended tree — is 2^μ, for the same reason that this is the number of nodes on the level for the case of known p_1. Thus the maximum weight of the $2^\mu - 1$ items in $\mathcal{A}' \cup \mathcal{N} \backslash \{j\}$ is at least its average, which is, in turn, at least $(1 - p_j)/(2^\mu - 1)$. If that item is in $\mathcal{N}$, it follows, as above, that $R^*(\boldsymbol{l}, \boldsymbol{p}) \geq \mu + \lg((1 - p_j)/(2^\mu - 1))$. If it is not, then it — along with $2^{\mu - l_i}$ items of the same weight — corresponds to an item i with $p_i \geq (1 - p_j)/(2^{l_i} - 2^{l_i - \mu})$. In this case, too,

$$\begin{aligned} R^*(\boldsymbol{l}, \boldsymbol{p}) &\geq l_i + \lg((1 - p_j)/(2^{l_i} - 2^{l_i - \mu}) \\ &= \mu + \lg((1 - p_j)/(2^\mu - 1)). \end{aligned}$$

Thus the lower bound is identical for any p_j as it is for p_1.

For $p_1 = p_j \in [1/(2^{\mu+1} - 1), 1/2^\mu)$ 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 has a complete coding tree — in this case with l_1 one bit shorter than the other lengths — and thus achieves the lower bound for this range ($\lambda_j = \mu + 1$). 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 ($\lambda_j = \mu$). ■

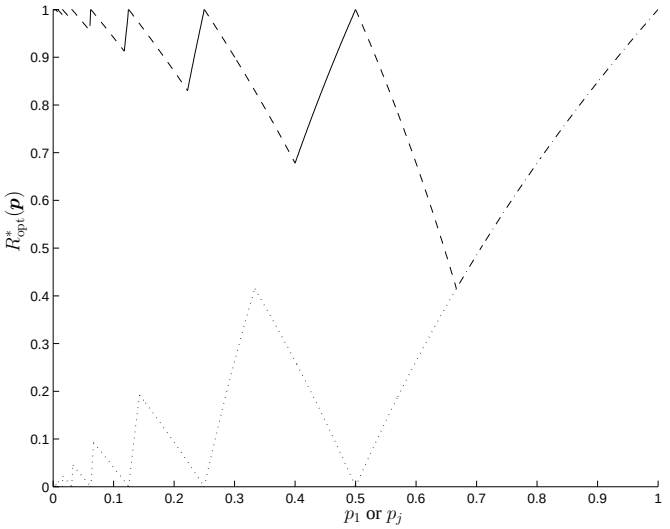

Fig. 1. Tight bounds on minimum maximum pointwise redundancy, including achievable upper bounds (solid), approachable upper bounds (dashed), achievable lower bounds (dotted), and fully determined values for $p_1 = p_j \geq 2/3$ (dot-dashed).

The unit-sized bounds of (9) are identical to the tight bounds at (negative integer) powers of two. In addition, the tight bounds clearly approach 0 and 1 as $p_j \downarrow 0$, similarly to those for average redundancy [10]. Bounds found knowing p_1 are different for the two utilities, however, the average redundancy upper and lower bounds being very close (about 0.086 apart) [3], [6], [8]. For larger given probabilities, note that, above 0.5, p_1 and p_j bounds are identical since any such probability must be the most probable. 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 $p_1 \geq 2/3$.

B. Minimized maximum pointwise redundancy codeword lengths

In addition to finding redundancy bounds in terms of p_1 or p_j, it is also often useful to find bounds on the behavior of l_j in terms of p_j (for $j = 1$ or general j), as was done for optimal average redundancy in [36] (for $j = 1$).

Lengths, Maximum Pointwise Redundancy:

Theorem 2: Any code with lengths $\boldsymbol{l}$ minimizing $\max_{i\in\mathcal{X}}(l_i + \lg p_i)$ over probability mass function $\boldsymbol{p}$, where $p_j \geq 2^{-\nu}$, must have $l_j \leq \nu$. 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 $p_j \leq 1/(2^\nu - 1)$, there is an optimal code with $l_j \geq \nu$, and this bound is also tight.

Proof: Suppose $p_j \geq 2^{-\nu}$ and $l_j \geq 1 + \nu$. Then $R^*_{\rm opt}(\boldsymbol{p}) = R^*(\boldsymbol{l}, \boldsymbol{p}) \geq l_j + \lg p_j \geq 1$, contradicting the unit-sized bounds of (9). Thus $l_j \leq \nu$.

For tightness of the bound, suppose $p_1 \in (2^{-\nu-1}, 2^{-\nu})$ and consider $n = 2^{\nu+1}$ and

$$\boldsymbol{p} = \left(p_1, \underbrace{2^{-\nu-1}, \ldots, 2^{-\nu-1}}_{n-2}, 2^{-\nu} - p_1\right).$$

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 unit-sized bounds of (9). 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 $p_j \leq 1/(2^\nu - 1)$ and consider an optimal length distribution with $l_j < \nu$. As in the Theorem 1 proof, we consider the weights of the nodes of the corresponding extension to the code tree at level l_j: $\mathcal{N}$ is the set of nodes on that level, while $\mathcal{A}'$ is a set of nodes not in the tree, where each leaf i above the level has $2^{l_j - l_i}$ nodes in $\mathcal{A}'$, each of weight $p_i 2^{l_i - l_j}$. Again, the sum of the $2^{l_j} - 1$ weights in $\mathcal{A}' \cup \mathcal{N} \backslash \{j\}$ is no less than $1 - p_j$, so there is one node k' such that

$$w(k') \geq \frac{1 - p_j}{2^{l_j} - 1} \geq \frac{1 - p_j}{2^{l_j} - 2^{l_j + 1 - \nu}}. \tag{11}$$

If this is in $\mathcal{N}$, taking the logarithm and adding l_j to the right-hand side,

$$R^*(\boldsymbol{l}, \boldsymbol{p}) \geq \nu - 1 + \lg \frac{1 - p_j}{2^{\nu-1} - 1} \tag{12}$$

the right-hand side being an upper bound to its pointwise redundancy (based on the right-hand side of (11)). If k' is in $\mathcal{A}'$, then, using the right-hand side of (11), the corresponding leaf (codeword) k at level $l_k < l_j$ has at least probability

$$2^{l_j - l_k} \cdot \frac{1 - p_j}{2^{l_j} - 2^{l_j + 1 - \nu}} = \frac{1 - p_j}{2^{l_k} - 2^{l_k + 1 - \nu}}$$

and (12) thus still holds.

Consider adding a bit to codeword j. Note that

$$\begin{aligned} l_j + 1 + \lg p_j &\leq \nu + \lg p_j \\ &\leq \nu - 1 + \lg \frac{1 - p_1}{2^{\nu-1} - 1} \end{aligned}$$

a direct consequence of $p_j \leq 1/(2^\nu - 1)$. Thus, if we replace this code with one for which $l_j = \nu$, maximum redundancy is not increased and thus the new code is also optimal. The tightness of the bound is seen by applying Lemma 1 to distributions of the form

$$\boldsymbol{p} = \left(p_1, \underbrace{\frac{1 - p_1}{2^\nu - 2}, \ldots, \frac{1 - p_1}{2^\nu - 2}}_{2^\nu - 2}\right)$$

for $p_1 \in (1/(2^\nu - 1), 1/2^{\nu-1})$. This distribution results in $l_1 = \nu - 1$ and thus $R^*_{\rm opt}(\boldsymbol{p}) = \nu + \lg(1 - p_1) - \lg(2^\nu - 2)$, which no code with $l_1 > \nu - 1$ could achieve. ∎

In particular, if $p_j \geq 0.5$, $l_j = 1$, while if $p_j \leq 1/3$, there is an optimal code with $l_j > 1$. One might wonder about $\boldsymbol{p} = (0.99, 0.01)$, for which two 1-length codewords are optimal, yet $p_2 \leq 1/3$. In this case, any code with $l_2 \leq 6$ is optimal, having item 1 (with $l_1 = 1$) as the item with maximum pointwise redundancy. Thus there is no contradiction, although this does illustrate how this lower bound on length is not as tight as it might at first appear, only applying to *an* optimal code rather than *all* optimal codes.

IV. $d^{\rm TH}$ Exponential Redundancy

We now briefly address the $d^{\rm th}$ exponential redundancy problem. Recall that this is the minimization of (4),

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

A straightforward application of Lyapunov's inequality for moments — an application of Hölder's inequality, e.g., [37, p. 27] or [38, p. 54] — yields $R^{d'}(\boldsymbol{p}, \boldsymbol{l}) \leq R^d(\boldsymbol{p}, \boldsymbol{l})$ for $d' \leq d$. Taking limits to 0 and ∞, this results in

$$\begin{aligned} &0 \leq \bar{R}(\boldsymbol{p}, \boldsymbol{l}) \leq R^d(\boldsymbol{p}, \boldsymbol{l}) \leq R^*(\boldsymbol{p}, \boldsymbol{l}) < 1, \quad d > 0 \\ &0 \leq R^d(\boldsymbol{p}, \boldsymbol{l}) \leq \bar{R}(\boldsymbol{p}, \boldsymbol{l}) \leq R^*(\boldsymbol{p}, \boldsymbol{l}) < 1, \quad d \in (-1, 0) \end{aligned}$$

for any valid $\boldsymbol{p}$ and any $\boldsymbol{l}$ satisfying the Kraft inequality with equality; the lower bound in the negative case is a result of

$$R^{-1}(\boldsymbol{p}, \boldsymbol{l}) = -\lg \sum_{i\in\mathcal{X}} 2^{-l_i} = 0, \text{ given } \sum_{i\in\mathcal{X}} 2^{-l_i} = 1.$$

This results in an extension of (9):

$$\begin{aligned} &0 \leq \bar{R}_{\rm opt}(\boldsymbol{p}) \leq R^d_{\rm opt}(\boldsymbol{p}) \leq R^*_{\rm opt}(\boldsymbol{p}) < 1, \quad d > 0 \\ &0 \leq R^d_{\rm opt}(\boldsymbol{p}) \leq \bar{R}_{\rm opt}(\boldsymbol{p}) \leq R^*_{\rm opt}(\boldsymbol{p}) < 1, \quad d \in (-1, 0) \end{aligned}$$

where $R^d_{\rm opt}(\boldsymbol{p})$ is the optimal $d^{\rm th}$ exponential redundancy, an improvement on the bounds found in [18]. These inequalities lead directly to:

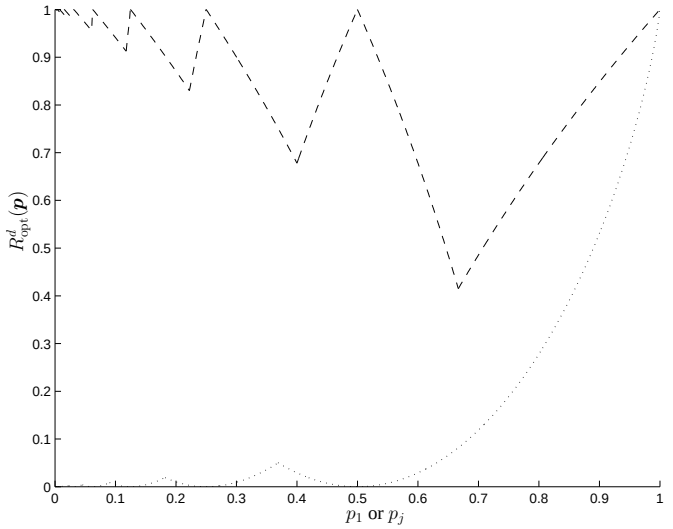

Fig. 2. Bounds on d^{th} exponential redundancy, valid for any $d > 0$. Upper bounds dashed, lower bounds dotted.

Bounds, d^{th} Exponential Redundancy:

Corollary 1: The upper bounds of Theorem 1 are upper bounds for $R^d_{\rm opt}(\boldsymbol{p})$ with any d, while for $d < 0$, any upper bounds for average redundancy (Huffman) coding will also suffice (e.g., [3], [8] for known p_1 or [9], [10] for any known p_j). If $d > 0$, the tight lower bounds of average redundancy coding are lower bounds for $R^d_{\rm opt}(\boldsymbol{p})$ with $d > 0$. These lower bounds — whether or not we know that $j = 1$ — are

$$\bar{R}_{\rm opt}(\boldsymbol{p}) \geq \xi - (1-p_j)\lg(2^\xi - 1) - H(p_j) \quad \text{[6], [10]} \tag{13}$$

where

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

for $p_j \in (0,1)$ and

$$H(x) \triangleq -x \lg x - (1-x)\lg(1-x). \tag{14}$$

This result is illustrated for $d > 0$ in Fig. 2, showing an improvement on the original unit bounds for values of p_j other than (negative integer) powers of two.

V. Bounds on Exponential-Average Problems

A. Previously known exponential-average bounds

While the average, maximum, and d^{th} average redundancy problems yield performance bounds in terms of p_1 (or any p_j) alone, here we seek to find any bounds on $L_a(\boldsymbol{p}, \boldsymbol{l})$ in terms of p_1 and an appropriate entropy measure. Such a more general form is needed because, unlike the other objectives discussed here, this is not a redundancy objective.

Note that $a \leq 0.5$ is a trivial case, always solved by a finite unary code,

$$\boldsymbol{c}^{\rm u}(n) \triangleq (0, 10, 110, \ldots, 1^{n-2}0, 1^{n-1}).$$

This can be seen by applying the exponential combination rule (7) of the associated Huffman-like algorithm; at each step, the combined weight will be the lowest weight of the reduced problem, being strictly less than the higher of the two combined weights, thus leading to a unary code.

For $a > 0.5$, there is a relationship between this problem and Rényi entropy. Rényi entropy [39] is defined as

$$H_\alpha(\boldsymbol{p}) \triangleq \frac{1}{1-\alpha} \lg \sum_{i=1}^n p_i^\alpha \tag{15}$$

for $\alpha > 0$, $\alpha \neq 1$. It is often defined for $\alpha \in \{0, 1, \infty\}$ via limits, that is,

$$H_0(\boldsymbol{p}) \triangleq \lim_{\alpha \downarrow 0} H_\alpha(\boldsymbol{p}) = \lg \|\boldsymbol{p}\|$$

(the logarithm of the cardinality of $\boldsymbol{p}$),

$$H_1(\boldsymbol{p}) \triangleq \lim_{\alpha \to 1} H_\alpha(\boldsymbol{p}) = -\sum_{i=1}^n p_i \lg p_i$$

(the Shannon entropy of $\boldsymbol{p}$), and

$$H_\infty(\boldsymbol{p}) \triangleq \lim_{\alpha \uparrow \infty} H_\alpha(\boldsymbol{p}) = -\lg p_1$$

(the min-entropy).

Campbell first proposed exponential utility functions for coding in [26], [27]. He observed the simple lower bound for $a > 0.5$ in [27]; the simple upper bound was subsequently shown, e.g., in [19, p. 156] and [21]. These bounds are similar to the minimum average redundancy bounds. In this case, however, the bounds involve Rényi's entropy, not Shannon's.

Defining

$$\alpha(a) \triangleq \frac{1}{\lg 2a} = \frac{1}{1 + \lg a}$$

and

$$L_a^{\rm opt}(\boldsymbol{p}) \triangleq \min_{\boldsymbol{l} \in \mathcal{L}_n} L_a(\boldsymbol{p}, \boldsymbol{l})$$

the unit-sized bounds for $a > 0.5$, $a \neq 1$ are

$$0 \leq L_a^{\rm opt}(\boldsymbol{p}) - H_{\alpha(a)}(\boldsymbol{p}) < 1. \tag{16}$$

In the next subsection we show how this bound follows from a result introduced there.

As an example of these bounds, consider the probability distribution implied by Benford's law [40], [41]:

$$p_i = \log_{10}(i+1) - \log_{10}(i), \;\; i = 1, 2, \ldots 9 \tag{17}$$

that is,

$$\boldsymbol{p} \approx (0.30, 0.17, 0.12, 0.10, 0.08, 0.07, 0.06, 0.05, 0.05).$$

At $a = 0.6$, for example, $H_{\alpha(a)}(\boldsymbol{p}) = 2.259\ldots$, so the optimal code cost is between 2.259 and 3.260. In the application given in [29] with (8), these bounds correspond to an optimal solution with probability of success (codeword transmission) between 0.189 and 0.316. Running the algorithm, the optimal lengths are $\boldsymbol{l} = (1, 2, 3, 4, 5, 6, 7, 8, 8)$, resulting in cost $2.382\ldots$ (probability of success $0.296\ldots$). At $a = 2$, $H_{\alpha(a)}(\boldsymbol{p}) = 3.026\ldots$, so the optimal code cost is bounded by 3.026 and 4.027, while the algorithm yields an optimal code with $\boldsymbol{l} = (2, 3, 3, 3, 3, 4, 4, 4, 4)$, resulting in cost $3.099\ldots$.

The optimal cost in both cases is quite close to entropy, indicating that better upper bounds might be possible. In looking for better bounds, recall first that the inequalities in (16) — like the use of the exponential Huffman algorithm — apply for both $a \in (0.5, 1)$ and $a > 1$. Improved bounds on the optimal solution for the $a > 1$ case are given in [21], but not in closed form or in terms of a single probability and entropy. Closed-form bounds for a related objective are given in [33]. However, the proof for the latter set of bounds is incorrect in that it uses the assumption that we will always have an exponential-average-optimal l_1 equal to 1 if $p_1 \geq 0.4$. We shortly disprove this assumption for $a > 1$, showing the need for modified entropy bounds. Before this, we derive bounds based on the results of the prior section.

B. Better exponential-average bounds

Any exponential-average minimization can be transformed into a R^d minimization problem, so we can apply Corollary 1: Given an exponential-average minimization problem with $\boldsymbol{p}$ and a, if we define $\tilde{\alpha} \triangleq \alpha(a) = 1/(1 + \lg a)$ and

$$\hat{p}_i \triangleq \frac{p_i^{\tilde{\alpha}}}{\sum_{k=1}^{n} p_k^{\tilde{\alpha}}} = \frac{p_i^{\tilde{\alpha}}}{2^{(1-\tilde{\alpha})H_{\tilde{\alpha}}(\boldsymbol{p})}}$$

we have

$$\begin{aligned} R^{\lg a}(\hat{\boldsymbol{p}}, \boldsymbol{l}) &= \frac{1}{\lg a} \lg \sum_{i=1}^{n} \hat{p}_i^{1+\lg a} a^{l_i} \\ &= \log_a \sum_{i=1}^{n} p_i a^{l_i} - \log_a \left(\sum_{i=1}^{n} p_i^{\tilde{\alpha}} \right)^{\frac{1}{\tilde{\alpha}}} \\ &= L_a(\boldsymbol{l}, \boldsymbol{p}) - H_{\tilde{\alpha}}(\boldsymbol{p}) \end{aligned} \tag{18}$$

where $H_\alpha(\boldsymbol{p})$ is Rényi entropy, as in (15). This transformation — shown previously in [21] — provides a reduction of exponential-average minimization to d^{th} exponential redundancy. Thus improving bounds for the redundancy problem improves them for the exponential-average problem, and we can show similarly strict improvements to the unit-sized bounds (16); because $\hat{p}_j$ can be expressed as a function of p_j, a, and $H_{\tilde{\alpha}}(\boldsymbol{p})$, so can this bound:

Bounds, Exponential-Average Objective:

Corollary 2: Denote the known lower bound for optimal average redundancy (Huffman) coding as $\bar{o}(p_j) \geq 0$ — which is that of Corollary 1 [6], [10] — and the Theorem 1 upper redundancy bound for minimized maximum pointwise redundancy coding as $\omega^*(p_j) \leq 1$. Further, denote the known upper redundancy bound for optimal average redundancy given p_1 as $\bar{\omega}(p_1) \leq 1$ [8] and that given p_j as $\bar{\omega}'(p_j) \leq 1$ [10]. Then, for $a > 1$, we have

$$\begin{aligned} \bar{o}\left(p_j^{\tilde{\alpha}} 2^{(\tilde{\alpha}-1)H_{\tilde{\alpha}}(\boldsymbol{p})}\right) &\leq L_a^{\text{opt}}(\boldsymbol{p}) - H_{\tilde{\alpha}}(\boldsymbol{p}) \\ &\leq \omega^*\left(p_j^{\tilde{\alpha}} 2^{(\tilde{\alpha}-1)H_{\tilde{\alpha}}(\boldsymbol{p})}\right) \end{aligned}$$

Similarly, for $a \in (0.5, 1)$, we have

$$0 \leq L_a^{\text{opt}}(\boldsymbol{p}) - H_{\tilde{\alpha}}(\boldsymbol{p}) \leq \bar{\omega}\left(p_1^{\tilde{\alpha}} 2^{(\tilde{\alpha}-1)H_{\tilde{\alpha}}(\boldsymbol{p})}\right)$$

and

$$0 \leq L_a^{\text{opt}}(\boldsymbol{p}) - H_{\tilde{\alpha}}(\boldsymbol{p}) \leq \bar{\omega}'\left(p_j^{\tilde{\alpha}} 2^{(\tilde{\alpha}-1)H_{\tilde{\alpha}}(\boldsymbol{p})}\right).$$

Proof: This is a direct result of Corollary 1 and equation (18). ∎

Recall the example of Benford's distribution in (17) for $a = 2$. In this case, adding knowledge of p_1 improves the bounds from $[3.026\ldots, 4.026\ldots)$ to $[3.039\ldots, 3.910\ldots]$ using the ω^* from Theorem 1 and $\bar{o}$ from [6] given as (13) here. For $a = 0.6$, the bounds on cost are reduced from $[2.259\ldots, 3.259\ldots)$ to $[2.259\ldots, 2.783\ldots]$ using $\bar{\omega}$ given as (10) in [3]:

$$\bar{R}_{\text{opt}}(\tilde{\boldsymbol{p}}) \leq 2 - H(\tilde{p}_1) - \tilde{p}_1$$

with argument

$$\tilde{p}_1 = p_1^{\tilde{\alpha}} 2^{(\tilde{\alpha}-1)H_{\tilde{\alpha}}(\boldsymbol{p})} = 0.8386\ldots.$$

Recall from (14) that

$$H(x) = -x \lg x - (1 - x) \lg(1 - x).$$

Although the bounds derived from Huffman coding are close for $a \approx 1$ (the most common case), these are likely not tight bounds; we introduce another bound for $a < 1$ after deriving a certain condition in the next section.

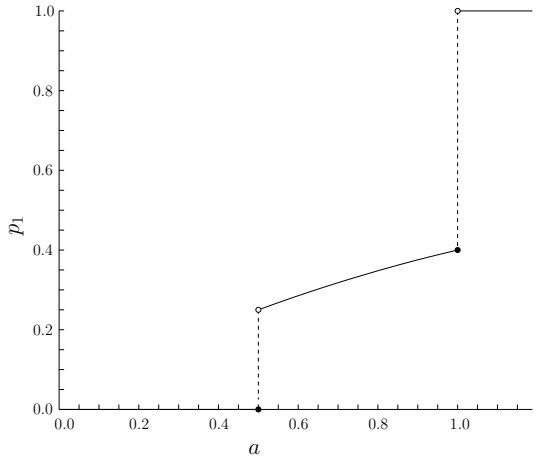

Fig. 3. Minimum p_1 sufficient for the existence of an optimal l_1 not exceeding 1.

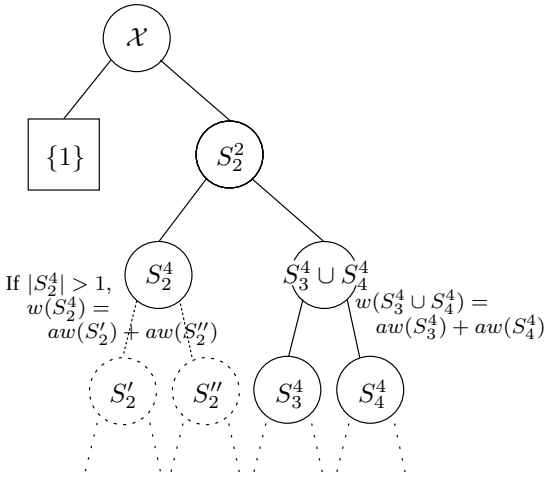

Fig. 4. Tree in last steps of the exponential Huffman algorithm.

C. Exponential-average shortest codeword length

Techniques for finding Huffman coding bounds do not always translate readily to exponential generalizations because Rényi entropy's very definition [39] involves a relaxation of a property used in finding bounds such as Gallager's entropy bounds [3], namely

$$H_1[tp_1, (1-t)p_1, p_2, \ldots, p_n] = H_1[p_1, p_2, \ldots, p_n] + p_1 H_1(t, 1-t)$$

for Shannon entropy H_1 and $t \in [0,1]$. This fails to hold for Rényi entropy. The penalty function L_a differs from the usual measure of expectation in an analogous fashion, and we cannot know the weight of a given subtree in the optimal code (merged item in the coding procedure) simply by knowing the sum probability of the items included. However, we can improve upon the Corollary 2 bounds for the exponential problem when we know that $l_1 = 1$; the question then becomes when we can know this given only a and p_1:

Length $l_1 = 1$***, Exponential-Average Objective****:*

Theorem 3: There exists a code minimizing $L_a(\boldsymbol{p}, \boldsymbol{l}) \triangleq \log_a \sum_{i \in \mathcal{X}} p_i a^{l_i}$ with $l_1 = 1$ for a and $\boldsymbol{p}$ if either $a \leq 0.5$ or both $a \in (0.5, 1]$ and $p_1 \geq 2a/(2a+3)$. Conversely, given $a \in (0.5, 1]$ and $p_1 < 2a/(2a+3)$, there exists a $\boldsymbol{p}$ such that any code with $l_1 = 1$ is suboptimal. Likewise, given $a > 1$ and $p_1 < 1$, there exists a $\boldsymbol{p}$ such that any code with $l_1 = 1$ is suboptimal.

Proof: Recall that the exponential Huffman algorithm combines the items with the smallest weights, w' and w'', yielding a new item of weight $w = aw' + aw''$, and this process is repeated on the new set of weights, the tree thus being constructed up from the leaves to the root. This process makes it clear that, as mentioned, the finite unary code (with $l_1 = 1$) is optimal for all $a \leq 0.5$. This leaves the two nontrivial cases.

1) $a \in (0.5, 1]$*:* The proof in this case is a generalization of [4] and is only slightly more complex to prove. 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 both a (possibly merged) item of weight $w(S_j^k)$ and the set of (individual) items combined in item S_j^k. Since $\{1\}$ is combined in this step, all but one S_j^k has at least weight p_1 (reflected in the second inequality below). Note too that all weights $w(S_j^k)$ must be less than or equal to the sums of probabilities $\sum_{i \in S_j^k} p_i$ (reflected in the third inequality below); equality only occurs for when S_j^k has a single item, due to multiplication by $a < 1$ upon each merge step. Then

$$\begin{array}{rcll} \frac{2a(k-1)}{2a+3} & \leq & (k-1)p_1 & \\ & < & p_1 + \sum_{j=2}^{k} w(S_j^k) & \\ & \leq & p_1 + \sum_{j=2}^{k} \sum_{i \in S_j^k} p_i & \\ & = & \sum_{i=1}^{n} p_i & = \quad 1 \end{array}$$

which, since $a > 0.5$, means that $k < 5$. Thus we can ignore all merging steps prior to having four items and begin with this step; if we start out with fewer than four items ($n \leq 3$), we are guaranteed an optimal code with $l_1 = 1$. Four items remain, one of which is item $\{1\}$ and the others of which are S_2^4, S_3^4, and S_4^4. We show that, if $p_1 \geq 2a/(2a+3)$, these items are combined as shown in Fig. 4.

We assume without loss of generality that weights

$w(S_2^4)$, $w(S_3^4)$, and $w(S_4^4)$ are in descending order. From

$$\begin{aligned} w(S_2^4) + w(S_3^4) + w(S_4^4) &\leq \sum_{i=2}^{n} p_i \\ &\leq \frac{3}{2a+3}, \\ w(S_2^4) &\geq w(S_3^4), \\ \text{and } w(S_2^4) &\geq w(S_4^4) \end{aligned}$$

it follows that $w(S_3^4) + w(S_4^4) \leq 2/(2a+3)$. Consider set S_2^4. If its cardinality is 1, then

$$p_1 \geq w(S_2^4) \geq w(S_3^4) \geq w(S_4^4) \tag{19}$$

so the next step merges the least two weighted items S_3^4 and S_4^4. Since the merged item has weight at most $2a/(2a+3)$, this item can then be combined with S_2^4, then $\{1\}$, so that $l_1 = 1$. If S_2^4 is a merged item, let us call the two items (sets) that merged to form it S_2' and S_2'', indicated by the dashed nodes in Fig. 4. Because these were combined prior to this step,

$$w(S_2') + w(S_2'') \leq w(S_3^4) + w(S_4^4)$$

so

$$w(S_2^4) \leq aw(S_3^4) + aw(S_4^4) \leq \frac{2a}{2a+3}.$$

Thus (19) still applies, and, as in the other case, $l_1 = 1$.

This can be shown to be tight by noting that, for any $\epsilon \in (0, (2a-1)/(8a+12))$,

$$\boldsymbol{p}^{(\epsilon)} \triangleq \left(\tfrac{2a}{2a+3} - 3\epsilon, \tfrac{1}{2a+3} + \epsilon, \tfrac{1}{2a+3} + \epsilon, \tfrac{1}{2a+3} + \epsilon\right)$$

achieves optimality only with length vector $\boldsymbol{l} = (2,2,2,2)$. The result extends to smaller p_1.

2) $a > 1$: Given $a > 1$ and $p_1 < 1$, we wish to show that a probability distribution always exists such that there is no optimal code with $l_1 = 1$. We first show that, for the exponential penalties as for the traditional Huffman penalty, every optimal $\boldsymbol{l}$ can be obtained via the (modified) Huffman procedure. That is, if multiple length vectors are optimal, each optimal length vector can be obtained by the Huffman procedure as long as ties are broken in a certain manner.

Clearly the optimal code is obtained for $n = 2$. Let n' be the smallest n for which there is an $\boldsymbol{l}$ that is optimal but cannot be obtained via the algorithm. Since $\boldsymbol{l}$ 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. Were the latter not the case, the codeword of the more probable item could be exchanged with one of a less probable item, resulting in a better code. Were the former not the case, the longest codeword length could be decreased by one without violating the Kraft inequality, resulting in a better code. Either way, the code would no longer be optimal. Thus we can find two such smallest items with largest codewords (by breaking any ties properly), which, without loss of generality, can be considered siblings. Therefore the problem can be reduced to one of size $n'-1$ via the exponential Huffman algorithm. But since all problems of size $n'-1$ can be solved via the algorithm, this is a contradiction, and the Huffman algorithm can thus find any optimal code.

Note that this is not true for minimizing maximum pointwise redundancy, as the exchange argument no longer holds. This is why the sufficient condition of Section III was not verified using Huffman-like methods.

Now we can show that there is always a code with $l_1 > 1$ for any $p_1 \in (0.2, 1)$; $p_1 \leq 0.2$ follows easily. Let

$$m = \left\lfloor \log_a \left(\frac{4p_1}{1-p_1} \right) \right\rfloor$$

and suppose $n = 1 + 2^{2+m}$ and $p_i = (1-p_1)/(n-1)$ for all $i \in \{2, 3, \ldots, n\}$. Although item 1 need not be merged before the penultimate step, at this step its weight is strictly less than either of the two other remaining weights, which have values $w' = a^{1+m}(1-p_1)/2$. This distribution has an optimal code only with $l_1 \geq 2$. (This must be an equality unless m is equal to the logarithm from which it is derived, in which case l_1 can be either 2 or 3.) Thus, knowing merely the values of $a > 1$ and $p_1 < 1$ is not sufficient to ensure that $l_1 = 1$. ■

These relations are illustrated in Fig. 3, a plot of the minimum value of p_1 sufficient for the existence of an optimal code $\boldsymbol{l}^{\text{opt}}$ with l_1^{opt} not exceeding 1.

Similarly to minimum maximum pointwise redundancy, we can observe that, for $a \geq 1$ (that is, $a > 1$ and traditional Huffman coding), a necessary condition for $l_1^{\text{opt}} = 1$ is $p_1 \geq 1/3$. The sum of the last three combined weights is at least 1, and p_1 must be no less than the other two. However, for $a < 1$, there is no such necessary condition for p_1. Given $a \in (0.5, 1)$ and $p_1 \in (0, 1)$, consider the probability distribution consisting of one item with probability p_1 and $n = 1 + 2^{1+g}$ items with equal probability, where

$$g = \max\left(\left\lfloor \log_a \frac{2ap_1}{1-p_1} \right\rfloor, \left\lfloor \lg \frac{1-2p_1}{p_1} \right\rfloor, 0 \right)$$

and, by convention, we define the logarithm of negative numbers to be $-\infty$. Setting $p_i = (1-p_1)/(n-1)$ for all $i \in \{2, 3, \ldots, n\}$ results in a monotonic probability mass function in which $(1-p_1)a^g/2 < p_1$, which means that the generalized Huffman algorithm will have in its penultimate step three items: One of weight p_1 and two

of weight $(1-p_1)a^g/2$; these two will be complete subtrees with each leaf at depth g. Since $(1-p_1)a^g/2 < p_1$, $l_1^{\text{opt}} = 1$. Again, this holds for any $a \in (0.5, 1)$ and $p_1 \in (0, 1)$, so no nontrivial necessary condition exists for $l_1^{\text{opt}} = 1$. It is also the case for $a \leq 0.5$, since the unary code is optimal for any probability mass function.

D. Exponential-average bounds for $a \in (0.5, 1)$, $p_1 \geq 2a/(2a+3)$

Entropy bounds derived from Theorem 3, although rather complicated, are, in a certain sense, tight:

Further Bounds, Exponential-Average Objective:

Corollary 3: In the minimization of $L_a(\boldsymbol{p}, \boldsymbol{l}) \triangleq \log_a \sum_{i \in \mathcal{X}} p_i a^{l_i}$, if $a \in (0.5, 1)$ and a minimizing $\boldsymbol{l}$ has $l_1 = 1$ (i.e., all $p_1 \geq 2a/(2a+3)$), the following inequalities hold, where $\tilde{\alpha} = \alpha(a) \triangleq 1/(1 + \lg a)$:

$$\sum_{i=1}^{n} p_i a^{l_i} > a^2 \left[a^{\tilde{\alpha} H_{\tilde{\alpha}}(\boldsymbol{p})} - p_1^{\tilde{\alpha}} \right]^{\frac{1}{\tilde{\alpha}}} + a p_1$$

or, equivalently,

$$L_a(\boldsymbol{p}) < 1 + \log_a \left(a \left[a^{\tilde{\alpha} H_{\tilde{\alpha}}(\boldsymbol{p})} - p_1^{\tilde{\alpha}} \right]^{\frac{1}{\tilde{\alpha}}} + p_1 \right)$$

and

$$\sum_{i=1}^{n} p_i a^{l_i} \leq a \left[a^{\tilde{\alpha} H_{\tilde{\alpha}}(\boldsymbol{p})} - p_1^{\tilde{\alpha}} \right]^{\frac{1}{\tilde{\alpha}}} + a p_1$$

or, equivalently,

$$L_a(\boldsymbol{p}) \geq 1 + \log_a \left(\left[a^{\tilde{\alpha} H_{\tilde{\alpha}}(\boldsymbol{p})} - p_1^{\tilde{\alpha}} \right]^{\frac{1}{\tilde{\alpha}}} + p_1 \right).$$

This upper bound is tight for $p_1 \geq 0.5$ in the sense that, given values for a and p_1, we can find $\boldsymbol{p}$ to make the inequality arbitrarily close. Probability distribution $\boldsymbol{p} = (p_1, 1 - p_1 + \epsilon, \epsilon)$ does this for small ϵ, while the lower bound is tight (in the same sense) over its full range, since $\boldsymbol{p} = (p_1, (1-p_1)/4, (1-p_1)/4, (1-p_1)/4, (1-p_1)/4)$ achieves it (with a zero-redundancy subtree of the weights excluding p_1).

Proof: We apply the simple unit-sized coding bounds (16) for the subtree that includes all items but item $\{1\}$. Let $B = \{2, 3, \ldots, n\}$ with $p_i^B = \mathbb{P}[i \mid i \in B] = p_i/(1-p_1)$ and with Rényi α-entropy

$$H_{\tilde{\alpha}}(\boldsymbol{p}^B) = \frac{1}{1-\tilde{\alpha}} \lg \sum_{i=2}^{n} \left(\frac{p_i}{1-p_1} \right)^{\tilde{\alpha}}.$$

$H_{\tilde{\alpha}}(\boldsymbol{p}^B)$ is related to the entropy of the original source $\boldsymbol{p}$ by

$$2^{(1-\tilde{\alpha}) H_{\tilde{\alpha}}(\boldsymbol{p})} = (1-p_1)^{\tilde{\alpha}} 2^{(1-\tilde{\alpha}) H_{\tilde{\alpha}}(\boldsymbol{p}^B)} + p_1^{\tilde{\alpha}}$$

or, equivalently, since $2^{1-\tilde{\alpha}} = a^{\tilde{\alpha}}$,

$$a^{H_{\tilde{\alpha}}(\boldsymbol{p}^B)} = \frac{1}{1-p_1} \left[a^{\tilde{\alpha} H_{\tilde{\alpha}}(\boldsymbol{p})} - p_1^{\tilde{\alpha}} \right]^{\frac{1}{\tilde{\alpha}}}. \tag{20}$$

Applying (16) to subtree B, we have

$$a^{H_{\tilde{\alpha}}(\boldsymbol{p}^B)} \geq \frac{1}{(1-p_1)a} \sum_{i=2}^{n} p_i a^{l_i} > a^{H_{\tilde{\alpha}}(\boldsymbol{p}^B)+1}.$$

The bounds for $\sum_i p_i a^{l_i}$ are obtained by substituting (20), multiplying both sides by $(1-p_1)a$, and adding the contribution of item $\{1\}$, ap_1. ∎

A Benford distribution (17) for $a = 0.6$ yields $H_{\alpha(a)}(\boldsymbol{p}) \approx 2.260$. Since $p_1 > 2a/(2a+3)$, l_1 is 1 and the probability of success is between 0.250 and 0.298; that is, $L_a^{\text{opt}} \in [2.372\ldots, 2.707\ldots)$. Recall that the bounds found using (18) were $\mathbb{P}[\text{success}] \in (0.241, 0.316)$ and $L_a^{\text{opt}} \in [2.259\ldots, 2.783\ldots]$, an improvement on the unit-sized bounds, but not as good as those of Corollary 3. The optimal code $\boldsymbol{l} = (1, 2, 3, 4, 5, 6, 7, 8, 8)$ yields a probability of success of 0.296 ($L_a^{\text{opt}} = 2.382\ldots$).

Note that these arguments fail for $a > 1$ due to the lack of sufficient conditions for $l_1 = 1$. For $a < 1$, other cases likely have improved bounds that could be found by bounding l_1 — as with the use of lengths in [42] to come up with general bounds [7], [8] — but new bounds would each cover a more limited range of p_1 and be more complicated to state and to prove.

ACKNOWLEDGMENT

The author would like to thank J. David Morgenthaler for discussions on this topic.

REFERENCES

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

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

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

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

[5] 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.

[6] 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.

[7] 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.

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

[9] 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.

[10] 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.

[11] W. Szpankowski, "Asymptotic redundancy of Huffman (and other) block codes," *IEEE Trans. Inf. Theory*, vol. IT-46, no. 7, pp. 2434–2443, Nov. 2000.

[12] J. Abrahams, "Code and parse trees for lossless source encoding," *Communications in Information and Systems*, vol. 1, no. 2, pp. 113–146, Apr. 2001.

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

[14] 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.

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

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

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

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

[19] J. Aczél and Z. Daróczy, *On Measures of Information and Their Characterizations*. New York, NY: Academic, 1975.

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

[21] 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.

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

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

[24] P. Gawrychowski and T. Gagie, "Minimax trees in linear time with applications," in *Proc., 20th International Workshop on Combinatorial Algorithms (IWOCA)*. Springer-Verlag, June 29, 2009.

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

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

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

[28] 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.

[29] 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.

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

[31] 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.

[32] 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.

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

[34] R. G. Gallager and D. C. van Voorhis, "Optimal source codes for geometrically distributed integer alphabets," *IEEE Trans. Inf. Theory*, vol. IT-21, no. 2, pp. 228–230, Mar. 1975.

[35] N. Merhav, G. Seroussi, and M. Weinberger, "Optimal prefix codes for sources with two-sided geometric distributions," *IEEE Trans. Inf. Theory*, vol. IT-46, no. 2, pp. 121–135, Mar. 2000.

[36] 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.

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

[38] D. S. Mitrinović, *Analytic Inequalities*. Berlin, Germany: Springer-Verlag, 1970.

[39] A. Rényi, "On measures of entropy and information," in *Proc. 4th Berkeley Symposium on Mathematical Statistics and Probability*, vol. 1, 1961, pp. 547–561.

[40] S. Newcomb, "Note on the frequency of use of the different digits in natural numbers," *Amer. J. Math.*, vol. 4, no. 1/4, pp. 39–40, 1881.

[41] F. Benford, "The law of anomalous numbers," *Proc. Amer. Phil. Soc.*, vol. 78, no. 4, pp. 551–572, Mar. 1938.

[42] B. L. Montgomery and B. V. K. V. Kumar, "On the average codeword length of optimal binary codes for extended sources," *IEEE Trans. Inf. Theory*, vol. IT-33, no. 2, pp. 293–296, Mar. 1987.