On Optimal Permutation Codes

Vivek K Goyal, *Member, IEEE*, Serap A. Savari, *Member, IEEE*, and Wei Wang, *Student Member, IEEE*

***Abstract*—Permutation codes are vector quantizers whose codewords are related by permutations and, in one variant, sign changes. Asymptotically, as the vector dimension grows, optimal Variant I permutation code design is identical to optimal entropy-constrained scalar quantizer (ECSQ) design. However, contradicting intuition and previously published assertions, there are finite block length permutation codes that perform better than the best ones with asymptotically large length; thus, there are Variant I permutation codes whose performances cannot be matched by any ECSQ. Along similar lines, a new asymptotic relation between Variant I and Variant II permutation codes is established but again demonstrated to not necessarily predict the performances of short codes. Simple expressions for permutation code performance are found for memoryless uniform and Laplacian sources. The uniform source yields the aforementioned counterexamples.**

***Index Terms*—Entropy-constrained scalar quantization, vector quantization.**

I. INTRODUCTION

PERMUTATION codes are an elegant type of structured vector quantizer in which the codebook is comprised entirely of permutations of a single starting vector. The structure of the codebook allows optimal (nearest neighbor) encoding of an n-dimensional vector with $O(n \log n)$ operations and $O(n)$ memory. As a means of vector quantization, permutation codes were introduced by Dunn [1] for memoryless Gaussian sources and the mean-squared error (MSE) distortion measure. This was a natural dual to Slepian's modulation codes for additive white Gaussian noise channels based on permutations [2]. The subsequent development of permutation codes for more general sources and distortion measures is due to Berger *et al.* [3]–[5].

A key result of Berger [4] is the "equivalence" between entropy-constrained scalar quantizers (ECSQs) and permutation codes. The quotes are to emphasize that while Berger shows that the performance of any ECSQ can be approached by a sequence of permutation codes, he asserts without proof that no permutation code can do better than an optimal ECSQ. Assuming equivalence of performance, the primary advantage of a permutation code is the generation of a fixed-rate output, eliminating the need for buffering.

The main contribution of this paper is to exhibit a set of permutation codes whose performance cannot be equaled with ECSQ, contradicting an assertion in [4]. The result does not rely on long block lengths in the permutation code; in fact, the advantage disappears as the block length approaches infinity. Exhibiting these codes demonstrates that there are finite block length permutation codes with performance better than the best asymptotically long permutation codes, which contradicts an assertion in [5]. The counterexamples are quantizers for a memoryless uniform source subject to the MSE fidelity criterion. Several results and simple expressions describing the performance of permutation codes for this source are obtained, along with a new asymptotic relation between Variant I and Variant II codes.

II. PERMUTATION CODES

A. Structures

A fixed-rate vector quantizer, or block source code, represents a random vector $\boldsymbol{x} = (x_1, x_2, \ldots, x_n)$ in $\mathbb{R}^n$ with an element of the codebook $\mathcal{C} = \{\boldsymbol{y}_i\}_{i=1}^M$, where each codeword $\boldsymbol{y}_i$ is in $\mathbb{R}^n$. The *rate* of the vector quantizer is defined by[1]

$$R = n^{-1} \log_2 M \qquad \text{(bits per scalar sample).} \tag{1}$$

To minimize the squared error per component

$$n^{-1} \|\boldsymbol{x} - \hat{\boldsymbol{x}}\|^2 = n^{-1} \sum_{i=1}^{n} (x_i - \hat{x}_i)^2$$

the optimal encoder computes the nearest element in the codebook

$$\hat{\boldsymbol{x}} = \alpha(\boldsymbol{x}) = \arg\min_{\boldsymbol{y} \in \mathcal{C}} \|\boldsymbol{x} - \boldsymbol{y}\|.$$

The resulting per-sample distortion is given by

$$D = n^{-1} E\left[\|\boldsymbol{x} - \alpha(\boldsymbol{x})\|^2\right].$$

The complexity of optimal encoding can grow very quickly with the dimension n. Without constraints on the codebook, $\alpha(\cdot)$ is generally implemented with an exhaustive search. Since the size of the codebook is 2^{nR}, the complexity is exponential in the dimension n. Other implementations reduce running time while increasing memory usage [6]. To reduce complexity, it is common to either constrain the codebook so that searching for the nearest codeword is much simpler or use a search technique that does not necessarily find the nearest codeword. The former is more popular.

[1]A fixed-length indexing of the codewords may require slightly more bits: $n^{-1}\lceil \log_2 M \rceil$. The definition of rate above, consistent with [3]–[5], is maintained because it can easily be approached with block indexing of a sequence of codewords. The effect of the ceiling operation is discussed in Section IV-A1.

Manuscript received November 20, 2000; revised June 6, 2001. The material in this paper was presented in part at the IEEE International Symposium on Information Theory, Washington, DC. June 2001.

V. K Goyal is with Digital Fountain, Fremont, CA 94538 USA (e-mail: v.goyal@ieee.org).

S. A. Savari is with Bell labs, Lucent Technology, Murray Hill, NJ 07974 USA (e-mail: savari@research,bell-labs.com).

W. Wang is with the Department of Electrical Engineerign and Computer Science, University of California, Berkeley, Berkeley, CA 94720-1770 USA (e-mail: wangwei@eecs.berkeley.edu).

Communicated by P. A. Chou, Associate Editor for Source Coding.

Publisher Item Identifier S 0018-9448(01)08585-6.

In permutation codes, the codebook is either all the distinct permutations of an initial reproduction vector $\boldsymbol{y}_1$ (Variant I) or the distinct permutations combined with all distinct sign choices (Variant II). For either variant, the optimal encoder can be implemented by sorting, which has complexity $O(n \log n)$. (Allowing approximate sorting, and thus slightly suboptimal encoding, lowers the complexity further.) This complexity is similar to the complexities of the best lattice vector quantizer encoders [7]. At high rates, good lattice quantizers will certainly outperform permutation codes because of their space-filling properties [8]; at low rates their performance is uncertain. Regardless of the rate, lattice codes generally should be used in conjunction with variable-rate lossless codes because the codewords have unequal probabilities. For independent and identically distributed (i.i.d.) sources, permutation codes obviate this need since each permutation has the same probability. We confine our attention here to i.i.d. sources and to "static" quantizers, i.e., quantizers with codebooks that are designed with knowledge of the distribution of the source and not changed based on a specific realization.

Variant I: Let $\mu_1, \mu_2, \ldots, \mu_K$ denote K real numbers satisfying $\mu_1 > \mu_2 > \cdots > \mu_K$, and let $n_1, n_2, \ldots, n_K$ be positive integers with sum equal to n. Define the first codeword $\boldsymbol{y}_1$ of the permutation code to have components given by

$$y_{1,j} = \mu_i, \qquad \text{if } \sum_{\ell=1}^{i-1} n_\ell < j \le \sum_{\ell=1}^{i} n_\ell. \tag{2}$$

More simply

$$\boldsymbol{y}_1 = (\mu_1, \ldots, \mu_1, \mu_2, \ldots, \mu_2, \ldots, \mu_K, \ldots, \mu_K) \tag{3}$$

where each μ_i appears n_i times.

The Variant I code specified by $\{\mu_i\}_{i=1}^K$ and $\{n_i\}_{i=1}^K$ has a codebook consisting of all the distinct permutations of $\boldsymbol{y}_1$. This code has

$$M = \frac{n!}{\prod_{i=1}^{K} n_i!} \tag{4}$$

codewords. Optimal encoding with this codebook is accomplished with a very simple procedure [3]. Replace the n_1 largest components of $\boldsymbol{x}$ with μ_1, the next n_2 largest components of $\boldsymbol{x}$ with μ_2, and so on. The index into the codebook can be based on any enumeration of the permutations.

Variant II: Variant II codes are very similar. The initial codeword $\boldsymbol{y}_1$ is given by (2) or (3) as before, but the μ_i's are required to be nonnegative, so $\mu_1 > \mu_2 > \cdots > \mu_K \ge 0$.

The codebook now consists not only of distinct permutations of $\boldsymbol{y}_1$, but in addition has all distinct sign choices for each component. There are two sign choices for each nonzero component of $\boldsymbol{y}_1$, so the number of codewords is

$$M = 2^h \cdot \frac{n!}{\prod_{i=1}^{K} n_i!}$$

where $h = n$ if $\mu_K > 0$ and $h = n - n_K$ if $\mu_K = 0$.

Optimal encoding is again extremely simple [3]. There are two differences from Variant I: the components of $\boldsymbol{x}$ are taken in order of their absolute values; and $\pm\mu_i$ is used, with the sign chosen to match the component it replaces.

B. Optimal Codebook Entries

For either variant, the design parameters are the block length n, the number of distinct codeword components K, the numbers of repetitions $\{n_i\}_{i=1}^K$, and the codebook entries themselves $\{\mu_i\}_{i=1}^K$. Note that the n_i's determine the rate; the μ_i's affect only the distortion.

The optimal μ_i's, which are determined in this section, can be expressed using order statistics means, assuming the other parameters are fixed. The distortion can then be expressed in terms of the remaining parameters, which will be optimized in subsequent sections. Ultimately, we would like to find permutation codes that minimize distortion for given n and rate at most R.

Variant I: Let π be a permutation that puts the random vector $\boldsymbol{x}$ in decreasing order and let $(\xi_1, \xi_2, \ldots, \xi_n) = \pi(\boldsymbol{x})$. The ξ_i's are called *order statistics*.[2] Using the notation

$$S_0 = 0, \quad S_i = \sum_{j=1}^{i} n_j, \qquad i \in \{1, 2, \ldots, K\}$$

the optimal encoder replaces ξ_j with μ_i for $j = S_{i-1}+1, S_{i-1}+2, \ldots, S_i$. Thus, the distortion incurred by the optimal encoder can be written as

$$D = n^{-1} E\left[\sum_{i=1}^{K} \sum_{j=S_{i-1}+1}^{S_i} (\xi_j - \mu_i)^2\right].$$

It is shown in [3] that for any given $\{n_i\}_{i=1}^K$, the distortion is minimized with

$$\mu_i = n_i^{-1} \sum_{j=S_{i-1}+1}^{S_i} E[\xi_j]. \tag{5}$$

This can be interpreted as follows. The numbers $\{\xi_j\}_{j=S_{i-1}+1}^{S_i}$ are together quantized to a single value μ_i, so to minimize the distortion μ_i should be the mean of this collection. With μ_i's given by (5), the MSE is

$$D = n^{-1}\left(E\left[\|x\|^2\right] - \sum_{i=1}^{K} n_i \mu_i^2\right). \tag{6}$$

The distortion can be expressed in another, perhaps more enlightening way. To begin with, assume $K = n$ and $n_1 = n_2 = \cdots = n_K = 1$. This is the highest rate and (with optimal μ_i's) lowest distortion Variant I permutation code. According to (5), each sorted component is reconstructed to its mean: $\mu_i = E[\xi_i]$, $i = 1, 2, \ldots, n$. The distortion is then the average of the variances of the order statistics

$$D_{\min} = n^{-1} \sum_{i=1}^{n} E\left[(\xi_i - E[\xi_i])^2\right]. \tag{7}$$

This component of the distortion decreases as n increases because the density of each order statistic becomes more peaked, i.e., has lower variance. Note also that it lower-bounds the total

[2]Order statistics are usually sorted smallest-to-largest, but we use the reverse ordering for consistency with earlier papers on permutation codes. Many properties of order statistics are given in [9].

distortion so it can be used to lower-bound the block length n needed to achieve a given distortion.

Increasing any n_i increases the distortion because at least one order statistic is reconstructed to a value other than its mean. Since

$$E\left[(\xi_i - c)^2\right] = E\left[(\xi_i - E[\xi_i])^2\right] + (c - E[\xi_i])^2$$

for any constant c, the ith group of coefficients contributes an additional

$$n^{-1} \sum_{j=S_{i-1}+1}^{S_i} (\mu_i - E[\xi_j])^2$$

to the distortion. The overall distortion is thus

$$D = D_{\min} + n^{-1} \sum_{i=1}^{K} \sum_{j=S_{i-1}+1}^{S_i} (\mu_i - E[\xi_j])^2. \quad (8)$$

Variant II: The optimization of the μ_i's for Variant II codes is very similar. Let σ be a permutation that puts $|\boldsymbol{x}|$ in decreasing order and let $(\eta_1, \eta_2, \ldots, \eta_n) = \sigma(|\boldsymbol{x}|)$. The η_i's are the order statistics of $|\boldsymbol{x}|$. The μ_i's that minimize the distortion are given by

$$\mu_i = n_i^{-1} \sum_{j=S_{i-1}+1}^{S_i} E[\eta_j] \quad (9)$$

and again yield distortion (6) [3].

III. Asymptotic Analysis and Optimization

Optimization of permutation codes requires the selection of parameters $n, K, \{n_i\}_{i=1}^K$, and $\{\mu_i\}_{i=1}^K$. For Variant I codes, the μ_i's affect only the distortion—not the rate—so they naturally should be chosen to minimize the distortion as in (5). Almost the same thing can be said for Variant II codes except that whether or not μ_K is zero affects the rate by n_K/n. In either case, the optimization is inherently difficult because the parameter choices are discrete.

When n is large, $p_i = n_i/n, i = 1, 2, \ldots, K$, can be considered a set of continuous parameters. This makes many analytical computations and optimizations easier, though one should remember that only asymptotic conclusions can be drawn. Section III-A reviews an equivalence from [4], with some additional explanation of why optimal ECSQs and long permutation codes are so similar. A new asymptotic equivalence is presented in Section III-B.

A. Equivalence of Variant I Codes and ECSQs

When n and each n_i are large, the multinomial expression (4) that determines the rate can be approximated with Stirling's formula (see, for example, [10, p. 530]):

$$R \approx -\sum_{i=1}^{K} p_i \log_2 p_i \qquad \text{for Variant I.} \quad (10)$$

The distortion (8) is simplified by noting that $D_{\min} \to 0$ as $n \to \infty$.[3] Furthermore, the deterministic quantities $\{E[\xi_j]\}_{j=1}^n$ are "distributed" identically to a generic source variable x since

$$\frac{\#\{j | E[\xi_j] \le x\}}{n} \approx F(x) \quad (11)$$

where $F(x)$ is the cumulative distribution function (cdf) of x.

With these observations, establishing a connection between permutation codes and ECSQ is straightforward. The rate (10) is the entropy of the output of a quantizer with bin probabilities $\{p_i\}_{i=1}^K$. The distortion (8) can be written as

$$D \approx \sum_{i=1}^{K} \frac{n_i}{n} \cdot \frac{1}{n_i} \sum_{j=S_{i-1}+1}^{S_i} (\mu_i - E[\xi_j])^2$$
$$= \sum_{i=1}^{K} p_i \left[\frac{1}{n_i} \sum_{j=S_{i-1}+1}^{S_i} (\mu_i - E[\xi_j])^2\right]. \quad (12)$$

The bracketed term is the mean of n_i squared errors and, because of (11), is like the squared error in a quantizer that maps

$$x \in \left(F^{-1}\left(\frac{S_{i-1}+1}{n}\right), F^{-1}\left(\frac{S_i}{n}\right)\right)$$

to μ_i. Thus, the permutation code is like an ECSQ with codebook $\{\mu_i\}_{i=1}^K$ and thresholds selected such that the entries have respective probabilities $\{p_i\}_{i=1}^K$.

It is shown in [4, Theorem 2] that the performance of any ECSQ can be approached by a sequence of Variant I permutation codes of increasing block length. This result is established by identifying the ECSQ codebook with the μ_i's and by choosing n_i's so that each n_i/n approaches the probability of the ith ECSQ codebook entry. Through this construction, an ECSQ gives a (long) permutation code, so optimal permutation codes are at least as good as optimal ECSQ. We can say more, however. The rate (10) and distortion (12) show that, asymptotically, the design problems for ECSQ and permutation codes are identical. So *long* permutation codes are no better than ECSQ. This analysis does not tell us anything about short permutation codes.

B. Relation Between Variant I and Variant II Codes

For Variant II codes, Stirling's formula can again be applied to approximate the rate, yielding

$$R \approx 1 - \sum_{i=1}^{K} p_i \log_2 p_i, \qquad \text{for Variant II with } \mu_K \neq 0 \quad (13)$$

$$R \approx 1 - p_K - \sum_{i=1}^{K} p_i \log_2 p_i, \qquad \text{for Variant II with } \mu_K = 0. \quad (14)$$

Manipulations of the distortion are similar to those in the previous section and yield expressions analogous to ECSQ of $|x|$. These are omitted.

[3] One can show that for most sources with invertible cumulative distribution functions (cdfs), the sum of variances in (7) approaches a constant or grows very slowly as $n \to \infty$ (see [9, p. 80]). $D_{\min}$ thus decays roughly as n^{-1}.

The following theorem establishes that, asymptotically, Variant I codes are at least as good as Variant II codes. Note the asymptotic nature of the result; for fixed n, Variant II codes are superior at certain rates.

Theorem 1: For a source with a density that is symmetric about zero and an arbitrary distortion measure, any rate and distortion pair asymptotically achievable by a Variant II code can also asymptotically be achieved by a Variant I code. For a source which is not symmetric about zero, the distortion asymptotically achievable by a Variant II code at a given rate can be asymptotically achieved by a Variant I code at a lower rate.

Proof: Suppose that $\{\mu_i\}_{i=1}^K$ and $\{n_i\}_{i=1}^K$ specify a Variant II code with K distinct components. We will construct a Variant I code with the same distortion. For $i \in \{1, 2, \ldots, K-1\}$, let r_i denote the probability that a component of the original source random vector $\boldsymbol{x}$ replaced by μ_i in the permutation code is positive. In the case of a source that is symmetric about the origin, $r_i = \frac{1}{2}$, $i \in \{1, 2, \ldots, K-1\}$.

Case 1: If $\mu_K = 0$, the corresponding Variant I code has $K' = 2K - 1$ components. Let $\{\mu_i'\}_{i=1}^{2K-1}$ and $\{n_i'\}_{i=1}^{2K-1}$ specify the code. We choose

$$\mu_i' = \begin{cases} \mu_i, & i \in \{1, \ldots, K\} \\ -\mu_{2K-i}, & i \in \{K+1, \ldots, 2K-1\} \end{cases}$$

$$n_i' = \begin{cases} \lceil r_i n_i \rceil, & i \in \{1, \ldots, K-1\} \\ n_K, & i = K \\ n_{2K-i} \\ \quad -\lceil r_{2K-i} n_{2K-i} \rceil, & i \in \{K+1, \ldots, 2K-1\}. \end{cases}$$

By the choice of r_i, the per-symbol distortion of both codes are the same. The rate of the new code R' asymptotically approaches

$$R' \approx -\sum_{i=1}^{K'} \frac{n_i'}{n} \log_2 \left(\frac{n_i'}{n}\right) \approx R - \sum_{i=1}^{K-1} \frac{n_i}{n} \left[1 - h(r_i)\right]$$

where

$$h(x) = -x \log_2 x - (1-x) \log_2 (1-x)$$

is the binary entropy function. For $0 \le x \le 1$, we have the inequality $h(x) \le 1$ with equality only at $x = \frac{1}{2}$.

Case 2: If $\mu_K \neq 0$, the corresponding Variant I code has $K' = 2K$ components. Let $\{\mu_i'\}_{i=1}^{2K}$ and $\{n_i'\}_{i=1}^{2K}$ specify the code. We choose

$$\mu_i' = \begin{cases} \mu_i, & i \in \{1, \ldots, K\} \\ -\mu_{2K+1-i}, & i \in \{K+1, \ldots, 2K\} \end{cases}$$

$$n_i' = \begin{cases} \lceil r_i n_i \rceil, & i \in \{1, \ldots, K\} \\ n_{2K+1-i} \\ \quad -\lceil r_{2K+1-i} n_{2K+1-i} \rceil, & i \in \{K+1, \ldots, 2K\}. \end{cases}$$

Because of the choice of r_i, the per-symbol distortion is the same for both Variant I and Variant II. The rate of the Variant I code asymptotically approaches

$$R' \approx -\sum_{i=1}^{2K} \frac{n_i'}{n} \log_2 \left(\frac{n_i'}{n}\right) \approx R - \sum_{i=1}^{K} \frac{n_i}{n} \left[1 - h(r_i)\right]. \quad \square$$

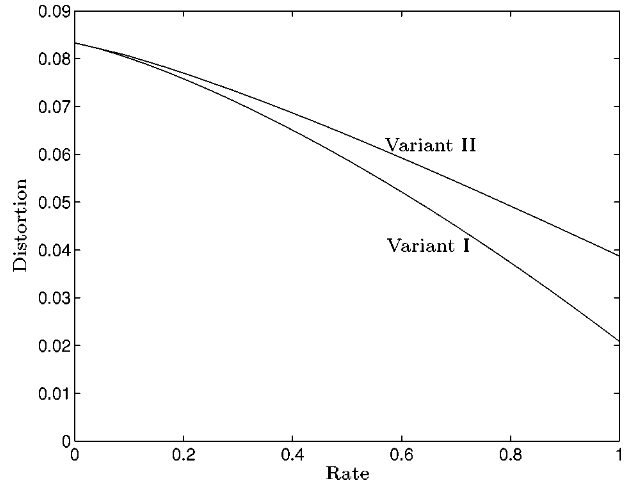


Fig. 1. The performances of the best Variant I and Variant II permutation codes, asymptotically for large n.

The converse of Theorem 1 is not true even for sources that are symmetric about zero; i.e., asymptotically there are Variant I codes that cannot be matched by Variant II codes. An example is for uniform sources at low rates. One can show that optimal Variant I and Variant II codes with $R < 1$ have $K = 2$. Furthermore, it is clear from (13) and (14) that Variant II codes must have $\mu_K = 0$ to attain this low rate. By sweeping p_1 over the interval $(0, 1)$, one obtains the rate and distortion pairs shown in Fig. 1.

IV. Permutation Codes for Uniform Random Variables

The remainder of the paper concerns analyses and optimizations which do not rely on asymptotic approximations. In this section, we assume an information source emitting a sequence of i.i.d. random variables $\{x_k, k = 1, 2, \ldots\}$, each uniformly distributed over the interval $[-\frac{1}{2}, \frac{1}{2}]$. With this source, we are able to not only obtain simple distortion expressions, but also to exhibit codes that contradict some previous general statements about permutation codes.

A. Variant I

First assume n and $\{n_i\}_{i=1}^K$ are fixed. Analytical distortion computations are facilitated by the simplicity of order statistic means and variances for a uniformly distributed source. The means of the order statistics are

$$E[\xi_j] = \frac{n+1-2j}{2(n+1)}, \qquad j = 1, 2, \ldots, n.$$

Substituted in (5), this yields

$$\mu_i = \frac{n - S_i - S_{i-1}}{2(n+1)}, \qquad i = 1, 2, \ldots, K.$$

The variances of the order statistics are

$$E\left[(\xi_j - E[\xi_j])^2\right] = \frac{j(n-j+1)}{(n+1)^2(n+2)}, \qquad j = 1, 2, \ldots, n.$$

Averaging these variances gives the minimum distortion

$$D_{\min} = \frac{1}{6(n+1)}$$

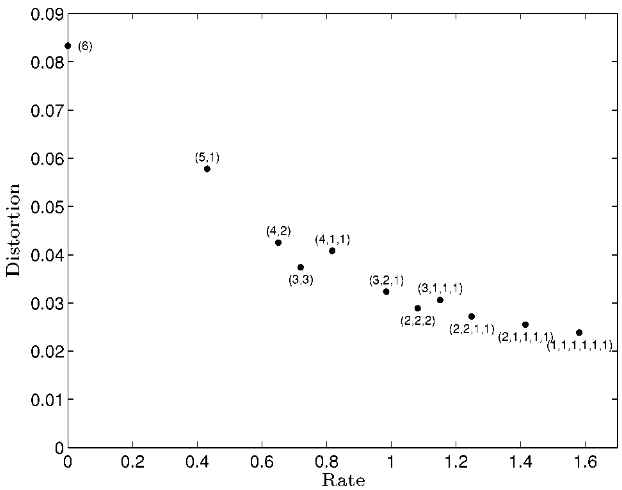


Fig. 2. The performances of all Variant I permutation codes for a uniform source with $n = 6$ and optimal μ_i's. Each point is labeled by its vector of n_i's. For this source, the order of the n_i's affects neither the rate nor the distortion.

and simplifying (8) then gives

$$D = D_{\min} + \frac{\left(\sum_{i=1}^{K} n_i^3\right) - n}{12n(n+1)^2} = \frac{2n^2 + n + \sum_{i=1}^{K} n_i^3}{12n(n+1)^2}. \qquad (15)$$

The distortion expression (15) shows that for a uniform source, the order of the n_i's does not affect the distortion. (Of course, the order of the n_i's never affects the rate.) Below, specific permutation codes are denoted by listing out the n_i's in nonincreasing order.

For a given value of block length n and maximal rate R, we would ideally like to select K and $\{n_i\}_{i=1}^K$ to minimize D. For example, all the operating points obtained with $n = 6$ are shown in Fig. 2. The following choices of n_i's are optimal at their respective rates: $(1,1,1,1,1,1)$, $(2,1,1,1,1)$, $(2,2,1,1)$, $(2,2,2)$, $(3,2,1)$, $(3,3)$, $(4,2)$, $(5,1)$, (6). (Only $(3,1,1,1)$ and $(4,1,1)$ are *not* optimal choices.) We have no efficient method for generating this set of optimal parameter values; in general, the exact solution seems to require an exhaustive search of the possibilities. However, with a more restrictive sense of optimality—seeking points on the lower convex hull of (R, D) pairs—we can greatly restrict the candidate parameters. For $n = 6$, Fig. 2 shows that $(3,2,1)$, $(4,2)$, and $(5,1)$ codes do not give points on the convex hull of (R, D) pairs.

Proposition 2: Consider the set of Variant I permutation codes for a uniform source with fixed block length n. A code specified by $\{n_i\}_{i=1}^K$ and optimal μ_i's cannot lie on the convex hull of possible operating points if there exist i and j such that n_i and n_j differ by more than one.

Proof: Choose i and j such that $n_i > n_j + 1$. That the (R, D) operating point is not on the convex hull is established by considering the operating points associated with three permutation codes: A) the given code; B) the code with (n_i, n_j) replaced with $(n_i', n_j') = (n_i + 1, n_j - 1)$; and C) the code with (n_i, n_j) replaced with $(n_i'', n_j'') = (n_i - 1, n_j + 1)$.[4] We will show that Point A lies above a straight line connecting Point B and Point C by comparing the slopes of $\overline{BA}$ and $\overline{AC}$ (see Fig. 3).

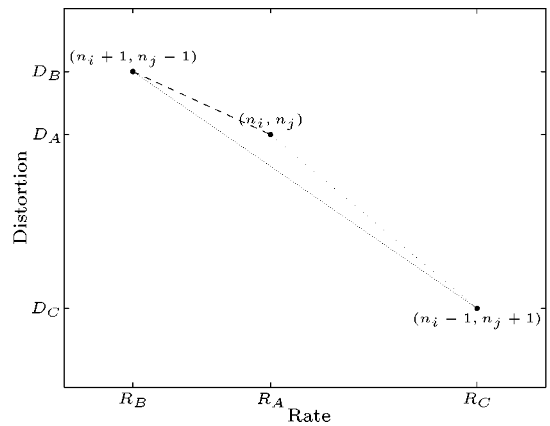


Fig. 3. Operating points used in the proof of Proposition 2.

The changes in rates and distortions can be computed from (4) and (15):

$$n(R_B - R_A) = \log_2 \frac{n_j}{n_i + 1}$$
$$n(R_A - R_C) = \log_2 \frac{n_j + 1}{n_i}$$
$$4n(n+1)^2(D_B - D_A) = (n_i + n_j)(n_i - n_j + 1)$$
$$4n(n+1)^2(D_A - D_C) = (n_i + n_j)(n_i - n_j - 1).$$

We would like to show

$$\frac{D_B - D_A}{R_B - R_A} > \frac{D_A - D_C}{R_A - R_C}$$

which by substitution and cancellation of like factors is equivalent to

$$(n_i - n_j + 1)\log_2 \frac{n_j + 1}{n_i} > (n_i - n_j - 1)\log_2 \frac{n_j}{n_i + 1}.$$

After further rearrangement, this can be transformed to

$$\log_2 \left[\left(\frac{n_i + 1}{n_i}\right)^{n_i} \frac{1}{n_i(n_i+1)}\right] < \log_2 \left[\left(\frac{n_j + 1}{n_j}\right)^{n_j} \frac{1}{n_j(n_j+1)}\right] \qquad (16)$$

for all $n_i > n_j + 1 \geq 0$.

To establish (16), it is sufficient to show that

$$f(z) = \ln \left[\left(\frac{z+1}{z}\right)^{z} \frac{1}{z(z+1)}\right] = z \ln\left(\frac{z+1}{z}\right) - \ln\left(z(z+1)\right)$$

[4]In the case that $n_j = 1$, there is an abuse of notation in having $n_j' = 0$. We mean that K is replaced by $K' = K - 1$, n_i is replaced by $n_i' = n_i + 1$, and the repetition count n_j is removed. However, it is easy to verify that the expressions for codebook size (4) and distortion (15) hold with any n_i equal to zero. Thus, all the computations in this proof hold without modification for the $n_j = 1$ case.

is a strictly decreasing function for $z \in [1, \infty)$. Observe that

$$f'(z) = \ln\left(\frac{z+1}{z}\right) - \frac{3z+1}{z(z+1)}$$

satisfies $\lim_{z\to\infty} f'(z) = 0$ and that

$$f''(z) = z^{-2}(z+1)^{-2}(2z^2+z+1)$$

is strictly positive. Thus, $f'(z)$ must be negative. □

Proposition 2 indicates that, after taking into account the insensitivity to permutation of the n_i's, there is one *candidate* for giving an operating point on the convex hull for each value of K. We conjecture these candidate points are always on the convex hull.

Conjecture 3: There are precisely n parameter choices that give operating points on the convex hull, one for each $K \in \{1, 2, \ldots, n\}$. For a given K, a point on the convex hull is obtained with n_i's given by

$$(\lceil n/K \rceil, \lceil n/K \rceil, \ldots, \lceil n/K \rceil, \lfloor n/K \rfloor, \lfloor n/K \rfloor, \ldots, \lfloor n/K \rfloor)$$

where the number of repetitions of each makes $\sum_{i=1}^{K} n_i = n$. Explicitly, $\lceil n/K \rceil$ is repeated $K + n - K\lceil n/K \rceil$ times and $\lfloor n/K \rfloor$ is repeated $K\lceil n/K \rceil - n$ times.

Returning to the $n = 6$ example, Proposition 2 indicates that

$$(1, 1, 1, 1, 1, 1), (2, 1, 1, 1, 1), (2, 2, 1, 1), (2, 2, 2), (3, 3), (6)$$

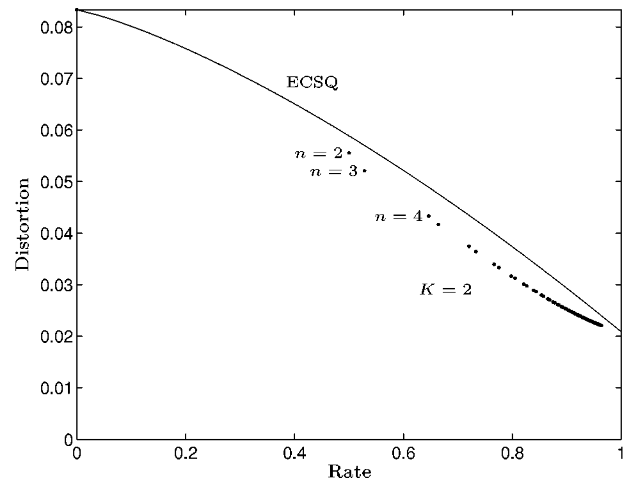
are the only candidates for giving operating points on the optimal convex hull. Conjecture 3 is true for $n = 6$, as these operating points are all on the convex hull. Arguments in support of the conjecture for general n appear in the Appendix.

Conjecture 3 suggests that the point on the convex hull with smallest positive rate is obtained with $(\lceil n/2 \rceil, \lfloor n/2 \rfloor)$. Fig. 4(a) shows the performances of $(\lceil n/2 \rceil, \lfloor n/2 \rfloor)$ codes for $n = 1, 2, \ldots, 100$. (The $n = 1$ point is at $R = 0$, $D = 1/12$.) The curve given for comparison is the performance of optimal ECSQ. For $R \le 1$, an optimal ECSQ is regular and has two output points. It can be shown that

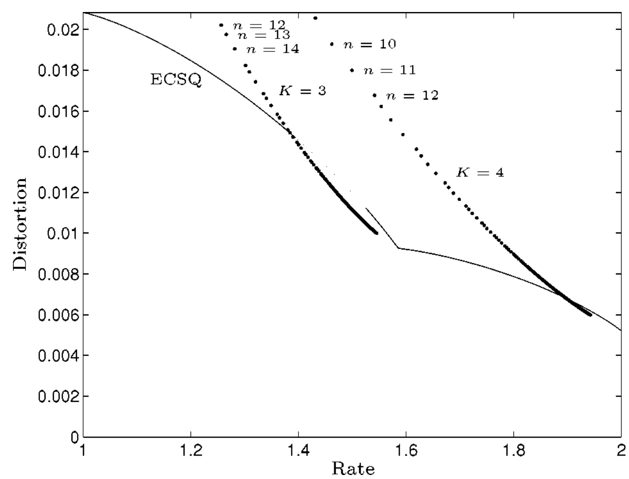
$$R_{\text{ECSQ}} = h\left(\frac{1}{2} - 2\sqrt{D_{\text{ECSQ}} - \frac{1}{48}}\right)$$

where h is the binary entropy function. Note that for all $n > 1$, the $(\lceil n/2 \rceil, \lfloor n/2 \rfloor)$ permutation code is superior to the best ECSQ at the same rate or distortion. This contradicts Berger's "equivalence" between permutation codes and ECSQ.

As $n \to \infty$, the performance of the $(\lceil n/2 \rceil, \lfloor n/2 \rfloor)$ permutation codes approaches $R = 1$ and $D = 1/48$. (This is an operating point also attainable with ECSQ.) However, for all $n \ge 5$, the (R, D) points for the permutation codes lie below a straight line connecting $(0, 1/12)$ and $(1, 1/48)$. Using Proposition 2, all permutation codes with large n will have operating points on or above this line. Therefore, we have constructed finite-length permutation codes with performance better than the best possible in the limit of large block length; this contradicts an assertion of Berger [5]. If Conjecture 3 holds, the operating points shown in Fig. 4(a) indicate that the best convex hull performance at low rates is obtained with $n = 14$.



(a)



(b)

Fig. 4. Comparisons between Variant I permutation codes and ECSQ for uniform source. In both graphs, the connected curve is the performance of optimal ECSQ. (a) Operating points with $(n_1, n_2) = (\lceil n/2 \rceil, \lfloor n/2 \rfloor)$ are marked. For all $n > 1$, these operating points are better than optimal ECSQ. (b) Operating points of $(n/3, n/3, n/3)$ and $(n/4, n/4, n/4, n/4)$ codes (with appropriate rounding) for $n < 200$.

Performance beating ECSQ is not limited to rates under 1 bit per sample. As shown in Fig. 4(b), $(n/3, n/3, n/3)$ and $(n/4, n/4, n/4, n/4)$ codes (with appropriate rounding) also give infinitely many codes better than ECSQ. For $K = 3$, the codes with $n \ge 26$ are better than ECSQ for $1.390 < R < \log_2 3 \approx 1.585$. Also, codes with $K = 4$ and $n \ge 96$ beat ECSQ at rates $1.897 < R < 2$.

Limiting attention to the convex hull of achievable operating points does not allow us to find optimal permutation codes at arbitrary rates. As discussed in Section III-A, we can formulate an optimization problem to generate optimal *long* permutation codes. For the uniform source, this optimization reveals that the optimal K is $\lceil 2^R \rceil$. Furthermore, $\{n_i/n\}_{i=1}^{K}$ should asymptotically take only two values.

Remarks on the Definition of Rate: A fixed-length binary indexing of M codewords requires $\lceil \log_2 M \rceil$ bits. Thus, one appropriate definition of the rate of an n-dimensional vector quantizer with M codewords is $R' = n^{-1}\lceil \log_2 M \rceil$. In this correspondence, the rate definition (1), without a ceiling operation, has been used. Obviously, R' exceeds R by at most n^{-1} bits.

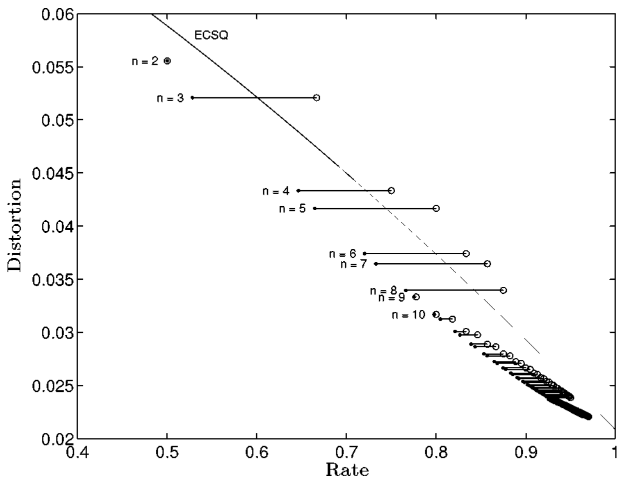


Fig. 5. Portion of Fig. 4(a) shown by circles added to show increases in rate from using $R' = n^{-1}\lceil \log_2 M \rceil$. With the ceiling operation, the Variant I permutation codes with $(n_1, n_2) = (\lceil n/2 \rceil, \lfloor n/2 \rfloor)$ give operating points better than ECSQ for $n = 2$ and $n \geq 9$.

Even for fixed n, the rate R can be approached by jointly indexing consecutive quantizer outputs. For example, binary indexing of pairs of quantizer outputs requires $(2n)^{-1}\lceil \log_2 M^2 \rceil$ bits per scalar sample, which exceeds R by at most $(2n)^{-1}$ bits. This is a trivial block entropy code that requires no storage.

The effect of the ceiling operation on the results in Fig. 4(a) is shown in Fig. 5. Operating points obtained with the original rate R (shown by dots) are connected to new points obtained with the modified rate R' (shown by circles). Block indexing would give intermediate rates.

B. Variant II

The absolute value of a random variable uniformly distributed on $[-\frac{1}{2}, \frac{1}{2}]$ is another uniform random variable, this time distributed on $[0, \frac{1}{2}]$. Thus, especially for uniform random variables, Variant II codes are very similar to Variant I codes. The means of the order statistics are simply translated and scaled, and the variances of the order statistics are reduced by a factor of 4. The only complicating factor is to pay careful attention to whether μ_K is zero.

To minimize the distortion without regard to rate, μ_K should be chosen according to (9), which always gives a nonzero value. The resulting distortion is precisely one-fourth the distortion of (15). This is intuitive because the rate is increased by 1 bit per sample.

The reason to force $\mu_K = 0$ even though this is not the mean of the relevant order statistics is to reduce the rate by n_K/n bits per sample. The distortion obtained in this way is

$$D = \frac{2n^2 + 2n + 2n_K(2n_K^2 + 3n_K + 1) + \sum_{i=1}^{K-1} n_i(n_i^2 - 1)}{48n(n+1)^2}. \tag{17}$$

Up to a multiplicative factor of 4, the dependence of the distortion on $\{n_i\}_{i=1}^{K-1}$ is unchanged. Thus, the analog of Proposition 2 holds for $\{n_i\}_{i=1}^{K-1}$.

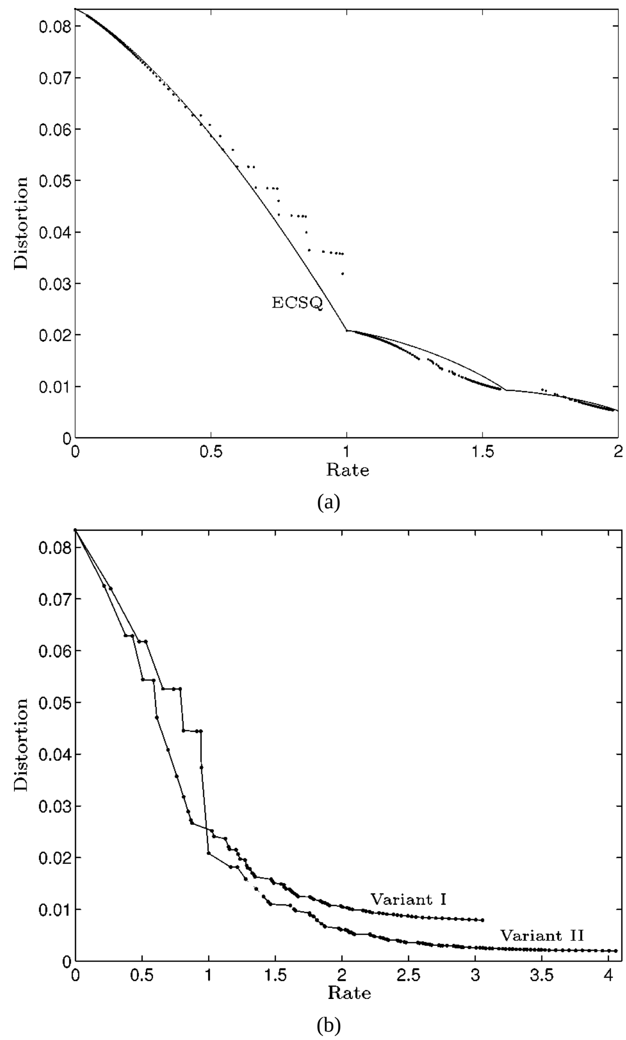


Fig. 6. (a) Comparison between Variant II permutation codes with $n \leq 200$ and ECSQ for uniform source. (b) Comparison between the best Variant I and Variant II codes with $n = 20$. Operating points are connected with line segments to distinguish the two variants. The intermediate points are not attainable.

With $\mu_K \neq 0$, the relative performance of permutation codes and ECSQ for Variant II is very similar to that of Variant I. For example, $(\lceil n/2 \rceil, \lfloor n/2 \rfloor)$ codes are better than ECSQ for all $n \geq 11$ and as $n \to \infty$ they have $(R, D) \to (2, 1/192)$, which is attainable with ECSQ.

Operating points better than ECSQ are more readily attained with $\mu_K = 0$. In the range $0 < R < 1$, codes with $(n_1, n_2) = (1, n-1)$ are better than ECSQ for $n \geq 8$. Many codes better than ECSQ can be obtained with $1 < R < \log_2 3$; in particular, $(n-1, 1)$ for $n \geq 2$, $(n-2, 2)$ for $n \geq 3$, $(n-3, 3)$ for $n \geq 4$, and $(n-4, 4)$ for $n \geq 5$. Many of the best operating points with $n \leq 200$ are gathered and compared to ECSQ in Fig. 6(a). This plot includes operating points obtained with $\mu_K = 0$ and $\mu_K \neq 0$.

For fixed n, Variant II codes are superior to Variant I codes at certain rates. In particular, the $D_{\min}$ component is lower for Variant II codes by a factor of 4, thus the high-rate performance is always better. A comparison for $n = 20$ is made in Fig. 6(b). In this graph, we also see the emergence of the low-rate superiority of Variant I codes, as discussed following Theorem 1.

V. Permutation Codes for Laplacian Random Variables

Now assume an information source emitting i.i.d. random variables with the Laplacian distribution $f(x) = \frac{1}{2}e^{-|x|}$. Though this is more difficult to handle than a uniform source, some analytical computations are still possible. Numerical results facilitate some final qualitative observations.

A. Variant I

Given n and $\{n_i\}_{i=1}^K$, the optimal μ_i's can be computed with (5), where the mean of the jth-order statistic is (see [9])

$$E[\xi_j] = 2^{-n}\left[\sum_{i=0}^{n-j}\binom{n}{i}\sum_{\ell=j}^{n-i}\ell^{-1} - \sum_{i=n-j+1}^{n}\binom{n}{i}\sum_{\ell=n-j+1}^{i}\ell^{-1}\right].$$

Assuming $j \le (n+1)/2$, this can be simplified further to

$$E[\xi_j] = 2^{-n}\sum_{i=0}^{n-j}\binom{n}{i}\sum_{\ell=j}^{n-\max(i,j)}\ell^{-1}.$$

The remaining means can be computed by symmetry: $E[\xi_j] = -E[\xi_{n-j+1}]$.

We have not found a mechanism for easily determining optimal n_i's. One possibility is to use the algorithm of [3, App. II], which finds good—but not necessarily optimal—parameter choices. An exhaustive search is simplified by noting that the increment $E[\xi_j] - E[\xi_{j+1}]$ is decreasing for $1 \le j \le n/2$ and increasing for $n/2 \le j < n$. Thus, we should have $n_i \le n_{i+1}$ for all i such that $S_{i+1} \le n/2$; similarly, $n_i \ge n_{i+1}$ for all i such that $S_i \ge n/2$. More detailed arguments along these lines are given for Variant II.

B. Variant II

For our Laplacian source, Variant II codes are somewhat easier to analyze and design than Variant I codes. The absolute value of the source is an exponential random variable with mean 1. Using the order statistic means

$$E[\eta_j] = \sum_{i=j}^{n} i^{-1}, \qquad j = 1, 2, \ldots, n$$

in (9), the optimal nonzero codebook entries are

$$\mu_i = n_i^{-1}\sum_{j=S_{i-1}+1}^{S_i} E[\eta_j] = n_i^{-1}\sum_{j=S_{i-1}+1}^{S_i}\sum_{\ell=j}^{n}\frac{1}{\ell}, \qquad i = 1, 2, \ldots, K.$$

As always with Variant II codes, μ_K may be chosen as above to minimize distortion without regard to rate, or set to zero to reduce the rate.

The variances of the order statistics are given by

$$E\left[(\eta_j - E[\eta_j])^2\right] = \sum_{i=j}^{n} i^{-2}, \qquad j = 1, 2, \ldots, n.$$

Averaging these gives

$$D_{\min} = n^{-1}\sum_{j=1}^{n}\sum_{i=j}^{n} i^{-2} = n^{-1}\sum_{i=1}^{n}\frac{1}{i}. \tag{18}$$

The harmonic sum diverges slowly, with

$$1 + \frac{1}{2}\log_2 n \le \sum_{i=1}^{n} i^{-1} \le 1 + \log_2 n$$

[11]. Thus,

$$\frac{1 + \frac{1}{2}\log_2 n}{n} \le D_{\min} \le \frac{1 + \log_2 n}{n}$$

explaining the dependence of the high-rate performance on n in [12, Fig. 1] and in Fig. 7.

Substituting (18) in (8) gives overall distortion

$$\begin{aligned} D &= D_{\min} + n^{-1}\sum_{i=1}^{K}\sum_{j=S_{i-1}+1}^{S_i}(\mu_i - E[\eta_j])^2 \\ &= \left(n^{-1}\sum_{i=1}^{n}\frac{1}{i}\right) + n^{-1}\sum_{i=1}^{K}\sum_{j=S_{i-1}+1}^{S_i}\left(\mu_i - \sum_{\ell=j}^{n}\frac{1}{\ell}\right)^2. \end{aligned}$$

This form is useful in deducing the best candidates for $\{n_i\}_{i=1}^K$. In contrast to the case of a uniform source, the order of the n_i's does affect the distortion. The strictly decreasing nature of the increments $\{E[\eta_j] - E[\eta_{j+1}]\}_{j=1}^{K-1}$ implies that $\{n_i\}_{i=1}^{K-1}$ should be a nondecreasing sequence. In addition, one should have $n_K \ge n_{K-1}$ if $\mu_K \ne 0$. We have generated all of the Variant II codes satisfying the necessary condition for optimality for $n = 10, 20, 30, 40$, and 50.

The performances of the best codes are shown in Fig. 7 along with the estimated performance of optimal ECSQ.[5] This graph allows us to summarize some of our findings. Because of asymptotic equivalence, Variant I permutation codes perform similarly to ECSQ for sufficiently large n. Though we have only proved a converse, it seems that except at low rates Variant II codes also perform similarly to ECSQ for symmetric sources. For any fixed n, the performances of permutation codes and ECSQ separate for high rates, with the separation occurring at a lower rate for smaller values of n. The separation must occur because the distortion of a permutation code is lower-bounded by $D_{\min}$, which is a function of n but not of R. Though not shown as dramatically as for the uniform source, performance does *not* strictly improve as n is increased.

Berger, Jelinek, and Wolf [3] attribute the poor performance of permutation codes at high rates (for fixed n) to the close clustering of the $E[\xi_j]$'s (or $E[\eta_j]$'s). They suggest that closely clustered order statistic means must be quantized to a single value to avoid a large increase in rate with only a small decrease in distortion. This explanation is accurate, but we would like to complement it with another. Viewed in spherical coordinates, a permutation code allocates all the rate to the angular components, with no rate to the radial component. With the

[5]The curve labeled "Optimum Quantizer" in [12, Fig. 1] is incorrect. This is partly explained by Berger in the discussion of [5, Fig. 4], where it is noted that at some rates optimal ECSQs have codewords at zero. Berger [5] uses the envelope of several parametric curves to suggest the performance of optimal ECSQ. We have used an iterative design algorithm that minimizes a weighted sum of distortion and output entropy [8]. The algorithm is easy to implement because all needed integrations can be computed in closed form.

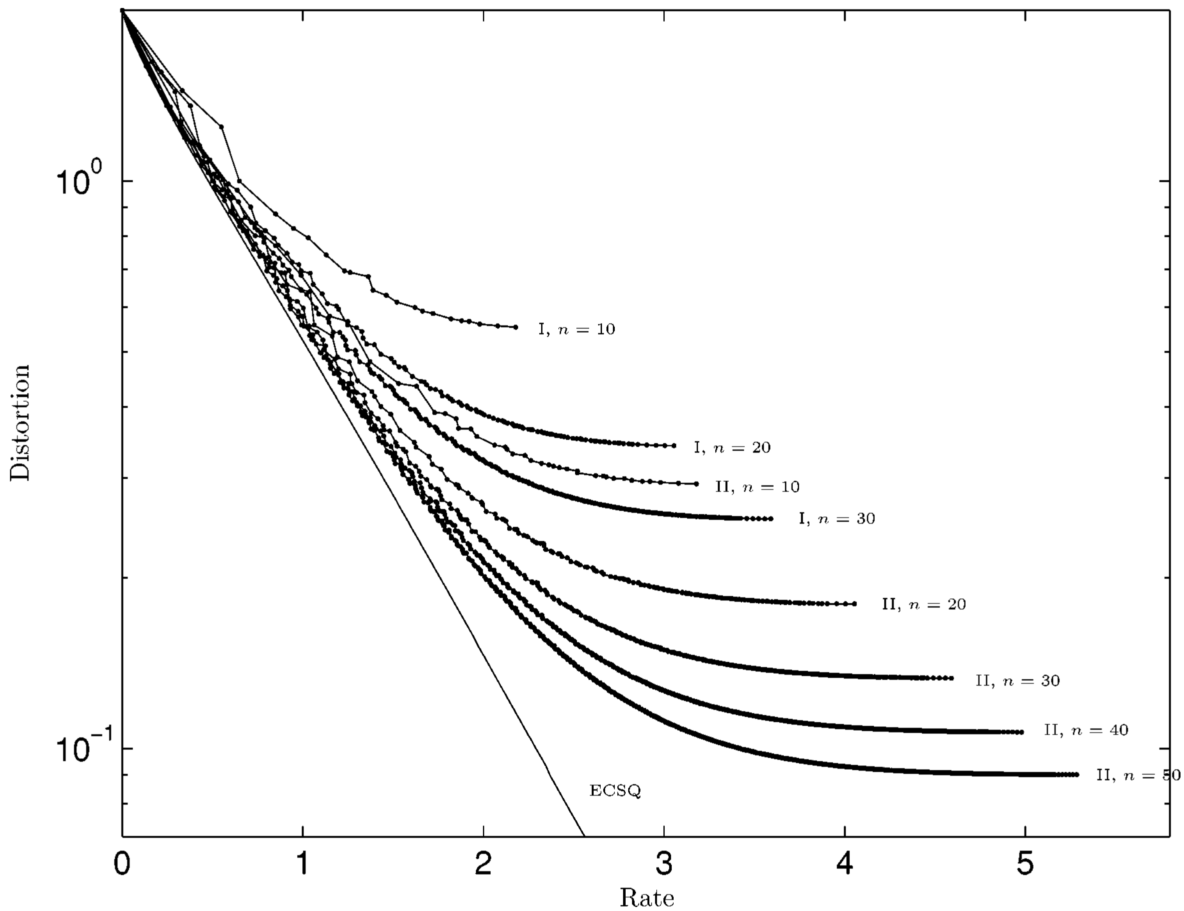


Fig. 7. Operating points of the best Variant I and Variant II codes for the Laplacian source. Operating points are connected with line segments to distinguish the three values of n. The intermediate points are not attainable. Note the logarithmic scale for distortion.

rate fixed and n very large, this is reasonable because the typical set approaches a spherical shell. However, for any given n, there is a rate above which it becomes critical to allocate positive rate to the radial component. Finally, one should note that the performance of ECSQ is attained in an actual implementation only when lossless coding at the entropy bound is achieved.

Appendix
Plausibility Argument for Conjecture 3

Let D_K and R_K denote the per-symbol distortion and the rate associated with the code having K distinct codeword components. Conjecture 3 could be established by showing that the slope magnitudes $(D_K - D_{K+1})/(R_{K+1} - R_K)$ are nonincreasing as K increases from 1 to $n-1$. We rigorously demonstrate in Case 1 below that the slope magnitudes are nonincreasing for $K \geq \sqrt{n+\frac{1}{4}} - \frac{1}{2}$. Later, in Case 2, we provide heuristic arguments that the slope magnitudes are nonincreasing for smaller values of K.

Case 1: Suppose

$$\lceil n/K \rceil \leq \lfloor n/(K+1) \rfloor + 2.$$

It is easy to verify that this condition holds for all K such that $K(K+1) \geq n$. We will break this case into two subcases.

a) Suppose that $\lceil n/(j+1) \rceil \leq K \leq \lfloor n/j \rfloor - 1$ for some positive integer j. Then the difference between the code with K codeword components and the code with $K+1$ codeword components is that j values of $\{n_i\}_{i=1}^{K}$ that are set to $j+1$ in the former are converted to $j+1$ values that are set to j in the latter. Hence,

$$D_K - D_{K+1} = \frac{j(j+1)^3 - (j+1)j^3}{12n(n+1)^2} = \frac{j(j+1)(2j+1)}{12n(n+1)^2}$$

$$R_{K+1} - R_K = \frac{1}{n}\log_2\left(\frac{[(j+1)!]^j}{[j!]^{j+1}}\right) = \frac{1}{n}\log_2\left(\frac{(j+1)^j}{j!}\right).$$

In the next subcase, we will demonstrate that

$$\frac{j(j+1)(2j+1)}{\log_2\left(\frac{(j+1)^j}{j!}\right)}$$

is increasing with j. Hence, $(D_K - D_{K+1})/(R_{K+1} - R_K)$ is nonincreasing with K for these cases.

b) For the remaining subcase, there is an integer $j \geq 2$ and nonnegative integers α, β, γ, and δ such that

- the code with K distinct codeword components has $n_i = j$ for α components and $n_i = j+1$ for β components, and
- the code with $K+1$ distinct codeword components has $n_i = j$ for γ components and $n_i = j-1$ for δ components.

We have the relationships

$$\begin{aligned}\alpha+\beta&=K\\ \gamma+\delta&=K+1\\ \alpha\cdot j+\beta\cdot(j+1)&=Kj+\beta=n\\ \gamma\cdot j+\delta\cdot(j-1)&=(K+1)j-\delta=n.\end{aligned}$$

Hence,

$$\begin{aligned}\beta+\delta&=j\\ \alpha+\gamma&=2K+1-j.\end{aligned}$$

It follows that

$$\begin{aligned}&12n(n+1)^2(D_K-D_{K+1})\\ &\quad=\alpha j^3+\beta(j+1)^3-\gamma j^3-\delta(j-1)^3\\ &\quad=(2K+1-j-\gamma)j^3+(j-\delta)(j+1)^3-\gamma j^3-\delta(j-1)^3\\ &\quad=j(2j^2+3j+1-6\,\delta).\end{aligned}$$

For the rate increment, we have that

$$\begin{aligned}&R_{K+1}-R_K\\ &\quad=\frac{1}{n}\log_2\left(\frac{n!}{[j!]^{\gamma}\,[(j-1)!]^{\delta}}\right)-\frac{1}{n}\log_2\left(\frac{n!}{[j!]^{\alpha}\,[(j+1)!]^{\beta}}\right)\\ &\quad=\frac{1}{n}\log_2\left(\frac{[j!]^{2K+1-j-2\gamma}\,[(j+1)!]^{j}}{[(j+1)!(j-1)!]^{\delta}}\right)\\ &\quad=\frac{1}{n}\log_2\left(\frac{(j+1)^j}{j!}\left(\frac{j}{j+1}\right)^{\delta}\right).\end{aligned}$$

Hence,

$$\frac{D_K-D_{K+1}}{R_{K+1}-R_K}=\frac{j(2j^2+3j+1-6\,\delta)}{12(n+1)^2\log_2\left(\frac{(j+1)^j}{j!}\left(\frac{j}{j+1}\right)^{\delta}\right)}. \tag{19}$$

We would like to show that the right-hand side of (19) decreases as δ increases from 0 to j. We have that

$$\frac{d}{d\,\delta}\,\frac{2j^2+3j+1-6\,\delta}{\log_2\left(\frac{(j+1)^j}{j!}\left(\frac{j}{j+1}\right)^{\delta}\right)}=\frac{\log_2\left(\frac{(1+j^{-1})^{(j+1)(2j+1)}[(j+1)!]^6}{(j+1)^{6(j+1)}}\right)}{\left(\log_2\left(\frac{(j+1)^j}{j!}\left(\frac{j}{j+1}\right)^{\delta}\right)\right)^2}. \tag{20}$$

For all positive integers j, $(1+j^{-1})^j<e$. Hence,

$$\begin{aligned}&\left(1+\frac{1}{j}\right)^{(j+1)(2j+1)}\\ &\quad=\left(\left(1+\frac{1}{j}\right)^{j}\right)^{2j+1}\left(\left(1+\frac{1}{j}\right)^{j}\right)^{2}\left(1+\frac{1}{j}\right)\le\frac{3}{2}\,e^{2j+3}.\end{aligned} \tag{21}$$

Since $j\ge 2$, it follows from Stirling's formula (see [10, p. 530]) that

$$(j+1)!<e^{1/36}\sqrt{2\pi(j+1)}\left(\frac{j+1}{e}\right)^{j+1}. \tag{22}$$

Now it follows from (21) and (22) that

$$\begin{aligned}\frac{(1+j^{-1})^{(j+1)(2j+1)}\,[(j+1)!]^6}{(j+1)^{6(j+1)}}&\le\frac{3}{2}\,(2\pi)^3e^{7/6}e^{-4(j+1)}(j+1)^3\\ &\le\frac{3}{2}\,(6\pi)^3e^{7/6}e^{-12}<0.2.\end{aligned} \tag{23}$$

By (23) and (20), we have that

$$\frac{d}{d\,\delta}\left(\frac{j(2j^2+3j+1-6\,\delta)}{12(n+1)^2\log_2\left(\frac{(j+1)^j}{j!}\left(\frac{j}{j+1}\right)^{\delta}\right)}\right)<0 \tag{24}$$

as desired. Therefore, if we substitute $\delta=0$ and $\delta=j$ into (19), we find that

$$\frac{j(j+1)(2j+1)}{\log_2\left(\frac{(j+1)^j}{j!}\right)}>\frac{(j-1)j(2j-1)}{\log_2\left(\frac{j^{j-1}}{(j-1)!}\right)}.$$

Hence, if K and $K+1$ both belong to subcase a), then it follows that

$$\begin{aligned}&(D_K-D_{K+1})/(R_{K+1}-R_K)\\ &\qquad\ge(D_{K+1}-D_{K+2})/(R_{K+2}-R_{K+1}).\end{aligned}$$

If either

- K falls into subcase a) and $K+1$ falls into subcase b); or
- K falls into subcase b) and $K+1$ falls into subcase a);

then (19)–(24) imply that

$$\begin{aligned}&(D_K-D_{K+1})/(R_{K+1}-R_K)\\ &\qquad\ge(D_{K+1}-D_{K+2})/(R_{K+2}-R_{K+1}).\end{aligned}$$

Case 2: In the case where $\lceil n/K\rceil\ge\lfloor n/(K+1)\rfloor+3$, our argument is less precise. For each K, define $\alpha_K\in[0,(K-1)/K]$ by

$$\alpha_K=\left\lceil\frac{n}{K}\right\rceil-\frac{n}{K}.$$

Then

$$\begin{aligned}&12n(n+1)^2D_K\\ &\quad=2n^2+n+K(1-\alpha_K)\left\lceil\frac{n}{K}\right\rceil^3+K\alpha_K\left\lfloor\frac{n}{K}\right\rfloor^3\end{aligned}$$

$$R_K=\frac{1}{n}\log_2\left(\frac{n!}{\left(\lceil\frac{n}{K}\rceil!\right)^{K(1-\alpha_K)}\left(\lfloor\frac{n}{K}\rfloor!\right)^{K\alpha_K}}\right).$$

We will assume that $\lceil n/K\rceil\approx\lfloor n/K\rfloor\approx n/K$ to make further calculations. As n grows, the supremum of the error in the approximation to $(D_K-D_{K+1})/(R_{K+1}-R_K)$ approaches zero with increasing K. We will approximate D_K by

$$D_K\approx\frac{1}{12n(n+1)^2}\left(2n^2+n+\frac{n^3}{K^2}\right).$$

To approximate R_K, we will use Stirling's formula

$$x!\approx\sqrt{2\pi x}\left(\frac{x}{e}\right)^x$$

to obtain

$$R_K\approx\left(1+\frac{K}{2n}\right)\log_2 K-\frac{K-1}{2n}\,\log_2(2\pi n).$$

With these approximations, we have

$$\frac{12(n+1)^2}{n^2}\cdot\frac{D_K-D_{K+1}}{R_{K+1}-R_K} \approx \frac{2K+1}{K^2(K+1)^2\left(\left(1+\frac{K}{2n}\right)\log_2\left(\frac{K+1}{K}\right)+\frac{1}{2n}\log_2\left(\frac{K+1}{2\pi n}\right)\right)}. \quad (25)$$

We will sketch how to demonstrate that the right-hand side of (25) is decreasing with increasing K. Its derivative with respect to K is $(Y\ln 2)/Z$, where

$$Y = 2K+1-\frac{K(K+1)(2K+1)}{2n}\ln\left(\frac{K+1}{K}\right) - (6K^2+6K+2)\left(\left(1+\frac{K}{2n}\right)\ln\left(\frac{K+1}{K}\right) + \frac{1}{2n}\ln\left(\frac{K+1}{2\pi n}\right)\right),$$

$$Z = K^3(K+1)^3\left(\left(1+\frac{K}{2n}\right)\ln\left(\frac{K+1}{K}\right) + \frac{1}{2n}\ln\left(\frac{K+1}{2\pi n}\right)\right)^2.$$

It is possible to show that the terms with $2n$ in the denominator provide a positive contribution to Y. The magnitude of these terms is maximized by selecting n as small as possible subject to the constraint $K(K+1) < n$. Finally, when this value of n is incorporated into Y, the derivative is negative.

Acknowledgment

The authors wish to thank Dr. C. Weidmann for introducing them to permutation codes and the anonymous reviewers for their thorough work.

References

[1] J. G. Dunn, "Coding for continuous sources and channels," Ph.D. dissertation, Columbia Univ., New York, 1965.
[2] D. Slepian, "Permutation modulation," *Proc. IEEE*, vol. 53, pp. 228–236, Mar. 1965.
[3] T. Berger, F. Jelinek, and J. K. Wolf, "Permutation codes for sources," *IEEE Trans. Inform. Theory*, vol. IT-18, pp. 160–169, Jan. 1972.
[4] T. Berger, "Optimum quantizers and permutation codes," *IEEE Trans. Inform. Theory*, vol. IT-18, pp. 759–765, Nov. 1972.
[5] ——, "Minimum entropy quantizers and permutation codes," *IEEE Trans. Inform. Theory*, vol. IT-28, pp. 149–157, Mar. 1982.
[6] N. Moayeri and D. L. Neuhoff, "Time-memory tradeoffs in vector quantizer codebook searching based on decision trees," *IEEE Trans. Speech Audio Processing*, vol. 2, pp. 490–506, Oct. 1994.
[7] J. H. Conway and N. J. A. Sloane, "Fast quantizing and decoding algorithms for lattice quantizers and codes," *IEEE Trans. Inform. Theory*, vol. IT-28, pp. 227–232, Mar. 1982.
[8] R. M. Gray and D. L. Neuhoff, "Quantization," *IEEE Trans. Inform. Theory*, vol. 44, pp. 2325–2383, Oct. 1998.
[9] H. A. David, *Order Statistics*, 2nd ed. New York: Wiley, 1982.
[10] R. G. Gallager, *Information Theory and Reliable Communication*. New York: Wiley, 1968.
[11] K. H. Rosen, *Elementary Number Theory and Its Applications*, 3rd ed. Reading, MA: Addison-Wesley, 1993.
[12] S. A. Townes and J. B. O'Neal, Jr., "Permutation codes for the Laplacian source," *IEEE Trans. Inform. Theory*, vol. IT-30, pp. 553–559, May 1984.
[13] A. Györgi and T. Linder, "Optimal entropy-constrained scalar quantizationof a uniform source," *IEEE Trans. Inform. Theory*, vol. 46, pp. 2704–2711, Nov. 2000.