

## Noise Prediction for Channels with Side Information at the Transmitter

Uri Erez and Ram Zamir, *Senior Member, IEEE*

**Abstract**—The computation of channel capacity with side information at the transmitter side (but not at the receiver side) requires, in general, extension of the input alphabet to a space of “strategies,” and is often hard. We consider the special case of a discrete memoryless modulo-additive noise channel  $Y = X + Z_S$ , where the encoder observes *causally* the random state  $S \in \mathcal{S}$  that governs the distribution of the noise  $Z_S$ . We show that the capacity of this channel is given by

$$C = \log |\mathcal{X}| - \min_{t: \mathcal{S} \rightarrow \mathcal{X}} H(Z_S - t(S)).$$

This capacity is realized by a state-independent code, followed by a shift by the “noise prediction”  $t_{\min}(S)$  that minimizes the entropy of  $Z_S - t(S)$ . If the set of conditional noise distributions  $\{p(z | s), s \in \mathcal{S}\}$  is such that the optimum predictor  $t_{\min}(\cdot)$  is independent of the state weights, then  $C$  is also the capacity for a *noncausal* encoder, that observes the entire state sequence in advance. Furthermore, for this case we also derive a simple formula for the capacity when the state process has memory.

**Index Terms**—Optimum transmitter, prediction with minimum error entropy, side information, time-varying channels.

### I. INTRODUCTION

The model of an underlying state process that governs the channel behavior fits well many communication links. Examples include wireless communication (fading channel), telephone lines (time-varying filter channel), partial time/band jamming, magnetic memory with defective cells, and more. To analyze such channels it is often useful, and sometimes makes sense, to assume that the channel states are available as Side Information (SI) to the encoder, to the decoder, or to both [16], [11], [2], [8].

One “good” reason for the latter two assumptions is that they simplify the calculation of channel capacity, and the design of optimum encoding/decoding strategies. A common example for that is a *slow* fading Gaussian channel, in which case the receiver can adapt to the fading level and perform “soft” decision [2], [15].

Before turning to the case of side information at the *encoder side only*, let us consider the channel model illustrated in Fig. 1. We use  $\mathcal{X}$ ,  $\mathcal{Y}$ , and  $\mathcal{S}$  to denote the input, output, and state alphabets of the channel, respectively. Given the states  $s_1, s_2, \dots$ , the channel is memoryless with transition distribution  $p(y | x, s)$ , i.e.,

$$p(y_1^n | x_1^n, s_1^n) = \prod_{i=1}^n p(y_i | x_i, s_i)$$

where  $y_1^n$  denotes the sequence  $y_1 \dots y_n$ ; furthermore, the stationary state process  $S_1, S_2, \dots$  is statistically independent of the message to be sent, and it completely captures the memory, if there is any, in the channel, i.e.,

$$p(s_{n+1} | x_1^n, y_1^n, w, s_1^n) = p(s_{n+1} | s_1^n).$$

Manuscript received May 18, 1998; revised October 20, 1999. The material in this paper was presented in part at the IEEE International Symposium on Information Theory, MIT, Cambridge, MA, August 1998.

The authors are with the Department of Electrical Engineering—Systems, Tel-Aviv University, Ramat-Aviv 69978 Israel (e-mail: eretz@eng.tau.ac.il; zamir@eng.tau.ac.il).

Communicated by I. Csizsár, Associate Editor for Shannon Theory.  
Publisher Item Identifier S 0018-9448(00)04646-0.

These assumptions model well a “pure noisy state,” e.g., as in the interference, fading, or jamming channels, but they exclude an “input-dependent state,” like in the intersymbol interference (ISI) channel [6] (in the latter case the state sequence depends on the message  $w$ ).

The respective states of switches  $A$  and  $B$  in Fig. 1 give rise to the following distinct cases<sup>1</sup>:

- **Case I—SI not available:** The channel is regarded as an ordinary channel, given by the mixture of the component channels, e.g., for a memoryless state sequence it is a DMC with transition probability

$$p(y | x) = \sum_s p(s)p(y | x, s).$$

The capacity is denoted  $C_{\text{NOSI}}$ .

- **Case II—SI available at receiver only (A closed, B open):** The state is regarded as an additional channel output, and the capacity is given by (see, e.g., [16])

$$C_{\text{SI@REC}} = \max_{p(x)} I(X; S, Y) = \max_{p(x)} I(X; Y | S). \quad (1)$$

- **Case III—SI available at transmitter and receiver alike (A, B closed):** The capacity is given by the weighted sum of the individual capacities (see, e.g., [16])

$$C_{\text{SI@BOTH}} = \sum_{s \in \mathcal{S}} p(s) \max_{p(x|s)} I(X; Y | S = s) \\ \triangleq \sum_{s \in \mathcal{S}} p(s) C_s \quad (2)$$

where  $C_s$  is the capacity at state  $s$ . (Note that in Cases II and III the capacity depends only on the marginal distribution of the states.)

- **Case IV—SI available at transmitter only (A open, B closed):** The capacity is denoted  $C_{\text{SI@TR}}$ ; we recall its formula in the next section.

The situation where the transmitter has access to side information which is not available to the decoder (Case IV above) is less common in applications. Furthermore, its solution, in terms of capacity and coding, is rather involved.

This problem divides into two categories, according to whether the encoder observes the state process *causally* or *anticipates future states*. In the causal case, considered by Shannon [14], the encoder maps the message  $w \in \{1, 2, \dots, 2^{nR}\}$  into  $\mathcal{X}^n$  using functions

$$x_i = f_i(w, s_1^i), \quad 1 \leq i \leq n \quad (3)$$

where  $s_1^i = s_1, \dots, s_i$  are the states up to time  $i$ . In the noncausal case, considered by Gelfand and Pinsker [7], the encoder observes the entire state sequence before generating the code sequence, thus

$$x_i = f_i(w, s_1^n), \quad 1 \leq i \leq n. \quad (4)$$

In both cases, the receiver decodes the message  $w$  from the whole received vector as  $\hat{w} = g(y_1^n)$ .

Very few explicit solutions exist for the capacity of channels with side information at the encoder; see, e.g., [3], [2], [12], and [8]. This is due in part to the computational complexity of the solution. For the case of discrete memoryless channels with a causal encoder, Shannon [14] showed that  $C_{\text{SI@TR}}$  is given by the ordinary capacity of a discrete

<sup>1</sup>As shown in [2], a “mixture” of these cases, i.e., a configuration where the encoder and the decoder have access to (possibly different) noisy versions of the state process, is effectively equivalent to one of Cases I–IV.

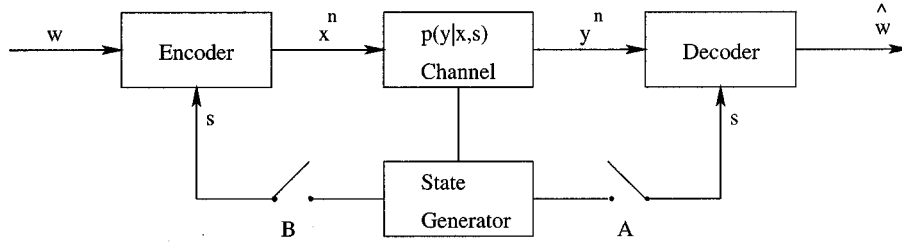


Fig. 1. Channel configuration.

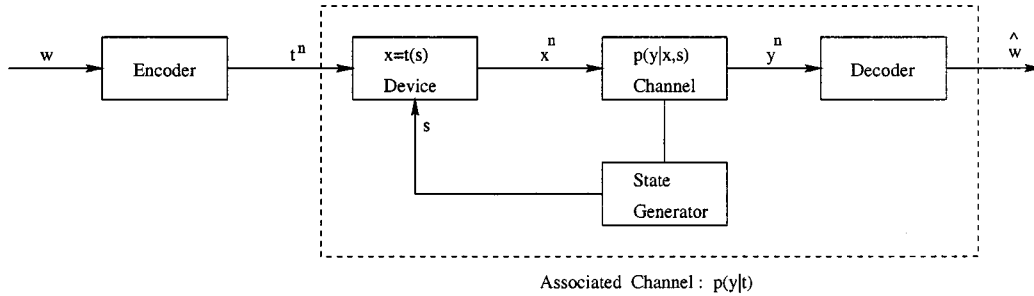


Fig. 2. The associated discrete memoryless channel.

memoryless channel (DMC) with an *extended input alphabet* of size  $|\mathcal{X}|^{|\mathcal{S}|}$ , commonly interpreted as a space of “strategies.” Even more involved are the general solutions for a state process with memory [9], [10], and for noncausal side information [7].

In this correspondence we find a simple and operational solution for the capacity of discrete channels of the form  $Y = X + Z$ , where the additive noise  $Z$  is correlated with a random variable  $S$ , which is available as side information to the encoder. Here  $Z \in \mathcal{X}$ , and the  $+$  sign denotes modulo- $|\mathcal{X}|$  addition. The variable  $S$  may be thought of as the “channel state,” or as a “noisy version” of the channel noise  $Z$ . No “power” constraint is imposed on the transmitter.

For a memoryless state process available causally to the encoder, we show that

$$C_{\text{SI@TR}} = \log|\mathcal{X}| - \min_{t: \mathcal{S} \rightarrow \mathcal{X}} H(Z - t(S))$$

where  $H(\cdot)$  denotes entropy. Furthermore, the optimal encoder is composed of a state-independent code, followed by a shift by the “minimum-error-entropy noise prediction”  $t_{\min}(S)$  that achieves the minimum above. Prediction with minimum error entropy was introduced by Elias [4], in the context of predictive source coding. In general, the optimum predictor  $t_{\min}(\cdot)$  depends on the set of conditional noise distributions  $\{p(z|s), s \in \mathcal{S}\}$ , and on the state distribution (“state weights”)  $p(s)$ . We identify the property of State Weight Independent Prediction (SWIP), which means that  $\{p(z|s), s \in \mathcal{S}\}$  is such that the optimizing function  $t_{\min}(\cdot)$  does not depend on  $p(s)$ . For example, if  $p(z|s)$  is unimodal and symmetric for each  $S$  then  $p(z|s)$  is SWIP. If the pair  $(S, Z)$  satisfies the SWIP property, then *noncausal side information does not increase capacity*, i.e., the anticipatory encoder coincides capacity-wise and structure-wise with the causal encoder. Furthermore, in this case the same predictor is also optimal when there is memory in the state sequence.

In the next section we present our main result regarding the capacity of a memoryless channel with causal state information at the encoder. In Section III we examine the properties of the minimum error entropy prediction function  $t_{\min}(\cdot)$ , and discuss the SWIP property. For channels satisfying the SWIP property, we extend in Section IV the capacity

formula for noncausal state information at the encoder and for channels with a stationary state process.

## II. CAUSAL SIDE INFORMATION AT THE ENCODER

Consider encoding with causal side information at the encoder as defined in (3). For the general memoryless channel  $p(y|x, s)$  shown in Fig. 1, Shannon [14] showed that  $C_{\text{SI@TR}}$  is equal to the regular capacity of the associated DMC illustrated in Fig. 2. The input alphabet of the associated channel, denoted  $\mathcal{T}$ , is the set of all possible mappings

$$t: \mathcal{S} \rightarrow \mathcal{X}$$

which we refer to as *strategies* or *strategy functions*. We may describe each strategy  $t(s) \in \mathcal{T}$  by the vector  $(x^1, x^2, \dots, x^{|\mathcal{S}|})$ , i.e.,  $t(s) = x^s$  for  $s = 1, \dots, |\mathcal{S}|$ . Therefore,  $|\mathcal{T}| = |\mathcal{X}|^{|\mathcal{S}|}$ . The output  $y$  of the associated channel is related to the input  $t$  according to the transition probability

$$p(y|t) \triangleq \sum_s p(s)p(y|x=t(s), s) \quad (5)$$

and also

$$p(y_1^n | t_1^n) = \prod_{i=1}^n p(y_i | t_i). \quad (6)$$

Thus the capacity with side information at the transmitter is given by [14]

$$C_{\text{SI@TR}} = \max_{p(t)} I(T; Y) \quad (7)$$

where the maximization is taken over the distribution  $p(t)$  of the random variable  $T \in \mathcal{T}$ .

Note that at most  $|\mathcal{Y}|$  of the strategies need be given positive probability in order to achieve capacity [6, Ch. 4]. Therefore, if  $\mathcal{X} = \mathcal{Y}$  then at most  $|\mathcal{X}|$  of the  $|\mathcal{X}|^{|\mathcal{S}|}$  strategies need be given positive probability. However, in the general case one does not know in advance which of the strategies are to be used to achieve capacity. We next treat a class

of channels, namely, the class of symmetric, or modulo-additive noise channels, for which it is easy to identify this set of “active” strategies.

Let  $\mathcal{X} = \mathcal{Y} = \mathcal{Z} = \{0, \dots, |\mathcal{X}| - 1\}$ . A symmetric, or modulo-additive noise channel can be concisely described by

$$Y = X + Z \quad (8)$$

where  $Z$  is conditionally independent of  $X$  given the state  $S$ , and addition (and, in the sequel, subtraction) is understood to be performed modulo- $|\mathcal{X}|$ . Denoting by  $p_{Z_s}$  the distribution of the noise given the side information  $S = s$ , we have by the additivity of the channel

$$p(y | x, s) = p_{Z_s}(y - x).$$

Thus the transition probability of the associated channel defined in (5) is given by

$$p(y | t) = \sum_s p(s) p_{Z_s}(y - t(s)) = \Pr(Z + t(S) = y). \quad (9)$$

**Theorem 1 (Memoryless Causal Case):** The capacity of the discrete memoryless additive noise channel  $Y = X + Z$  defined above, with causal side information  $S$  at the encoder, is given by

$$C_{\text{SI@TR}} = \log |\mathcal{X}| - H_{\min} \quad (10)$$

where

$$H_{\min} \triangleq \min_{t \in \mathcal{T}} H(Z - t(S)). \quad (11)$$

Note that (9) implies that

$$H_{\min} = \min_{t \in \mathcal{T}} H(Y | T = t) = \min_{t \in \mathcal{T}} H(Z + t(S)) \quad (12)$$

because the minimization of the entropy of  $Z + t(S)$  and of  $Z - t(S)$  is the same.

*Proof:* We first show the converse part, i.e.,

$$C_{\text{SI@TR}} \leq \log |\mathcal{X}| - H_{\min}.$$

Since

$$H(Y) \leq \log |\mathcal{X}| \quad \text{and} \quad H(Y | T) \geq \min_{t \in \mathcal{T}} H(Y | T = t) = H_{\min} \quad (13)$$

we have

$$I(T; Y) = H(Y) - H(Y | T) \leq \log |\mathcal{X}| - H_{\min}$$

for any distribution on  $\mathcal{T}$ , and the converse follows from (7).

We next show the direct part, i.e.,

$$C_{\text{SI@TR}} \geq \log |\mathcal{X}| - H_{\min}.$$

Let  $t^*$  denote a strategy, i.e., a mapping from  $\mathcal{S}$  to  $\mathcal{X}$ , for which  $H(Y | T = t^*) = H_{\min}$ . Define the following class of strategies:

$$\mathcal{T}^* \triangleq \{t_j\} \quad \text{where} \quad t_j(s) = t^*(s) + j, \quad j = 1 \cdots |\mathcal{X}|. \quad (14)$$

From (9) we see that

$$p(y | t_j) = \Pr(Z + t_j(S) = y) = \Pr(Z + t^*(S) = y - j) \quad (15)$$

i.e.,  $p(y | t_j)$  is the transition probability  $p(y | t^*)$  shifted (modulo- $|\mathcal{X}|$ ) by  $j$ . This clearly implies  $H(Y | t_j) = H(Y | t^*) = H_{\min}$  for all  $j$ . Furthermore, choosing  $T$  distributed uniformly within  $\mathcal{T}^*$  (and zero on strategies not in  $\mathcal{T}^*$ ) induces a uniform distribution on  $\mathcal{Y}$ . Thus for such  $T$  we have equality in both inequalities in (13), and the direct part follows.  $\square$

Note that by restricting the input alphabet to the set  $\mathcal{T}^*$ , the resulting channel from  $\mathcal{T}^*$  to  $\mathcal{Y}$  can be viewed as an additive noise channel, whose alphabet is  $\mathcal{X}$ , and whose noise is distributed as

$$\tilde{Z} \triangleq Z + t^*(S) \triangleq Z - t_{\min}(S) \quad (16)$$

where

$$t_{\min}(\cdot) \triangleq \arg \min_{t: \mathcal{S} \rightarrow \mathcal{X}} H(Z - t(S)). \quad (17)$$

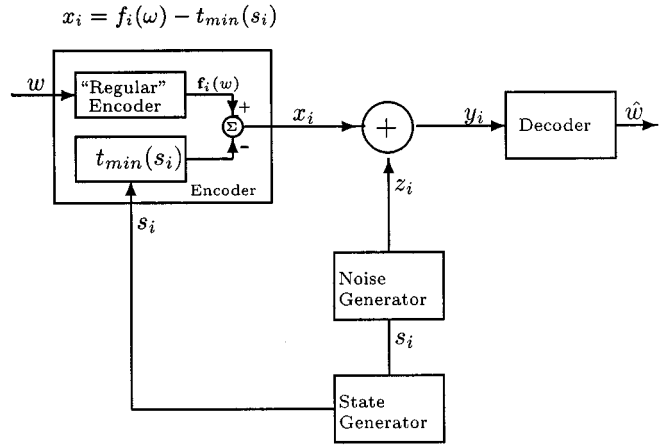


Fig. 3. Instantaneous prediction encoding scheme.

Thus we have simplified the associated channel into a  $|\mathcal{X}|$ -input/ $|\mathcal{X}|$ -output channel. Note that actually any single strategy function  $t$ , not necessarily  $t^*$ , generates a class of  $|\mathcal{X}|$  strategies as in (14), which induces an additive noise channel as in (16).

Theorem 1 implies that the optimal transmitter with side information has a simple modular structure, consisting of an ordinary (i.e., state-independent) Shannon code for a symmetric channel, followed by a shift by  $t_{\min}(s)$ , as shown in Fig. 3. Specifically, suppose a code  $\mathcal{C} = \{(c_{w1}, \dots, c_{wn})\}_{w=1}^{2^{nR}}$  is drawn at random, using a uniform distribution over  $\mathcal{X}$ , with rate  $R < C_{\text{SI@TR}}$ . Given a message  $1 \leq w \leq 2^{nR}$ , and a channel state  $s_i$  at time  $i$ , the transmitter outputs

$$x_i = c_{wi} - t_{\min}(s_i), \quad i = 1 \cdots n. \quad (18)$$

The receiver decodes  $w$  from  $Y_1 \cdots Y_n$  as if the channel were an ordinary DMC with additive noise  $\tilde{Z}$ . It then follows from Theorem 1 and the forward channel coding theorem [6] that  $w$  is decoded reliably with high probability. In practice, any “good” code for a symmetric channel (with a decoder optimized to the noise  $\tilde{Z}$ ) can replace the random code in the scheme above. We call the configuration of (18) “the instantaneous-predictor encoder”; the relation to prediction will become clearer in the next section.

This structure of the optimum transmitter leads to the following interpretation. Since the receiver does not know  $S$ , it sees a channel with an effective noise  $\tilde{Z} = Z - t_{\min}(S)$ . Shifting  $Z$  by  $t_{\min}(S)$  thus makes the effective noise the least harmful for the uninformed receiver. Note also that since

$$I(S; \tilde{Z}) = H(\tilde{Z}) - H(\tilde{Z} | S) = H(\tilde{Z}) - H(Z | S)$$

minimizing  $H(Z - t(S)) = H(\tilde{Z})$  is equivalent to minimizing the information carried by the effective noise about the channel state.

To conclude this section, we return to Cases I–IV of the switches in Fig. 1, and compare the various capacities with and without side information of an additive noise channel. Since choosing a uniform distribution for  $X$  achieves capacity for any discrete additive noise channel, we have for Cases II and III

$$\begin{aligned} C_{\text{SI@REC}} = C_{\text{SI@BOTH}} &= \log |\mathcal{X}| - \sum_s p(s) H(Z_s) \\ &= \log |\mathcal{X}| - H(Z | S) \end{aligned} \quad (19)$$

and for Case I

$$C_{\text{NOSI}} = \log |\mathcal{X}| - H(Z). \quad (20)$$

We thus have the following chain of inequalities:

$$C_{\text{NOSI}} \leq C_{\text{SI@TR}} \leq C_{\text{SI@REC}} = C_{\text{SI@BOTH}} \quad (21)$$

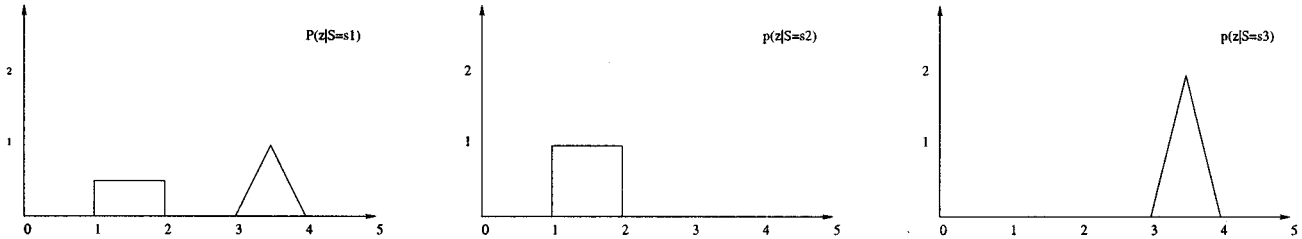


Fig. 4. Example of conditional noise distributions which do not satisfy the SWIP property.

where

$$\begin{aligned} C_{\text{SI@BOTH}} - C_{\text{SI@TR}} &= I(S; \tilde{Z}) \\ C_{\text{SI@BOTH}} - C_{\text{NOSI}} &= I(S; Z). \end{aligned} \quad (22)$$

We have equality in the second inequality in (21), i.e.,  $C_{\text{SI@REC}} = C_{\text{SI@TR}}$ , iff the distributions of  $Z_s$ ,  $s \in \mathcal{S}$  differ by a shift only, in which case the optimum  $\tilde{Z}$  is statistically independent of  $S$ . We have equality in the first inequality in (21), i.e.,  $C_{\text{SI@TR}} = C_{\text{NOSI}}$ , iff  $H(\tilde{Z}) = H(Z)$ , i.e., iff the optimal shifts  $\{t_{\min}(s), s \in \mathcal{S}\}$  are the set of zero shifts.

### III. PREDICTION WITH MINIMUM ERROR ENTROPY AND EXAMPLES

As discussed above, the function  $t_{\min}(\cdot)$  minimizes the entropy of the difference  $Z - t_{\min}(S)$ ; we view  $t_{\min}(S)$  as the ‘‘prediction of  $Z$  from  $S$  with minimum error entropy.’’ In this section we review some properties of minimum error entropy prediction that will be useful in the next section, in particular the SWIP property. We then illustrate these concepts with two examples.

Elias [4] arrives at this very solution in his treatment of entropy-coded predictive source coding. Specifically, for a discrete stationary ergodic source  $X_i$ , one wishes to find the prediction function  $f(x_{i-1}, x_{i-2}, \dots)$  such that the entropy of the error,  $E_i \triangleq X_i - f(X_{i-1}, X_{i-2}, \dots)$ , is minimized. This will minimize the coding rate assuming that the process  $E_1, E_2, \dots$  is entropy-coded according to its *marginal* distribution (ignoring the possible residual memory in the sequence  $E_1, E_2, \dots$ ). The analogy to the problem we study is clear. The past source samples play the role of the current state, while the conditional random variable  $X_i | x_{i-1}, x_{i-2}, \dots$  plays the role of  $Z_s$ .

Prediction under the minimum error entropy criterion features some interesting properties, not found in the more common criteria for prediction and estimation, the minimum mean-squared error criterion  $\min_g E[Z - g(S)]^2$  (continuous case), and the minimum error frequency criterion  $\min_g \Pr[Z \neq g(S)]$  (discrete case). While for these criteria the optimum predictor  $g(s)$  is a function only of the conditional noise distribution  $p(z | s)$  for each  $s$ , the optimal shift  $t(s)$  of the minimum error entropy predictor depends, in general, on the *entire* set of conditional noise distributions  $\{p(z | s)\}$ , and on the *state probabilities*  $p(s)$  (the ‘‘state weights’’).

To illustrate this property, consider the set of conditional noise distributions shown in Fig. 4. Define the following weight distributions on the states:

$$\mathbf{w}_1 = \left(\frac{1}{2}, \frac{1}{2}, 0\right) \quad \mathbf{w}_2 = \left(\frac{1}{2}, 0, \frac{1}{2}\right) \quad \mathbf{w}_3 = \left(0, \frac{1}{2}, \frac{1}{2}\right)$$

We shall now find the corresponding optimal shift functions and show that no single strategy can be optimal for all three weights. Without loss of generality we set  $t_{\min}(s_1) = 0$  for all strategies below. For

$(p(s_1), p(s_2), p(s_3)) = \mathbf{w}_1$ , the vector  $(t_{\min}(s_1), t_{\min}(s_2), t_{\min}(s_3))$  is given by  $(0, 0, *)$ , where  $*$  means that the corresponding letter may be arbitrarily chosen. Any other strategy will induce strictly higher entropy for  $Z - t(S)$ . For  $\mathbf{w}_2$ , the vector of  $t_{\min}(\cdot)$  is given by  $(0, *, 0)$ . Finally, for  $\mathbf{w}_3$ , the vector of  $t_{\min}(\cdot)$  is given by  $(0, * - 2, *)$  (e.g.,  $(0, -2, 0)$  or  $(0, 0, 2)$ ). It is easy to see that no single strategy is consistent with all three cases. Note that by the continuity of the entropy function  $H(Z - t(S))$  as a function of the state weight vector, the above argument can easily be extended to examples of state weight vectors whose components are all nonzero.

*Definition 1 (The State Weight Independent Prediction (SWIP) Property):* Fix the set of conditional noise distributions  $\{p(z | s), z \in \mathcal{X}, s \in \mathcal{S}\}$ . If the same function  $t = t_{\min}(\cdot)$  minimizes the entropy of  $Z - t(S)$  for any state weights  $\{p(s), s \in \mathcal{S}\}$ , then we say that the noise satisfies the State Weight Independent Prediction Property, or in short, ‘‘the noise is SWIP.’’

We shall now demonstrate the existence of SWIP noises. For that we recall the concept of *ordered average*, introduced by Elias [4]. Consider a set of  $n$  discrete distributions, each of  $m$  letters. Denote by  $p_{i,j}$  the probability of the  $j$ th letter in the  $i$ th distribution. Let  $w_i$  be some averaging weight on the distributions

$$w_i \geq 0 \quad \sum_{i=1}^n w_i = 1.$$

Define the ordered distribution  $\tilde{p}_{i,j}$  of the  $i$ th distribution to be the set of probabilities  $p_{i,j}$  indexed according to decreasing probability. The ordered average is defined by

$$p_j^{\text{ord}} = \sum_{i=1}^n w_i \tilde{p}_{i,j}.$$

Thus  $p_1^{\text{ord}}$  is formed by taking  $w_1$  times the largest probability in the first distribution,  $w_2$  times the largest probability in the second distribution, and so on. It can be shown that the entropy of the ordered average distribution is less than or equal to that of any other average, formed from the same distributions with the same weights but with the terms of one or more of the distributions not arranged in order of decreasing probability.

Following Elias, we identify two simple cases of interest in which the optimal shifts are independent of  $p(s)$ , i.e., cases of SWIP noises

- $p_{Z_s}$  is a symmetric and unimodal distribution for each  $s$ .
- $p_{Z_s}$  is zero to the left (right) of a point,  $x_0(s)$ , and is monotonically nonincreasing to the right (left) of  $x_0(s)$

$$\begin{aligned} p_{Z_s}(x) &= 0, & \text{for } x < x_0(s) \\ p_{Z_s}(x) &\geq p_{Z_s}(y), & \text{for } x_0(s) \leq x < y. \end{aligned}$$

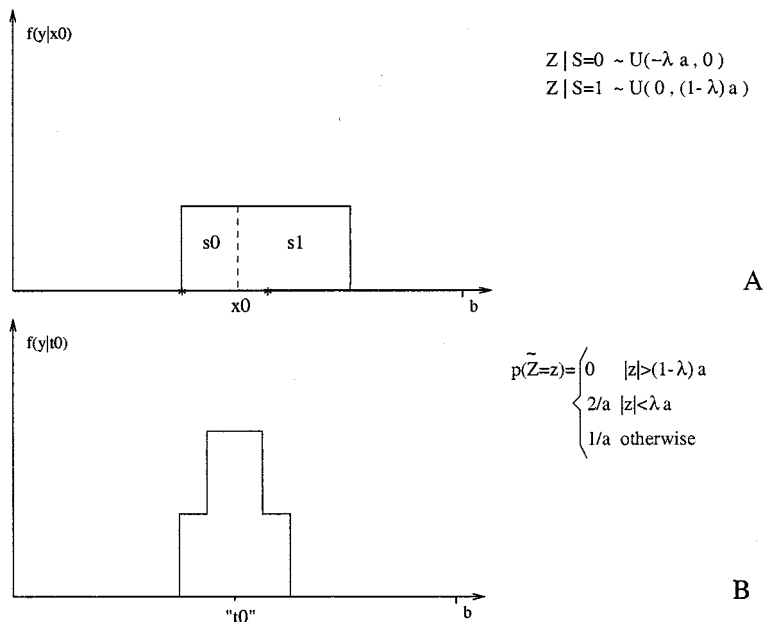


Fig. 5. Example II.

In both cases the assertion follows from the observation that choosing  $t_{\min}(s)$  so as to align the maxima together yields an ordered average irrespective of  $p(s)$ .

The following two lemmas characterize some important properties of minimum error entropy prediction which we use for proving our main results in the next section.

*Lemma 1 (Irrelevancy):* Let  $U$  be a random variable defined over the finite alphabet  $\mathcal{U}$  such that  $U, S, Z$  form a Markov chain  $U \leftrightarrow S \leftrightarrow Z$ . Then, for any function  $g : \mathcal{U} \times S \rightarrow \mathcal{X}$

$$H(Z - g(S, U)) \geq H(Z - t_{\min}(S))$$

where  $t_{\min}$  is defined in (17). Thus  $U$  is irrelevant for minimum error entropy prediction of  $Z$ .

The surprisingly nonstraightforward proof is given in the Appendix.

*Lemma 2 (Conditional Irrelevancy for SWIP Noise):* Let  $U, V$  be random variables defined over the finite alphabets  $\mathcal{U}, \mathcal{V}$ , respectively, such that  $U, V, S, Z$  form a Markov chain  $(U, V) \leftrightarrow S \leftrightarrow Z$ . If the noise  $Z$  satisfies the SWIP property above, then for any function  $g : \mathcal{U} \times S \rightarrow \mathcal{X}$

$$H(Z - g(S, U) | V) \geq H(Z - t_{\min}(S) | V) \tag{23}$$

where  $t_{\min}$  is defined in (17). Thus for SWIP noise  $U$  is conditionally irrelevant for minimum error entropy prediction of  $Z$ .

*Proof:* By Lemma 1 we have for any  $v \in \mathcal{V}$

$$H(Z - g(S, U) | V = v) \geq H(Z - t_{\min}(S) | V = v)$$

and since the noise is SWIP the function  $t_{\min}$  is independent of  $p(s | v)$  and therefore independent of the value of  $v$ . Thus (23) follows by taking expectation over  $V$ .  $\square$

Note that (23) does not hold in general for non-SWIP noises, i.e., for noises for which  $t_{\min} = t_{\min}(s, p(s))$  does depend on the state weights. For such noises, if the state weights  $p(s | v)$  depend on the value of the condition  $V$ , and if  $U$  depends on  $V$ , then the function

$g(s, u)$  can take advantage of these dependencies in order to reduce the conditional entropy of  $Z - g(U, S)$  given  $V$  beyond that achieved by a function of  $S$  alone.

A. Examples

We turn to consider a few examples of finite-state additive noise channels, their associated minimum error entropy predictors, and their capacities. We first note that the results of Theorem 1 hold equally well when the alphabet  $\mathcal{X}$  is a continuous interval  $[0, b] \in \mathcal{R}^1$ , with addition performed modulo  $b$ , and regular entropy replaced by differential entropy.

*Example I: Binary-Symmetric Channel (BSC) with crossover probability  $\theta_s$ .* Here

$$C_{\text{SI@REC}} = 1 - \sum_s p(s) h(\theta_s)$$

while

$$C_{\text{NOSI}} = 1 - h\left(\sum_s p(s) \theta_s\right).$$

In the case where the SI consists of two states,  $\theta_1$  and  $\theta_2$ , we have

$$C_{\text{SI@TR}} = \begin{cases} C_{\text{NOSI}}, & \text{if } \theta_1, \theta_2 \leq \frac{1}{2} \\ 1 - h[p(s=1)\theta_1 + p(s=2)\bar{\theta}_2], & \text{if } \theta_1 \leq \frac{1}{2} \leq \theta_2 \end{cases} \tag{24}$$

where  $\bar{\theta}_2 = 1 - \theta_2$ . This follows since binary noise falls into Case b) of SWIP noise above, so Elias' ordered average condition applies. The extension of the result to more than two states is obvious; capacity is achieved by "flipping" each  $\theta_i$  that is greater than  $\frac{1}{2}$ .

*Example II:* A channel with  $\mathcal{X} = [0, b]$  and transition distribution as shown in Fig. 5(A). We have

$$\begin{aligned} H(Z | S) &= p(s_1)H(Z | S = s_1) + p(s_0)H(Z | S = s_0) \\ &= \lambda \log \lambda a + (1 - \lambda) \log (1 - \lambda) a = \log a - h(\lambda) \end{aligned}$$

while

$$\begin{aligned} H(\tilde{Z}) &= -\int_0^{\lambda a} \frac{2}{a} \log \frac{2}{a} - \int_0^{(1-2\lambda)a} \frac{1}{a} \log \frac{1}{a} \\ &= -2\lambda \log \frac{2}{a} - (1-2\lambda) \log \frac{1}{a} = \log a - 2\lambda \end{aligned}$$

where to obtain  $\tilde{Z}$  we optimized the shifts according to Elias' conditions for symmetric unimodal distributions (Case a) above). Therefore,

$$C_{\text{SI@REC}} = \log \frac{b}{a} + h(\lambda) \quad \text{while} \quad C_{\text{SI@TR}} = \log \frac{b}{a} + 2\lambda.$$

The corresponding distribution of the noise  $\tilde{Z}$  is depicted in Fig. 5(B).

#### IV. NONCAUSAL SIDE INFORMATION AND STATE PROCESS WITH MEMORY

In this section we generalize our results in two directions, first to the case where the state process has memory and secondly to a noncausal encoder as discussed in Section I. Jelinek treated channels having memory with causal side information at the transmitter in [9], [10]. However, the expressions he arrives at are given in terms of complex double limits. This remains the case even when they are applied to additive noise channels, thus giving little new insight. Gelfand and Pinsker [7] studied the problem of coding for a discrete memoryless channel with channel states available noncausally to the encoder. Their solution, as Shannon's solution for a causal encoder, involves maximization over an extended alphabet, and moreover the optimization is done over conditional input probabilities (given the state). We shall confine our treatment to SWIP noise which will allow us to treat both problems together and to obtain simple expressions for the capacities for both cases.

For our purpose we can proceed directly from the definition of the encoding function (4). Consider an encoder which observes the entire state sequence  $\mathbf{S} = S_1, \dots, S_n$  prior to encoding. The state process is a general stationary process, such that the noise sample  $Z_i$  is conditionally independent of  $(X_1 \dots X_n, S_1 \dots S_{i-1}, S_{i+1} \dots S_n)$  given  $S_i$ . The encoding and decoding functions now assume the form

$$\mathbf{x} = \mathbf{f}(w, \mathbf{s}) \quad \text{and} \quad \hat{w} = g(\mathbf{y}) \quad (25)$$

where boldface denotes  $n$ -vectors, i.e.,

$$\begin{aligned} \mathbf{x} &= x_1, \dots, x_n \\ \mathbf{y} &= y_1, \dots, y_n \end{aligned}$$

and

$$\mathbf{f}(\cdot) = f_1(\cdot), \dots, f_n(\cdot).$$

We first give a (*non* "single-letter") upper bound on the capacity of general additive noise channels.

*Lemma 3:* Assume the discrete additive noise channel defined in (8). The rate  $R$  of any code with noncausal side information  $\mathbf{S}$  at the encoder (as defined in (25)) and error probability  $\text{Pr}(\hat{W} \neq W) \leq \epsilon$ , satisfies

$$R \leq \log |\mathcal{X}| - H_{\min}^{(n)} + \epsilon'$$

where  $\epsilon' \rightarrow 0$  as  $\epsilon \rightarrow 0$  uniformly in the code length  $n$ , and

$$H_{\min}^{(n)} \triangleq \frac{1}{n} \min_{\mathbf{t}: \mathbf{S}^n \in \mathcal{X}^n} H(\mathbf{Z} - \mathbf{t}(\mathbf{S})). \quad (26)$$

*Proof:* By Fano's inequality [6], we have

$$H(W | \mathbf{Y}) \leq h(\epsilon) + \epsilon n R \leq h(\epsilon) + \epsilon n \log |\mathcal{X}| \triangleq n \epsilon'.$$

Clearly,  $\epsilon' \rightarrow 0$  as  $\epsilon \rightarrow 0$  uniformly in  $n$ . On the other hand,

$$\begin{aligned} H(\mathbf{Y} | W) &= H(\mathbf{f}(W, \mathbf{S}) + \mathbf{Z} | W) \\ &\stackrel{\text{a)}}{=} \sum_w p(w) H(\mathbf{f}(w, \mathbf{S}) + \mathbf{Z}) \\ &\geq \min_w H(\mathbf{f}(w, \mathbf{S}) + \mathbf{Z}) \\ &\geq n H_{\min}^{(n)} \end{aligned} \quad (27)$$

where in a) we used the fact that  $W$  is statistically independent of  $(\mathbf{Z}, \mathbf{S})$ . Combining the above and the fact that  $H(\mathbf{Y}) \leq n \log |\mathcal{X}|$ , we have

$$\begin{aligned} nR &= H(W) \\ &= H(\mathbf{Y}) - H(\mathbf{Y} | W) + H(W | \mathbf{Y}) \\ &\leq n \log |\mathcal{X}| - n H_{\min}^{(n)} + n \epsilon' \end{aligned} \quad (28)$$

which proves the lemma.  $\square$

Lemma 3 implies that the capacity of this channel is upper-bounded by

$$\log |\mathcal{X}| - \inf_n H_{\min}^{(n)},$$

a quantity which may be hard to compute in general. Nevertheless, we now show that for SWIP noise the instantaneous prediction scheme of Section II is still optimal.

*Lemma 4:* For SWIP noise we have for every  $n$

$$H_{\min}^{(n)} = \frac{1}{n} H(Z_1 - t_{\min}(S_1), \dots, Z_n - t_{\min}(S_n)) \quad (29)$$

where  $t_{\min}$  achieves  $H_{\min}^{(1)}$  in (26).

*Proof:* let  $\tilde{\mathbf{Z}} = \mathbf{Z} - \mathbf{t}(\mathbf{S})$  for some function  $\mathbf{t}: \mathcal{S}^n \rightarrow \mathcal{X}^n$ . We have

$$H(\tilde{Z}_1^n) = H(\tilde{Z}_1) + H(\tilde{Z}_2 | \tilde{Z}_1) + \dots + H(\tilde{Z}_n | \tilde{Z}_1^{n-1}) \quad (30)$$

$$= H(\tilde{Z}_1) + H(\tilde{Z}_2 | \tilde{Z}_1) + \dots + H(Z_n + t_n(S_1^n) | \tilde{Z}_1^{n-1}). \quad (31)$$

Since  $Z_n \leftrightarrow S_n \leftrightarrow (\tilde{Z}_1^{n-1}, S_1^{n-1})$  form a Markov chain, we have by Lemma 2 and the SWIP property

$$H(Z_n - t_n(S_1^n) | \tilde{Z}_1^{n-1}) \geq H(Z_n - t_{\min}(S_n) | \tilde{Z}_1^{n-1})$$

i.e., assigning  $t_n(s_1^n) = t_{\min}(s_n)$  can only reduce  $H(\tilde{Z}_1^n)$ . Since the ordering of the set  $\{\tilde{Z}_1, \dots, \tilde{Z}_n\}$  can be arbitrarily changed, the above argument holds for all the  $\{t_i\}$  and we obtain

$$H(\mathbf{Z} - \mathbf{t}(\mathbf{S})) \geq H(Z_1 - t_{\min}(S_1), \dots, Z_n - t_{\min}(S_n)). \quad (32)$$

Since (32) is true for any function  $\mathbf{t}$ , the optimum prediction function and the minimum noise entropy are

$$\mathbf{t}_{\min}(\mathbf{s}) = (t_{\min}(s_1), t_{\min}(s_2), \dots, t_{\min}(s_n))$$

and

$$H_{\min}^{(n)} = \frac{1}{n} H(Z_1 - t_{\min}(S_1), \dots, Z_n - t_{\min}(S_n))$$

respectively, and the proof follows.  $\square$

As a consequence we have the following results.

*Theorem 2 (SWIP Noise with Memory: Causal and Noncausal Case):* For SWIP noise and stationary state process, the instantaneous shift function

$$t^n(s^n) = (t_{\min}(s_1), \dots, t_{\min}(s_n))$$

where  $t_{\min}$  achieves  $H_{\min}^{(1)}$  in (26), is optimal for both causal and non-causal side information at the encoder. Thus

$$\begin{aligned} C_{\text{SI@TR}}^{\text{n caus}} &= C_{\text{SI@TR}}^{\text{caus}} \\ &= \log |\mathcal{X}| - \lim_{n \rightarrow \infty} \frac{1}{n} H(Z_1 - t_{\min}(S_1), \dots, Z_n - t_{\min}(S_n)) \end{aligned} \quad (33)$$

i.e., the optimum (causal or noncausal) encoder reduces to the instantaneous-prediction encoder of (18).

Note that unlike the general expression of Jelinek [9], for SWIP noise the optimization leading to the optimal strategy ( $t_{\min}$ ) is done with respect to single-letter quantities.

*Proof:* Lemma 3 implies that

$$C_{\text{SI@TR}}^{\text{n caus}} \leq \log |\mathcal{X}| - \inf_n H_{\min}^{(n)}.$$

Furthermore, by Lemma 4

$$H_{\min}^{(n)} \geq (1/n) H(Z_1 - t_{\min}(S_1), \dots, Z_n - t_{\min}(S_n))$$

which is further lower bounded by taking the limit as  $n$  goes to infinity (since  $Z_i - t_{\min}(S_i)$ ,  $i = 1, 2, \dots$  is a stationary process). Thus the noncausal capacity is *upper*-bounded by (33). Achievability follows by noting that the effective noise channel, resulting from the instantaneous encoding scheme, is a stationary additive noise channel having (33) as its capacity; see, e.g., [1].  $\square$

*Corollary 1 (Memoryless Noncausal Case):* For the memoryless additive noise channel (8), if the noise satisfies the SWIP property, then

$$C_{\text{SI@TR}}^{\text{n caus}} = C_{\text{SI@TR}}^{\text{caus}} = \log |\mathcal{X}| - H(Z - t_{\min}(S)).$$

## V. DISCUSSION

We identified the role of *instantaneous* noise prediction as the main ingredient in optimum transmission with side information over discrete additive-noise channels. This allows us to derive formulas for the capacity of memoryless channels with causal side information, and for SWIP noise, for the capacity with noncausal side information and for channels with memory.

In a future paper [5] we extend these concepts to rates below capacity and obtain expressions for the error exponent for additive noise channels with side information at the transmitter.

Our results do not extend to channels where the input is subject to some average input constraint. One reason for that is that under such a constraint each shift  $t(s)$  of the input may have a different "cost," and this changes the optimization of  $t(s)$ . Another reason can be seen in the case of a power-constrained Gaussian fading channel, with fading level known (say, causally) to the transmitter. Here, for a zero-mean noise our theory suggests that a fading independent code with zero shift is optimal. However, due to the average power constraint, allocating different power to code symbols at different fading levels improves the system performance [13].

### APPENDIX PROOF OF LEMMA 1

Suppose  $\mathcal{S} = \{s_1, \dots, s_k\}$ . Define

$$g_i(s, u) = \begin{cases} g(s_i, u), & \text{if } s = s_i \\ 0, & \text{if } s \neq s_i \end{cases}$$

so that

$$g(s, u) = \sum_{i=1}^k g_i(s, u).$$

Let us introduce the auxiliary random variables  $\{U_1, \dots, U_k\}$ , which are mutually independent and independent of  $(S, Z)$ , and are distributed as

$$p_{u_i}(u) \triangleq p_{U|S}(u | s_i).$$

With these definitions we have

$$\begin{aligned} &H(Z + g(S, U)) \\ &\stackrel{\text{a)}}{=} H\left(Z + \sum_{i=1}^k g_i(S, U)\right) \\ &\stackrel{\text{b)}}{=} H\left(Z + \sum_{i=1}^k g_i(S, U_i)\right) \\ &\stackrel{\text{c)}}{\geq} H\left(Z + \sum_{i=1}^k g_i(S, U_i) \middle| U_1 \cdots U_k\right) \\ &\stackrel{\text{d)}}{\geq} \min_{u_1 \cdots u_k} H\left(Z + \sum_{i=1}^k g_i(S, u_i) \middle| U_1 = u_1 \cdots U_k = u_k\right) \\ &\stackrel{\text{e)}}{=} \min_{u_1 \cdots u_k} H\left(Z + \sum_{i=1}^k g_i(S, u_i)\right) \\ &\stackrel{\text{f)}}{\geq} \min_{t: \mathcal{S} \rightarrow \mathcal{X}} H(Z + t(S)) \\ &\stackrel{\text{g)}}{=} H(Z + t_{\min}(S)) \end{aligned}$$

where b) follows from the Markov relation  $U - S - Z$ ; c) since conditioning reduces the entropy; and e) since  $(Z, S)$  is independent of  $(U_1, \dots, U_k)$ .

## ACKNOWLEDGMENT

The authors wish to thank Shlomo Shamai for his valuable comments, and for pointing their attention to the noncausal side information case and the work by Gelfand, Pinsker, and Costa. They also wish to thank Meir Feder for bringing to their attention the work of Elias on minimum error entropy prediction.

## REFERENCES

- [1] F. Alajaji, "Feedback does not increase the capacity of discrete channels with additive noise," *IEEE Trans. Inform. Theory*, vol. 41, pp. 546–549, Mar. 1995.
- [2] G. Caire and S. Shamai, "On the capacity of some channels with channel state information," *IEEE Trans. Inform. Theory*, vol. 45, pp. 2007–2019, Sept. 1999.
- [3] M. H. M. Costa, "Writing on dirty paper," *IEEE Trans. Inform. Theory*, vol. IT-29, pp. 439–441, May 1983.
- [4] P. Elias, "Predictive coding," *IRE Trans. Inform. Theory*, vol. IT-1, pp. 16–33, Mar. 1955.
- [5] U. Erez and R. Zamir, "Error exponents of modulo additive noise channels with side information at the transmitter," *IEEE Trans. Inform. Theory*, May 2000, revised, submitted for publication.
- [6] R. G. Gallager, *Information Theory and Reliable Communication*. New York: Wiley, 1968.
- [7] S. I. Gelfand and M. S. Pinsker, "Coding for channel with random parameters," *Probl. Pered. Inform. (Probl. Inform. Transm.)*, vol. 9, no. 1, pp. 19–31, 1980.
- [8] C. Heegard and A. El Gamal, "On the capacity of computer memory with defects," *IEEE Trans. Inform. Theory*, vol. IT-29, pp. 731–739, Sept. 1983.
- [9] F. Jelinek, "Indecomposable channels with side information at the transmitter," *Inform. Contr.*, vol. 8, pp. 36–55, 1965.
- [10] —, "Determination of capacity achieving input probabilities for a class of finite state channels with side information," *Inform. Contr.*, vol. 9, pp. 101–129, 1966.
- [11] R. J. McEliece and W. E. Stark, "Channels with block interference," *IEEE Trans. Inform. Theory*, vol. IT-30, pp. 44–53, Jan. 1984.

- [12] M. Salehi, "Capacity and coding for memories with real-time noisy defect information at encoder and decoder," *Proc. Inst. Elect. Eng.*, vol. 139, pp. 113–117, Apr. 1992.
- [13] S. Shamai, private communication, 1997.
- [14] C. E. Shannon, "Channels with side information at the transmitter," *IBM J. Res. Devel.*, vol. 2, pp. 289–293, Oct. 1958.
- [15] H. Viswanathan, "Capacity of Markov channels with receiver CSI and delayed feedback," *IEEE Trans. Inform. Theory*, vol. 45, pp. 761–771, Mar. 1999.
- [16] J. Wolfowitz, *Coding Theorems of Information Theory*. New York: Springer-Verlag, 1964.

## On the Training Distortion of Vector Quantizers

Tamás Linder, *Member, IEEE*

**Abstract**—The in-training-set performance of a vector quantizer as a function of its training set size is investigated. For squared error distortion and independent training data, worst case type upper bounds are derived on the minimum training distortion achieved by an empirically optimal quantizer. These bounds show that the training distortion can underestimate the minimum distortion of a truly optimal quantizer by as much as a constant times  $n^{-1/2}$ , where  $n$  is the size of the training data. Earlier results provide lower bounds of the same order.

**Index Terms**—Empirical design, training distortion, vector quantization, worst case bounds.

### I. INTRODUCTION

Vector quantizer design is usually based on a collection of example vectors, called the training set or training data. In general, the objective of a design algorithm (such as the popular generalized Lloyd algorithm [1]) is to find an empirically optimal quantizer, that is, a quantizer of a given codebook size whose distortion in quantizing the training data is minimum. The underlying principle of empirical design is that good performance inside the training set will imply good performance on other data produced by the source if the training set size is sufficiently large to represent well the source statistics. But training vectors may be costly to obtain and the computational cost of design may become prohibitive for large training sets. Therefore, it is of interest to quantify how the performance of the designed vector quantizer improves as the size of the training set increases.

Assume that the quantizer dimension and the codebook size are fixed. For any quantizer  $Q_n$  trained on  $n$  vectors, let  $D_n(Q_n)$  denote the training distortion of  $Q_n$  (its average distortion inside the training set) and let  $D(Q_n)$  denote the test distortion of  $Q_n$  (its distortion in coding independent test data). Note that both  $D_n(Q_n)$  and  $D(Q_n)$  are functions of the training set and therefore are random quantities. The quantity  $D(Q_n)$  is the "true" distortion of the designed quantizer; it is the performance figure one wants to be as close as possible to  $D(Q^*)$ , the distortion of a truly optimal quantizer  $Q^*$ . A design procedure is called consistent if the test distortion  $D(Q_n)$  of the resulting quantizer

$Q_n$  converges (in some sense) to its lower bound  $D(Q^*)$  as  $n \rightarrow \infty$ . Of particular interest are the empirically optimal quantizers  $Q_n^*$  minimizing the training distortion:  $D_n(Q_n^*) = \min_{Q_n} D_n(Q_n)$ . The consistency of empirically optimal quantizers was first investigated by Pollard [2], [3] for the case of mean-squared quantizer distortion. His results show, among other things, that for a stationary and ergodic training sequence, the test distortion  $D(Q_n^*)$  of an empirically optimal quantizer converges to  $D(Q^*)$  with probability one as  $n \rightarrow \infty$ .

Pollard's results imply that the performance of empirically optimal quantizers will approach the optimum performance as the training set size increases without bound. On the other hand, to determine the training set size sufficient for achieving a preassigned level of performance, one needs to study the dependence of  $D(Q_n^*)$  on finite  $n$ . Assume that the training set consists of  $n$  independent sample vectors drawn from the source distribution and let  $E[D(Q_n^*)]$  denote the expected value (taken over the training sequence) of the mean-squared test distortion  $D(Q_n^*)$ . In [4] it was shown that for all source distributions supported by a given bounded region, the test distortion of the empirically optimal quantizer satisfies  $E[D(Q_n^*)] - D(Q^*) \leq cn^{-1/2}$  for some positive constant  $c$ . This upper bound was shown to have the right order in a minimax sense in [5], where it was demonstrated that for any quantizer design method, there exist "bad" source distributions for which the test distortion of the resulting quantizer  $Q_n$  is lower-bounded as  $E[D(Q_n)] - D(Q^*) \geq c_1 n^{-1/2}$  for another positive constant  $c_1$ . The sample behavior of  $D(Q_n^*) - D(Q^*)$  for a class of smooth source densities was studied by Chou [6], and upper bounds on  $E[D(Q_n^*)] - D(Q^*)$  for dependent (mixing) training data were developed by Zeevi [7]. The dependence of the test distortion on the training set size was also empirically investigated by Cosman *et al.* [8] and Cohn *et al.* [9] in the context of image coding.

In this correspondence, the focus of attention is the less studied training distortion  $D_n(Q_n^*)$ . Since the value of  $D_n(Q_n^*)$  is obtained as a by-product of the design procedure without requiring additional test data, it can be considered an inexpensive estimate of  $D(Q_n^*)$  or  $D(Q^*)$ . For an empirically optimal quantizer minimizing  $D_n(Q_n)$ , one always has

$$E[D_n(Q_n^*)] \leq D(Q^*) \leq E[D(Q_n^*)].$$

The exact relationship between the training, test, and optimal distortions is only known in the special case of quantizers with codebook size  $k = 1$ . In this case, it is easy to see that the single unique codepoint of the empirically optimal quantizer is the arithmetic average of the  $n$  training vectors, and therefore,

$$E[D_n(Q_n^*)] = D(Q^*) \left(1 - \frac{1}{n}\right)$$

and

$$E[D(Q_n^*)] = D(Q^*) \left(1 + \frac{1}{n}\right)$$

for all source distributions with finite second moment.

The problem becomes nontrivial when quantizers with more than one codepoint are considered, and in general little is known about the size of the difference  $D(Q^*) - E[D_n(Q_n^*)]$ . In this respect, Abaya and Wise [10] proved that under general conditions the expected training distortion is a consistent estimate of the optimal distortion in the sense that  $D(Q^*) - E[D_n(Q_n^*)] \rightarrow 0$  as  $n \rightarrow \infty$ . The size of the bias of  $D_n(Q_n^*)$  in estimating  $D(Q^*)$  was first investigated in a recent work by Kim and Bell [11] who showed that for squared error distortion

$$E[D_n(Q_n^*)] \leq D(Q^*) \left(1 - \frac{1}{n}\right) \quad (1)$$

Manuscript received July 14, 1999; revised February 1, 2000. This work was supported in part by Queen's University, Kingston, Ont., Canada, and by the Natural Sciences and Engineering Research Council (NSERC) of Canada.

The author is with the Department of Mathematics and Statistics, Queen's University, Kingston, Ont., Canada K7L 3N6 (e-mail: linder@mast.queensu.ca).

Communicated by P. A. Chou, Associate Editor for Source Coding.  
Publisher Item Identifier S 0018-9448(00)04282-6.

for any source distribution with a finite second moment. No matching lower bounds or sharper upper bounds seem to be available in the literature.

In this correspondence, we apply techniques developed in [5] for proving minimax bounds in quantizer design to show that in the worst case, the difference  $D(Q^*) - E[D_n(Q_n^*)]$  is proportional to  $n^{-1/2}$ . After introducing the necessary definitions in Section II, three results concerning the mean-squared training distortion of an empirically optimal quantizer are given in Section III. Theorem 1 proves the existence of “badly behaved” distributions on a bounded support set for which

$$E[D_n(Q_n^*)] \leq D(Q^*) - \frac{c}{\sqrt{n}} \quad (2)$$

for a constant  $c > 0$  which depends on the quantizer dimension, the codebook size (which is assumed to be at least 3), and the diameter of the support set. Theorem 2 reformulates this bound in terms of the *training ratio*  $\beta = n/k$  (where  $k \geq 3$  is the codebook size) by showing that there exist source distributions for which

$$E[D_n(Q_n^*)] \leq D(Q^*) \left(1 - \frac{c_0}{\sqrt{\beta}}\right) \quad (3)$$

where  $c_0 > 0$  is a universal constant. Theorem 3 presents an improved, explicit form of an earlier result in [4] to show that the lower bound

$$E[D_n(Q_n^*)] \geq D(Q^*) - \frac{\hat{c}}{\sqrt{n}}$$

holds for a  $\hat{c} > 0$ , uniformly for all sources supported on a given bounded set. This shows that bound (2) is tight in the sense that only the constants may be improved. The proofs of these results are given in Section IV.

The bounds (2) and (3) immediately demonstrate that for larger values of  $n$  any bound in the form of (1) will be very loose for some source distributions. On the other hand, note that (1) holds for all source distributions while (2) and (3) are worst case bounds. Thus our results do not exclude the possibility that the  $n^{-1}$  term in (1) has the right order for a restricted class of “smooth” source distributions. Potential candidates are the source densities satisfying Pollard’s central limit theorem [12] for empirical quantizer design.

## II. PRELIMINARIES AND PROBLEM FORMULATION

A vector quantizer  $Q$  of dimension  $d$  and codebook size  $k$  is a (measurable) mapping of the  $d$ -dimensional Euclidean space  $\mathbb{R}^d$  into a finite set of points  $\{y_1, \dots, y_k\}$ . The points  $y_i \in \mathbb{R}^d$ ,  $i = 1, \dots, k$  are called the *codepoints* or *codevectors* and the collection  $\{y_1, \dots, y_k\}$  is called the codebook.

For any  $x \in \mathbb{R}^d$ , let  $\|x\|$  denote its Euclidean norm. Given a  $d$ -dimensional random vector  $X$  with probability distribution  $\mu_X$  and finite second moment  $E\|X\|^2 < \infty$ , the mean-squared distortion of a vector quantizer  $Q$  is

$$D(Q) = E\|X - Q(X)\|^2 = \int_{\mathbb{R}^d} \|x - Q(x)\|^2 \mu_X(dx).$$

A vector quantizer  $Q$  with codebook  $\{y_1, \dots, y_k\}$  is called a *nearest neighbor quantizer* if for all  $x \in \mathbb{R}^d$

$$\|x - Q(x)\|^2 = \min_{1 \leq i \leq k} \|x - y_i\|^2.$$

For any source distribution, a nearest neighbor quantizer has minimum distortion among all other quantizers with the same codebook. This fact allows us to consider only nearest neighbor quantizers in this correspondence without loss of generality.

For any  $k \geq 1$ , let  $\tilde{\mathcal{Q}}_k$  be the family of all  $d$ -dimensional nearest neighbor vector quantizers with  $k$  codevectors. A quantizer  $Q^* \in \tilde{\mathcal{Q}}_k$  is called an *optimal  $k$ -point quantizer* for  $\mu_X$  if it has minimum distortion

$$D(Q^*) = \min_{Q \in \tilde{\mathcal{Q}}_k} E\|X - Q(X)\|^2.$$

(An optimal  $Q^*$  always exists if  $E\|X\|^2 < \infty$ , see, e.g., [3].)

Let  $X_1, X_2, \dots, X_n$  be independent and identically distributed (i.i.d.)  $d$ -dimensional random vectors drawn according to  $\mu_X$ . The collection  $\{X_i\}_{i=1}^n$  is called the *training data* or training set. The average squared distortion of a vector quantizer  $Q$  on the training set is

$$D_n(Q) = \frac{1}{n} \sum_{i=1}^n \|X_i - Q(X_i)\|^2.$$

Let  $Q_n^*$  denote an *empirically optimal* quantizer in  $\tilde{\mathcal{Q}}_k$ , that is,  $Q_n^*$  is a  $k$ -point quantizer which has minimum average squared distortion

$$D_n(Q_n^*) = \min_{Q \in \tilde{\mathcal{Q}}_k} \frac{1}{n} \sum_{i=1}^n \|X_i - Q(X_i)\|^2$$

over the training set. The random quantity  $D_n(Q_n^*)$  is called the *training distortion* of the empirically optimal quantizer. Note that the dependence of  $Q_n^*$  and  $D_n(Q_n^*)$  on the training data  $\{X_i\}_{i=1}^n$  is suppressed in the notation.

Our goal is to compare the expected training distortion

$$E[D_n(Q_n^*)] = E \left\{ \min_{Q \in \tilde{\mathcal{Q}}_k} \frac{1}{n} \sum_{i=1}^n \|X_i - Q(X_i)\|^2 \right\}$$

of an empirically optimal quantizer  $Q_n^*$  with the distortion  $D(Q^*)$  of an optimal quantizer  $Q^*$ . Since

$$D_n(Q^*) \geq D_n(Q_n^*)$$

and

$$D(Q^*) = E[D_n(Q^*)]$$

we always have

$$D(Q^*) \geq E[D_n(Q_n^*)].$$

Moreover, it is easy to see that strict inequality holds whenever  $D(Q^*) > 0$ . Let  $\mathcal{P}$  denote the collection of all source distributions which are supported by a given bounded set. Our results concern the maximum deviation over  $\mathcal{P}$  of the expected training distortion from the optimal distortion, that is, the quantity

$$\sup_{\mu_X \in \mathcal{P}} (D(Q^*) - E[D_n(Q_n^*)]).$$

In order to be consistent with earlier work [13], [5] on worst case bounds in vector quantization, we will formulate our results in terms of classes  $\mathcal{P}(B)$  containing all source distributions which satisfy the peak power constraint  $P\{(1/d)\|X\|^2 \leq B\} = 1$ . In other words, for any  $B > 0$ , the class  $\mathcal{P}(B)$  consists of all source distributions whose support is contained in the ball  $\{x : \|x\| \leq \sqrt{dB}\}$ .

## III. RESULTS

Our first result shows that for training data of size  $n$ , the difference  $D(Q^*) - E[D_n(Q_n^*)]$  of the minimum distortion of an optimal quantizer and the expected training distortion of the empirically optimal quantizer can be as large as constant times  $n^{-1/2}$ .

*Theorem 1:* For any quantizer dimension  $d \geq 1$  and codebook size  $k \geq 3$  there exists a distribution  $\mu_X \in \mathcal{P}(B)$  such that for all training set size  $n \geq \frac{2}{3}k$

$$E[D_n(Q_n^*)] \leq D(Q^*) - \frac{c(B, d, k)}{\sqrt{n}} \quad (4)$$

where

$$c(B, d, k) = \frac{Bd\sqrt{k^{1-\frac{4}{d}}}}{2^{83}}.$$

In the next result the relative difference of the training and optimal distortions is considered, in which case a very simple bound can be obtained in terms of the training ratio  $\beta = n/k$ .

*Theorem 2:* For any quantizer dimension  $d \geq 1$  and codebook size  $k \geq 3$  there exists a distribution  $\mu_X \in \mathcal{P}(B)$  such that for all training set size  $n \geq \frac{2}{3}k$

$$E[D_n(Q_n^*)] \leq D(Q^*) \left(1 - \frac{c_0}{\sqrt{\beta}}\right)$$

where  $c_0 = \frac{1}{4} \sqrt{\frac{7}{6}} \approx 0.27$ .

Theorems 1 and 2 are proved by using a construction of “bad” distributions introduced in [5]. This method uses discrete distributions supported by a finite number of points, although a modified construction using distributions with smooth densities is possible at the expense of complicating an already somewhat involved argument. An important point is that in [5] the choice of these “bad” distributions depends on the training set size  $n$ . In our case, due apparently to the fact that we deal with the training distortion instead of the test distortion, we are able to construct one “bad” distribution which works for all large enough  $n$ . Therefore, Theorem 1 guarantees the existence of at least one *fixed* source distribution in  $\mathcal{P}(B)$  such that

$$\liminf_{n \rightarrow \infty} \sqrt{n}(D(Q^*) - E[D_n(Q_n^*)]) > 0.$$

Next we examine in what sense (if any) the bound of Theorem 1 is tight. The constant  $c(B, d, k)$  is rather small and can probably be improved. But the more fundamental question is whether  $n^{-1/2}$  can be replaced with something larger. To answer this question in the negative, we note that for all  $\mu_X \in \mathcal{P}(B)$

$$D(Q^*) - E[D_n(Q_n^*)] \leq E \left\{ \sup_{Q \in \mathcal{Q}_k} [D(Q) - D_n(Q)] \right\} \quad (5)$$

where  $\mathcal{Q}_k$  denotes the family of all  $k$ -point nearest neighbor quantizers with codepoints inside the sphere  $\{x: \|x\| \leq \sqrt{dB}\}$  (see the proof of Theorem 3). Any uniform upper bound on the expectation on the right-hand side will result in a uniform lower bound on  $E[D_n(Q_n^*)]$ . The existence of such an upper bound of order  $n^{-1/2}$  has been pointed out in [4] (see [4, the discussion following Corollary 1]) although in an asymptotic form and without explicit constants. Nevertheless, such a bound implies that the bound of Theorem 1 is essentially tight.

The following theorem presents a new form of this lower bound which is tighter than those given by existing results and has a more attractive, nonasymptotic form.

*Theorem 3:* For any quantizer dimension  $d \geq 1$ , codebook size  $k \geq 1$ , training set size  $n \geq 1$ , we have

$$E[D_n(Q_n^*)] \geq D(Q^*) - \frac{\hat{c}(B, d, k)}{\sqrt{n}}$$

for all  $\mu_X \in \mathcal{P}(B)$ , where  $\hat{c}(B, d, k) = 96Bd^{3/2}\sqrt{k}$ .

The result is based on a nonasymptotic upper bound on

$$E \left\{ \sup_{Q \in \mathcal{Q}_k} [D(Q) - D_n(Q)] \right\}.$$

At the core of the proof is a simple and elegant version of the classic “metric entropy” bound [14], [15] of empirical process theory, recently proved by Cesa-Bianchi and Lugosi [16], which allows us to provide an explicit form of the constant  $\hat{c}(B, d, k)$ .

In summary, Theorems 1 and 3 show that for independent training data of size  $n$ , the maximum difference  $D(Q^*) - E[D_n(Q_n^*)]$  of the distortion of an optimal quantizer and the expected training distortion of the empirically optimal quantizer is of order  $n^{-1/2}$ . More formally, these results imply that for all  $k \geq 3$  and large enough  $n$

$$\frac{c}{\sqrt{n}} \leq \sup_{\mu_X \in \mathcal{P}(B)} (D(Q^*) - E[D_n(Q_n^*)]) \leq \frac{\hat{c}}{\sqrt{n}}$$

for some constants  $c, \hat{c} > 0$  depending on  $d, k$ , and  $B$ .

#### IV. PROOFS

*Proof of Theorem 1:* We simplify the notation by assuming that  $B = 1$  (that is,  $\mu_X$  has to satisfy  $P\{\|X\|^2 \leq \sqrt{d}\} = 1$ ). Since we consider mean-squared distortion, for arbitrary  $B > 0$  the result follows by straightforward scaling.

To demonstrate the existence of a  $\mu_X$  satisfying the bound of the theorem, we will use a modified form of a construction introduced in [5, the proof of Theorem 1]. Just as in [5], the basic idea is to construct a source distribution such that with constant positive probability, the empirically optimal quantizer is sufficiently “far” from the optimal quantizer. However, new techniques are needed to derive the desired bound since we consider the training distortion (the empirically optimal quantizer is a function of the data on which its distortion is evaluated), while in [5] the test distortion was considered (the distortion is evaluated on independent data).

Assume that  $k \geq 3$  is divisible by 3 (we will relax this assumption later) and let  $m = \frac{2}{3}k$  (note that  $m$  is even). Let  $\Delta > 0$  be a constant to be specified later and let  $z_1, \dots, z_m$  be  $m$  points in  $\mathbb{R}^d$  satisfying  $\|z_i - z_j\| \geq 3\Delta$  for all  $i \neq j$ . Let  $w$  denote the  $d$ -vector  $w = (\Delta, 0, \dots, 0)$ . The proposed  $\mu_X$  is the uniform distribution concentrated on the  $2m$  points  $\{z_i, z_i + w; i = 1, \dots, m\}$ , that is,

$$\mu_X(\{z_i\}) = \mu_X(\{z_i + w\}) = \frac{1}{2m}, \quad 1 \leq i \leq m. \quad (6)$$

The parameters of  $\mu_X$  are  $\Delta$  and the points  $z_1, \dots, z_m$ . We assume that  $z_1, \dots, z_m$  and  $\Delta$  are such that  $\mu_X \in \mathcal{P}(1)$ , i.e.,

$$\max_{1 \leq i \leq m} (\|z_i\|, \|z_i + w\|) \leq \sqrt{d}.$$

Clearly, if  $\Delta$  is small enough this is always possible; the specific choice of  $\Delta$  will be given later. A key feature of  $\mu_X$  is that an optimal quantizer  $Q^*$  for  $\mu_X$  with  $k = \frac{3}{2}m$  codepoints has a very simple structure.

*Lemma 1:* Let  $\mu_X$  be defined by (6) and assume that  $\|z_i - z_j\| \geq 3\Delta$  for all  $i \neq j$ . Let  $S$  be any subset of  $\{1, \dots, m\}$  of cardinality  $|S| = m/2$ . Then the quantizer which has one codepoint at  $z_i + \frac{1}{2}w$  for each  $i \in S$  and has codepoints at both  $z_i$  and  $z_i + w$  for each  $i \in \{1, \dots, m\} \setminus S$  is an optimal  $k$ -point quantizer for  $\mu_X$ .

The assertion of the lemma is intuitively clear; the proof is given in [5, the Appendix]. Note that the optimal quantizer is not unique, and in fact there are  $\binom{m}{m/2}$  optimal quantizers for  $\mu_X$ .

Let the training data  $X_1, X_2, \dots, X_n$  be drawn independently from  $\mu_X$  and let  $N_i$  be the number of training data points falling in the set  $\{z_i, z_i + w\}$ , i.e.,

$$N_i = |\{j: X_j = z_i \text{ or } X_j = z_i + w, j = 1, \dots, n\}|.$$

Let  $Q^*$  have one codepoint at  $z_i + \frac{1}{2}w$  for each  $i \leq m/2$  and two codepoints at  $z_i$  and  $z_i + w$  for each  $m/2 + 1 \leq i \leq m$ . Then  $Q^*$  is an optimal  $k$ -point quantizer by Lemma 1, and its distortion is given in terms of the  $N_i$  by

$$\begin{aligned} D(Q^*) &= E \left\{ \frac{1}{n} \sum_{j=1}^n \|X_j - Q^*(X_j)\|^2 \right\} \\ &= E \left\{ \frac{\Delta^2}{4} \frac{1}{n} \sum_{i=1}^{m/2} N_i \right\} \end{aligned} \quad (7)$$

where the second equality holds because  $\|Q^*(X_j) - X_j\|^2 = \Delta^2/4$  if  $X_j$  takes value in  $\bigcup_{i=1}^{m/2} \{z_i, z_i + w\}$  and  $\|Q^*(X_j) - X_j\|^2 = 0$  otherwise.

We now define a training-set-dependent quantizer  $Q_n$  to approximate the empirically optimal  $k$ -point quantizer  $Q_n^*$ . Let  $\sigma(1), \dots, \sigma(m)$  be the permutation of  $1, \dots, m$  obtained by switching the positions of the indices  $i$  and  $m/2 + i$  (i.e., letting  $\sigma(i) = m/2 + i$  and  $\sigma(m/2 + i) = i$ ) for each  $i \leq m/2$  such that  $N_i > N_{m/2+i}$ . Furthermore, let  $Q_n$  be the  $k$ -point quantizer whose codepoints are  $z_{\sigma(i)} + \frac{1}{2}w$  for  $i \leq m/2$ , and  $z_{\sigma(i)}, z_{\sigma(i)} + w$  for  $m/2 + 1 \leq i \leq m$ . Then we have

$$\begin{aligned} E[D_n(Q_n)] &= E \left\{ \frac{1}{n} \sum_{j=1}^n \|X_j - Q_n(X_j)\|^2 \right\} \\ &= E \left\{ \frac{\Delta^2}{4} \frac{1}{n} \sum_{i=1}^{m/2} N_{\sigma(i)} \right\} \end{aligned} \quad (8)$$

since  $\|Q_n(X_j) - X_j\|^2 = \Delta^2/4$  if

$$X_j \in \bigcup_{i=1}^{m/2} \{z_{\sigma(i)}, z_{\sigma(i)} + w\}$$

and  $\|Q_n(X_j) - X_j\|^2 = 0$  otherwise. Since the empirically optimal quantizer  $Q_n^*$  minimizes the training distortion over all  $k$ -point quantizers, we have

$$E[D_n(Q_n)] \geq E[D_n(Q_n^*)].$$

Therefore, using (7) and (8), we can lower-bound the difference  $D(Q^*) - E[D_n(Q_n^*)]$  as

$$D(Q^*) - E[D_n(Q_n^*)] \geq \frac{\Delta^2}{4} \frac{1}{n} E \left\{ \sum_{i=1}^{m/2} (N_i - N_{\sigma(i)}) \right\}. \quad (9)$$

In the rest of the proof we will demonstrate that the expectation on the right-hand side is of order  $n^{-1/2}$ . First note that for all  $i \leq m/2$  we have  $N_{\sigma(i)} = N_i$  if  $N_i \leq N_{m/2+i}$ , and  $N_{\sigma(i)} = N_{m/2+i}$  otherwise. Therefore,  $N_i - N_{\sigma(i)} = (N_i - N_{m/2+i})^+$ , where  $x^+ = \max(x, 0)$ . Thus

$$\begin{aligned} E \left\{ \sum_{i=1}^{m/2} (N_i - N_{\sigma(i)}) \right\} &= E \left\{ \sum_{i=1}^{m/2} (N_i - N_{m/2+i})^+ \right\} \\ &= \frac{m}{2} E[(N_1 - N_{m/2+1})^+] \end{aligned} \quad (10)$$

since the pairs  $(N_i, N_{m/2+i})$  have the same distribution. For each  $j \in \{1, \dots, n\}$  define the random variable  $Y_j$  as follows:  $Y_j = 1$  if the training vector  $X_j$  contributes to  $N_1$ ,  $Y_j = -1$  if the training vector  $X_j$  contributes to  $N_{m/2+1}$ , and  $Y_j = 0$  otherwise. Then

$$P\{Y_j = 1\} = P\{Y_j = -1\} = \frac{1}{m}$$

and

$$P\{Y_j = 0\} = 1 - \frac{2}{m}.$$

Define

$$S_n = \sum_{j=1}^n Y_j.$$

Then  $S_n = N_1 - N_{m/2+1}$  and since  $S_n$  is distributed symmetrically about zero

$$E[(N_1 - N_{m/2+1})^+] = \frac{1}{2} E|S_n|. \quad (11)$$

To lower-bound the last expectation we will use the following useful inequality: for any random variable  $Z$  with finite fourth moment

$$E|Z| \geq \frac{(E[Z^2])^{3/2}}{(E[Z^4])^{1/2}} \quad (12)$$

(see [17, p. 194] or [18, Lemma A.4]). Since the  $Y_j$  are independent and identically distributed, and have zero mean, we have

$$E[S_n^2] = nE[Y_1^2] = \frac{2n}{m}.$$

On the other hand, expanding

$$S_n^4 = \left( \sum_{j=1}^n Y_j \right)^4$$

yields

$$\begin{aligned} E[S_n^4] &= nE[Y_1^4] + 3n(n-1)(E[Y_1^2])^2 \\ &= \frac{2n}{m} + 3n(n-1) \left( \frac{2}{m} \right)^2 \\ &\leq 4 \left( \frac{2n}{m} \right)^2 \end{aligned}$$

where the inequality holds if  $n \geq m$ . Hence (12) gives

$$E|S_n| \geq \frac{1}{\sqrt{2}} \sqrt{\frac{n}{m}}.$$

Combine this with (11), (10), and (9) to obtain

$$D(Q^*) - E[D_n(Q_n^*)] \geq \frac{\Delta^2}{2^4 \sqrt{2}} \sqrt{\frac{m}{n}}. \quad (13)$$

To maximize this lower bound, we need to make  $\Delta$  as large as possible under the constraint  $\mu_X \in \mathcal{P}(1)$ . A simple packing argument shows (see [5, Step 14, proof of Theorem 1]) that the choice

$$\Delta = \frac{\sqrt{d}}{4m^{1/d}}$$

is possible while maintaining the separation condition

$$\|z_i - z_j\| \geq 3\Delta, \quad i \neq j$$

and also satisfying

$$\max_{1 \leq i \leq m} (\|z_i\|, \|z_i + w\|) \leq \sqrt{d}.$$

Substituting  $\Delta = \frac{\sqrt{d}}{4m^{1/d}}$  and  $m = \frac{2}{3}k$  in (13) we can conclude that

$$D(Q^*) - E[D_n(Q_n^*)] \geq \frac{d}{2^8 \sqrt{3}} \sqrt{\frac{k^{1-\frac{4}{d}}}{n}} \quad (14)$$

which proves the statement of the theorem for all  $k \geq 3$  divisible by 3 and  $n \geq \frac{2}{3}k$ .

The proof for the case when  $k$  is not a multiple of 3 involves a slightly modified construction. In this case, we let  $m$  be the unique even positive integer satisfying  $k = 3m/2 + p$ , where  $p$  is either 1 or 2. In the definition of the modified  $\mu_X$  the points  $z_i, z_i + w$  are assigned probability  $\frac{1}{2(m+1)}$ , and we augment the support of  $\mu_X$  by one additional point with probability  $\frac{1}{(m+1)}$  (when  $p = 1$ ), or a pair of points, each having probability  $\frac{1}{2(m+1)}$  (when  $p = 2$ ). Since we now have  $m + 1$  pairs, we set

$$\Delta = \frac{\sqrt{d}}{4(m+1)^{1/d}}.$$

The details of the derivation are omitted since these are almost identical to the case when  $k$  is divisible by 3. Instead of (13), in this case we obtain the slightly weaker bound

$$\begin{aligned} D(Q^*) - E[D_n(Q_n^*)] &\geq \frac{\Delta^2}{2^4 \sqrt{2}} \frac{m}{\sqrt{m+1}} \frac{1}{\sqrt{n}} \\ &\geq \frac{\Delta^2}{2^4 \sqrt{2}} \sqrt{\frac{2}{3}} \sqrt{\frac{m}{n}} \\ &\geq \frac{d}{2^8 3} \sqrt{\frac{k^{1-\frac{4}{d}}}{n}} \end{aligned} \quad (15)$$

where the second inequality holds because  $m \geq 2$  and the third holds because  $m \geq \frac{2}{3}(k-2)$  and  $k \geq 4$ .  $\square$

*Proof of Theorem 2:* The construction in the proof of Theorem 1 is used again. Assume first that  $k$  is divisible by 3. Then by (7) we have

$$D(Q^*) = \frac{\Delta^2}{8}$$

since  $E(N_i) = \frac{n}{m}$ . Hence (13) can be rewritten as

$$\begin{aligned} E[D_n(Q_n^*)] &\leq D(Q^*) - \frac{\Delta^2}{2^4 \sqrt{2}} \sqrt{\frac{m}{n}} \\ &= D(Q^*) \left(1 - \frac{1}{2\sqrt{2}} \sqrt{\frac{m}{n}}\right) \\ &= D(Q^*) \left(1 - \frac{1}{2\sqrt{3}} \sqrt{\frac{k}{n}}\right). \end{aligned}$$

If  $k \geq 3$  is not divisible by 3, then  $\mu_X$  is modified as in the last part of the proof of Theorem 1. In this case, the distortion of  $Q^*$  is

$$D(Q^*) = \frac{\Delta^2}{8} \frac{m}{m+1}$$

where  $m$  is the unique even positive integer such that  $k = 3m/2 + p$ , where  $p$  is either 1 or 2. Then (15) implies

$$\begin{aligned} E[D_n(Q_n^*)] &\leq D(Q^*) - \frac{\Delta^2}{2^4 \sqrt{2}} \frac{m}{\sqrt{m+1}} \frac{1}{\sqrt{n}} \\ &= D(Q^*) \left(1 - \frac{1}{2\sqrt{2}} \sqrt{\frac{m+1}{n}}\right) \\ &\leq D(Q^*) \left(1 - \frac{1}{4} \sqrt{\frac{7}{6}} \sqrt{\frac{k}{n}}\right) \end{aligned}$$

where the second inequality holds since  $m \geq \frac{2}{3}(k-2)$  and  $k \geq 4$ .  $\square$

*Proof of Theorem 3:* As in the proof of Theorem 1, we assume that  $B = 1$  and obtain the result for general  $B$  by scaling. Since  $X, X_1, \dots, X_n$  is an i.i.d. sequence and  $Q^*$  is a  $k$ -point quantizer with minimum distortion, we can write

$$\begin{aligned} D(Q^*) - E[D_n(Q_n^*)] &\leq E\{E[\|X - Q_n^*(X)\|^2 | X_1, \dots, X_n] - D_n(Q_n^*)\} \\ &\leq E\left\{\sup_{Q \in \mathcal{Q}_k} [D(Q) - D_n(Q)]\right\} \end{aligned} \quad (16)$$

where  $\mathcal{Q}_k$  denotes the family of all  $k$ -point nearest neighbor quantizers with codepoints inside the sphere  $S(\sqrt{d}) = \{x: \|x\| \leq \sqrt{d}\}$ . The second inequality holds since  $P\{\|X_i\| \leq \sqrt{d}\} = 1$  for all  $i$  and therefore the codepoints of  $Q_n^*$  are inside  $S(\sqrt{d})$  with probability one.

For any  $Q \in \mathcal{Q}_k$  let the random variable  $T_n^{(Q)}$  be defined by

$$\begin{aligned} T_n^{(Q)} &= \frac{1}{2} \sum_{i=1}^n (E[\|X_i - Q(X_i)\|^2] - \|X_i - Q(X_i)\|^2) \\ &= \frac{n}{2} (D(Q) - D_n(Q)) \end{aligned}$$

so that by (16)

$$D(Q^*) - E[D_n(Q_n^*)] \leq \frac{2}{n} E\left\{\sup_{Q \in \mathcal{Q}_k} T_n^{(Q)}\right\}. \quad (17)$$

We will use a standard but effective technique of empirical process theory to upper-bound the expectation on the right-hand side.

First we recall some definitions. Let  $(S, \rho)$  be a totally bounded metric space. For any  $F \subset S$  and  $\epsilon > 0$  the  $\epsilon$ -covering number  $N_\rho(F, \epsilon)$  of  $F$  is defined as the minimum number of closed balls with radius  $\epsilon$  whose union covers  $F$ .

A family  $\{T_s : s \in S\}$  of zero-mean random variables indexed by the metric space  $(S, \rho)$  is called *subgaussian* in the metric  $\rho$  if for any  $\lambda > 0$  and  $s, s' \in S$  we have

$$E[e^{\lambda(T_s - T_{s'})}] \leq e^{\lambda^2 \rho(s, s')/2}.$$

The family  $\{T_s : s \in S\}$  is called *sample continuous* if for any sequence  $s_1, s_2, \dots \in S$  such that  $s_j \rightarrow s \in S$  we have  $T_{s_j} \rightarrow T_s$  with probability one.

The following result gives an upper bound on the expected supremum of the random variables  $\{T_s : s \in S\}$  in terms of the covering number of the index space. It provides a version of a classical result in empirical process theory (see, e.g., [15]) with an explicit constant.

*Lemma 2 ([16, Proposition 3]):* If  $\{T_s : s \in S\}$  is subgaussian and sample continuous in the metric  $\rho$ , then

$$E\left\{\sup_{s \in S} T_s\right\} \leq 12 \int_0^{\text{diam}(S)/2} \sqrt{\ln N_\rho(S, \epsilon)} d\epsilon$$

where  $\text{diam}(S) = \sup_{s, s' \in S} \rho(s, s')$  is the diameter of  $S$ .

To apply the above result we need to show that when  $\mathcal{Q}_k$  is equipped with a suitable metric, the family of random variables  $\{T_n^{(Q)} : Q \in \mathcal{Q}_k\}$  is subgaussian and sample continuous. For any  $Q, Q' \in \mathcal{Q}_k$  define

$$\rho_n(Q, Q') = \sqrt{n} \sup_{\|x\| \leq d} \left| \|x - Q(x)\|^2 - \|x - Q'(x)\|^2 \right|.$$

Clearly,  $\rho_n$  is a metric on  $\mathcal{Q}_k$ . Also, for any  $Q, Q' \in \mathcal{Q}_k$  we have

$$|T_n^{(Q)} - T_n^{(Q')}| \leq \sqrt{n} \rho_n(Q, Q') \quad (18)$$

with probability one, which implies that  $\{T_n^{(Q)} : Q \in \mathcal{Q}_k\}$  is sample continuous. To show that  $\{T_n^{(Q)} : Q \in \mathcal{Q}_k\}$  is subgaussian in  $\rho_n$ , we recall Hoeffding's inequality [19] which states that if  $Y_1, \dots, Y_n$

are independent zero-mean random variables such that  $a \leq Y_i \leq b$ ,  $i = 1, \dots, n$  with probability one, then for all  $\lambda > 0$

$$E \left[ e^{\lambda \sum_{i=1}^n Y_i} \right] \leq e^{\lambda^2 n(b-a)^2/8}.$$

For  $i = 1, \dots, n$  let

$$Y_i = \frac{1}{2} (D(Q) - \|X_i - Q(X_i)\|^2) - \frac{1}{2} (D(Q') - \|X_i - Q'(X_i)\|^2).$$

Then

$$T_n^{(Q)} - T_n^{(Q')} = \sum_{i=1}^n Y_i$$

where the  $Y_i$  are independent, have zero mean, and

$$|Y_i| \leq \frac{1}{\sqrt{n}} \rho_n(Q, Q')$$

for all  $i$ . Hence Hoeffding's inequality implies

$$E \left[ e^{\lambda (T_n^{(Q)} - T_n^{(Q')})} \right] \leq e^{\lambda^2 \rho_n(Q, Q')^2/2}$$

proving that  $\{T_n^{(Q)}; Q \in \mathcal{Q}_k\}$  is subgaussian in  $\rho_n$ . Therefore, Lemma 2 gives

$$E \left\{ \sup_{Q \in \mathcal{Q}_k} T_n^{(Q)} \right\} \leq 12 \int_0^{\text{diam}(\mathcal{Q}_k)/2} \sqrt{\ln N_{\rho_n}(\mathcal{Q}_k, \epsilon)} d\epsilon. \quad (19)$$

To evaluate the integral we need the following bound on the covering number of  $\mathcal{Q}_k$ .

*Lemma 3 ([5, Corollary 1]):* For any  $0 < \epsilon \leq 4d$  and  $k \geq 1$ , the covering number of  $\mathcal{Q}_k$  in the metric

$$\rho(Q, Q') = \sup_{\|x\|^2 \leq d} \|\|x - Q(x)\|^2 - \|x - Q'(x)\|^2\|$$

is bounded as

$$N_{\rho}(\mathcal{Q}_k, \epsilon) \leq \left( \frac{16d}{\epsilon} \right)^{kd}.$$

Since  $\rho_n(Q, Q') = \sqrt{n} \rho(Q, Q')$ , the preceding lemma implies that

$$N_{\rho_n}(\mathcal{Q}_k, \epsilon) \leq \left( \frac{16d\sqrt{n}}{\epsilon} \right)^{kd}$$

for all  $0 < \epsilon \leq \sqrt{n} 4d$ . Moreover, since

$$\sup_{\|x\|^2 \leq d} \|x - Q(x)\|^2 \leq 4d$$

for all  $Q \in \mathcal{Q}_k$ , we have  $\text{diam}(\mathcal{Q}_k) \leq \sqrt{n} 4d$ . Therefore, (17) and (18) imply

$$\begin{aligned} D(Q^*) - E[D_n(Q_n^*)] &\leq \frac{24}{n} \int_0^{\sqrt{n} 2d} \sqrt{\ln \left( \frac{16d\sqrt{n}}{\epsilon} \right)^{kd}} d\epsilon \\ &= \frac{24\sqrt{kd}}{n} \int_0^{\sqrt{n} 2d} \sqrt{\ln \left( \frac{16d\sqrt{n}}{\epsilon} \right)} d\epsilon. \end{aligned} \quad (20)$$

We can upper-bound the last integral as

$$\begin{aligned} \int_0^{\sqrt{n} 2d} \sqrt{\ln \left( \frac{16d\sqrt{n}}{\epsilon} \right)} d\epsilon &= 16d\sqrt{n} \int_0^{1/8} \sqrt{\ln \left( \frac{1}{x} \right)} dx \\ &\leq 2d\sqrt{n} \sqrt{8} \int_0^{1/8} \ln \left( \frac{1}{x} \right) dx \\ &= 2d\sqrt{n} \sqrt{\ln 8 + 1} \\ &\leq 4d\sqrt{n} \end{aligned}$$

where we first used the change of variable  $x = \epsilon/(16d\sqrt{n})$  and then applied Jensen's inequality to the concave function  $f(t) = \sqrt{t}$ . Combining this bound with (20) proves Theorem 3.  $\square$

#### ACKNOWLEDGMENT

The author wishes to thank G. Lugosi for helpful discussions. Also, thanks are due to an anonymous reviewer for pointing out how to simplify the proof of Theorem 1 in a way that improved the bound.

#### REFERENCES

- [1] Y. Linde, A. Buzo, and R. M. Gray, "An algorithm for vector quantizer design," *IEEE Trans. Commun.*, vol. COM-28, pp. 84–95, Jan. 1980.
- [2] D. Pollard, "Strong consistency of  $k$ -means clustering," *Ann. Statist.*, vol. 9, no. 1, pp. 135–140, 1981.
- [3] —, "Quantization and the method of  $k$ -means," *IEEE Trans. Inform. Theory*, vol. IT-28, pp. 199–205, Mar. 1982.
- [4] T. Linder, G. Lugosi, and K. Zeger, "Rates of convergence in the source coding theorem, in empirical quantizer design, and in universal lossy source coding," *IEEE Trans. Inform. Theory*, vol. 40, pp. 1728–1740, Nov. 1994.
- [5] P. Bartlett, T. Linder, and G. Lugosi, "The minimax distortion redundancy in empirical quantizer design," *IEEE Trans. Inform. Theory*, vol. 44, pp. 1802–1813, Sept. 1998.
- [6] P. A. Chou, "The distortion of vector quantizers trained on  $n$  vectors decreases to the optimum as  $O_p(1/n)$ ," in *Proc. IEEE Int. Symp. Information Theory*, Trondheim, Norway, June 27–July 1 1994, p. 457.
- [7] A. J. Zeevi, "On the performance of vector quantizers empirically designed from dependent sources," in *Proc. Data Compression Conf. DCC'98*, J. Storer and M. Cohn, Eds. Los Alamitos, CA: IEEE Computer Soc. Press, 1998, pp. 73–82.
- [8] P. C. Cosman, K. O. Perlmutter, S. M. Perlmutter, R. A. Olshen, and R. M. Gray, "Training sequence size and vector quantizer performance," in *Proc. Asilomar Conf. Signals, Systems, and Computers*, 1991, pp. 434–438.
- [9] D. Cohn, E. Riskin, and R. Ladner, "Theory and practice of vector quantizers trained on small training sets," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 16, pp. 54–65, Jan. 1994.
- [10] E. A. Abaya and G. L. Wise, "Convergence of vector quantizers with applications to optimal quantization," *SIAM J. Appl. Math.*, vol. 44, pp. 183–189, 1984.
- [11] D. S. Kim and M. R. Bell, "Bounds on the trained vector quantizer distortion measured using training data," Purdue Univ., Tech. Rep. TR-ECE 98-6, Apr. 1998.
- [12] D. Pollard, "A central limit theorem for  $k$ -means clustering," *Ann. Probab.*, vol. 10, no. 4, pp. 919–926, 1982.
- [13] N. Merhav and J. Ziv, "On the amount of side information required for lossy data compression," *IEEE Trans. Inform. Theory*, vol. 43, pp. 1112–1121, July 1997.
- [14] R. Dudley, "Central limit theorems for empirical measures," *Ann. Probab.*, vol. 6, pp. 899–929, 1978.
- [15] D. Pollard, "Empirical processes: Theory and applications," in *NSF-CBMS Regional Conf. Ser. Probability and Statistics*. Hayward, CA: Inst. Math. Statist., 1990.
- [16] N. Cesa-Bianchi and G. Lugosi, "Minimax regret under log loss for general classes of experts," in *Proc. 10th Annu. Conf. Computational Learning Theory*, 1999, pp. 12–18.
- [17] L. Devroye and L. Györfi, *Nonparametric Density Estimation: The  $L_1$  View*. New York: Wiley, 1985.
- [18] L. Devroye, L. Györfi, and G. Lugosi, *A Probabilistic Theory of Pattern Recognition*. New York: Springer-Verlag, 1996.

- [19] W. Hoeffding, "Probability inequalities for sums of bounded random variables," *J. Amer. Statist. Assoc.*, vol. 58, pp. 13–30, 1963.

## Transform Coding with Backward Adaptive Updates

Vivek K Goyal, *Member, IEEE*, Jun Zhuang, and  
Martin Vetterli, *Fellow, IEEE*

**Abstract**—The Karhunen–Loève transform (KLT) is optimal for transform coding of a Gaussian source. This is established for all scale-invariant quantizers, generalizing previous results. A backward adaptive technique for combating the data dependence of the KLT is proposed and analyzed. When the adapted transform converges to a KLT, the scheme is universal among transform coders. A variety of convergence results are proven.

**Index Terms**—Dithered quantization, lossy data compression, transform coding, universal source coding.

### I. INTRODUCTION

The essence of transform coding is to apply a linear transform to a source vector and then apply scalar quantization, as opposed to applying scalar quantization directly to the source vector. Heuristically, transform coding works because the transform can eliminate correlation between components of the source vector, producing a vector of transform coefficients more amenable to scalar quantization and entropy coding. Transform codes are popular because they provide an attractive compromise between computational complexity and performance. In the parlance of vector quantization, the point-density and oblongity losses of scalar quantization are eliminated or reduced, leaving predominantly only a space-filling loss [1].

With a Gaussian source model, the optimal transform is a Karhunen–Loève (KLT), an orthonormal transform that produces uncorrelated transform coefficients. The optimality of the KLT is well known for high rates [2] or when optimal fixed-rate quantizers are employed [3], but holds more generally (see Appendix I). However, the KLT is rarely used in practice for a variety of reasons. One prominent reason is that the KLT is signal-dependent; the transform used in the encoder and decoder must be adjusted to correspond to the covariance of the source in order to maintain optimality. A second reason is that since the KLT has no special structure, it requires more operations to compute than a harmonic transform such as a discrete cosine transform. For vectors

of length of  $N$ , the complexity difference is roughly  $N^2$  compared to  $N \log N$ , which is not overwhelming for small values of  $N$ .

This correspondence addresses only the first issue—the matching of transform to source. A *backward adaptive* method for transform adaptation is proposed and analyzed. In backward adaptation the encoder and decoder adapt in unison based on the coded data without the explicit transmission of coder parameters. Backward adaptation is also called *adaptation without side information* or *on-line adaptation*.

The use of backward adaptation for transform adaptation in transform coding seems to be unprecedented, though backward adaptive techniques have a long history. For example, adaptation of prediction filters in speech coders is often backward adaptive [4], [5] and ADPCM includes not only backward adaptation of filter taps but also of quantizer scaling [6]. Similar to the quantizer scaling in ADPCM is the backward adaptive context modeling and quantizer scaling of the EQ image coder [7]. It is also possible to adapt a quantizer more generally without side information [8].

The incompletely realized aim of our work is to show that backward adaptation can result in a transform code that is *universal* for Gaussian sources. "Universal" is used here to mean that the performance approaches that of an ideal *transform code* designed with *a priori* knowledge of the source distribution. The results along these lines are asymptotic in the data length, but the transform or block size is fixed. Empirical evidence and partial analyses are provided. Such a code would be an "on-line" alternative to the "universal codebook" approach to universal transform coding by Effros and Chou [9].<sup>1</sup> Forward adaptive techniques that are not necessarily universal are discussed, e.g., in [11].

The results of [9] were inspiring to this study because they indicated superior performance of weighted universal transform coding over weighted universal vector quantization for image compression with reasonable vector dimensions. It was also shown that there are sizable gains to be realized by varying the transform, a result that runs counter to the conventional wisdom in image compression.

In the remainder of the correspondence, the aforementioned ideas are made more precise. The sources and coding structures under consideration are described in Section II. Unable to satisfactorily analyze the original coding structure, we give several analyses based on simplifying assumptions. The main results are stated in Section III and proven in Appendix II. Section IV describes ways in which the encoding algorithms can be modified to reduce computational complexity or to track a varying source. Concluding comments appear in Section V.

### II. PROPOSED BACKWARD ADAPTIVE CODING STRUCTURE

Let  $\{x_n\}_{n \in \mathbb{Z}^+}$  be a sequence of independent and identically distributed (i.i.d.), zero-mean Gaussian random vectors of dimension  $N$  with covariance matrix  $R_x = E[xx^T]$ .<sup>2</sup> If  $R_x$  is not diagonal, i.e., the components of  $x$  are correlated, one obtains better rate-distortion performance with transform coding than with direct scalar quantization and scalar entropy coding of the source vectors.

In transform coding, a square, invertible linear transform  $T$  is applied to each source vector to get a vector of *transform coefficients*  $y_n = Tx_n$ . The transform coefficients undergo scalar quantization

<sup>1</sup>See the taxonomy of universal coding methods by Zhang and Wei [10] for explanations of the quoted terms.

<sup>2</sup>Throughout the correspondence,  $R_v$  will be used to denote the (exact) covariance matrix  $E[vv^T]$  of a random vector  $v$ .  $\widehat{R}_v$  denotes an estimate of  $R_v$  obtained from a finite-length observation. Aside from this convention, subscripts indicate the time index of a variable, except where two subscripts are given to indicate the row and column indices of a matrix. A superscript  $T$  indicates a transpose.

Manuscript received April 20, 1998; revised December 1, 1999. This work was initiated while the first and second authors were with the University of California, Berkeley. The material in this correspondence was presented in part at the IEEE International Conference on Image Processing, Lausanne, Switzerland, September 16–19, 1996 and at the IEEE Data Compression Conference, Snowbird, UT, March 25–27, 1997.

V. K. Goyal is with the Mathematics of Communications Research Department, Bell Labs, Lucent Technologies, Murray Hill, NJ 07974 USA (e-mail: v.goyal@ieee.org).

J. Zhuang is with SBC Technology Resources, Inc., Pleasanton, CA 94588 USA (e-mail: jxzhua1@tri.sbc.com).

M. Vetterli is with the Laboratoire de Communications Audiovisuelles, École Polytechnique Fédérale de Lausanne, CH-1015 Lausanne, Switzerland, and the Department of Electrical Engineering and Computer Science, University of California, Berkeley, Berkeley, CA USA (e-mail: Martin.Vetterli@epfl.ch).

Communicated by R. Laroia, Associate Editor for Source Coding.

Publisher Item Identifier S 0018-9448(00)04641-1.

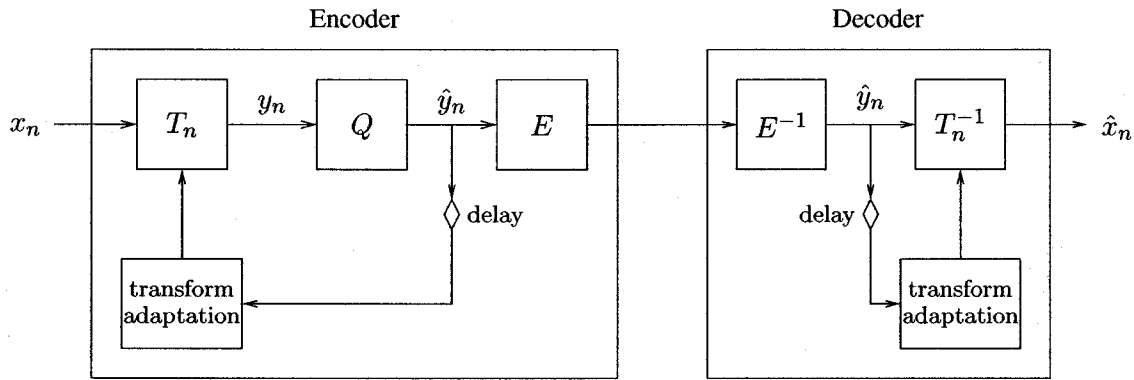


Fig. 1. Block diagram of transform coding system with backward adaptive transform updates.  $T_n$  is a time-varying orthogonal transform,  $Q$  is a scalar quantizer, and  $E$  is a universal scalar entropy coder.

and scalar entropy coding. Ideally, the transform should be selected such that the transform coefficients are uncorrelated and hence, since the source is Gaussian, independent. This was first shown by Huang and Schultheiss [3] under assumptions of optimal fixed-rate quantization and a mild, common sense condition on the bit allocation. (Earlier work by Kramer and Mathews [12] did not involve quantization and was not in an operational rate-distortion framework.) Using high-resolution quantization theory, the same result can be obtained for optimal variable-rate (entropy-coded) quantization or uniform quantization [2]. A new extension is given in Appendix I that relies only on the scalar quantizers having performance invariant to scaling (Theorem 6).

To mathematically describe an optimal transform  $T$ , simply note that by linearity of the expectation operator

$$R_y = E[(Tx)(Tx)^T] = TR_xT^T.$$

Thus  $T$  may be an orthonormal similarity transform composed of eigenvectors of  $R_x$ . This makes  $R_y$  a matrix with the eigenvalues of  $R_x$  on its main diagonal and zeros elsewhere. Such a transform is a Karhunen-Loève transform (KLT) of the source.

Since the optimal transform  $T$  depends on an ensemble average  $R_x$ , it is generally unknown at the encoder. (It may also be the case that  $E[x_n x_n^T]$  varies slowly with  $n$ , though we will deal with this case only in passing.) We consider here systems that periodically adjust the transform at the encoder and decoder in a backward adaptive manner. A block diagram for such a system is shown in Fig. 1. In this system, the quantizer  $Q$  is a scalar quantizer with uniform quantizer  $q$  applied to each component

$$q(x_i) = k_i \Delta, \quad \text{for } \left(k_i - \frac{1}{2}\right) \Delta \leq x_i < \left(k_i + \frac{1}{2}\right) \Delta, \\ k_i \in \mathbb{Z}, \quad i = 1, 2, \dots, N. \quad (1)$$

The entropy coder has  $N$  separate universal lossless codes for the  $N$  transform coefficient streams.

In this work we concentrate on the update mechanism for the transform and the effect of the transform updates. This is partly a matter of taste, but it is also motivated by the insensitivity of the optimal quantizer to the source and transform. The use of uniform scalar quantization with equal step sizes for each component is discussed in Section II-A and transform update procedures are considered in Section II-B.

#### A. Focusing on the Transform

Consider the quantization and entropy coding of a single transform coefficient branch in Fig. 1. Since the quantizer indices are entropy-coded, the proper optimization criterion for the quantizer is to minimize

the distortion for a given entropy coder output rate. Assuming that the transform and the universal lossless codes converge, this rate is well-approximated by the entropy rate of the quantizer output sequence. With this approximation one is left with an *entropy-constrained scalar quantizer* to design.

Even assuming that the variance of the transform coefficient is known, the best quantizer will generally be known only through a numerical optimization procedure. However, a uniform quantizer is optimal asymptotically for high rates [13] and, more importantly, is close to optimal at moderate rates [14]. This is an important distinction between fixed-rate and variable-rate scalar quantization that partially justifies our use of fixed uniform quantizers. (Alternatively, it was shown in [8] that backward adaptation of fixed-rate quantizers can be successful, but this is not pursued here.)

Now consider the joint optimization of the set of scalar quantizers. Using high-resolution analysis, it is easy to show that the optimal allocation of rates between the transform coefficients results in equal distortions and equal quantization step sizes for each transform coefficient [2]. Though this result is well known, the minimum rate at which this is a good approximation is not; thus we present some numerical calculations. At high rates, the operational distortion-rate performance of entropy-coded uniform quantization (ECUQ) of a Gaussian source with variance  $\sigma^2$  is given approximately by

$$D = \frac{\pi e}{6} \sigma^2 2^{-2R}. \quad (2)$$

This is easily obtained by combining the  $D \approx \Delta^2/12$  distortion of fine, uniform quantization with Rényi's relation between the differential entropy of a continuous source and its uniformly quantized version [15]

$$H(q(X)) \approx h(X) - \log_2 \Delta. \quad (3)$$

The inaccuracy of (2) at low rates is apparent from the fact that the maximum distortion should be  $\sigma^2$ ; the distortion given by (2) exceeds  $\sigma^2$  for rates below  $\approx 0.255$  bits. The actual distortion-rate behavior is compared to (2) in Fig. 2(a).

The simplicity of bit allocation using (2) is due to the form of  $\partial D/\partial R$ . Consider the allocation of  $R_1$  and  $R_2$  bits between components with variances  $\sigma_1^2$  and  $\sigma_2^2$ , respectively. Since

$$\frac{\partial D_i}{\partial R_i} = -\frac{\pi e \log 2}{3} \sigma_i^2 2^{-2R_i}, \quad i = 1, 2 \quad (4)$$

operating at equal slopes demands  $\sigma_1^2 2^{-2R_1} = \sigma_2^2 2^{-2R_2}$ . This in turn makes the component distortions equal and, again using high-resolution approximations, the quantization step sizes equal. This analysis

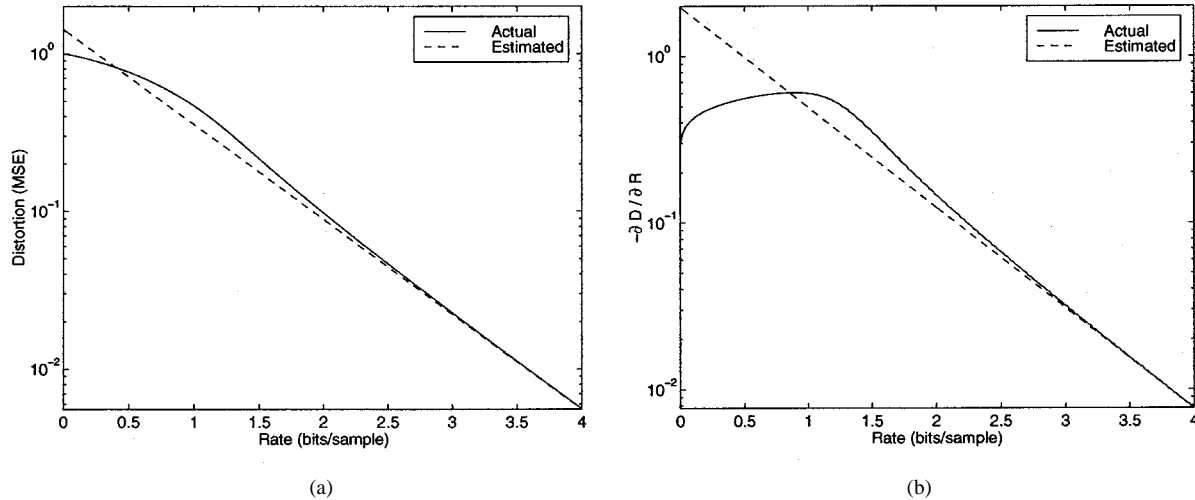


Fig. 2. Comparisons between the actual performance of entropy coded uniform scalar quantization and high-resolution approximations. (a) Actual distortion-rate performance compared to (2). (b) Derivative of the actual distortion-rate performance compared to (4).

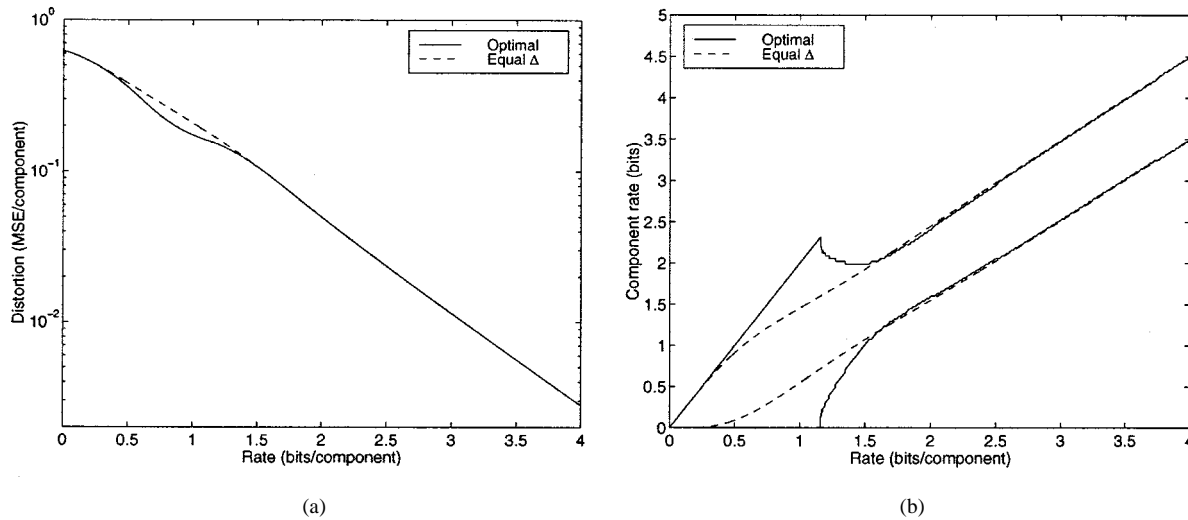


Fig. 3. Comparisons between the performance of optimal bit allocation and equal quantization step sizes for variables with variances  $\sigma_1^2 = 1$  and  $\sigma_2^2 = 1/4$ : (a) distortion-rate performances; and (b) bit allocations.

demonstrates that using equal quantization step sizes is a good approximation to optimal bit allocation when (4) is accurate. This is true for rates above about 1 bit per sample (see Fig. 2(b)).

To conclude the discussion of bit allocation, let us look at the effect of optimal bit allocation in one simple example. Variables with variances  $\sigma_1^2 = 1$  and  $\sigma_2^2 = 1/4$  are quantized by ECUQ either with optimal bit allocation or with equal quantization step sizes. Fig. 3(a) compares the distortion-rate performances and Fig. 3(b) compares the bit allocation. It is apparent that optimal bit allocation provides little improvement. Note also that the optimal bit allocation is predicted well by the high-resolution analysis when the lower rate is at least 1 bit per sample.

For the remainder of the paper, ECUQ with equal quantization step sizes for all components is employed exclusively. With this restriction, we may fix the quantization step size  $\Delta$  and focus on the entropies of the quantizer outputs; for small  $\Delta$  the distortion is insensitive to the choice of transform. In the limit as  $\Delta$  approaches zero, this insensitivity is clear because the distortion approaches  $\Delta^2/12$  per component. It turns out that the deviation from this approximation is less than 5% for rates above 1 bit per sample. This is demonstrated in two dimensions by Fig. 4. Sources with covariance matrices

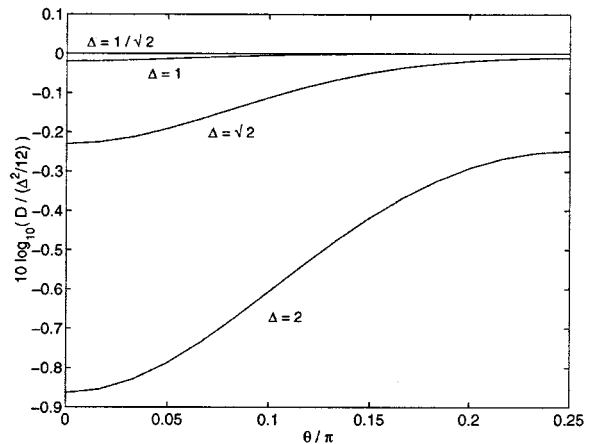


Fig. 4. Dependence of overall distortion on the choice of transform for a two-dimensional source. The dependence is mild and vanishes as the quantization step size  $\Delta$  shrinks.

$R_x = J(\theta)^T \text{diag}(1, 1/4)J(\theta)$ , where  $J(\theta)$  is a Jacobi rotation of  $\theta$  radians,<sup>3</sup> were quantized with various quantization step sizes. The distortion, normalized by  $\Delta^2/12$ , is shown on a logarithmic scale as a function of  $\theta$ . In this example, the distortion differs little from, and is bounded above, by  $\Delta^2/12$ .

### B. Transform Update Mechanisms

Referring again to Fig. 1, for decoder tracking without side information it is necessary that the transform  $T_{n+1}$  depend only on  $\{T_k\}_{k=1}^n$  and  $\{\hat{y}_k\}_{k=1}^n$ . We assume that the covariance estimate

$$\widehat{R}_{\hat{x}}^{(n)} = \frac{1}{n} \sum_{k=1}^n \hat{x}_k \hat{x}_k^T \quad (5)$$

is computed and that  $T_{n+1}$  is chosen such that  $T_{n+1} \widehat{R}_{\hat{x}}^{(n)} T_{n+1}^T$  is diagonal with nonincreasing diagonal elements. This amounts to using  $\widehat{R}_{\hat{x}}^{(n)}$  as an estimate for  $R_x$ . The calculation of  $T_{n+1}$  will have sign ambiguities<sup>4</sup> and if the eigenvalues of  $\widehat{R}_{\hat{x}}^{(n)}$  are not distinct, there will be additional ambiguities; these can be resolved arbitrarily. The initial transform  $T_1$  can also be arbitrary.

More complicated update mechanisms are possible, but using an eigendecomposition of (5) has the attractive property of requiring only constant storage: As the data vectors are coded, only the  $N(N+1)/2$  independent components of (5) must be stored. Adjustments to (5) to compensate for quantization effects are possible, but are not used so as to not rely too heavily on the Gaussian model for the source data.

At first glance it may seem that we expect  $\widehat{R}_{\hat{x}}^{(n)}$  to converge to  $R_x$ , which would result in the transform converging to the desired KLT. In fact, we do not need  $\widehat{R}_{\hat{x}}^{(n)} \rightarrow R_x$  to have the desired transform convergence. Suppose for the moment that the effect of quantization is to add a zero-mean signal  $z$  independent of  $x$  with  $E[zz^T] = (\Delta^2/12)I_N$ . Then  $R_{\hat{x}} = R_x + (\Delta^2/12)I_N$  and since  $R_{\hat{x}}$  and  $R_x$  have the same eigenvectors, the transform converges to the correct transform. Of course, this is an overly simplistic model of quantization. As detailed below, the difference between  $E[xx^T]$  and  $E[Q(x)Q(x)^T]$  is generally not a scaled identity. Nevertheless, we assert that the system works: The transform converges to the optimal transform, resulting in a universal system. We cannot prove this convergence precisely, but results suggesting the observed convergence are given in the following section.

## III. MAIN RESULTS

The main results of the correspondence are summarized in this section. Proofs are given in Appendix II.

### A. Transform Convergence Implies Universality

*Theorem 1:* Fix a quantization step size  $\Delta$  and suppose  $\{T_n\}$  converges elementwise to  $T$ , a KLT of the source. Let  $L_n$  denote the per-component code length for coding the first  $n$  vectors using the adaptive scheme and let  $L_n^*$  denote the per-component code length for coding the first  $n$  vectors with the fixed, optimal transform  $T$ . Then the average excess rate  $n^{-1}(L_n - L_n^*)$  converges in mean square to zero.

As discussed in Section II-A, given a quantization step size, the distortion of a transform coder depends only slightly on the transform. Thus Theorem 1 indicates that the backward adaptive scheme will have performance asymptotically almost equal to an optimal transform coder whenever the transform converges to a KLT. Transform convergence can be established when using an independence assumption similar to

<sup>3</sup>Jacobi rotations are defined in equation (12) of Appendix I

<sup>4</sup>If  $T_{n+1} \widehat{R}_{\hat{x}}^{(n)} T_{n+1}^T$  is diagonal, then negating any row of  $T_{n+1}$  will not change the product  $T_{n+1} \widehat{R}_{\hat{x}}^{(n)} T_{n+1}^T$ .

that used in heuristic analyses of the LMS algorithm. In such an analysis the sequence of transforms is assumed to be independent, though this assumption is clearly false [16, Appendix 3.B].

The following two sections give different types of convergence results that are suggestive of the convergence seen in simulations. In Section III-B the stochastic variation of (5) is ignored. The transform updates are then described by a deterministic iteration. As an alternative, the quantizer can be replaced by a subtractive dithered quantizer in order to insure nice behavior of the transform sequence. This is considered in Section III-C.

### B. Deterministic Analysis

In the original system, the distribution of  $\hat{x}_n$  depends on  $T_n$ , which in turn depends on  $T_1$  and  $\{x_k\}_{k=1}^{n-1}$ . Because of this complicated interdependence between quantization and stochastic effects, it is very difficult to analyze the convergence of the transform.

One way to reduce the complexity of the analysis is to neglect the stochastic aspect, meaning to assume there is no variance in moment estimates despite the fact that moments are estimated from finite-length observations. The effect is to replace (5) with

$$R_{\hat{x}}^{(n)} = E \left[ \hat{x}_n \hat{x}_n^T \right] \quad (6)$$

and update the transform such that  $T_{n+1} R_{\hat{x}}^{(n)} T_{n+1}^T$  is diagonal with nonincreasing diagonal elements. We are left with a deterministic iteration summarized by

$$R_{\hat{x}}^{(n)} = T_n^T R_{\hat{y}}^{(n)} T_n = T_n^T \tilde{Q}(R_y^{(n)}) T_n = T_n^T \tilde{Q}(T_n R_x T_n^T) T_n$$

$$T_{n+1} R_{\hat{x}}^{(n)} T_{n+1}^T = \Lambda_n (\text{diagonal with nonincreasing diagonal elements})$$

where  $\tilde{Q}: \mathbb{R}^{N \times N} \rightarrow \mathbb{R}^{N \times N}$  gives the effect of quantization on the covariance matrix.  $\tilde{Q}$  depends on the source distribution and  $\Delta$  and can be described by evaluating expressions from [17].

Since  $R_x$  and  $R_{\hat{x}}^{(n)}$  generally have different eigenvectors, it is not obvious that this iteration will converge. The following theorem gives a limited convergence result.

*Theorem 2:* Let  $R_x$  and  $T_1$  be given. Then there exists a sequence of quantization step sizes  $\{\Delta_n\} \subset \mathbb{R}^+$  such that the deterministic iteration described above converges to a KLT of the source. Since the KLT is ambiguous if the eigenvalues of  $R_x$  are not distinct, convergence is indicated by  $R_{\hat{y}}^{(n)}$  approaching a diagonal matrix in Frobenius norm.

Theorem 2 does not preclude the possibility that the iteration will converge only with  $\inf \Delta_n = 0$ . However, numerical calculations suggest that the iteration actually converges for constant sequences of sufficiently small step sizes. Fig. 5 shows numerical results for a four-dimensional Gaussian source with  $(R_x)_{ij} = 0.9^{|i-j|}$ ,  $T_1 = I$ , and various values of  $\Delta$ . To show the degree to which  $T_n$  diagonalizes  $R_x$ ,  $\|R_y^{(n)}\|$  is plotted as a function of the iteration number  $n$ , where  $\|A\| = \sum_{i \neq j} a_{ij}^2$ . An approximate correspondence between quantization step size and rate is also given.

Starting from an arbitrary initial transform,  $\|R_y^{(n)}\|$  becomes small after a single iteration (note the logarithmic vertical axis). Then, to the limits of machine precision, it converges exponentially to zero with a rate of convergence that depends on  $\Delta$ . (For  $\Delta > 3$ , loss of significance problems in the computation combined with very slow convergence make it difficult to ascertain convergence numerically.)

The results shown in Fig. 5 are representative of the performance with an arbitrary  $R_x$ . The convergence, as measured by  $\|R_y^{(n)}\|$ , is unaffected by the multiplicities of the eigenvalues of  $R_x$ . The eigenspace associated with a multiple eigenvalue can be rotated arbitrarily without affecting  $\|R_y^{(n)}\|$  or the decorrelation and energy compaction properties of the transform.

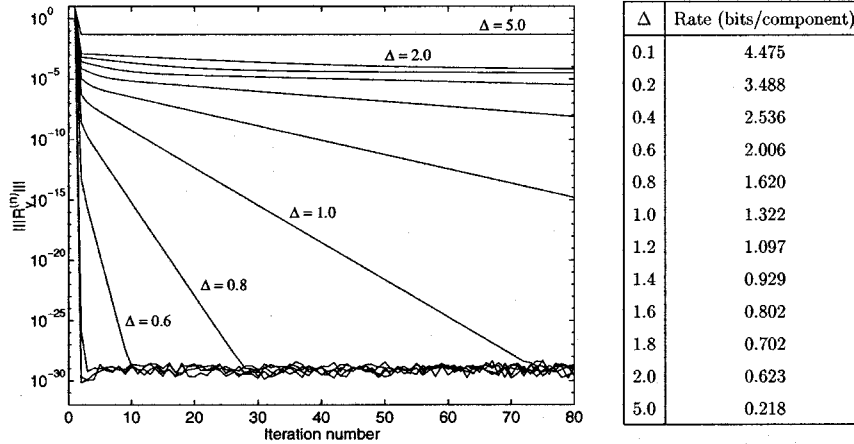


Fig. 5. Simulations for various fixed quantization step sizes suggest that the deterministic iteration converges more generally than predicted by Theorem 2. The source vector length is  $N = 4$  and the initial transform is the identity transform. The accompanying table provides the approximate correspondence between quantization step sizes and rates.

*Theorem 3:* Let  $N = 2$  and let  $R_x$  be given. There exists  $\Delta_{\max} > 0$  such that for any  $\Delta < \Delta_{\max}$  the deterministic iteration converges, in the same sense as before, for any initial transform  $T_1$ .

### C. Using Dithered Quantization

For the sake of analysis, let us alter the system to use subtractive dithered quantization [19]. Replace the quantizer  $q(\cdot)$  defined in (1) by

$$q_{\text{dither}}((y_n)_i) = q((y_n)_i + z_{ni}) - z_{ni} \quad (7)$$

where the  $z_{ni}$ 's are independent and each is uniformly distributed on  $[-\Delta/2, \Delta/2]$ . We assume that the dither signal  $\{z_{ni}\}_{n \in \mathbb{Z}^+, 1 \leq i \leq N}$  is somehow available at the decoder so that each component of the quantizer input can be reconstructed up to an error of magnitude  $\Delta/2$ . The dither signal is not used in the entropy coder.

The effect of the dither is to make the quantization error independent of the data and transform sequences. The following result is then straightforward.

*Theorem 4:* With the dithered quantizer (7) and any initial transform  $T_1$

$$\widehat{R}_x^{(n)} \text{ converges in mean square to } R_x + \frac{\Delta^2}{12} I \text{ as } n \rightarrow \infty.$$

Also, the sequence of transforms  $\{T_n\}$  converges in mean square to a KLT for the source.

Although we are assuming Gaussian signals throughout, the proof of the theorem does not depend on the distribution of the source. The transform converges to a transform that maximizes coding gain for any i.i.d. source; however, for non-Gaussian sources maximizing coding gain may not be ideal.

When the source is Gaussian, the KLT is the optimal transform and the entropies of the quantized variables can be easily estimated. This leads to the following theorem.

*Theorem 5:* Denote the eigenvalues of  $R_x$  by  $\lambda_1, \lambda_2, \dots, \lambda_N$ . Define  $L_n$  and  $L_n^*$  as in Theorem 1. Then the average excess rate  $n^{-1}(L_n - L_n^*)$  converges in mean square to a constant  $\rho$ . Estimating discrete entropies with (3)

$$\rho < \frac{1}{2N} \sum_{i=1}^N \log_2 \left( 1 + \frac{\Delta^2}{12\lambda_i} \right). \quad (8)$$

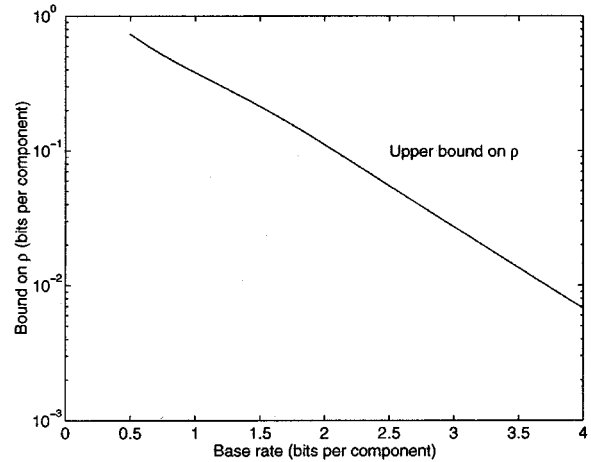


Fig. 6. Bound (8) on the excess rate  $\rho$  as a function of the coding rate for a Gaussian source with  $(R_x)_{ij} = 0.8^{|i-j|}$ .

The constant  $\rho$  can be interpreted as the asymptotic redundancy of the system. It is the excess rate, in bits per source component, of the adaptive system, as compared to a fixed, optimal transform code designed with knowledge of  $R_x$ . The bound (8) comes simply from the variance added by the dither signal.<sup>5</sup> As  $\Delta$  approaches zero, the power of the dither signal vanishes and accordingly  $\rho$  approaches 0. Thus the dithered system is universal for high-rate coding.

At moderate rates,  $\rho$  is quite small. For example, consider the coding of an eight-dimensional Gaussian source with  $(R_x)_{ij} = 0.8^{|i-j|}$ . By computing the bound (8) and the correspondence between  $\Delta$  and the rate of a KLT coder for this particular source we get the curve shown in Fig. 6. This roughly indicates that  $\rho$  must decay exponentially with the overall coding rate. In fact, using the high-rate approximation

$$\frac{\Delta^2}{12} \approx \frac{\pi e}{6} \left( \prod_{i=1}^N \lambda_i \right)^{1/N} 2^{-2R}$$

<sup>5</sup>When the dither signal is known at the entropy coder, performance better than the worst case given by (8) can be expected [20].

where  $R$  is the rate of the optimal transform coder, (8) can be written as

$$\begin{aligned} \rho &< \frac{1}{2N} \sum_{i=1}^N \log_2 \left( 1 + \frac{\Delta^2}{12\lambda_i} \right) < \frac{1}{2N \ln 2} \sum_{i=1}^N \frac{\Delta^2}{12\lambda_i} \\ &\approx \frac{\pi e}{12 \ln 2} \left( \prod_{i=1}^N \lambda_i \right)^{1/N} \left( \frac{1}{N} \sum_{i=1}^N \frac{1}{\lambda_i} \right) 2^{-2R}. \end{aligned}$$

At rates of 2 or 3 bits per component, the excess rate is less than 6% or 1%, respectively.

#### D. Remark

The deterministic analyses and the analysis of the system with dither can be combined to form a heuristic argument for convergence. Soon after the system is initialized, the variance of (5) is high and thus the variation of the transform is also high; this has the effect of a dither. Later the changes to the transform are much smaller, but the transform cannot settle at an incorrect value because incorrect transforms are not fixed points of the deterministic iteration.

### IV. VARIATIONS ON THE BASIC ALGORITHMS

Certain modifications to the basic algorithms can be made to reduce the computational complexity or to facilitate the coding of non-i.i.d. sources. All of the modifications mentioned in this section apply equally well to the dithered and undithered systems.

The most complicated step in these algorithms is the computation of the updated transform; thus the complexity can be reduced by suppressing this computation. Instead of computing an eigendecomposition of  $\widehat{R}_{\hat{x}}^{(n)}$  at each step, one can compute the eigendecomposition every  $L$  steps, holding the transform constant in between.  $L$  need not be constant, but if it is to vary it must be computable from coded data. Having constant  $L > 1$  does not affect the conclusions in Theorem 4.

The coding of a non-i.i.d. source poses many problems. First of all, we must assume that  $R_{\hat{x}}^{(n)}$  varies slowly, or that the source is ‘‘locally stationary.’’ If this is not the case, an on-line algorithm will fail because the coding of  $x_n$  is based on an estimate of  $R_{\hat{x}}^{(n)}$  from (recent) past samples.<sup>6</sup> Secondly, the covariance matrix estimate  $\widehat{R}_{\hat{x}}^{(n)}$  should be local, e.g.,

$$\widehat{R}_{\hat{x}}^{(n)} = \frac{1}{K} \sum_{k=n-K+1}^n \hat{x}_k \hat{x}_k^T \quad (9)$$

or

$$\widehat{R}_{\hat{x}}^{(n)} = \omega \widehat{R}_{\hat{x}}^{(n-1)} + (1 - \omega) \hat{x}_n \hat{x}_n^T \quad (10)$$

with appropriate initialization. If the update interval  $L$  divides  $K$  in (9), it is not necessary to store a full window of  $K$  past samples [22].

A technique which simultaneously reduces the computational complexity and introduces a covariance estimate equivalent to (10) is to replace the eigendecomposition computation with an *incremental* change in the transform based on  $\hat{x}_n$ . This is explored in [16, Ch. 4], [23].

### V. CONCLUSIONS

This correspondence has proposed a backward adaptive structure for transform adaptation in transform coding. Since there is no side information, the system is universal for Gaussian sources when the transform converges to a Karhunen-Loève transform. Simulations indicate

<sup>6</sup>Without local stationarity, a forward adaptive method would presumably be superior; see [9], [21].

convergence, and convergence can be shown under certain simplifying assumptions such as when the estimation noise is ignored or when the quantization is dithered. The problem of optimally combining forward and backward adaptive methods remains open.

Gaussian sources were assumed throughout. This assumption was used in two ways: to justify maximizing coding gain and to concretely describe the effect of quantization on moment estimation. The availability of universal lossless coders is assumed, but, in contrast to [24], they are applied only to sequences of scalars. This potentially decreases the memory requirement and speeds convergence.

### APPENDIX I

#### OPTIMALITY OF THE KARHUNEN-LOÈVE TRANSFORM

This appendix provides a new result, with two proofs, on the optimality of the Karhunen-Loève transform (KLT) for transform coding of Gaussian sources. It is more general than earlier results relying on optimal fixed-rate quantization [3] or high-resolution quantization theory [2], [3] because it relies only on a scale-invariance property of quantizer distortion-rate performance; in particular, it encompasses the earlier results and applies to entropy-coded uniform scalar quantization with equal step sizes for each component, as utilized in this correspondence.

*Theorem 6 (Optimality of Karhunen-Loève Transform):* Consider the transform coding of a Gaussian source subject to minimum square error (MSE) distortion. Assume that the distortion-rate performance of a scalar quantizer applied to a component with variance  $\sigma^2$  is  $D = \sigma^2 f(R)$ . Then a KLT is an optimal transform, i.e., for any given maximum per-component rate, it minimizes the distortion.

First note that if  $f(\cdot)$  is not nonincreasing, there will be rates that are useless: if  $R_1 > R_2$  but  $f(R_1) > f(R_2)$ , rate  $R_1$  can be replaced in any purportedly optimal solution by rate  $R_2$  without increasing the distortion and without violating the rate constraint. Thus we henceforth assume that  $f(\cdot)$  is nonincreasing.

Two proofs are given: The first is simple to describe from first principles but relies on an iterative construction. The second, more elegant proof relying on the theory of majorization (see [25]) is due to Telatar [26].

*Proof 1:* Let  $(R_1, R_2, \dots, R_N)$  be any bit allocation vector, i.e., suppose that  $R_i$  bits are allocated to transform coefficient  $y_i$ . Given any orthogonal transform  $T$ , we will show that there exists a KLT  $\hat{T}$  that yields distortion at most as high as yielded by  $T$ .

Before proceeding with more complicated constructions, note that the variances of the transform coefficients should have the same ordering as the rates. If  $\sigma_{y_i}^2 > \sigma_{y_j}^2$  but  $R_i < R_j$ , then the distortion is reduced or unchanged by swapping the  $i$ th and  $j$ th rows of  $T$ . The resulting change in distortion is

$$\begin{aligned} &(\sigma_{y_i}^2 f(R_j) + \sigma_{y_j}^2 f(R_i)) - (\sigma_{y_i}^2 f(R_i) + \sigma_{y_j}^2 f(R_j)) \\ &= \underbrace{(\sigma_{y_i}^2 - \sigma_{y_j}^2)}_{>0} \underbrace{(f(R_j) - f(R_i))}_{\leq 0} \leq 0. \end{aligned}$$

In the remainder of the proof we assume that  $T$  has the property

$$\text{for any } i \text{ and } j, \quad \sigma_{y_i}^2 > \sigma_{y_j}^2 \text{ implies } R_i \geq R_j. \quad (11)$$

There is nothing to prove if  $T$  is a KLT, so we may assume that the  $(i, j)$  component of  $R_y = TR_x T^T$  is nonzero for some  $(i, j)$  pair.

Construct a new transform  $T_1 = J(i, j, \theta)^T T$ , where  $J(i, j, \theta)$  is a Jacobi rotation defined by

$$J(i, j, \theta) = \begin{bmatrix} 1 & \cdots & 0 & \cdots & 0 & \cdots & 0 \\ \vdots & \ddots & \vdots & & \vdots & & \vdots \\ 0 & \cdots & \cos \theta & \cdots & \sin \theta & \cdots & 0 \\ \vdots & & \vdots & \ddots & \vdots & & \vdots \\ 0 & \cdots & -\sin \theta & \cdots & \cos \theta & \cdots & 0 \\ \vdots & & \vdots & & \vdots & \ddots & \vdots \\ 0 & \cdots & 0 & \cdots & 0 & \cdots & 1 \end{bmatrix} \begin{matrix} i \\ j \end{matrix},$$

$$\theta \in [-\pi/4, \pi/4] \quad (12)$$

and  $\theta$  is chosen such that the  $(i, j)$  element of  $T_1 R_x T_1^T$  is zero.

This choice of transform has a few important properties. The first is that  $T_1 R_x T_1^T$  is closer to a diagonal matrix than  $T R_x T^T$ , where closeness is measured by the Euclidean norm of the off-diagonal elements. Thus repeatedly cycling through all  $(i, j)$  pairs, defining  $T_{k+1} = J^T T_k$ , eventually yields a diagonal matrix  $\tilde{T} R_x \tilde{T}^T$ , where  $\tilde{T} = \lim_{k \rightarrow \infty} T_k$ .

The second property is that among the diagonal elements, only the  $i$ th and  $j$ th are altered. These are altered such that the larger of the two is increased by a positive increment  $\delta$  and the smaller is decreased by the same amount. This is easily verified by expanding

$$\begin{bmatrix} \cos \theta & \sin \theta \\ -\sin \theta & \cos \theta \end{bmatrix}^T \begin{bmatrix} a_1 & a_3 \\ a_3 & a_2 \end{bmatrix} \begin{bmatrix} \cos \theta & \sin \theta \\ -\sin \theta & \cos \theta \end{bmatrix} = \begin{bmatrix} b_1 & 0 \\ 0 & b_2 \end{bmatrix}$$

and solving for  $\theta \in [-\pi/4, \pi/4]$ . If  $a_1 \geq a_2$ , one finds

$$\begin{aligned} b_1 &= a_1 + \delta \\ b_2 &= a_2 - \delta \end{aligned}$$

$$\text{where } \delta = \frac{-(a_1 - a_2) + \sqrt{(a_1 - a_2)^2 + 4a_3^2}}{2} \geq 0$$

with equality if and only if  $a_3 = 0$ .

Suppose  $\sigma_{y_i}^2 \geq \sigma_{y_j}^2$ . Then using the second property, the change in distortion by using  $J(i, j, \theta)^T T$  in place of  $T$  is

$$\begin{aligned} &((\sigma_{y_i}^2 + \delta)f(R_i) + (\sigma_{y_j}^2 - \delta)f(R_j)) - (\sigma_{y_i}^2 f(R_i) + \sigma_{y_j}^2 f(R_j)) \\ &= \underbrace{\delta}_{>0} \underbrace{(f(R_i) - f(R_j))}_{\leq 0} \leq 0. \end{aligned}$$

Thus as we iterate to find  $\tilde{T}$ , the distortion is decreased or unchanged at each step. The  $\tilde{T}$  thusly constructed is both a KLT and at least as good as  $T$  in distortion-rate performance.  $\square$

*Proof 2:* The second proof is based on elementary properties of majorization, which are detailed in [25]. A vector  $(\alpha_1, \alpha_2, \dots, \alpha_N)$  is said to be *majorized* by another vector  $(\beta_1, \beta_2, \dots, \beta_N)$  if

$$\sum_{i=1}^k \alpha_{[i]} \leq \sum_{i=1}^k \beta_{[i]}, \quad k = 1, 2, \dots, N-1$$

and

$$\sum_{i=1}^N \alpha_{[i]} = \sum_{i=1}^N \beta_{[i]}$$

where the  $[i]$  notation indicates a decreasing ordering  $\alpha_{[1]} \geq \alpha_{[2]} \geq \dots \geq \alpha_{[N]}$ .

<sup>7</sup>This is the well-known classical Jacobi algorithm for computing eigendecompositions of symmetric matrices; for details, including convergence results, see [27, Sec. 8.5].

Again let  $(R_1, R_2, \dots, R_N)$  be any bit allocation vector. The problem is to minimize the function

$$D = \sum_{i=1}^N \sigma_{y_i}^2 f(R_i)$$

by manipulating the  $\sigma_{y_i}^2$ 's through the choice of  $T$ . Let

$$\sigma = (\sigma_{y_1}^2, \sigma_{y_2}^2, \dots, \sigma_{y_N}^2) = \text{diag}(T R_x T^T).$$

For a Hermitian matrix, the diagonal elements are majorized by the eigenvalues, so  $\sigma$  is majorized by a vector  $\lambda$  of eigenvalues of  $R_x$ . Now the majorization of  $\sigma$  by  $\lambda$  is equivalent to  $\sigma$  being in the convex hull of the  $N!$  permutations of  $\lambda$ . We are thus left with minimizing  $D$  over the convex polytope defined by the permutations of  $\lambda$ .<sup>8</sup> In minimizing a linear function over a convex polytope, the optimum is always attained at a corner point. This establishes that the optimal transform is a KLT. Furthermore, the arguments given in Proof 1 indicate that the optimal KLT (the optimal permutation) is one that satisfies (11).  $\square$

## APPENDIX II PROOFS

### A. Proof of Theorem 1

Let  $N\ell_n$  denote the number of bits used to code  $x_n$ . Because of the convergence of the sequence of transforms and the universality of the entropy coder,  $E[\ell_n]$  converges to some limit, say  $\bar{\ell}$ . For the static coder, the number of bits used by the optimal coder  $N\ell_n^*$  will satisfy  $E[\ell_n^*] = \bar{\ell}^*$ . Since  $\{T_n\}$  converges to  $T$  and the entropy rate of the quantizer output depends only on the transform,  $\bar{\ell} = \bar{\ell}^*$ . Now since  $\{E[\ell_k - \ell_k^*]\}$  converges to zero, the mean of the sequence

$$n^{-1} \sum_{k=1}^n (\ell_k - \ell_k^*) = n^{-1} (L_n - L_n^*)$$

converges to zero in mean square.

### B. Proof of Theorem 2

The proofs of Theorems 2 and 3 rely on properties of  $\tilde{Q}$ , the function that describes the effects of quantization on the covariance matrix.

Let  $\eta_1$  and  $\eta_2$  be jointly Gaussian with

$$\begin{aligned} E[\eta_1] &= E[\eta_2] = 0 \\ E[\eta_1^2] &= \nu_1^2 & E[\eta_2^2] &= \nu_2^2 \end{aligned}$$

and

$$E[\eta_1 \eta_2] = \nu_{12}.$$

Define

$$\hat{\nu}_2^2 = E[q(\eta_1)^2] \quad \hat{\nu}_2^2 = E[q(\eta_2)^2]$$

and

$$\hat{\nu}_{12} = E[q(\eta_1)q(\eta_2)]$$

where  $q(\cdot)$  was defined by (1). Then using expressions from [17], one can show

$$\hat{\nu}_i^2 = \nu_i^2 + \frac{\Delta^2}{12} + \sum_{m=1}^{\infty} (-1)^m e^{-2m^2 \pi^2 \nu_i^2 / \Delta^2} \left( \frac{\Delta^2}{m^2 \pi^2} + 4\nu_i^2 \right), \quad i = 1, 2 \quad (13)$$

and

$$\hat{\nu}_{12} = (1 + \delta)\nu_{12} + \mu \quad (14)$$

<sup>8</sup>We have not carefully argued that all points in the polytope are feasible, but the achievability of the optimizing value will be clear.

where

$$\delta = 2 \left( \sum_{m_1=1}^{\infty} (-1)^{m_1} e^{-2m_1^2 \pi^2 \nu_1^2 / \Delta^2} + \sum_{m_2=1}^{\infty} (-1)^{m_2} e^{-2m_2^2 \pi^2 \nu_2^2 / \Delta^2} \right)$$

and

$$\mu = \sum_{m_1, m_2=1}^{\infty} (-1)^{m_1+m_2} \frac{\Delta^2}{m_1 m_2 \pi^2} \cdot \exp\left(\frac{-2\pi^2(m_1^2 \nu_1^2 + m_2^2 \nu_2^2)}{\Delta^2}\right) \sinh\left(\frac{4\pi^2 \nu_{12} m_1 m_2}{\Delta^2}\right). \quad (15)$$

For any covariance matrix  $R$ , the diagonal elements of  $\tilde{Q}(R)$  are described by (13) and the off-diagonal elements are described by (14). For the purpose of this theorem, we need only the following simple property of  $\tilde{Q}$ :

$$\tilde{Q}(R) = R + \frac{\Delta^2}{12} I + C, \text{ where } C \rightarrow 0 \text{ elementwise as } \Delta \rightarrow 0. \quad (16)$$

To measure the degree to which  $T_n$  diagonalizes  $R_x$ , define a distance measure  $\| \| \cdot \| \|$  between a matrix  $A$  and the set of diagonal matrices by  $\| \| A \| \| = \sum_{i \neq j} a_{ij}^2$ . The strategy of the proof is to show that for sufficiently small  $\Delta$ , the inequality  $\| \| R_y^{(n+1)} \| \| \leq \frac{1}{2} \| \| R_y^{(n)} \| \|$  holds for all  $n \geq 1$ .

Combining  $R_{\hat{x}}^{(n)} = T_n^T R_y^{(n)} T_n$  with  $R_{\hat{x}}^{(n)} = T_{n+1}^T \Lambda_n T_{n+1}$  gives  $T_{n+1}^T \Lambda_n T_{n+1} = T_n^T R_y^{(n)} T_n$ . Define  $H_n = T_n T_{n+1}^T$  so that

$$R_y^{(n)} = H_n \Lambda_n H_n^T. \quad (17)$$

Also notice that

$$R_y^{(n+1)} = T_{n+1} R_x T_{n+1}^T = T_{n+1} T_n^T T_n R_x T_n^T T_{n+1} = H_n^T R_y^{(n)} H_n.$$

As a final preparation, define  $Z_n = R_y^{(n)} - R_{\hat{y}}^{(n)}$ .

We can now make the calculation

$$\begin{aligned} \| \| R_y^{(n+1)} \| \| &= \| \| H_n^T R_y^{(n)} H_n \| \| = \| \| H_n^T (Z_n + R_{\hat{y}}^{(n)}) H_n \| \| \\ &= \| \| H_n^T Z_n H_n \| \| \end{aligned}$$

where the last equality follows from  $H_n^T R_{\hat{y}}^{(n)} H_n$  being diagonal (see (17)). From (16), it is clear that if  $\Delta$  is small enough,

$$\| \| Z_n \| \| \leq \frac{1}{4} \| \| R_y^{(n)} \| \|.$$

It remains now to relate  $\| \| Z_n \| \|$  and  $\| \| H_n^T Z_n H_n \| \|$ .

Substitute  $R_{\hat{y}}^{(n)} = R_y^{(n)} + \Delta^2 I/12 + C_1$ , where  $\| \| C_1 \| \| \rightarrow 0$  as  $\Delta \rightarrow 0$ , in (17) to get

$$R_y^{(n)} + \frac{\Delta^2}{12} I + C_1 = H_n \Lambda_n H_n^T. \quad (18)$$

Decrementing the index and rearranging gives

$$H_{n-1}^T R_y^{(n-1)} H_{n-1} + \frac{\Delta^2}{12} I + H_{n-1}^T C_1 H_{n-1} = \Lambda_{n-1}. \quad (19)$$

Since  $H_{n-1}^T R_y^{(n-1)} H_{n-1} = R_y^{(n)}$ , comparing (18) and (19) gives

$$H_n \Lambda_n H_n^T = \Lambda_{n-1} + C_1 - H_{n-1}^T C_1 H_{n-1}. \quad (20)$$

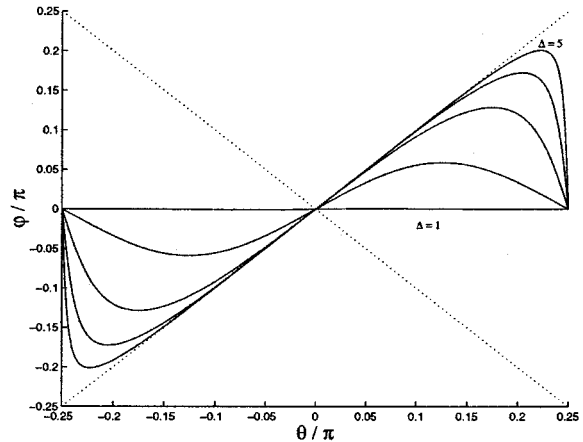


Fig. 7. Simulations of the deterministic iteration for  $N = 2$  suggest convergence for any quantization step size  $\Delta$ . The eigenvalues of  $R_x$  are 1 and  $1/4$ . The step sizes shown are  $\Delta = 1, 2, \dots, 5$ .  $\varphi$  is the iterate that follows after  $\theta$ . Global convergence is indicated by the curves lying inside the cone  $|\varphi| \leq |\theta|$ , which is marked by dotted lines.

Now let  $C_2 = H_n - I$ . Substituting in (20) and expanding, we conclude that  $\| \| C_2 \| \| \rightarrow 0$  as  $\Delta \rightarrow 0$ . Thus by expanding  $H_n^T Z_n H_n$  we see that

$$\| \| H_n^T Z_n H_n \| \| - \| \| Z_n \| \| \rightarrow 0$$

faster than  $\| \| Z_n \| \| \rightarrow 0$  as  $\Delta \rightarrow 0$ , so by choosing  $\Delta$  small enough we have the bound  $\| \| H_n^T Z_n H_n \| \| \leq 2 \| \| Z_n \| \|$ .

Combining all these calculations gives

$$\begin{aligned} \| \| R_y^{(n+1)} \| \| &= \| \| H_n^T Z_n H_n \| \| \leq 2 \| \| Z_n \| \| \leq 2 \cdot \frac{1}{4} \| \| R_y^{(n)} \| \| \\ &= \frac{1}{2} \| \| R_y^{(n)} \| \|. \end{aligned}$$

### C. Proof of Theorem 3

Without loss of generality (rotating the coordinate system and initial transform, if necessary), assume  $R_x = \text{diag}(\sigma_1^2, \sigma_2^2)$ ,  $\sigma_1 \geq \sigma_2$ . The transform iterates are all in  $SO_2(\mathbb{R})$  and can be parameterized as

$$T_\theta = \begin{bmatrix} \cos \theta & -\sin \theta \\ \sin \theta & \cos \theta \end{bmatrix}$$

where  $\theta \in [-\pi/4, \pi/4]$ . We assume  $\sigma_1 > \sigma_2$ ; if  $\sigma_1 = \sigma_2$  the situation is uninteresting because  $R_y$  is diagonal for any  $T_\theta$ .

Denote the transform iterate that follows after  $\theta$  by  $\varphi$ . The proof will be completed by showing that there is a constant  $\Delta_{\max}$ , independent of  $\theta$ , such that  $\Delta \leq \Delta_{\max}$  implies  $\sin^2 2\varphi \leq \sin^2 2\theta$  with equality only when  $\theta = 0$ . This will show global convergence to the fixed point zero, which is an optimal transform. As a preview of this result—and to motivate the rest of the proof—we compute and plot the “next iterate” map  $\theta \mapsto \varphi$ . Fig. 7 shows the map  $\theta \mapsto \varphi$  when  $\sigma_1 = 1$  and  $\sigma_2 = 1/2$  for  $\Delta = 1, 2, \dots, 5$ . The iteration globally converges as long as the graph of  $\varphi(\theta)$  lies inside the cone  $|\varphi| \leq |\theta|$ . From the plot, it seems this may be true for any  $\Delta$ ; we endeavor to show this for  $\Delta$  less than some  $\Delta_{\max}$ .

The first step is to relate  $\varphi$  to  $\theta$ . By looking at the general form of  $T_\theta R_{\hat{x}} T_\theta^T$ , one can show that

$$\varphi = \frac{1}{2} \arctan \left( \frac{-2(R_{\hat{x}})_{1,2}}{(R_{\hat{x}})_{1,1} - (R_{\hat{x}})_{2,2}} \right). \quad (21)$$

$R_{\hat{x}}$  is related to  $\theta$  through  $R_y$  and  $R_{\hat{y}}$

$$\begin{aligned} R_y &= T_\theta R_x T_\theta^T \\ &= \begin{bmatrix} \sigma_1^2 \cos^2 \theta + \sigma_2^2 \sin^2 \theta & \frac{1}{2}(\sigma_1^2 - \sigma_2^2) \sin 2\theta \\ \frac{1}{2}(\sigma_1^2 - \sigma_2^2) \sin 2\theta & \sigma_1^2 \sin^2 \theta + \sigma_2^2 \cos^2 \theta \end{bmatrix} \\ &= \begin{bmatrix} \nu_1^2 & \nu_{12} \\ \nu_{12} & \nu_2^2 \end{bmatrix} \\ R_{\hat{y}} &= \tilde{Q}(R_y) = R_y + \frac{\Delta^2}{12}I + \begin{bmatrix} \alpha & \beta \\ \beta & \gamma \end{bmatrix} \end{aligned} \quad (22)$$

where  $\alpha$ ,  $\beta$ , and  $\gamma$  depend on  $\theta$ ,  $\sigma_1$ , and  $\sigma_2$  as given by (13), (14), and (22). Now, after computing  $R_{\hat{x}} = T_\theta^T R_{\hat{y}} T_\theta$ , one finds

$$(R_{\hat{x}})_{1,1} - (R_{\hat{x}})_{2,2} = \sigma_1^2 - \sigma_2^2 + (\alpha - \gamma) \cos 2\theta + 2\beta \sin 2\theta$$

and

$$(R_{\hat{x}})_{1,2} = \frac{1}{2}(\gamma - \alpha) \sin 2\theta + \beta \cos 2\theta. \quad (23)$$

Since  $\sin^2(\arctan \phi) = \phi^2 / (1 + \phi^2)$ , we get (24) at the bottom of this page, where the inequality follows from minimizing the denominator over  $\theta$ . The following three lemmas allow us to complete the bounding of  $\sin^2 2\varphi$ .

*Lemma 1:*  $|\alpha - \gamma| < c_1(\Delta, \sigma_1, \sigma_2)$  uniformly in  $\theta$ , with  $c_1(\Delta, \sigma_1, \sigma_2) \rightarrow 0$  as  $\Delta \rightarrow 0$ .

*Proof:* The series in (13) is an alternating series with terms that monotonically decrease in absolute value. Thus it can be bounded (with appropriate sign) by any partial sum [28]. Using simply the first term

$$-e^{-2\pi^2(R_{\hat{y}})_{1,1}/\Delta^2} \left( \frac{\Delta^2}{\pi^2} + 4(R_{\hat{y}})_{1,1} \right) < \alpha < 0$$

and similarly for  $\gamma$ . Finally,

$$|\alpha - \gamma| \leq \max\{|\alpha|, |\gamma|\} < e^{-2\pi^2\sigma_2^2/\Delta^2} \left( \frac{\Delta^2}{\pi^2} + 4\sigma_1^2 \right) = c_1(\Delta, \sigma_1, \sigma_2)$$

since  $\alpha$  and  $\gamma$  have the same sign and  $\sigma_1 < \sigma_2$ .  $\square$

*Lemma 2:*  $|\beta| \leq c_2(\Delta, \sigma_1, \sigma_2) |\sin 2\theta|$  uniformly in  $\theta$ , with  $c_2(\Delta, \sigma_1, \sigma_2) \rightarrow 0$  as  $\Delta \rightarrow 0$ .

*Proof:* Rearranging (14) gives  $\beta = \delta(R_{\hat{y}})_{1,2} + \mu$ , where the definitions of  $\delta$  and  $\mu$  must use  $(R_{\hat{y}})_{1,1}$ ,  $(R_{\hat{y}})_{2,2}$ , and  $(R_{\hat{y}})_{1,2}$  in place of  $\nu_1^2$ ,  $\nu_2^2$ , and  $\nu_{12}$ . As in the proof of Lemma 1,  $\delta$  can be bounded by using the first term in each series

$$|\delta| < 2(e^{-2\pi^2(R_{\hat{y}})_{1,1}/\Delta^2} + e^{-2\pi^2(R_{\hat{y}})_{2,2}/\Delta^2}) < 4e^{-2\pi^2\sigma_2^2/\Delta^2}. \quad (25)$$

Assume for the moment that the absolute value of the summand of (15) decreases monotonically with both  $m_1$  and  $m_2$ . Then computing

the double summation (15) in either order gives alternating series, so the same bounding technique can be used. We get

$$\begin{aligned} |\mu| &\leq \frac{\Delta^2}{\pi^2} \exp\left(\frac{-2\pi^2((R_{\hat{y}})_{1,1} + (R_{\hat{y}})_{2,2})}{\Delta^2}\right) \\ &\quad \cdot \sinh\left(\frac{4\pi^2(R_{\hat{y}})_{1,2}}{\Delta^2}\right) \\ &= \frac{\Delta^2}{\pi^2} \exp\left(\frac{-2\pi^2(\sigma_1^2 + \sigma_2^2)}{\Delta^2}\right) \\ &\quad \cdot \sinh\left(\frac{2\pi^2(\sigma_1^2 - \sigma_2^2)}{\Delta^2}\right) \\ &\leq \frac{\Delta^2}{\pi^2} \exp\left(\frac{-2\pi^2(\sigma_1^2 + \sigma_2^2)}{\Delta^2}\right) \\ &\quad \cdot \sinh\left(\frac{2\pi^2(\sigma_1^2 - \sigma_2^2)}{\Delta^2}\right) \sin 2\theta \\ &\leq \frac{\Delta^2}{2\pi^2} e^{-2\pi^2\sigma_2^2/\Delta^2} \sin 2\theta. \end{aligned} \quad (26)$$

Combining (25) and (26) gives

$$\begin{aligned} |\beta| &= |\delta(R_{\hat{y}})_{1,2} + \mu| = \left| \frac{1}{2}\delta(\sigma_1^2 - \sigma_2^2) \sin 2\theta \right| \\ &\leq \frac{1}{2}(\sigma_1^2 - \sigma_2^2) |\delta| |\sin 2\theta| + |\mu| \\ &< \underbrace{\left( 2(\sigma_1^2 - \sigma_2^2) + \frac{\Delta^2}{2\pi^2} \right)}_{c_2(\Delta, \sigma_1, \sigma_2)} e^{-2\pi^2\sigma_2^2/\Delta^2} |\sin 2\theta|. \end{aligned}$$

In general, the terms of (15) are not monotonically decreasing. However, the terms are monotonically decreasing (in absolute value) outside of  $(m_1, m_2) \in \{1, 2, \dots, M\}^2$  for some  $M < \infty$ . Since each individual term for  $(m_1, m_2) \in \{1, 2, \dots, M\}^2$  can be bounded as above, the bound can be extended to the general case.  $\square$

*Lemma 3:*  $|\beta| < c_2(\Delta, \sigma_1, \sigma_2)$  uniformly in  $\theta$ , with  $c_2(\Delta, \sigma_1, \sigma_2) \rightarrow 0$  as  $\Delta \rightarrow 0$ .

*Proof:* This follows immediately from Lemma 2.  $\square$

By combining Lemmas 1 and 3, there exists  $\Delta_1 > 0$  such that  $\Delta < \Delta_1$  implies  $(\alpha - \gamma)^2 + 4\beta^2 \leq (\sigma_1^2 - \sigma_2^2)^2/4$ , uniformly in  $\theta$ . Thus assuming  $\Delta < \Delta_1$  we have

$$\sin^2 2\varphi \leq \frac{[(\alpha - \gamma)^2 \sin 2\theta - 2\beta \cos 2\theta]^2}{\frac{1}{4}(\sigma_1^2 - \sigma_2^2)^2}.$$

Applying Lemmas 1 and 2

$$\sin^2 2\varphi \leq (c_1 + 2c_2)^2 \sin^2 2\theta$$

$$\begin{aligned} \sin^2 2\varphi &= \frac{\left( \frac{-2(R_{\hat{x}})_{1,2}}{(R_{\hat{x}})_{1,1} - (R_{\hat{x}})_{2,2}} \right)^2}{1 + \left( \frac{-2(R_{\hat{x}})_{1,2}}{(R_{\hat{x}})_{1,1} - (R_{\hat{x}})_{2,2}} \right)^2} = \frac{(-2(R_{\hat{x}})_{1,2})^2}{((R_{\hat{x}})_{1,1} - (R_{\hat{x}})_{2,2})^2 + (-2(R_{\hat{x}})_{1,2})^2} \\ &= \frac{[(\alpha - \gamma)^2 \sin 2\theta - 2\beta \cos 2\theta]^2}{(\sigma_1^2 - \sigma_2^2)^2 + 2(\sigma_1^2 - \sigma_2^2)[(\alpha - \gamma)^2 \cos 2\theta - 2\beta \sin 2\theta] + (\alpha - \gamma)^2 + 4\beta^2} \\ &\leq \frac{[(\alpha - \gamma)^2 \sin 2\theta - 2\beta \cos 2\theta]^2}{[\sigma_1^2 - \sigma_2^2 - \sqrt{(\alpha - \gamma)^2 + 4\beta^2}]^2} \end{aligned} \quad (24)$$

and there exists  $\Delta_2 > 0$  such that  $\Delta < \Delta_2$  implies  $(c_1 + 2c_2)^2 < 1$ . The proof is complete with  $\Delta_{\max} = \min\{\Delta_1, \Delta_2\}$ .

The bounds in this theorem are rather complicated but we can check that the requirements on  $\Delta$  are reasonable. Suppose  $\sigma_1 = 1$  and  $\sigma_2 = 1/2$ . Then  $\Delta_1 > 1.366$  and  $\Delta_2 > 1.565$ , so the theorem guarantees convergence for any  $\Delta < 1.366$ . (For this range of  $\Delta$ , (15) can be bounded by the  $m_1 = m_2 = 1$  term for any  $\theta$ .) As we found for Theorem 2, numerical calculations suggest convergence for any  $\Delta$  (see Fig. 7).

#### D. Proof of Theorem 4

The mean-square convergence of  $\widehat{R}_x^{(n)}$  follows from the Chebyshev law of large numbers [29] once we establish that each term of (5) has common expected value  $R_x + \Delta^2 I/12$ , has finite variance, and is elementwise uncorrelated with every other term. The second conclusion follows easily.

First note that

$$\hat{x}_k = T_k^T \hat{y}_k = T_k^T (y_k + (\hat{y}_k - y_k)) = x_k + T_k^T (\hat{y}_k - y_k).$$

Because of the use of subtractive dither,  $\hat{y}_k - y_k$  is uniformly distributed on the hypercube  $[-\Delta/2, \Delta/2]^N$  and independent of  $x_k$  and  $T_k$  [18], [19]. (The overall error  $\hat{x}_k - x_k$  is uniformly distributed on a rotated hypercube, independent of  $x_k$  but not independent of  $T_k$ . Its components are uncorrelated but not independent.) Now any term of (5) can be expanded as

$$\hat{x}_k \hat{x}_k^T = x_k x_k^T + x_k (\hat{y}_k^T - y_k^T) T_k + T_k^T (\hat{y}_k - y_k) x_k^T + T_k^T (\hat{y}_k - y_k) (\hat{y}_k^T - y_k^T) T_k. \quad (27)$$

Since  $\hat{y}_k - y_k$  depends on  $T_k$  but  $x_k$  and  $T_k$  are independent, computing the expectation of  $\hat{x}_k \hat{x}_k^T$  is simplified by first conditioning on  $T_k$

$$\begin{aligned} E[\hat{x}_k \hat{x}_k^T | T_k] &= E[x_k x_k^T | T_k] + E[x_k (\hat{y}_k^T - y_k^T) T_k | T_k] \\ &\quad + E[T_k^T (\hat{y}_k - y_k) x_k^T | T_k] \\ &\quad + E[T_k^T (\hat{y}_k - y_k) (\hat{y}_k^T - y_k^T) T_k | T_k] \\ &= R_x + 0 + 0 + T_k^T \frac{\Delta^2}{12} I T_k \\ &= R_x + \frac{\Delta^2}{12} I \end{aligned}$$

where we have used the independence of  $x_k$  and  $\hat{y}_k - y_k$  and the fact that each has mean zero.

The  $(i, j)$  element of  $\widehat{R}_x^{(n)}$  is the average of  $n$  random observations of  $(\hat{x}_k \hat{x}_k^T)_{ij}$ , which we denote  $A_{ij}^{(k)}$ . The calculation above shows that each  $A_{ij}^{(k)}$  has mean  $(R_x)_{ij} + \Delta^2 \delta_{ij}/12$ . It can furthermore be shown that each  $A_{ij}^{(k)}$  has variance bounded by a constant and that  $A_{ij}^{(k)}$  is uncorrelated with  $A_{ij}^{(\ell)}$  for  $k \neq \ell$  [16, Appendix 3.C]. Thus by the Chebyshev law of large numbers we have that  $\widehat{R}_x^{(n)} \rightarrow R_x + \Delta^2 I/12$  elementwise in mean-square. The second conclusion follows from the fact that  $R_x$  and  $R_x + \Delta^2 I/12$  have the same eigenvectors.

Note that the dither is essential to the proof because it makes the quality of the estimate  $\widehat{R}_x^{(n)}$  independent of the sequence of transforms.

#### E. Proof of Theorem 5

The convergence of  $n^{-1}(L_n - L_n^*)$  to a constant follows by mimicking the proof of Theorem 1. In this case, the constant  $\rho$  is not zero because the entropy rate of the quantizer output depends not only on the transform but on the dither. It remains to estimate  $\rho$ .

Using (3) and the differential entropy of a Gaussian random variable

$$h(\mathcal{N}(0, \sigma^2)) = \frac{1}{2} \log_2 2\pi e \sigma^2 \text{ bits}$$

the entropy of a Gaussian random variable with variance  $\sigma^2$ , uniformly quantized with bin width  $\Delta$ , is approximately  $2^{-1} \log_2 \Delta^{-2} 2\pi e$  bits. Thus the rate of the static optimal system is approximately

$$R_{\text{opt}} = \frac{1}{N} \sum_{i=1}^N \frac{1}{2} \log_2 \frac{2\pi e \lambda_i}{\Delta^2} \text{ bits/component.} \quad (28)$$

The adaptive scheme converges to an optimal transform. However, because of the dithering, the signal at the input to the quantizer is not Gaussian and does not have component variances equal to the  $\lambda_i$ 's. Since  $\{z_n\}$  is independent of  $\{y_n\}$ , the variances simply add, giving  $\lambda_i + \Delta^2/12$ ,  $i = 1, 2, \dots, N$ . Since a Gaussian probability density function has the largest differential entropy for a given variance, the asymptotic rate of the universal coder can be bounded as

$$R_{\text{univ}} < \frac{1}{N} \sum_{i=1}^N \frac{1}{2} \log_2 \frac{2\pi e (\lambda_i + \Delta^2/12)}{\Delta^2}. \quad (29)$$

Subtracting (28) from (29) and pairing terms gives (8).

#### ACKNOWLEDGMENT

The perceptive comments of an anonymous reviewer are gratefully acknowledged.

#### REFERENCES

- [1] R. M. Gray and D. L. Neuhoff, "Quantization," *IEEE Trans. Inform. Theory*, vol. 44, pp. 2325–2383, Oct. 1998.
- [2] A. Gersho and R. M. Gray, *Vector Quantization and Signal Compression*. Boston, MA: Kluwer, 1992.
- [3] J. J. Y. Huang and P. M. Schultheiss, "Block quantization of correlated Gaussian random variables," *IEEE Trans. Commun. Syst.*, vol. COM-11, pp. 289–296, Sept. 1963.
- [4] J. D. Gibson, "Adaptive prediction in speech differential encoding systems," *Proc. IEEE*, vol. 68, pp. 488–525, Apr. 1980.
- [5] A. Gersho, "Advances in speech and audio compression," *Proc. IEEE*, vol. 82, pp. 900–918, June 1994.
- [6] R. V. Cox, "Speech coding," in *The Digital Signal Processing Handbook*. Piscataway, NJ: IEEE Press, 1998, ch. 45, pp. 45.1–45.19.
- [7] S. LoPresto, K. Ramchandran, and M. T. Orchard, "Image coding based on mixture modeling of wavelet coefficients and a fast estimation-quantization framework," in *Proc. IEEE Data Compression Conf.*, J. A. Storer and M. Cohn, Eds., Snowbird, UT, Mar. 1997, pp. 221–230.
- [8] A. Ortega and M. Vetterli, "Adaptive scalar quantization without side information," *IEEE Trans. Image Processing*, vol. 6, pp. 665–676, May 1997.
- [9] M. Effros and P. A. Chou, "Weighted universal transform coding: Universal image compression with the Karhunen-Loève transform," in *Proc. IEEE Int. Conf. Image Processing*, vol. II, Washington, DC, Oct. 1995, pp. 61–64.
- [10] Z. Zhang and V. K. Wei, "An on-line universal lossy data compression algorithm via continuous codebook refinement—Part I: Basic results," *IEEE Trans. Inform. Theory*, vol. 42, pp. 803–821, May 1996.
- [11] R. D. Dony and S. Haykin, "Optimally adaptive transform coding," *IEEE Trans. Image Processing*, vol. 4, pp. 1358–1370, Oct. 1995.
- [12] H. P. Kramer and M. V. Mathews, "A linear coding for transmitting a set of correlated signals," *IRE Trans. Inform. Theory*, vol. IT-2, pp. 41–46, Sept. 1956.
- [13] H. Gish and J. P. Pierce, "Asymptotically efficient quantizing," *IEEE Trans. Inform. Theory*, vol. IT-14, pp. 676–683, Sept. 1968.
- [14] R. C. Wood, "On optimum quantization," *IEEE Trans. Inform. Theory*, vol. IT-15, pp. 248–252, Mar. 1969.
- [15] A. Rényi, "On the dimension and entropy of probability distributions," *Acta Math. Acad. Sci. Hungar.*, vol. 10, pp. 193–215, 1959.
- [16] V. K. Goyal, "Beyond Traditional Transform Coding," Univ. Calif., Berkeley, Ph.D. dissertation, 1998.

- [17] L. Cheded and P. A. Payne, "The exact impact of amplitude quantization on multi-dimensional, high-order moments estimation," *Signal Processing*, vol. 39, no. 3, pp. 293–315, Sept. 1994.
- [18] S. P. Lipshitz, R. A. Wannamaker, and J. Vanderkooy, "Quantization and dither: A theoretical survey," *J. Audio Eng. Soc.*, vol. 40, no. 5, pp. 355–375, May 1992.
- [19] R. M. Gray and T. G. Stockham, Jr., "Dithered quantizers," *IEEE Trans. Inform. Theory*, vol. 39, pp. 805–812, May 1993.
- [20] D. W. E. Schobben, R. A. Beuker, and W. Oomen, "Dither and data compression," *IEEE Trans. Signal Processing*, vol. 45, pp. 2097–2101, Aug. 1997.
- [21] V. K. Goyal, M. Vetterli, and N. T. Thao, "Quantized overcomplete expansions in  $\mathbb{R}^N$ : Analysis, synthesis, and algorithms," *IEEE Trans. Inform. Theory*, vol. 44, pp. 16–31, Jan. 1998.
- [22] V. K. Goyal, J. Zhuang, and M. Vetterli, "Universal transform coding based on backward adaptation," in *Proc. IEEE Data Compression Conf.*, J. A. Storer and M. Cohn, Eds., Snowbird, UT, Mar. 1997, pp. 231–240.
- [23] V. K. Goyal and M. Vetterli, "Block transform adaptation by stochastic transform coding of Gaussian vectors," in *Proc. IEEE Digital Signal Processing Workshop*, Bryce Canyon, UT, Aug. 9–12, 1998.
- [24] J. Ziv, "On universal quantization," *IEEE Trans. Inform. Theory*, vol. IT-31, pp. 344–347, May 1985.
- [25] A. W. Marshall and I. Olkin, *Inequalities: Theory of Majorizations and Its Applications*. San Diego, CA: Academic, 1979.
- [26] I. E. Telatar, private communication, July 1999.
- [27] G. H. Golub and C. F. Van Loan, *Matrix Computations*, 2nd ed. Baltimore, MD: Johns Hopkins Univ. Press, 1989.
- [28] T. M. Apostol, *Mathematical Analysis*, 2nd ed. Reading, MA: Addison-Wesley, 1974.
- [29] A. Papoulis, *Probability, Random Variables, and Stochastic Processes*. New York: McGraw-Hill, 1965.

## Universal Coding of Nonstationary Sources

Karthik Visweswariah, Sanjeev R. Kulkarni, *Senior Member, IEEE*,  
and Sergio Verdú, *Fellow, IEEE*

**Abstract**—In this correspondence we investigate the performance of the Lempel–Ziv incremental parsing scheme on nonstationary sources. We show that it achieves the best rate achievable by a finite-state block coder for the nonstationary source. We also show a similar result for a lossy coding scheme given by Yang and Kieffer which uses a Lempel–Ziv scheme to perform lossy coding.

**Index Terms**—Data compression, entropy, Lempel–Ziv algorithm, nonstationary sources, universal source coding.

### I. INTRODUCTION

We investigate the use of universal coding methods for coding nonstationary sources. It is widely known that Lempel–Ziv coding methods are asymptotically optimal for the coding of stationary ergodic sources. We will show that for lossless coding of finite possibly nonstationary sources Lempel–Ziv coding methods perform as well as any finite-state

Manuscript received March 6, 1999; revised December 6, 1999. This work was supported in part by the National Science Foundation under Grants NYI Award IRI-9457645 and NCR 9523805.

K. Visweswariah was with the Department of Electrical Engineering Princeton University, Princeton, NJ 08544 USA. He is now with IBM Thomas J. Watson Research Center, Yorktown Heights, NY 10598 USA.

S. R. Kulkarni and S. Verdú are with the Department of Electrical Engineering, Princeton University, Princeton, NJ 08544 USA.

Communicated by N. Merhav, Associate Editor for Source Coding.

Publisher Item Identifier S 0018-9448(00)04289-9.

block coding scheme. We will also consider as an example of a nonstationary source the Arbitrarily Varying Source and investigate the performance of universal noiseless coding schemes for this source. For lossy coding of finite sources we show that the Yang–Kieffer coding scheme asymptotically performs better than any block code even when applied to nonstationary sources.

Lempel–Ziv coding techniques are known to be asymptotically optimal for individual sequences in the sense that they perform as well as any finite-state coding scheme [11]. There are also *uniform* bounds on the performance of the incremental-parsing techniques in terms of the coding performance achievable by a finite-state coder [5]. Kieffer [3], [4] gave the optimal rate for coding nonstationary sources using finite-state coders. Given the performance of Lempel–Ziv techniques on individual sequences it is natural to investigate whether they achieve the optimal rates given by Kieffer for nonstationary sources.

### II. LOSSLESS CODING

Let us consider the lossless coding of a source  $\mathbf{X} = (X_1, X_2, \dots)$  with distribution  $P$ . Let  $X^n = (X_1, X_2, \dots, X_n)$  take values in  $A^n$  governed by the marginal distribution  $P_n$ , where  $A$  is a finite set. The probability measure  $P$  is possibly nonstationary. We code the source using the Lempel–Ziv incremental parsing scheme given in [11]. We will define the stationary hull of  $P$  as defined in [4] and denote it by  $S(P)$ . For the sake of completeness we repeat the definition here.

*Definition 1:* Consider a (possibly nonstationary) random process  $\mathbf{X}$  taking values in a finite set  $A$ . We say that a process  $\mathbf{Z}$  belongs to the stationary hull of  $\mathbf{X}$  if there exists a sequence of natural numbers  $n_0, n_1, \dots$  such that

$$\lim_{j \rightarrow \infty} \frac{1}{n_j} \sum_{i=1}^{n_j} E(f(X_i, X_{i+1}, \dots, X_{i+m-1})) = E(f(Z_1, Z_2, \dots, Z_m))$$

for all real-valued functions  $f$  that depend only on finitely many coordinates.

Processes in the stationary hull capture properties of finite-dimensional distributions along various convergent subsequences. In particular, if for a given size  $m$  the best  $m$ -block code has bad performance on the nonstationary source along a particular subsequence then there is a source in the stationary hull which reflects this. It is shown in [3] that the best possible average rate at which the source can be coded using a finite-state adaptive block to variable-length code is given by

$$R(P) = \sup_{Z \in S(P)} H(Z)$$

where  $H(Z)$  is the entropy rate of the stationary source  $Z$ . We will show that this rate is achieved asymptotically by the Lempel–Ziv coding scheme. Let  $LZ(x^n)$  denote the length of  $x^n$  when coded by the Lempel–Ziv algorithm. We will denote by  $x_i^j$  the string  $(x_i, x_{i+1}, \dots, x_j)$ . We will see that the key property we will require is a uniform bound on  $LZ(x^n)$  in terms of a certain finite-state code. One such bound is given by Lemma 1 in [10, Appendix], but we could have also used the result in [5].

*Theorem 1:* Suppose  $\mathbf{X}$  is a source with distribution  $P$  that takes values in a finite set, then

$$\limsup_{n \rightarrow \infty} \frac{E(LZ(X^n))}{n} \leq R(P).$$

*Proof:* Fix  $\epsilon > 0$ . From Lemma A2 in [3, Appendix] there exists a block length  $t$  and a prefix-free code  $\phi : A^t \rightarrow \{0, 1\}^\infty$  such that

$$E \left( \frac{l(\phi(Z_1^t))}{t} \right) \leq R(P) + \frac{\epsilon}{2} \quad (1)$$

for  $\mathbf{Z} \in S(P)$ . Using Lemma 1 in [10, Appendix] with block length  $t$  and code  $\phi$  we have a sequence  $\delta_n$  of positive numbers tending to zero as  $n \rightarrow \infty$  such that

$$LZ(x^n) \leq \min_{1 \leq j \leq t} \sum_{\substack{i=j \bmod t \\ 1 \leq i \leq n-t+1}} l(\phi(x_i^{i+t-1})) + n\delta_n.$$

This implies that

$$LZ(x^n) \leq \frac{1}{t} \sum_{1 \leq j \leq t} \sum_{\substack{i=j \bmod t \\ 1 \leq i \leq n-t+1}} l(\phi(x_i^{i+t-1})) + n\delta_n.$$

Dividing both sides by  $n$  and taking expectations and limits we have

$$\begin{aligned} & \limsup_{n \rightarrow \infty} \frac{1}{n} E(LZ(X^n)) \\ & \leq \limsup_{n \rightarrow \infty} \left( \frac{1}{nt} \sum_{1 \leq j \leq t} \sum_{\substack{i=j \bmod t \\ 1 \leq i \leq n-t+1}} E(l(\phi(X_i^{i+t-1}))) + \delta_n \right) \\ & \leq \limsup_{n \rightarrow \infty} \left( \frac{1}{nt} \sum_{1 \leq i \leq n-t+1} E(l(\phi(X_i^{i+t-1}))) \right) + \limsup_{n \rightarrow \infty} \delta_n \\ & = \frac{1}{t} E(l(\phi(Z_i^{i+t-1}))) \\ & \leq R(P) + \frac{\epsilon}{2} \end{aligned}$$

where the equality is for some  $\mathbf{Z} \in S(P)$  and follows from Lemma 1 in the Appendix, and the last inequality follows from (1). Now  $\epsilon > 0$  was arbitrary so we have that

$$\limsup_{n \rightarrow \infty} \frac{E(LZ(X^n))}{n} \leq R(P). \quad \square$$

### III. EXAMPLE: CODING AN ARBITRARILY VARYING SOURCE

In this section we will use bounds on the performance of universal codes on individual sequences to investigate their performance on the arbitrarily varying source, studied as an example in [2]. The arbitrarily varying source can be used to model, for example, the piecewise-stationary Bernoulli source studied in [9].

Let  $S$  be a finite set and  $A$  be the finite alphabet on which the source takes values. Let  $p(\cdot|s)$  be a probability distribution on  $A$  for each  $s \in S$ . An Arbitrarily Varying Source (AVS) is a nonstationary source defined by an infinite sequence  $\mathbf{s} \in S^\infty$ . The probability of a string  $x_1^n \in A^n$  occurring is given by

$$P(x_1^n | s_1^n) = \prod_{i=1}^n p(x_i | s_i).$$

If the underlying state sequence is known then the optimal fixed-variable coding rate is clearly

$$\limsup_{n \rightarrow \infty} \frac{1}{n} \sum_{i=1}^n H(P(\cdot|s_i)).$$

This is also the block-to-block coding rate (see [7]).

Consider a string  $x_1^n \in A^n$  and a string  $y_1^m \in A^m$  with  $m < n$ . Let us define the empirical probability distribution  $\hat{P}$  by

$$\hat{P}_m^m(y_1^m) \triangleq \frac{1}{n-m+1} \sum_{i=1}^{n-m+1} \delta(x_i^{i+m-1}, y_1^m)$$

where  $\delta(x, y) = 1$  if  $x = y$  and is zero otherwise. Let  $\hat{H}_m$  denote the entropy of the empirical distribution on  $m$  blocks defined above. We can also define the conditional empirical probability distribution and entropy by

$$\hat{P}^n(a|y_1^m) \triangleq \frac{\hat{P}(y_1^m a)}{\hat{P}(y_1^m)}$$

and

$$\hat{H}_{1|m} = \sum_{y_1^m \in A^m} \hat{P}(y_1^m) H(\hat{P}(\cdot|y_1^m)).$$

Similarly, we can define the empirical distribution of  $m$  blocks on the state sequence  $s_1^n$ , we denote this distribution by  $\hat{Q}_m^n$ . Note that when we write  $xy$  where  $x$  and  $y$  are two strings we mean  $y$  concatenated with  $x$ . We also drop the  $m$  and  $n$  in the notation for the empirical distribution, the dimension of the distribution, and the length of the underlying string when these are clear from the context.

Suppose we have a code  $\phi$  such that for any fixed  $m$

$$l(\phi(x_1^n)) \leq n\hat{H}_{1|m} + o(n). \quad (2)$$

Note that the  $o(n)$  term does not depend on the sequence  $x_1^n$ . Codes based on the Lempel–Ziv incremental parsing scheme and the infinite depth context tree weighting method can be shown to have such a property [6], [5], [8]. We can bound the performance of such codes on arbitrarily varying sources and the result is given in Theorem 2. The theorem implies that if the underlying state sequence has low complexity then codes can learn the source and perform as well asymptotically as if the underlying state sequence were known.

*Theorem 2:* If a code  $\phi$  satisfies (2) then for an arbitrarily varying source  $\mathbf{X}$  with an underlying state sequence  $\mathbf{s}$  and for any integer  $m > 0$  we have

$$\begin{aligned} \limsup_{n \rightarrow \infty} E_s \left( \frac{l(\phi(X^n))}{n} \right) & \leq \limsup_{n \rightarrow \infty} \sum_{z \in S} \hat{Q}^n(z) H(P(\cdot|z)) \\ & \quad + \limsup_{n \rightarrow \infty} \frac{1}{m+1} H(\hat{Q}_{m+1}^n) \end{aligned}$$

where  $\hat{Q}_m^n$  is the empirical  $m$ -dimensional distribution of  $s_1^n$ .

*Proof:* To prove the theorem we will require a bound on  $\hat{H}_{1|m}$  in terms of  $\hat{H}_{m+1}$ . This would be easy if the empirical distributions were stationary. We note that the empirical distributions defined above are not stationary, that is,

$$\sum_{y_2^m \in A^{m-1}} \hat{P}(ay_2^m) \neq \sum_{y_1 y_3^m \in A^{m-1}} \hat{P}(y_1 a y_3^m).$$

To make this distribution stationary we can define the empirical distribution by

$$\hat{P}'(y_1^m) \triangleq \frac{1}{n} \sum_{i=1}^n \delta(x_i^{i+m-1}, y_1^m)$$

where all indices are considered mod  $n$ . We then have for the corresponding entropies

$$\hat{H}'_{1|m} \leq \frac{1}{m+1} \hat{H}'_{m+1}.$$

Now

$$\begin{aligned}
 |\hat{H}_{1|m} - \hat{H}'_{1|m}| &\leq \sum_{y_1^m \in A^m} \hat{P}(y_1^m) |H(\hat{P}(\cdot|y_1^m)) - H(\hat{P}'(\cdot|y_1^m))| \\
 &\quad + \sum_{y_1^m \in A^m} |\hat{P}(y_1^m) - \hat{P}'(y_1^m)| H(\hat{P}'(\cdot|y_1^m)) \\
 &\leq \sum_{y_1^m \in A^m} \hat{P}(y_1^m) l_1(\hat{P}(\cdot|y_1^m), \\
 &\quad \hat{P}'(\cdot|y_1^m)) \log \frac{|A|}{l_1(\hat{P}(\cdot|y_1^m), \hat{P}'(\cdot|y_1^m))} \\
 &\quad + \sum_{y_1^m \in A^m} l_1(\hat{P}(\cdot|y_1^m), \hat{P}'(\cdot|y_1^m)) H(\hat{P}'(\cdot|y_1^m)) \quad (3)
 \end{aligned}$$

where  $l_1(\cdot, \cdot)$  denotes the  $l_1$  distance between two distributions on the same alphabet. The last inequality follows for sufficiently large  $n$  from [1, Theorem 16.3.2] since we can show that

$$l_1(\hat{P}(\cdot|y_1^m), \hat{P}'(\cdot|y_1^m)) \leq \frac{2m|A|}{n-m} \quad (4)$$

and hence satisfies the condition required in the theorem for sufficiently large  $n$ . Using (3) and (4) we have

$$|\hat{H}_{1|m} - \hat{H}'_{1|m}| \leq \frac{2m|A|}{n-m} \log \frac{n-m}{2m} + \frac{2m|A|2^m}{n-m} \log |A|.$$

Similarly we can show that

$$|\hat{H}_{m+1} - \hat{H}'_{m+1}| \leq \frac{2m2^m}{n-m} \log \frac{n-m}{2m}.$$

Now we can bound the performance of the code  $\phi$  as follows:

$$\begin{aligned}
 l(\phi(x^n)) &\leq n\hat{H}_{1|m} + o(n) \\
 &\leq n\hat{H}'_{1|m} \\
 &\quad + n \left( \frac{2m|A|}{n-m} \log \frac{n-m}{2m} + \frac{2m|A|2^m}{n-m} \log |A| \right) + o(n) \\
 &\leq \frac{n}{m+1} \hat{H}'_{m+1} \\
 &\quad + n \left( \frac{2m|A|}{n-m} \log \frac{n-m}{2m} + \frac{2m|A|2^m}{n-m} \log |A| \right) + o(n) \\
 &\leq \frac{n}{m+1} \hat{H}_{m+1} \\
 &\quad + n \left( \frac{2m(|A|+2^m)}{n-m} \log \frac{n-m}{2m} + \frac{2m|A|2^m}{n-m} \log |A| \right) + o(n).
 \end{aligned}$$

Let us consider the underlying state sequence  $\mathbf{s}$ . We can define, as before, a conditional empirical distribution  $\hat{W}(y^{m+1}|z^{m+1})$  where  $y^{m+1} \in A^{m+1}$  and  $z^{m+1} \in S^{m+1}$ . We will denote the empirical distribution of the underlying state sequence by  $\hat{Q}$ . Now we have

$$\begin{aligned}
 \sum_{z^{m+1} \in S^{m+1}} \hat{Q}(z^{m+1}) H(\hat{W}(\cdot|z^{m+1})) + I(\hat{P}_{m+1}, \hat{W}_{m+1}) \\
 = H(\hat{P}_{m+1}).
 \end{aligned}$$

Thus we have

$$\begin{aligned}
 l(\phi(x^n)) &\leq \frac{n}{m+1} \left( \sum_{z^{m+1} \in S^{m+1}} \hat{Q}(z^{m+1}) H(\hat{W}(\cdot|z^{m+1})) \right. \\
 &\quad \left. + I(\hat{Q}_{m+1}, \hat{W}_{m+1}) \right) \\
 &\quad + n \left( \frac{2m(|A|+2^m)}{n-m} \log \frac{n-m}{2m} + \frac{2m|A|2^m}{n-m} \log |A| \right) \\
 &\quad + o(n).
 \end{aligned}$$

Note that, although it is not explicit in the notation, all the empirical distributions depend on the sequence  $x_1^n$  and/or the underlying state sequence. We can take expectations on both sides of the previous inequality assuming a fixed underlying state sequence. Also since

$$I(\hat{Q}_{m+1}, \hat{W}_{m+1}) \leq H(\hat{Q}_{m+1})$$

we have

$$\begin{aligned}
 E_s \left( \frac{l(\phi(X^n))}{n} \right) &\leq \frac{1}{m+1} \left( \sum_{z^{m+1} \in S^{m+1}} \hat{Q}(z^{m+1}) \right. \\
 &\quad \left. E_s \left( H(\hat{W}(\cdot|z^{m+1})) \right) + H(\hat{Q}_{m+1}) \right) \\
 &\quad + \frac{2m(|A|+2^m)}{n-m} \log \frac{n-m}{2m} \\
 &\quad + \frac{2m|A|2^m}{n-m} \log |A| + \frac{o(n)}{n}.
 \end{aligned}$$

Since  $H(\cdot)$  is concave we have

$$\begin{aligned}
 \limsup_{n \rightarrow \infty} E_s \left( \frac{l(\phi(X^n))}{n} \right) &\leq \limsup_{n \rightarrow \infty} \frac{1}{m+1} \sum_{z^{m+1} \in S^{m+1}} \hat{Q}^n(z^{m+1}) \\
 &\quad \cdot H \left( E_s \hat{W}^n(\cdot|z^{m+1}) \right) + H(\hat{Q}_{m+1}^n) \\
 &= \limsup_{n \rightarrow \infty} \frac{1}{m+1} \sum_{z^{m+1} \in S^{m+1}} \hat{Q}^n(z^{m+1}) \\
 &\quad \cdot H(P_{m+1}(\cdot|z^{m+1})) + H(\hat{Q}_{m+1}^n) \\
 &\leq \limsup_{n \rightarrow \infty} \sum_{z \in S} \hat{Q}^n(z) H(P(\cdot|z)) \\
 &\quad + \limsup_{n \rightarrow \infty} \frac{1}{m+1} H(\hat{Q}_{m+1}^n). \quad (5)
 \end{aligned}$$

The last equality follows because  $P_{m+1}(\cdot|z^{m+1})$  has a product form so that the corresponding entropy is just the sum of the entropies of the one dimensional distributions.  $\square$

If we assume that the state sequence has zero empirical entropy rate (as defined in [11]) then as  $m \rightarrow \infty$  the second term in (5) goes to zero. Assuming that the underlying state has zero empirical entropy rate means that the state sequence has patterns that can be learned by the coding algorithm. The first term in (5) is independent of  $m$  and is equal to the coding performance that would be achieved if the underlying state sequence were known.

#### IV. LOSSY CODING

We consider a source  $\mathbf{X} = (X_1, X_2, \dots)$  with distribution  $P$ . As before, let the source  $X^n = (X_1, X_2, \dots, X_n)$  take values in  $A^n$ . Let  $P_n$  govern the probabilities of  $n$  strings. The probability measure  $P$  is possibly nonstationary. We assume that  $A$  is a finite set. We assume that the reproduction alphabet is also  $A$ . Let  $\rho : A \times A \rightarrow [0, \infty)$  be the distortion measure. Using  $\rho$  we can define distortion for  $n$  strings as

$$\rho_n((x_1, \dots, x_n), (y_1, \dots, y_n)) = \frac{1}{n} \sum_{i=1}^n \rho(x_i, y_i).$$

A block code of block size  $N$  is defined by a map  $\phi : A^N \rightarrow A^N$ . The average distortion for a code  $\phi$  is defined as

$$\bar{\rho}(P, \phi) = \limsup_{n \rightarrow \infty} \sum_{x^n \in A^n} P(x^n) \rho_n(x^n, \phi(x^n)).$$

Clearly, if  $\phi$  is a block code of size  $N$  and  $Q$  is a stationary source then

$$\bar{\rho}(Q, \phi) = \sum_{x^N \in A^N} Q(x^N) \rho_N(x^N, \phi(x^N)).$$

It is shown in [3] that the best possible distortion achievable for a source with measure  $P$  using block codes of rate  $R$  is given by

$$\mathcal{D}_b^*(R, P) = \sup_{Q \in \mathcal{S}(P)} D_b(R, Q).$$

Since  $Q$  is stationary  $D_b(R, Q)$  is known.

We will use the coding method given in [10]. As before, let  $LZ(x^n)$  denote the Lempel–Ziv coding length of a string  $x^n$ . By Lempel–Ziv coding we mean the incremental parsing scheme given in [11]. Let

$$B_n(R) = \{x^n \in A^n : LZ(x^n) \leq nR\}.$$

Now the size of  $B_n(R)$  is no more than  $2^{\lfloor nR \rfloor}$ . To code a string  $x^n$  we choose that string in  $B_n(R)$  that has minimum distortion with  $x^n$ . Thus we code the source using  $B_n(R)$  as our code book. We can code each  $n$  block with no more than  $\lfloor nR \rfloor$  bits.

Given a set  $C_n$  which is a subset of  $A^n$  let  $\rho(C_n)(x^n)$  denote the minimum distortion incurred when coding the string  $x^n$  using  $C_n$  as a code book.

**Theorem 3:** Suppose  $\mathbf{X}$  is a source with distribution  $P$  that takes values in a finite set then

$$\limsup_{n \rightarrow \infty} E(\rho(B_n(R))(X^n)) \leq \mathcal{D}_b^*(R, P).$$

*Proof:* We have from the discussion after [3, eq. (3.8)], that for any  $\epsilon > 0$  there exists a block code  $\phi : A^N \rightarrow A^N$  of rate less than  $R - \epsilon$  and block length  $N > 4/\epsilon$  such that

$$\bar{\rho}(Q, \phi) \leq \mathcal{D}_b^*(R - 2\epsilon, P) + \epsilon/4.$$

Since the rate of the code is at most  $R - \epsilon$  there exists a length function  $\sigma$  which satisfies Kraft's inequality and such that  $\sigma(y^N) \leq N(R - \epsilon) + 2$  for any string  $y^N$  in  $\phi(A^N)$ .

As in [10], from the block code  $\phi$  we can define a code  $\phi_n^j$  as follows:  $\phi_n^j(x^n)_{i^{j+N-1}} \triangleq \phi(x_{i^{j+N-1}})$  if  $1 \leq i \leq n - N + 1$  and  $i = j \bmod N$ . At coordinates not defined by the above equation let  $\phi_n^j(x^n)_k \triangleq a_0$  for some  $a_0$  in  $A$ . Now using Lemma 1 in [10, Appendix] we have

$$LZ(\phi_n^j(x^n)) \leq n(R - \epsilon + 2/N + \delta_n)$$

where  $\delta_n \rightarrow 0$  as  $n \rightarrow \infty$ . Once again we point out that we could use the bound in [5] instead of Lemma 1 in [10, Appendix] and obtain essentially the same result. Since  $2/N < \epsilon/2$  we have that for sufficiently large  $n$ ,  $\phi_n^j$  maps  $A^n$  into  $B_n(R)$ . Thus we have for  $1 \leq j \leq N$

$$\begin{aligned} n\rho(B_n(R))(X^n) &\leq \sum_{i=j \bmod N, 1 \leq i \leq n-N+1} N\rho_N \\ &\cdot \left( X_i^{i+N-1}, \phi_n^j(X_i^{i+N-1}) \right) + N\rho_{\max} \end{aligned}$$

where  $\rho_{\max} = \max_{x \in A} \rho(x, a_0)$ . Averaging over  $j$  and dividing by  $n$  we get

$$\begin{aligned} \rho(B_n(R))(X^n) &\leq \frac{1}{n} \sum_{i=1}^{n-N+1} \rho_N \left( X_i^{i+N-1}, \phi \left( X_i^{i+N-1} \right) \right) + \frac{N\rho_{\max}}{n}. \end{aligned}$$

Thus we have from Lemma 1 in the Appendix that

$$\limsup_{n \rightarrow \infty} E(\rho(B_n(R))(X^n)) \leq E(\rho_N(Z^N, \phi(Z^N)))$$

for some source  $\mathbf{Z}$  with measure  $Q$  in the stationary hull of  $P$ .

But  $E(\rho_N(Z^N, \phi(Z^N))) = \bar{\rho}(Q, \phi)$  and so

$$E(\rho_N(Z^N, \phi(Z^N))) \leq \mathcal{D}_b^*(R - 2\epsilon, P) + \epsilon/4.$$

Since  $\epsilon > 0$  was arbitrary and since  $\mathcal{D}_b^*$  is a convex (and hence continuous) function of  $R$  (from [3]) we have

$$\limsup_{n \rightarrow \infty} E(\rho(B_n(R))(X^n)) \leq \mathcal{D}_b^*(R, P). \quad \square$$

## APPENDIX

We now state and prove a lemma which is useful in proving Theorems 1 and 3. We are given a random process  $\mathbf{X}$  with distribution  $P$  which takes values in  $A$ .

**Lemma 1:** For any real function  $\phi$  defined on  $A^t$ , there exists  $\mathbf{Z} \in \mathcal{S}(P)$  such that

$$\limsup_{n \rightarrow \infty} \frac{1}{n} \sum_{i=1}^n E(\phi(X_i^{i+t-1})) = E(\phi(Z_1^t)).$$

*Proof:* It is clear that we can find a subset of the natural numbers  $N^0$  such that

$$\begin{aligned} \limsup_{n \rightarrow \infty} \frac{1}{n} \sum_{i=1}^n E(\phi(X_i^{i+t-1})) \\ = \lim_{n \rightarrow \infty, n \in N^0} \frac{1}{n} \sum_{i=1}^n E(\phi(X_i^{i+t-1})). \end{aligned} \quad (6)$$

The limit on the right-hand side denotes a limit along integers in  $N^0$ . Since the alphabet  $A$  is finite we can find  $N^1 \subseteq N^0$  such that

$$\lim_{n \rightarrow \infty, n \in N^1} \frac{1}{n} \sum_{i=1}^n P(X_i = a)$$

exists for each  $a \in A$ . Let this limiting one-dimensional distribution be  $P'_1$ . Continuing this argument we can find  $N^1 \subseteq N^2 \subseteq N^3 \subseteq \dots$  such that the  $k$ -dimensional distributions converge along integers in  $N^k$  to a distribution  $P'_k$ . Let  $\mathbf{Z}$  be a process defined by these stationary distributions. Now from the sets  $N^1 \subseteq N^2 \subseteq N^3 \subseteq \dots$  we can form a set  $N^*$  in the following way: Pick the smallest element  $n_0$  from  $N^0$ . For each  $k$  pick  $n_k \in N^k$  such that  $n_k > n_{k-1}$ . By the construction of  $N^*$  and  $\mathbf{Z}$  it is easy to see that

$$\lim_{n \rightarrow \infty, n \in N^*} \sum_{i=1}^n E(f(X_i, X_{i+1}, \dots)) = E(f(\mathbf{Z}))$$

holds for all real-valued functions  $f$  that depend only on finitely many coordinates. Thus  $\mathbf{Z}$  is in the stationary hull of  $\mathbf{X}$ . Also since  $N^* \subseteq N^0$  we have using (6)

$$\begin{aligned} \limsup_{n \rightarrow \infty} \frac{1}{n} \sum_{i=1}^n E(\phi(X_i^{i+t-1})) \\ = \lim_{n \rightarrow \infty, n \in N^*} \frac{1}{n} \sum_{i=1}^n E(\phi(X_i^{i+t-1})) \\ = E(\phi(Z_1^t)). \end{aligned} \quad \square$$

## REFERENCES

- [1] T. M. Cover and J. A. Thomas, *Elements of Information Theory* (Wiley Series in Telecommunications). New York: Wiley, 1991.
- [2] M. Feder and N. Merhav, "Hierarchical universal coding," *IEEE Trans. Inform. Theory*, vol. 42, no. 5, pp. 1354–1364, Sept. 1996.
- [3] J. C. Kieffer, "Fixed rate encoding of nonstationary sources," *IEEE Trans. Inform. Theory*, vol. IT-33, pp. 651–655, Sept. 1987.
- [4] —, "Finite-state adaptive block to variable-length noiseless coding of a nonstationary information source," *IEEE Trans. Inform. Theory*, vol. IT-35, pp. 1259–1263, Nov. 1989.
- [5] E. Plotnik, M. Weinberger, and J. Ziv, "Upper bounds on the probability of a sequence emitted from a finite-state source and on the redundancy of the Lempel–Ziv data compression algorithm," *IEEE Trans. Inform. Theory*, vol. 35, pp. 1259–1263, Nov. 1989.
- [6] S. Savari, "Redundancy of the Lempel–Ziv incremental parsing rule," *IEEE Trans. Inform. Theory*, vol. 43, pp. 9–21, Jan. 1997.

- [7] S. Verdú and T. S. Han, "A general formula for channel capacity," *IEEE Trans. Inform. Theory*, vol. 40, pp. 1147–1158, July 1994.
- [8] F. M. J. Willems, "The context-tree weighting method: Extensions," in *IEEE Int. Symp. Information Theory*, Trondheim, Norway, 1994.
- [9] —, "Coding for a binary independent piecewise-identically-distributed-source," *IEEE Trans. Inform. Theory*, vol. 42, pp. 2210–2217, Nov. 1996.
- [10] E. Yang and J. C. Kieffer, "Simple universal lossy data compression schemes derived from the Lempel–Ziv algorithm," *IEEE Trans. Inform. Theory*, vol. 42, pp. 239–245, Jan. 1996.
- [11] J. Ziv and A. Lempel, "Compression of individual sequence via variable rate coding," *IEEE Trans. Inform. Theory*, vol. IT-24, pp. 530–536, Sept. 1978.

$C_1$  : 10 11 000 001 010 011  
 $C_2$  : 01 11 001 101 0001 1001  
 $C_3$  : 1 01 001 0001 00001 000001

Fig. 1. Code  $C_1$  is an optimal prefix-free code for the distribution  $(1/6), (1/6), (1/6), (1/6), (1/6), (1/6)$ .  $C_2$  is an optimal *One-ended* prefix-free code for the same distribution.  $C_3$  is an optimal one-ended code for the distribution  $0.9, 0.09, 0.009, 0.0009, 0.00009, 0.000001$ .

## A Dynamic Programming Algorithm for Constructing Optimal "1"-Ended Binary Prefix-Free Codes

Sze-Lok Chan and Mordecai J. Golin, *Member, IEEE*

**Abstract**—The generic *Huffman-Encoding Problem* of finding a minimum cost prefix-free code is almost completely understood. There still exist many variants of this problem which are not as well understood, though. One such variant, requiring that each of the codewords ends with a "1," has recently been introduced in the literature with the best algorithms known for finding such codes running in exponential time. In this correspondence we develop a simple  $O(n^3)$  time algorithm for solving the problem.

**Index Terms**—Dynamic programming, one-ended codes, prefix-free codes.

### I. INTRODUCTION

In this correspondence we discuss the problem of efficiently constructing minimum-cost binary prefix-free codes having the property that each codeword ends with a "1."

We start with a quick review of basic definitions. A *code* is a set of binary words  $C = \{w_1, w_2, \dots, w_n\} \subset \{0, 1\}^*$ . A word  $w = \sigma_{i_1} \sigma_{i_2} \dots \sigma_{i_l}$  is a *prefix* of another word  $w' = \sigma'_{i_1} \sigma'_{i_2} \dots \sigma'_{i_{l'}}$  if  $w$  is the start of  $w'$ . Formally, this occurs if  $l \leq l'$  and, for all  $j \leq l$ ,  $\sigma_{i_j} = \sigma'_{i_j}$ . For example, 00 is a prefix of 00011. Finally, a code is said to be *prefix-free* if for all pairs  $w, w' \in C$ ,  $w$  is not a prefix of  $w'$ .

Let  $P = \{p_1, p_2, p_3, \dots, p_n\}$  be a discrete probability distribution, that is,  $\forall i, 0 \leq p_i \leq 1$  and  $\sum_i p_i = 1$ . The cost of code  $C$  with distribution  $P$  is

$$\text{Cost}(C, P) = \sum_i |w_i| \cdot p_i$$

where  $|w|$  is the length of word  $w$ ;  $\text{Cost}(C, P)$  is, therefore, the average length of a word under probability distribution  $P$ . The *prefix-coding problem* is, given  $P$ , to find a prefix-free code  $C$  that minimizes  $\text{Cost}(C, P)$ . It is well known that such a code can be found in

Manuscript received March 8, 1998; revised October 6, 1999. This work was supported in part by Hong Kong RGC/CRG under Grants HKUST652/95E, 6082/97E, and 6137/98E.

The authors are with the Department of Computer Science, Hong Kong University of Science and Technology, Clear Water Bay, Kowloon, Hong Kong (e-mail: SZELOK@cs.ust.hk; GOLIN@cs.ust.hk).

Communicated by D. Stinson, Associate Editor for Complexity and Cryptography.

Publisher Item Identifier S 0018-9448(00)04283-8.

$O(n \log n)$  time using the greedy *Huffman-Encoding* algorithm, see, e.g., [5] or even  $O(n)$  time if the  $p_i$  are already sorted [6].

In 1990, Berger and Yeung [1] introduced a new variant of this problem. They defined a *feasible* or *1-ended* code to be a prefix-free code in which every word is restricted to end with a "1." Such codes are used, for example, in the design of self-synchronizing codes [3] and testing. Given  $P$ , the problem is to find the minimum-cost 1-ended code. Fig. 1 gives some examples.

In their paper, Berger and Yeung derived properties of such codes, such as the relationship of a min-cost feasible code to the entropy of  $P$ , and then described an algorithm to construct them. Their algorithm works by examining all codes of a particular type, returning the minimum one. They noted that experimental evidence seemed to indicate that their algorithm runs in time exponential in  $n$ . A few years later, Capocelli, De Santis, and Persiano [4] noted that the min-cost code can be shown to belong to a *proper* subset of the code-set examined by Berger and Yeung. They, therefore, proposed a more efficient algorithm that examines only the codes in their subset. Unfortunately, even their restricted subset contains an exponential number of codes<sup>1</sup> so their algorithm also runs in exponential time.

In this correspondence we describe another approach to solving the problem. Instead of enumerating all of the codes of a particular type it uses dynamic programming to find an optimum one in  $O(n^3)$  time.

### II. TREES AND CODES

There is a very well-known standard correspondence between prefix-free codes and binary<sup>2</sup> trees. In this section we quickly discuss its restriction to the 1-ended code problem. This will permit us to reformulate the min-cost feasible code problem as one that finds a min-cost tree. In this new formulation we will require that  $p_1 \geq p_2 \geq \dots \geq p_n \geq 0$  but will no longer require that  $\sum_i p_i = 1$ .

**Definition 1:** Let  $T$  be a binary tree. A leaf  $u \in T$  is a *left leaf* if it is a left child of its parent; it is a *right leaf* if it is a right child of its parent.

The *depth* of a node  $v \in T$ , denoted by  $\text{depth}(v)$ , is the number of edges on the path connecting the root to  $v$ .

We build the correspondence between trees and codes as follows. First let  $T$  be a tree. Label every left edge in  $T$  with a 0 and every right edge with a 1. Associate with a leaf  $v$  the word  $w(v)$  read off by following the path from the root of  $T$  down to  $v$ . Now let  $v_1, v_2, \dots, v_n$  be the set of right leaves of  $T$ . Then  $C(T) = \{w(v_1), w(v_2), \dots, w(v_n)\}$  is the code associated with  $T$ . Note that this code is feasible since all of its words end with a 1. Note also that there can be many trees corresponding to the same feasible code. See Fig. 2 for an example.

<sup>1</sup>The proof of this fact is a straightforward argument that recursively builds an exponentially sized set of codes that belong to the restricted subset. Because of space considerations we do not include it here but the interested reader can find the details in [2].

<sup>2</sup>In this correspondence we use the slightly nonstandard convention that a binary tree is a tree in which every internal node has *one or two* children.

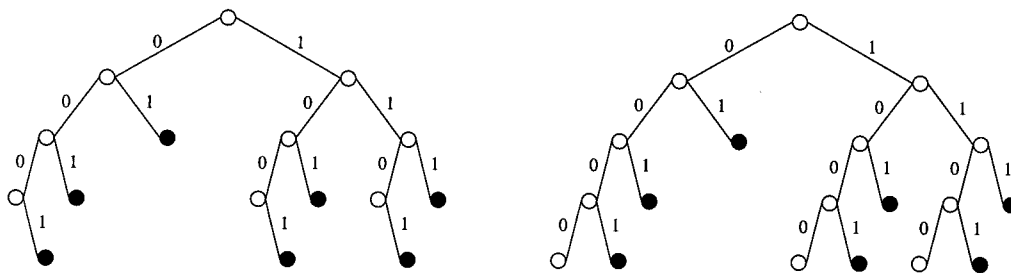


Fig. 2. Two trees with depth 4 having seven right leaves. Note that these two trees both correspond to the code {0001, 001, 01, 1001, 101, 1101, 111}. The left tree is nonfull while the right one is full.

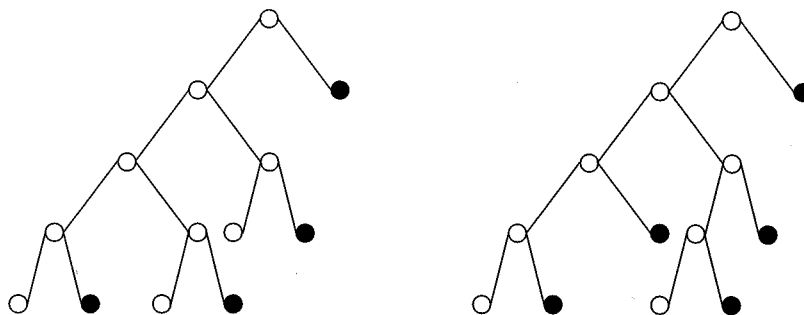


Fig. 3. The left tree is nonfeasible because, at depth 3 it contains both an internal right node and a left leaf. The right tree is feasible.

Now let  $C = \{w_1, w_2, \dots, w_n\}$  be any feasible code. Let  $T(C)$  be the smallest tree that contains all of the paths corresponding to the  $w_i$ . Since  $C$  is prefix-free we have that the right leaves of  $T(C)$  are exactly the nodes corresponding to the words of  $C$ .

Let  $T$  be a tree with  $n$  right leaves labeled  $v_1, v_2, \dots, v_n$ ,  $P = \{p_1, p_2, p_3, \dots, p_n\}$  and define

$$\text{Cost}(C, T) = \sum_i \text{depth}(v_i) \cdot p_i.$$

This is the *weighted-external path length* of  $T$  restricted to right leaves (external nodes). In all that follows  $P = \{p_1, p_2, p_3, \dots, p_n\}$  will be considered fixed and the dependence of quantities such as  $\text{Cost}(C, T)$  on  $P$  will be implicitly assumed.

Now suppose that  $T$  corresponds to some code  $C$  and  $v \in T$  is a right leaf corresponding to  $w \in C$ ; by definition  $\text{depth}(v) = |w|$ . Thus

$$\begin{aligned} \text{Cost}(C, T) &= \sum_i \text{depth}(v_i) \cdot p_i \\ &= \sum_i |w_i| \cdot p_i = \text{Cost}(C, P). \end{aligned}$$

Since every feasible code corresponds to some tree(s) and every tree corresponds to one feasible code this last equation tells us that we can find a min-cost code by constructing a min-code tree and returning the feasible code corresponding to it.

There is a technical problem that we need to address before proceeding. It is that our definition of cost formally requires that the right leaves of  $T$  be labeled  $1, 2, \dots, n$ . Different labelings of the right leaves could lead to different costs. We note though that, for a particular tree, the minimum cost over all labelings is *always* achieved when the highest node in the tree is assigned the largest weight  $p_1$ , the second highest node the second highest weight  $p_2$ , and in general the  $i$ th highest node (with height ties broken arbitrarily) the  $i$ th weight  $p_i$ . Since we are interested in finding a minimum cost tree we will always assume that the labeling used for any particular tree is the canonical labeling with  $v_i$  being the  $i$ th highest node. For example, if

the weights are 7, 6, 5, 4, 3, 2, 1, then the trees in Fig. 2 have cost  $2 \cdot 7 + 3 \cdot (6 + 5 + 4) + 4 \cdot (3 + 2 + 1) = 83$ .

The Optimal Feasible Coding Problem is now seen to be equivalent to the following tree problem.

**Definition 2:** The *Optimal Tree Problem* Given  $p_1 \geq p_2 \geq \dots \geq p_n$  find a tree  $\bar{T}$  with  $n$  right leaves with minimum cost over all trees with  $n$  right leaves, i.e.,

$$\text{cost}(\bar{T}) = \min \{ \text{cost}(T) : T \text{ has } n \text{ right leaves} \}.$$

We end this section by pointing out that there must be an optimal tree with a very specific structure.

**Definition 3:** A tree  $T$  is *full* if every internal node in  $T$  has two children.

A tree  $T$  is *feasible* if  $T$  is full and it also has the additional property: if  $u \in T$  is a right node and internal then *all* left nodes  $v \in T$  with  $\text{depth}(v) = \text{depth}(u)$  are also internal.

Fig. 2 illustrates a nonfull tree and a full one. Fig. 3 illustrates a nonfeasible tree and a feasible one.

**Lemma 1:** For every probability distribution

$$P = \{p_1, p_2, p_3, \dots, p_n\}$$

there exists an optimal tree  $T$  that is feasible.

The proof of the lemma is straightforward but technical. To avoid breaking the flow of the correspondence it has, therefore, been relegated to the Appendix.

### III. TRUNCATED TREES AND SIGNATURES

Our approach will be a modification of one developed in [7]. The problem considered there was to build a min-cost *lopsided tree* (tree in which edges have different length). The solution was to build trees from the top, root node, down, accumulating the cost as levels were added. We will follow the same approach in this correspondence to

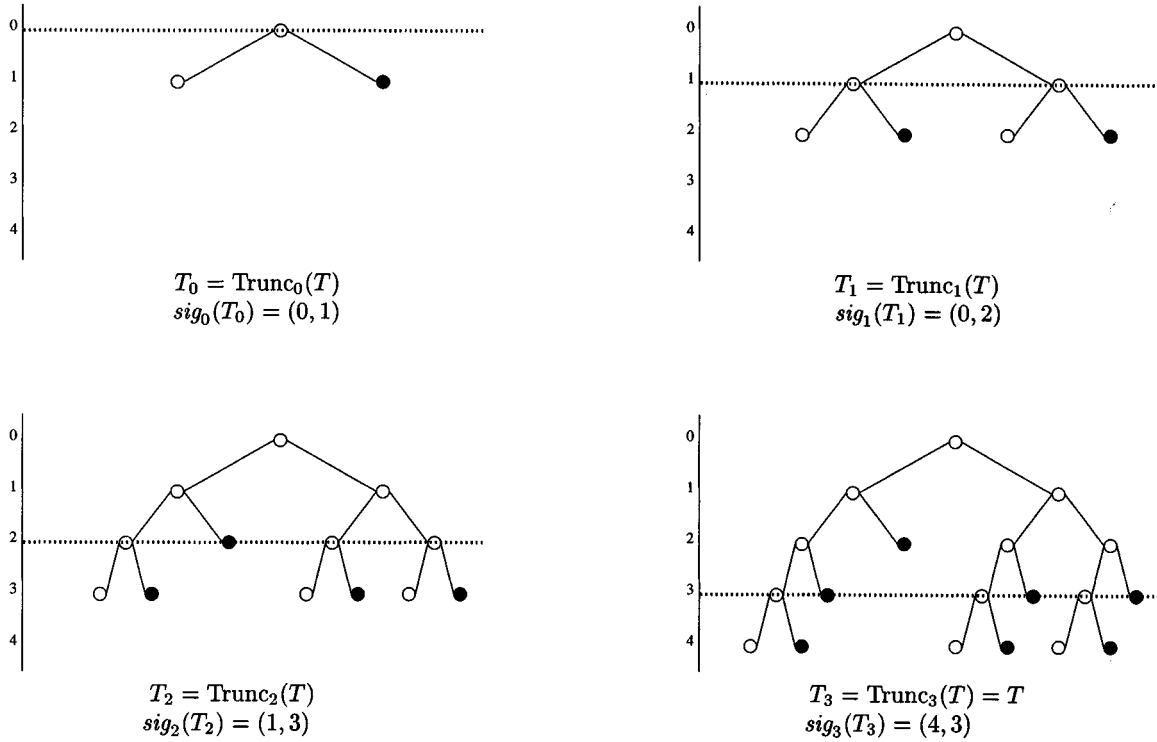


Fig. 4. The trees  $T_0, T_1, T_2, T_3$  are the truncations of the right tree in Fig. 2 which we will call  $T$ . Each  $T_i$  is an  $i$ -level tree for that value of  $i$  and the dotted horizontal line across each tree is the truncation level. Note that  $T = \text{Trunc}_4(T)$  with  $\text{sig}_4(T_4) = (7, 0)$ .

construct min-cost feasible trees. Since Lemma 1 guarantees that trees thus constructed will be min-cost trees among *all* trees we will have solved the problem.

To use this approach we need to define the following.

*Definition 4:* Let  $T$  be a tree and  $i$  a nonnegative integer. The  $i$ th-level truncation of  $T$  is the tree  $\text{Trunc}_i(T)$  containing all nodes in  $T$  of depth at most  $i + 1$

$$\text{Trunc}_i(T) = \{u \in T : \text{depth}(u) \leq i + 1\}.$$

A tree  $T$  is an  $i$ -level tree if all the internal nodes  $v \in T$  satisfy  $\text{depth}(v) \leq i$ .

See Fig. 4 for examples. We note that  $\text{Trunc}_i(T)$  is always an  $i$ -level tree and that truncation preserves feasibility, i.e., if  $T$  is a feasible tree then  $\text{Trunc}_i(T)$  is also a feasible tree. We also note that if  $T$  has depth  $d$  then  $\forall i \geq d - 1, \text{Trunc}_i(T) = T$ .

The dynamic programming algorithm will strongly use the idea of subproblem optimality, i.e., if a feasible tree  $T$  is optimal then all of its  $i$ -level truncations  $\text{Trunc}_i(T)$  are also optimal. In order for this observation to make sense we must define what it means for a feasible  $i$ -level tree (that might have fewer than  $n$  right leaves) to be optimal. That is, we must define a cost function on  $i$ -level trees.

*Definition 5:* Let  $T$  be a feasible  $i$ -level tree. The  $i$ -level signature of  $T$  is an ordered pair

$$\text{sig}_i(T) = (m, b)$$

in which

$$m = |\{v \in T : v \text{ is a right leaf, } \text{depth}(v) \leq i\}|$$

is the number of right leaves in  $T$  with depth at most  $i$  and

$$b = |\{v \in T : v \text{ is a right leaf, } \text{depth}(v) = i + 1\}|$$

is the number of right leaves in  $T$  at level  $i + 1$  (bottom level). Note that there are  $2b$  (left and right) leaves at level  $i + 1$ .

Now let  $T$  be an  $i$ -level tree with  $\text{sig}_i(T) = (m, b)$  with  $m \leq n$ . The  $i$ -level *partial cost* of  $T$  is

$$\text{Cost}_i(T) = \sum_{t=1}^m \text{depth}(v_t) \cdot p_t + i \cdot \sum_{t=m+1}^n p_t \quad (1)$$

where  $v_1, \dots, v_m$  are the  $m$  highest right leaves of  $T$  ordered by depth, e.g., those with  $\text{depth} \leq i$ .

Note that  $\text{Cost}_i(T)$  is not only dependent upon  $T$  but also upon  $i$ . For example, the right tree  $T$  in Fig. 2 is both a three-level and a four-level tree;  $\text{sig}_3(T) = (4, 3)$  and  $\text{sig}_4(T) = (7, 0)$ . Its associated  $i$ -level costs are

$$\text{Cost}_3(T) = 2p_1 + 3(p_2 + p_3 + p_4) + 3(p_5 + p_6 + p_7)$$

$$\text{Cost}_4(T) = 2p_1 + 3(p_2 + p_3 + p_4) + 4(p_5 + p_6 + p_7)$$

which are obviously not the same.

We can now define what it means for a tree with fewer than  $n$  right leaves to be optimal.

*Definition 6:* Let  $(m, b)$  be a valid signature, i.e.,  $m, b \geq 0$ . Set  $\text{OPT}[m, b]$  to be the minimum cost over all  $i$  and all feasible  $i$ -level trees  $T$  with signature  $(m, b)$ . More precisely

$$\text{OPT}[m, b] = \min\{\text{Cost}_i(T) : \exists i, T,$$

$$T \text{ is a feasible } i\text{-level tree with } \text{sig}_i(T) = (m, b)\}$$

A tree  $T$  is *min-cost* or *optimal* if, for some  $i$ , it is an  $i$ -level tree,  $\text{sig}_i(T) = (m, b)$  and  $\text{Cost}_i(T) = \text{OPT}[m, b]$ .

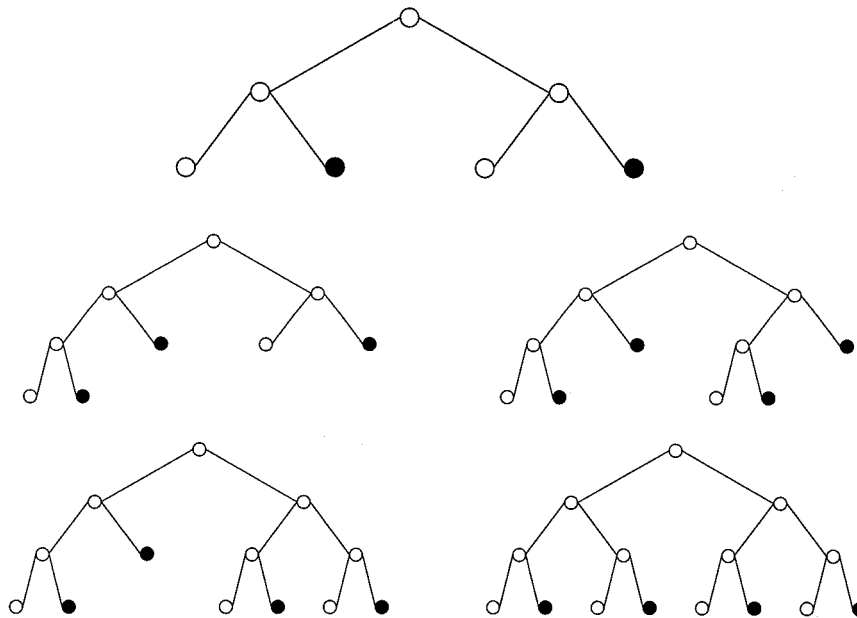


Fig. 5. The tree in the top row is our original tree  $T$  which is a one-level tree with  $\text{sig}_1(T) = (0, 2)$ . The next two rows are the four possible expansions  $\text{Expand}(T, 1)$ ,  $\text{Expand}(T, 2)$ ,  $\text{Expand}(T, 3)$ , and  $\text{Expand}(T, 4)$ . We do not draw  $\text{Expand}(T, 0)$  which is simply  $T$ .

Note that if  $T$  is a feasible tree with  $n$  right leaves and depth  $d \leq i$  then  $\text{sig}_i(T) = (n, 0)$  so, if  $T'$  is an optimal feasible tree (with  $n$  right leaves), then

$$\text{OPT}[n, 0] = \text{Cost}_i(T') = \sum_{t=1}^n \text{depth}(v_t) \cdot p_t.$$

Thus  $\text{OPT}[n, 0]$  is exactly the cost of the optimal tree that we are trying to calculate. We will calculate its value by using a dynamic programming approach to fill in the  $\text{OPT}$  table. Backtracking the dynamic programming will permit us to construct  $T'$ .

Before continuing we briefly digress to explain why we defined  $\text{OPT}[m, b]$  to be the minimum cost only among *feasible* trees and not among all trees.<sup>3</sup> The reason is that we will be building optimal trees level-by-level. Since Lemma 1 tells us that our final result is a feasible tree and we know that all truncations of feasible trees are feasible trees our construction will work by building feasible trees level by level, always storing the min-cost ones.

Now suppose that  $T$  is an  $i$ -level tree with  $\text{sig}_i(T) = (m, b)$ . What feasible  $(i + 1)$ -level trees can  $T$  be grown into? The only way to grow a feasible tree is by making some of the  $2b$  nodes on level  $i + 1$  internal and making the remainder of the nodes leaves. From Lemma 1 we know that all of the left nodes must be made internal before any of the right ones are. We therefore define an *Expansion* operator as follows.

**Definition 7:** Let  $T$  be an  $i$ -level tree with  $\text{sig}_i(T) = (m, b)$ . Let  $0 \leq q \leq 2b$ . The  $q$ th expansion of  $T$  is the tree

$$T' = \text{Expand}(T, q)$$

constructed by making  $q$  of the leaves at level  $i + 1$  (bottom level) of  $T$  internal nodes as follows:

- if  $q \leq b$ , make  $q$  left nodes at level  $i + 1$  internal.
- if  $q > b$ , make all  $b$  left nodes and  $q - b$  right nodes at level  $i + 1$  internal.

<sup>3</sup>The algorithm to be presented actually remains correct even if we optimized over *all* trees and not just all feasible ones. The reason for the restriction to feasible trees is that it makes the result both easier to understand and prove.

In Fig. 5 we see a tree and all of its expansions.

Once  $q$  is fixed both  $\text{Cost}_{i+1}(T')$ , the number of nodes at level  $i + 2$ , and the signature  $\text{sig}_{i+1}(T')$  of  $T'$  can be found.

**Lemma 2:** Suppose  $T$  is an  $i$ -level tree with  $\text{sig}_i(T) = (m, b)$ . Let  $T' = \text{Expand}(T, q)$  be its  $q$ th expansion. Then  $T'$  is an  $i + 1$ -level tree with

$$\text{Cost}_{i+1}(T') = \text{Cost}_i(T) + \sum_{m < t \leq n} p_t$$

and

- if  $0 \leq q \leq b$  then  $\text{sig}_{i+1}(T') = (m + b, q)$ ,
- if  $b + 1 \leq q \leq 2b$ , then  $\text{sig}_{i+1}(T') = (m + 2b - q, q)$ .

*Proof:* Let  $(m', b') = \text{sig}_{i+1}(T')$ . Since  $T'$  has exactly  $m' - m$  right leaves on level  $i + 1$  we find

$$\begin{aligned} \text{Cost}_{i+1}(T') &= \sum_{t=1}^{m'} \text{depth}(v_t) \cdot p_t + (i + 1) \cdot \sum_{t=m'+1}^n p_t \\ &= \sum_{t=1}^m \text{depth}(v_t) \cdot p_t + (i + 1) \cdot \sum_{t=m+1}^{m'} p_t \\ &\quad + (i + 1) \cdot \sum_{t=m'+1}^n p_t \\ &= \text{Cost}_i(T) + \sum_{m < t \leq n} p_t. \end{aligned}$$

The proof of the second part of the lemma follows directly from the definition of the *Expand* operator.  $\square$

This lemma tells us that to calculate the extra cost added by a level- $i$  expansion of  $T$  and the signature of the new expanded tree it is not necessary to know  $T$  or  $i$  but only  $\text{sig}_i(T)$ . We can, therefore, define a recurrence relationship for calculating  $\text{OPT}[\cdot, \cdot]$ . In what follows  $\mathcal{M}(m', b')$  is exactly the set of signatures  $(m, b)$  that have some expansion with signature  $(m', b')$ .

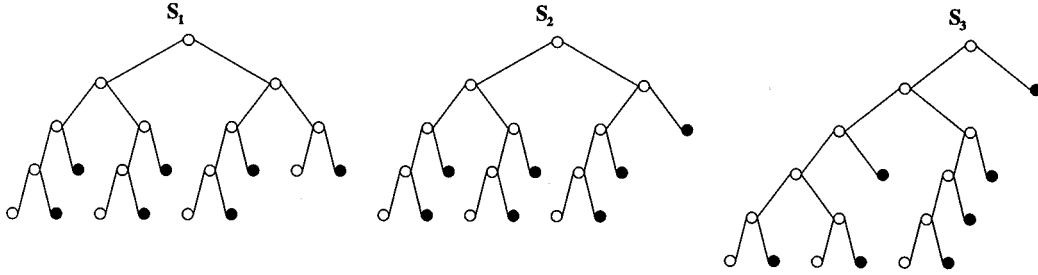


Fig. 6. These three trees have the property that  $\text{sig}_3(S_1) = \text{sig}_3(S_2) = \text{sig}_4(S_3) = (4, 3)$ . In fact, these three trees are the only way to realize the signature  $(4, 3)$  and thus  $\mathcal{M}(4, 3) = \{(0, 4), (1, 3), (3, 2)\}$ .

*Lemma 3:* Set

$$\mathcal{M}(m', b') = \left\{ (m, b) : (m, b) \neq (m', b') \text{ and} \right. \\ \left. \begin{array}{l} \exists q \text{ s.t. } 0 \leq q \leq b \text{ and } (m', b') = (m + b, q) \\ \text{or } \exists q \text{ s.t. } b + 1 \leq q \leq 2b \\ \text{and } (m', b') = (m + 2b - q, q). \end{array} \right\}.$$

Then

$$\text{OPT}[0, 1] = 0 \quad (2)$$

and, for  $(m', b') \neq (0, 1)$

$$\text{OPT}[m', b'] = \min_{(m, b) \in \mathcal{M}(m', b')} \left\{ \text{OPT}[m, b] + \sum_{m < t \leq n} p_t \right\}. \quad (3)$$

*Proof:* Fig. 6 illustrates an example of  $\mathcal{M}(m', b')$ .

To prove (2) we note that the *only* feasible tree with signature  $(0, 1)$  is the 0-level tree consisting of the root and its two children and this tree has 0-level cost 0.

To prove (3) first suppose that  $\text{OPT}[m', b']$  is realized by an  $i + 1$ -level tree  $T'$  with  $\text{sig}_{i+1}(T') = (m', b')$  and

$$\text{Cost}_{i+1}(T') = \text{OPT}[m', b'].$$

Now set  $T = \text{Trunc}_i(T')$  and let  $q$  be the number of internal nodes on level  $i + 1$  of  $T$ . Also set  $(m, b) = \text{sig}_i(T)$ . Then, by the definition of the *Trunc* and *Expand* operators we have that  $T' = \text{Expand}(T, q)$ . Thus from Lemma 2 and the definition of  $\text{OPT}[\cdot]$

$$\begin{aligned} \text{OPT}[m', b'] &= \text{Cost}_{i+1}(T') \\ &= \text{Cost}_i(T) + \sum_{m < t \leq n} p_t \\ &\geq \text{OPT}[m, b] + \sum_{m < t \leq n} p_t \\ &\geq \min_{(m, b) \in \mathcal{M}(m', b')} \left\{ \text{OPT}[m, b] + \sum_{m < t \leq n} p_t \right\}. \end{aligned}$$

To see the other direction let  $(m, b) \in \mathcal{M}(m', b')$  and set  $T$  to be such that

$$\text{sig}_i(T) = (m, b) \quad \text{and} \quad \text{OPT}[m, b] = \text{Cost}_i(T).$$

Let  $q$  be such that

$$\text{sig}_{i+1} \text{Expand}(T, q) = (m', b').$$

Such a  $q$  must exist by the definition of  $\mathcal{M}(m', b')$ . Let  $T' = \text{Expand}(T, q)$ . Then from Lemma 2  $\text{sig}_{i+1}(T') = (m', b')$  and

$$\begin{aligned} \text{Cost}_{i+1}(T') &= \text{Cost}_i(T) + \sum_{m < t \leq n} p_t \\ &= \text{OPT}[m, b] + \sum_{m < t \leq n} p_t. \end{aligned}$$

Since this is true for every  $(m, b) \in \mathcal{M}(m', b')$  we thus find that

$$\text{OPT}[m', b'] \leq \min_{(m, b) \in \mathcal{M}(m', b')} \left\{ \text{OPT}[m, b] + \sum_{m < t \leq n} p_t \right\}$$

completing the proof.  $\square$

#### IV. THE ALGORITHM

Using Lemma 3 we can directly design an algorithm for calculating the  $\text{OPT}[\cdot]$  values and constructing an optimal tree. Code for the algorithm is given in Fig. 7 and a worked example is shown in Table I and Fig. 8. In the algorithm, the entry  $Q[m', b']$  stores the pair  $(m, b) \in \mathcal{M}(m', b')$  such that

$$\text{OPT}[m', b'] = \text{OPT}[m, b] + \sum_{m < t \leq n} p_t.$$

We will now prove the correctness of the algorithm and then show that it runs in  $O(n^3)$  time. We start by recalling the definition of a *lexicographical ordering* on pairs.

*Definition 8:* Let  $(m, b), (m', b')$  be given. Then  $(m, b)$  is *lexicographically smaller* than  $(m', b')$

$$(m, b) \prec (m', b')$$

if and only if

$$m < m' \quad \text{or} \quad m = m' \quad \text{and} \quad b < b'.$$

It is now easy to see that

*Lemma 4:* Let  $(m, b), (m', b')$  be signatures such that  $(m, b) \in \mathcal{M}(m', b')$ . Then  $(m, b) \prec (m', b')$ .

*Proof:* This follows from the definition of  $\mathcal{M}(m', b')$ . We first point out that  $b \neq 0$  because it is impossible for  $(m, b) \in \mathcal{M}(m', b')$  if  $b = 0$ . Thus  $b \geq 1$ .

There are two cases. In the first case  $\exists q \leq b$  such that  $(m', b') = (m + b, q)$ . In this case, since  $b \geq 1$  we have that  $m < m'$  so  $(m, b) \prec (m', b')$ .

In the second case,  $\exists q$  with  $b + 1 \leq q \leq 2b$  such that  $(m', b') = (m + 2b - q, q)$ . In this case, if  $q < 2b$  then again  $m < m'$  so  $m < m'$

**The Algorithm**

**Initialize the  $OPT[,]$  table**

$\forall m, n, 0 \leq m \leq n, 1 \leq b \leq n - m,$   
 Set  $OPT[m, b] := \infty;$   
 $\forall m, 0 \leq m \leq n$  Set  $P_m := \sum_{m < t \leq n} p_t;$   
 $OPT[0, 1] := 0; OPT[n, 0] := \infty;$

**Calculate  $OPT[,]$  values**

for  $m := 0$  to  $n$   
 for  $b := 1$  to  $n - m$   
   **Process the pair  $(m, b)$**   
   for  $q := 0$  to  $b$   
      $X := \min(OPT[m, b] + P_m, OPT[m + b, q])$   
     if  $X < OPT[m + b, q]$   
        $\{OPT[m + b, q] := X; Q[m + b, q] := (m, b); \}$   
   for  $q := b + 1$  to  $2b$   
      $X := \min(OPT[m, b] + P_m, OPT[m + 2b - q, q])$   
     if  $X < OPT[m + 2b - q, q]$   
        $\{OPT[m + 2b - q, q] := X;$   
        $Q[m + 2b - q, q] := (m, b); \}$

**Backtracking and outputting tree**

$m := n; b := 0;$   
 repeat  $\{(m, b) = Q[m, b]; \text{print } q; \}$   
 until  $(m, b) = (1, 0)$

Fig. 7. The dynamic programming algorithm plus backtracking. The algorithm will output the number of right leaves on every level of some optimal tree.

TABLE I  
 VALUES FOR  $n = 7$  WITH WEIGHTS 7, 6, 5, 4, 3, 2, 1

	$b = 0$	1	2	3	4	5	6	7
$m = 0$	$\infty$	0	28 (0, 1)	$\infty$	56 (0, 2)	$\infty$	$\infty$	$\infty$
1	28 (0, 1)	28 (0, 1)	49 (1, 1)	56 (0, 2)	70 (1, 2)	$\infty$	77 (1, 3)	
2	49 (1, 1)	49 (1, 1)	56 (0, 2)	70 (1, 2)	71 (2, 2)	77 (1, 3)		
3	64 (2, 1)	64 (2, 1)	70 (1, 2)	71 (2, 2)	77 (1, 3)			
4	71 (2, 2)	71 (2, 2)	71 (2, 2)	77 (1, 3)				
5	77 (4, 1)	77 (4, 1)	80 (3, 2)					
6	77 (4, 2)	77 (4, 2)						
7	78 (6, 1)							

so  $(m, b) \prec (m', b')$ . If  $q = 2b$  then  $m' = m + 2b - q = m$  but then  $b < 2b = b'$  so we still have  $(m, b) \prec (m', b')$ .  $\square$

We now show that the algorithm correctly fills in the  $OPT(m, b)$  values (using a standard dynamic programming correctness proof).<sup>4</sup>

<sup>4</sup>We quickly sketch a second way of proving this, one similar to that developed in [7]. Essentially, one can create a weighted graph  $G = (V, E)$  in which  $V$  is the set of all signatures and  $((m, b), (m', b')) \in E$  if  $(m, b) \in \mathcal{M}(m', b')$ . The weight of this edge is defined to be  $\sum_{m < t \leq n} p_t$ . Then  $OPT[m, b]$  can be shown to be equal to the cost of the *minimum-cost path* connecting  $(0, 1)$  to  $(m, b)$  in  $G$ . The graph  $G$  can be shown to be acyclic and the algorithm presented in Fig. 8 is exactly the code for finding shortest paths in a directed acyclic graph, see, e.g., [8].

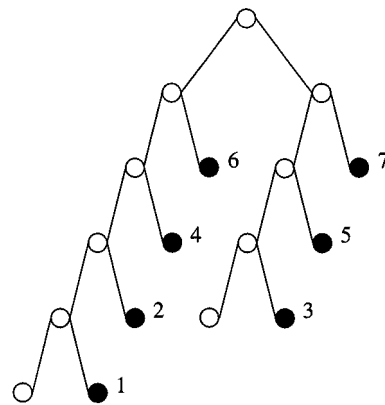


Fig. 8. An optimal tree for  $n = 7$  with weights 7, 6, 5, 4, 3, 2, 1. This tree is derived from Table I.

Note that if  $(m, b) \in \mathcal{M}(m', b')$  then either  $(m', b') = (m + b, q)$  or  $(m', b') = (m + 2b - q, q)$ . In both of these cases we find that  $m + b \leq m' + b'$ . Now note that the set of signatures processed by the algorithm is exactly

$$\begin{aligned} \mathcal{M} &= \{(m, b) : 0 \leq m \leq n, 0 \leq b \leq n - m\} \\ &= \{(m, b) : 0 \leq m, b, m + b \leq n\}. \end{aligned}$$

Thus if  $(m', b') \in \mathcal{M}$  and  $(m, b) \in \mathcal{M}(m', b')$  then  $m + b \leq m' + b' \leq n$  so  $(m, b) \in \mathcal{M}$ . Therefore,  $\mathcal{M}(m', b') \subseteq \mathcal{M}$ .

Next note that the algorithm actually processes the signatures in  $\mathcal{M}$  in *lexicographical order* so, from Lemma 4, *all* signatures in  $\mathcal{M}(m', b')$  are processed before  $(m', b')$  and no such signatures are processed after it.

Correctness now follows by induction on  $(m', b')$ , the induction order being the lexicographic order. The induction hypothesis is that, *at the time immediately preceding the processing of  $(m', b')$  the value of  $OPT[m, b]$  will already have been correctly set.*

The first signature processed is  $(0, 1)$  and since  $OPT[0, 1]$  is originally set to 0 and never changed afterwards, the statement is correct for  $(0, 1)$ . Now suppose all signatures preceding  $(m', b')$  in the lexicographic order have already been processed and it is now the turn of  $(m', b')$ . In particular, this implies that all  $(m, b)$  with  $(m, b) \in \mathcal{M}(m', b')$  have already been processed. By the induction hypothesis, at the time such an  $(m, b)$  was processed  $OPT[m, b]$  was already correctly set. Thus the statement executed at the time of processing  $(m, b)$  was equivalent to

$$OPT[m', b'] = \min\{OPT[m', b'], OPT[m, b] + P_m\}.$$

Since this is done for all  $(m, b) \in \mathcal{M}(m', b')$  but no other  $(m, b)$ 's, the value stored in  $OPT[m', b']$  is exactly

$$\min_{(m, b) \in \mathcal{M}(m', b')} \left\{ OPT[m, b] + \sum_{m < t \leq n} p_t \right\},$$

completing the proof that the  $OPT[,]$  table is filled in correctly.

While filling in the table, the algorithm also keeps track of where the optima came from by storing the appropriate  $Q[m', b'] = (m, b)$  such that

$$OPT[m', b'] = OPT[m, b] + \sum_{m < t \leq n} p_t.$$

After filling in the table, the algorithm then uses the  $Q[,]$  table to backtrack and print out the number of right nodes on every level of the

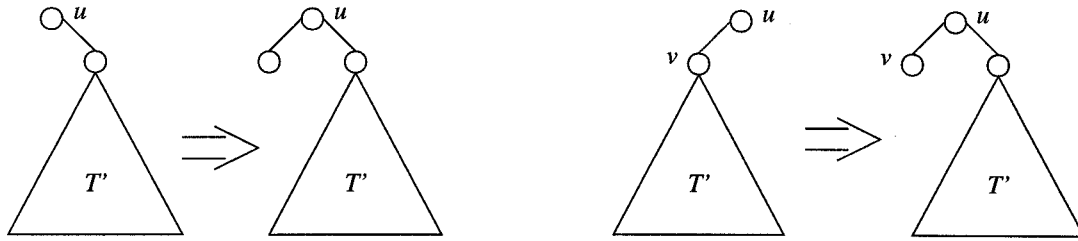


Fig. 9. Cases I and II in the proof.

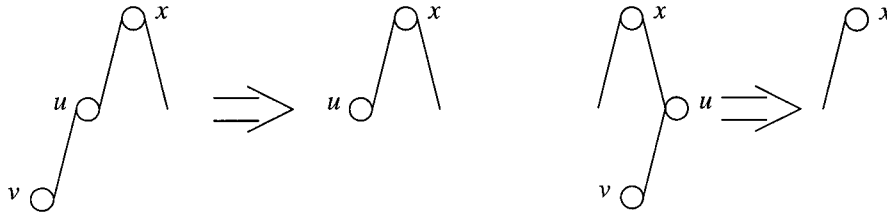


Fig. 10. Cases III and IV in the proof.

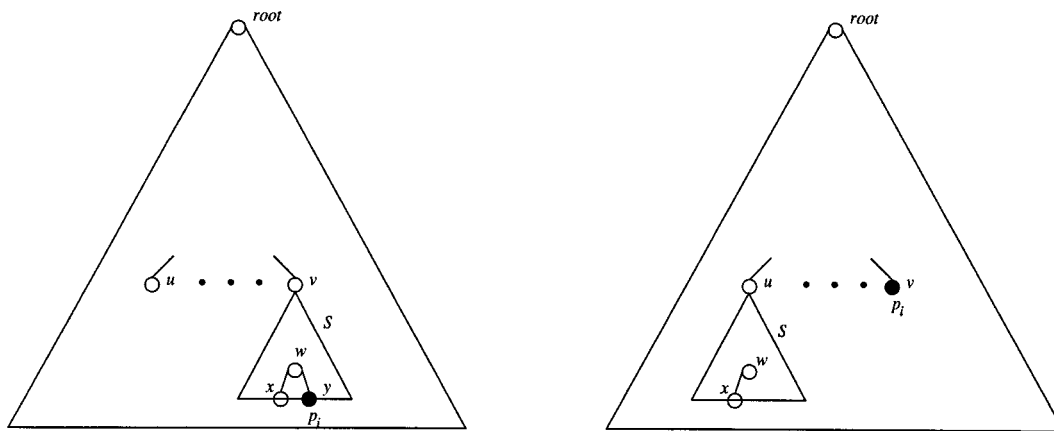


Fig. 11. Illustration of why some optimal tree must be feasible.

optimal tree. Since the tree is full with every right node being matched by some left node this gives the full tree and we are done.

Finally, we note that for each of the  $O(n^2)$  signatures  $(m, b)$  generated, the algorithm does  $O(n)$  work (the two **for** loops over  $q$ ). Thus the algorithm runs in  $O(n^3)$  total time.

Table I contains a worked example for  $n = 7$  with weights 7, 6, 5, 4, 3, 2, 1. The top element in each entry is  $\text{OPT}[m, b]$  and the bottom one is  $Q[m, b]$ . The  $\infty$  entries are signatures that are unrealizable by any feasible tree. The boldface entries are the ones that correspond to the optimal tree found by the backtracking section of Fig. 7. Reading them off we find that the number of right nodes on the levels of the optimal tree are, starting from the top level and working down, 1, 2, 2, 2, 1. The tree itself can be seen in Fig. 8. Note that this tree has (optimal) cost 78 as compared to the trees in Fig. 2. These have cost 83 for the same weights

### V. CONCLUSION

In this correspondence we have shown that it is possible to calculate optimal one-ended binary prefix-free codes in  $O(n^3)$  time improving upon the old exponential time algorithms. Our approach used

dynamic programming on an appropriate subproblem space. The main open question is whether it is possible to improve the algorithm, perhaps even to  $O(n)$  time, matching the linear time used by the standard Huffman-encoding algorithm.

### APPENDIX

In this section we prove Lemma 1, that there is always a feasible optimal tree.

*Proof (of Lemma 1):* We first show that there exists an optimal full tree. If  $T$  is a tree and  $u \in T$  an internal node we will call  $u$  *bad* if it has only one child. If a tree has no bad nodes it is a full tree.

Let  $B$  be the minimal number of bad nodes an optimal tree can have. If  $B = 0$  there is a full optimal tree and we are done. Otherwise, let  $T$  be an optimal tree with  $B$  bad nodes and the fewest total number of nodes among all optimal trees with  $B$  bad nodes. We will show a contradiction by building a new optimal tree with fewer bad nodes or the same number of bad nodes and fewer total nodes.

Let  $u$  be a highest bad node in  $T$ . Note that  $u$  cannot be the root because if the root were bad we could simply erase it; its (only) child then becomes the root of the new tree and, since the depth of every

leaf has been decreased by 1, this new tree is cheaper than the old one, contradicting optimality.

In what follows refer to Figs. 9 and 10 for illustration as we do a case-by-case analysis.

(Case I) If  $u$  had a right child but no left one we could simply add its left child to get a new tree with the same cost but fewer bad nodes, contradicting the definition of  $T$ . Thus  $u$  must have a left child  $v$  but no right child. There are two cases.

(Case II) If  $v$  is the root of some tree  $T'$  then we could move  $T'$  to be rooted at the right child of  $u$  and leave  $v$  a leaf. The new resulting tree has the same cost but fewer bad nodes, again leading to a contradiction.

Otherwise,  $v$  is itself a leaf. Let  $x$  be the parent of  $u$ .

(Case III) If  $u$  is a left child of  $x$  then we simply remove  $v$ , leaving  $u$  as a left leaf. The cost of the resulting tree is the same as before but it has one fewer bad node. Again a contradiction.

(Case IV) Otherwise,  $u$  is the right child of  $x$  and removing  $u$  could add a new right child to the tree, possibly even raising its cost. Therefore, in this case we remove *both*  $u$  and  $v$ . Since  $x$  was not bad before (because it is higher than  $u$ ) removing  $u$  does not add a new right leaf to the tree so the cost of the resulting tree remains the same. Since  $x$  has now become bad the new tree still has  $B$  bad nodes but it has fewer total nodes than  $T$ , again causing a contradiction.

We have just seen that there exists some optimal full tree  $T$ . We now prove that  $T$  is feasible. See Fig. 11 for illustration.

Suppose  $T$  is not feasible. Then there exists some right internal node  $v \in T$  and left leaf  $u \in T$  such that  $\text{depth}(v) = \text{depth}(u)$ . Let  $S$  be the subtree rooted at  $v$ ,  $y$  the deepest right node  $y \in S$ , and  $x$  the left sibling of  $y$  ( $x$  and  $y$  must exist because  $T$  is full). Also suppose that probability  $p_i$  is assigned to  $y$ . Now detach  $S$  from  $v$  and attach it to  $u$ , erase  $y$  and assign  $p_i$  to node  $v$ . Denote the new tree thus created by  $T'$ . Since the only probability whose assigned right leaf has changed is  $p_i$  we find that

$$\text{Cost}(T') = \text{Cost}(T) + (\text{depth}(v) - \text{depth}(y))p_i.$$

But  $\text{depth}(v) < \text{depth}(y)$  so  $\text{Cost}(T') < \text{Cost}(T)$  contradicting optimality of  $T$ . Thus  $T$  must be feasible.  $\square$

## REFERENCES

- [1] T. Berger and R. W. Yeung, "Optimum "1"-ended binary prefix codes," *IEEE Trans. Inform. Theory*, vol. 36, pp. 1435–1441, Nov. 1990.
- [2] C. Szelok, "Variations of prefix free codes," M.Phil. thesis, Dept. Comput. Sci., Hong Kong Univ. Sci. Technol., Dec. 1997.
- [3] R. M. Capocelli, A. D. Santis, L. Gargano, and U. Vaccaro, "On the construction of statistically synchronizable codes," *IEEE Trans. Inform. Theory*, vol. 38, pp. 407–414, Mar. 1992.
- [4] R. M. Capocelli, A. D. Santis, and G. Persiano, "Binary prefix codes ending in a "1"," *IEEE Trans. Inform. Theory*, vol. 40, pp. 1296–1302, July 1994.
- [5] T. H. Cormen, C. E. Leiserson, and R. L. Rivest, *Introduction to Algorithms*. Boston, MA: MIT Press, 1993.
- [6] S. Even, *Graph Algorithms*. Rockville, MD: Comput. Sci. Press, 1979.
- [7] M. Golin and G. Rote, "A dynamic programming algorithm for constructing optimal prefix-free codes for unequal letter costs," *IEEE Trans. Inform. Theory*, vol. 44, pp. 1770–1781, Sept. 1998.
- [8] K. Mehlhorn, *Data Structures and Algorithms 2: Graph Algorithms and NP-Completeness*. Berlin, Germany: Springer-Verlag, 1984.

## A Quantum Analog of Huffman Coding

Samuel L. Braunstein, Christopher A. Fuchs, Daniel Gottesman, and Hoi-Kwong Lo

**Abstract**—We analyze a generalization of Huffman coding to the quantum case. In particular, we notice various difficulties in using instantaneous codes for quantum communication. Nevertheless, for the storage of quantum information, we have succeeded in constructing a Huffman-coding-inspired quantum scheme. The number of computational steps in the encoding and decoding processes of  $N$  quantum signals can be made to be of polylogarithmic depth by a massively parallel implementation of a quantum gate array. This is to be compared with the  $O(N^3)$  computational steps required in the sequential implementation by Cleve and DiVincenzo of the well-known quantum noiseless block-coding scheme of Schumacher. We also show that  $O(N^2(\log N)^2)$  sequential computational steps are needed for the communication of quantum information using another Huffman-coding-inspired scheme where the sender must disentangle her encoding device before the receiver can perform any measurements on his signals.

**Index Terms**—Data compression, Huffman coding, instantaneous codes, quantum coding, quantum information, variable-length codes.

### I. INTRODUCTION

There has been much recent interest in the subject of quantum information processing. Quantum information is a natural generalization of classical information. It is based on quantum mechanics, a well-tested scientific theory in real experiments. This correspondence concerns quantum information.

The goal of this correspondence is to find a quantum source coding scheme analogous to Huffman coding in the classical source coding theory [3]. Let us recapitulate the result of classical theory. Consider the simple example of a memoryless source that emits a sequence of independent and identically distributed signals each of which is chosen from a list  $w_1, w_2, \dots, w_n$  with probabilities  $p_1, p_2, \dots, p_n$ . The task of source coding is to store such signals with a minimal amount of resources. In classical information theory, resources are measured in bits. A standard coding scheme to use is the optimally efficient Huffman coding algorithm, which is a well-known lossless coding scheme for data compression.

Apart from being highly efficient, it has the advantage of being instantaneous, i.e., unlike block coding schemes, the encoding and decoding of each signal can be done immediately. Note also that code-words of variable lengths are used to achieve efficiency. As we will see

Manuscript received May 7, 1999; revised January 11, 2000. This work was supported in part by EPSRC under Grants GR/L91344 and GR/L80676, by the Lee A. DuBridge Fellowship, by DARPA under Grant DAAH04-96-1-0386 through the Quantum Information and Computing (QUIC) Institute administered by ARO, and by the U.S. Department of Energy under Grant DE-FG03-92-ER40701. The work of H.-K. Lo was performed while the author was with Hewlett-Packard Laboratories, Bristol, U.K.

S. L. Braunstein is at SECS, University of Wales, Bangor LL57 1UT, U.K., and at Hewlett-Packard Labs, Filton Road, Stoke Gifford, Bristol BS34 8QZ, U.K. (e-mail: schmuel@sees.bangor.ac.uk).

C. A. Fuchs is at the Norman Bridge Laboratory of Physics 12-33, California Institute of Technology, Pasadena, CA 91125 USA.

D. Gottesman was with the California Institute of Technology, Pasadena and with the Los Alamos National Laboratory. He is now with Microsoft Research, Microsoft Corporation, Redmond, WA 98052 USA.

H.-K. Lo is with MagiQ Technologies, Inc., New York, NY 10001 USA (e-mail: hk1@magiqtech.com).

Communicated by A. M. Barg, Associate Editor for Coding Theory.  
Publisher Item Identifier S 0018-9448(00)04288-7.

below, these two features—instantaneousness and variable length—of Huffman coding are difficult to generalize to the quantum case.

Now let us consider quantum information. In the *quantum* case, we are given a quantum source which emits a time sequence of independent and identically distributed pure-state quantum signals each of which is chosen from  $|u_1\rangle, |u_2\rangle, \dots, |u_m\rangle$  with probabilities  $q_1, q_2, \dots, q_m$ , respectively. Notice that  $|u_i\rangle$ 's are normalized (i.e., unit vectors) but not necessarily orthogonal to each other. Classical coding theory can be regarded as a special case when the signals  $|u_i\rangle$  are orthogonal. The goal of quantum source coding is to minimize the number of dimensions of the Hilbert space needed for almost lossless encoding of quantum signals, while maintaining a high fidelity between input and output. For a pure input state  $|u_i\rangle$ , the fidelity of the output density matrix  $\rho_i$  is defined as the probability for it to pass a yes/no test of being the state  $|u_i\rangle$ . Mathematically, it is given by  $\langle u_i | \rho_i | u_i \rangle$  [4]. In particular, we will be concerned with the average fidelity  $F = \sum_i q_i \langle u_i | \rho_i | u_i \rangle$ . It is convenient to measure the dimensionality of a Hilbert space in terms of the number of qubits (i.e., quantum bits) composing it; that is, the base-2 logarithm of the dimension.

Though there has been some preliminary work on quantum Huffman coding [9], the most well-known quantum source coding scheme is a block coding scheme [10], [5]. The converse of this coding theorem was proven rigorously in [1]. In block coding, if the signals are drawn from an ensemble with density matrix  $\rho = \sum q_j |u_j\rangle\langle u_j|$ , Schumacher coding, which is almost lossless, compresses  $N$  signals into  $NS(\rho)$  qubits, where  $S(\rho) = -\text{tr } \rho \log \rho$  is the von Neumann entropy. To encode  $N$  signals *sequentially*, it requires  $O(N^3)$  computational steps [2]. The encoding and decoding processes are far from instantaneous. Moreover, the lengths of all the codewords are the same.

## II. DIFFICULTIES IN A QUANTUM GENERALIZATION

A notable feature of quantum information is that measurement of it generally leads to disturbance. While measurement is a passive procedure in classical information theory, it is an integral part of the formalism of quantum mechanics and is an active process. Therefore, the big challenge in quantum coding is: How to encode and decode without disturbing the signals too much by the measurements involved? To illustrate the difficulties involved, we shall first attempt a naive generalization of Huffman coding to the quantum case. Consider the density matrix for each signal  $\rho = \sum q_j |u_j\rangle\langle u_j|$  and diagonalize it into

$$\rho = \sum_i p_i |\phi_i\rangle\langle \phi_i| \quad (1)$$

where  $|\phi_i\rangle$  is an eigenstate and the eigenvalues  $p_i$ 's are arranged in decreasing order. Huffman coding of a corresponding classical source with the same probability distribution  $p_i$ 's allows one to construct a one-to-one correspondence between Huffman codewords  $h_i$  and the eigenstates  $|\phi_i\rangle$ . Any input quantum state  $|u_j\rangle$  may now be written as a sum over the complete set  $|\phi_i\rangle$ . Remarkably, this means that, for such a naive generalization of Huffman coding, the length of each signal is a quantum-mechanical variable with its value in a superposition of the length eigenstates. It is not clear what this really means nor how to deal with such an object. If one performs a measurement on the length variable, the statement that measurements lead to disturbance means that irreversible changes to the  $N$  signals will be introduced which disastrously reduce the fidelity.

Therefore, to encode the signals faithfully, the sender and the receiver are forbidden to measure the length of each signal. We emphasize that this difficulty—that the sender is ignorant of the length of the signals to be sent—is, in fact, very general. It appears in any distributed

scheme of quantum computation. It is also highly analogous to the synchronization problem in the execution of subroutines in a quantum computer: A quantum computer program runs various computational paths simultaneously. Different computational paths may take different numbers of computational steps. A quantum computer is, therefore, generally unsure whether a subroutine has been completed or not. We do not have a satisfactory resolution to those subtle issues in the general case. Of course, the sender can always avoid this problem by adding redundancies (i.e., adding enough zeros to the codewords to make them a fixed length). However, such a prescription is highly inefficient and is self-defeating for our purpose of efficient quantum coding. For this reason, we reject such a prescription in our current discussion.

In the hope of saving resources, the natural next step to try is to stack the signals in line in a single tape during the transmission. To greatly simplify our discussion we shall suppose that the read/write head of the machine is quantum-mechanical with its location given by an internal state of the machine (this head location could be thought of as being specified on a separate tape). But then the second problem arises. Assuming a fixed speed of transmission, the receiver can never be sure when a particular signal, say the seventh signal, arrives. This is because the *total* length of the signals up to that point (from the first to seventh signals) is a quantum-mechanical variable (i.e., it is in a superposition of many possible values). Therefore, Bob generally has a hard time in deciding when would be the correct instant to decode the seventh signal in an instantaneous quantum code.

Let us suppose that the above problem can be solved. For example, Bob may wait “long enough” before performing any measurements. We argue that there remains a third difficulty which is fatal for *instantaneous* quantum codes—that the head location of the encoder is *entangled* with the total length of the signals. If the decoder consumes the quantum signal (i.e., performs measurements on the signals) before the encoding is completed, the record of the total length of the signals in the encoder head will destroy quantum coherence. This decoherence effect is physically the same as a “which path” measurement that destroys the interference pattern in a double-slit experiment. One can also understand this effect simply by considering an example of  $N$  copies of a state  $a|0\rangle + b|1\rangle$ . It is easy to show that if the encoder couples an encoder head to the system and keeps a record of the total number of zeros, the state of each signal will become impure. Consequently, the fidelity between the input and the output is rather poor.

## III. STORAGE OF QUANTUM SIGNALS

Nevertheless, we will show here that Huffman-coding-inspired quantum schemes do exist for both storage and communication of quantum information. In this section we consider the problem of storage. Notice that the above difficulties are due to the requirement of instantaneousness. This leads in a natural way to the question of *storage* of quantum information, where there is no need for instantaneous decoding in the first place. In this case, the decoding does not start until the whole encoding process is done. This immediately gets rid of the second (namely, when to decode) and third (namely, the record in the encoder head) problems mentioned in the last section. However, the first problem reappears in a new incarnation: The *total* length of say  $N$  signals is unknown and the encoder is not sure about the number of qubits that he should use. A solution to this problem is to use essentially the law of large numbers. If  $N$  is large, then asymptotically the length variable of the  $N$  signals has a probability *amplitude* concentrated in the subspace of values between  $N(\bar{L} - \delta)$  and  $N(\bar{L} + \delta)$  for any  $\delta > 0$  [10], [5], [1]. Here  $\bar{L}$  is the weighted average length of a Huffman codeword. One can, therefore, truncate

the signal tape into one with a *fixed* length say  $N(\bar{L} + \delta)$  (“0’s” can be padded to the end of the tape to make up the number if necessary.). Of course, the whole tape is not of variable length anymore. Nonetheless, we will now demonstrate that this tape can be a useful component of a new coding scheme—which we shall call quantum Huffman coding—that shares some of the advantages of Huffman coding over block coding. In particular, assuming that quantum gates can be applied in *parallel*, the encoding and decoding of quantum Huffman coding can be done efficiently. While a sequential implementation of quantum source *block* coding [10], [5], [1] for  $N$  signals requires  $O(N^3)$  computational steps [2], a parallel implementation of quantum Huffman coding has only  $O((\log N)^a)$  depth for some positive integer  $a$ , and a sequential implementation still uses just  $O(N(\log N)^a)$  gates.

We will now describe our coding scheme for the storage of quantum signals. As before, we consider a quantum source emitting a sequence of independent and identically distributed quantum signals with a density matrix for each signal shown in (1) where  $p_i$ 's are the eigenvalues. Considering Huffman coding for a classical source with probabilities  $p_i$ 's allows one to construct a one-to-one correspondence between Huffman codewords  $h_i$  and the eigenstates  $|\phi_i\rangle$ . For parallel implementation, we find it useful to represent  $|\phi_i\rangle$  by two pieces,<sup>1</sup> the first being the Huffman codeword, padded by the appropriate number of zeros to make it into constant length,<sup>2</sup>  $|0 \cdots 0h_i\rangle$ , the second being the length of the Huffman codeword,  $|l_i\rangle$ , where  $l_i = \text{length}(h_i)$ . We also pad zeros to the second piece so that it becomes of fixed length  $\lceil \log l_{\max} \rceil$  where  $l_{\max}$  is the length of the longest Huffman codeword. Therefore,  $|\phi_i\rangle$  is mapped into  $|0 \cdots 0h_i\rangle|l_i\rangle$ . Notice that the length of the second tape is  $\lceil \log l_{\max} \rceil$  which is generally small compared to  $n$ . The usage of the second tape is a small price to pay for efficient parallel implementation.

In this section, we use the model of a quantum gate array for quantum computation. The complexity class **QNC** is the class of quantum computations that can be performed in polylogarithmic parallel depth [7]. We prove the following theorem.

*Theorem 1:* Encoding or decoding of a quantum Huffman code for storage is in the complexity class **QNC**. Solving the classical Huffman coding problem for the eigenvalues of the density matrix gives a coding scheme with average codeword length  $\bar{L}$  and maximum codeword length  $l_{\max}$ . For any  $\delta > 0$ , for large enough  $N$ , the quantum Huffman code stores data using less than  $N(\bar{L} + \delta + \lceil \log l_{\max} \rceil)$  qubits. The encoding network has depth  $O((\log N)^2)$ .

The proof follows in the next two subsections.

#### A. Encoding

Without much loss of generality, we suppose that the total number of messages is  $N = 2^r$  for some positive integer  $r$ . We propose to encode by divide and conquer. First, we divide the messages into pairs and apply a merging procedure to be discussed in (2) to each pair. The merging effectively reduces the total number of messages to  $2^{r-1}$ . We can repeat this process. Therefore, after  $r$  applications of the merging procedure below, we obtain a single tape containing all the messages (in addition to the various length tapes containing the length information).

<sup>1</sup>The second piece contains no new information. However, it is useful for a massively parallel implementation of the shifting operations, which is an important component in our construction.

<sup>2</sup>The encoding process to be discussed below will allow us to reduce the total length needed for  $N$  signals.

The first step is the merging of two signals into a single message. Let us introduce a message tape. For simplicity, we simply denote  $|0 \cdots 0h_{i_1}\rangle$  by  $|h_{i_1}\rangle$ , etc.,

$$\begin{array}{l} |h_{i_1}\rangle|l_1\rangle|h_{i_2}\rangle|l_2\rangle \quad |0\rangle_{\text{tape}} \\ \xrightarrow{\text{swap}} |0\rangle|l_1\rangle|h_{i_2}\rangle|l_2\rangle \quad |0 \cdots 0h_{i_1}\rangle_{\text{tape}} \\ \xrightarrow{\text{shift}} |0\rangle|l_1\rangle|h_{i_2}\rangle|l_2\rangle \quad |h_{i_1}0 \cdots 0\rangle_{\text{tape}} \\ \xrightarrow{\text{swap}} |0\rangle|l_1\rangle|0\rangle|l_2\rangle \quad |h_{i_1}0 \cdots 0h_{i_2}\rangle_{\text{tape}} \\ \xrightarrow{\text{shift}} |0\rangle|l_1\rangle|0\rangle|l_2\rangle \quad |h_{i_1}h_{i_2}0 \cdots 0\rangle_{\text{tape}}. \end{array} \quad (2)$$

We remark that the swap operation between any two qubits can be done efficiently by using an array of three XOR's with the two qubits alternately used as the control and the target.<sup>3</sup> The shift operation is just a permutation and therefore can be done in constant depth [7]. However, we actually need something slightly stronger: a controlled shift, controlled by functions of the lengths  $|l_1\rangle$  and  $|l_2\rangle$ , which are quantum variables. To do a shift controlled by the register  $|s\rangle$ , we expand  $s$  in binary, and perform a shift by  $2^i$  positions conditioned on the appropriate bit of  $s$ . When  $|s\rangle$  is a quantum register in a superposition, this operation performed coherently will entangle the register with the tape, just as in the third difficulty described above. It is no longer a problem here, since we will disentangle the register and the tape during decoding.

Now the encoder keeps the original length tape for *each* signal as well as the message tape for two messages, i.e.,

$$|l_1\rangle|l_2\rangle|h_{i_1}h_{i_2}0 \cdots 0\rangle_{\text{tape}}.$$

Notice that it is relatively fast to compute the length  $l_1 + l_2$  of the two messages from  $l_1$  and  $l_2$ — $O(\log l)$  steps for the obvious sequential method (where  $l$  is the larger of  $l_1$  and  $l_2$ ), and  $O(\log \log l)$  depth with a good parallel algorithm. Therefore, the merging procedure can be performed in polylogarithmic depth.

More concretely, at the end the encoder obtains

$$|l_1\rangle|l_2\rangle \cdots |l_N\rangle|h_{i_1}h_{i_2} \cdots h_{i_N}0 \cdots 0\rangle_{\text{tape}}. \quad (3)$$

He has performed  $\lceil \log N \rceil$  merges. Merging two messages of maximum length  $l$  requires  $\lceil \log l \rceil$  shifts (each of constant depth) plus swaps (of constant depth) and one addition (of depth  $O(\log \log l)$ ). The maximum length  $l = Nl_{\max}$ , so the full merging procedure requires depth  $O((\log N)^2 + \log N \log l_{\max})$ . In addition, there is a constant depth cost for performing the initial encoding, which we neglect in the large- $N$  limit. We will also neglect the  $\log N \log l_{\max}$  term.

Finally, the encoder truncates the message tape: He keeps only say the first  $N(\bar{L} + \delta)$  qubits in the message tape  $|h_{i_1}h_{i_2} \cdots h_{i_N}0 \cdots 0\rangle_{\text{tape}}$  for some  $\delta > 0$  and throws away the other qubits. This truncation minimizes the number of qubits needed. The only overhead cost compared to the classical case is the storage of the length tapes of the individual signals. This takes only  $N \lceil \log l_{\max} \rceil$  qubits.<sup>4</sup>

#### B. Decoding

Decoding can be done by adding an appropriate number of qubits in the zero state  $|0\rangle$  behind the truncated message tape and simply running the encoding process backward (again with only depth  $O((\log N)^a)$ ).

What about fidelity? The key observation is the following:

*Definition 2:* The typical subspace  $S_\delta$  is the subspace where the first  $N(\bar{L} + \delta)$  qubits are arbitrary, and any qubits beyond that are in the *fixed* state  $|0 \cdots 0\rangle$ .

<sup>3</sup>In (2), we do not include the position of the head, since it is simply dependent on the sum of the message lengths and can be reset to 0 after the process is completed.

<sup>4</sup>Further optimization may be possible. For instance, if  $\log l_{\max}$  is large, one can save storage space by repeating the procedure, i.e., one can now use quantum Huffman coding for the problem of storing the quantum signals  $|l_i\rangle$ 's.

*Proposition 3:*  $\forall \epsilon, \delta > 0, \exists N_0 > 0$  such that  $\forall N > N_0, F \geq 1 - \epsilon$  where  $F$  is the fidelity between the true state  $\rho$  of the  $N$  quantum signals and the projection of  $\rho$  on the typical subspace  $S_\delta$  in our quantum Huffman coding scheme.

*Proof:* The proof is identical to the case of Schumacher's noiseless quantum coding theorem [10], [5], [1].

Therefore, the truncation and subsequent replacement of the discarded portion by  $|0 \cdots 0\rangle$  still lead to a high fidelity in the decoding.

In conclusion, we have constructed an explicit parallel encoding and decoding scheme for the storage of  $N$  independent and identically distributed quantum signals that asymptotically has only  $O((\log N)^a)$  depth and uses  $N(\bar{L} + \delta + \lceil \log l_{\max} \rceil)$  qubits for storage where  $\bar{L}$  is the average length of the Huffman coding for the classical coding problem for the set of probabilities given by the eigenvalues of the density matrix of each signal. Here  $\delta$  can be any positive number and  $l_{\max}$  is the length of the longest Huffman codeword.

*Corollary 4:* A sequential implementation of the encoding algorithm requires only  $O(N(\log N)^a)$  gates.

*Proof:* This follows immediately from the fact that the encoding is in QNC and uses  $O(N)$  qubits: At each time step of a parallel implementation, only  $O(N)$  steps are implemented. Since the network has depth  $O((\log N)^a)$ , there can be at most  $O(N(\log N)^a)$  gates in the network.

#### IV. COMMUNICATION

We now attempt to use the quantum Huffman coding for communication rather than for the storage of quantum signals. By communication, we assume that Alice receives the signals *one by one* from a source and is compelled to encode them one by one. As we will show below, the number of qubits required is slightly more, namely,  $N(\bar{L} + \delta + \lceil \log l_{\max} \rceil) + \lceil \log(Nl_{\max}) \rceil$ . The code that we will construct is not instantaneous, but Alice and Bob can pay a small penalty in stopping the transmission any time. In fact, we have the following theorem.

*Theorem 5:* Sequential encoding and decoding of a quantum Huffman code for communication requires  $N(\bar{L} + \delta + \lceil \log l_{\max} \rceil) + \lceil \log(Nl_{\max}) \rceil$  qubits and only  $O(N^2(\log N)^a)$  computational gates.

The proof follows in the next three subsections.

##### A. Encoding

The encoding algorithm is similar to that of Section III except that the signals are encoded one by one. More concretely, it is done through alternating applications of the swap-and-shift operations.

$$\begin{aligned}
 & |h_1\rangle|l_1\rangle|h_2\rangle|l_2\rangle \cdots |h_N\rangle|l_N\rangle|\mathbf{0}\rangle_{\text{tape}} \\
 & \otimes |\mathbf{0}\rangle_{\text{total length}} \\
 \xrightarrow{\text{swap}} & |\mathbf{0}\rangle|l_1\rangle|h_2\rangle|l_2\rangle \cdots |h_N\rangle|l_N\rangle|0 \cdots 0h_1\rangle_{\text{tape}} \\
 & \otimes |\mathbf{0}\rangle_{\text{total length}} \\
 \xrightarrow{\text{shift}} & |\mathbf{0}\rangle|l_1\rangle|h_2\rangle|l_2\rangle \cdots |h_N\rangle|l_N\rangle|h_10 \cdots 0\rangle_{\text{tape}} \\
 & \otimes |\mathbf{0}\rangle_{\text{total length}} \\
 \xrightarrow{\text{add}} & |\mathbf{0}\rangle|l_1\rangle|h_2\rangle|l_2\rangle \cdots |h_N\rangle|l_N\rangle|h_10 \cdots 0\rangle_{\text{tape}} \\
 & \otimes |l_1\rangle_{\text{total length}} \\
 \xrightarrow{\text{swap}} & |\mathbf{0}\rangle|l_1\rangle|\mathbf{0}\rangle|l_2\rangle \cdots |h_N\rangle|l_N\rangle|h_10 \cdots 0h_2\rangle_{\text{tape}} \\
 & \otimes |l_1\rangle_{\text{total length}}
 \end{aligned}$$

$$\begin{aligned}
 \xrightarrow{\text{shift}} & |\mathbf{0}\rangle|l_1\rangle|\mathbf{0}\rangle|l_2\rangle \cdots |h_N\rangle|l_N\rangle|h_1h_20 \cdots 0\rangle_{\text{tape}} \\
 & \otimes |l_1\rangle_{\text{total length}} \\
 \xrightarrow{\text{add}} & |\mathbf{0}\rangle|l_1\rangle|\mathbf{0}\rangle|l_2\rangle \cdots |h_N\rangle|l_N\rangle|h_1h_20 \cdots 0\rangle_{\text{tape}} \\
 & \otimes |l_1 + l_2\rangle_{\text{total length}} \\
 \dots & \\
 \xrightarrow{\text{shift}} & |\mathbf{0}\rangle|l_1\rangle|\mathbf{0}\rangle|l_2\rangle \cdots |\mathbf{0}\rangle|l_N\rangle|h_1h_2 \cdots h_N0 \cdots 0\rangle_{\text{tape}} \\
 & \otimes |l_1 + \cdots + l_{N-1}\rangle_{\text{total length}} \\
 \xrightarrow{\text{add}} & |\mathbf{0}\rangle|l_1\rangle|\mathbf{0}\rangle|l_2\rangle \cdots |\mathbf{0}\rangle|l_N\rangle|h_1h_2 \cdots h_N0 \cdots 0\rangle_{\text{tape}} \\
 & \otimes |l_1 + \cdots + l_N\rangle_{\text{total length}}.
 \end{aligned} \tag{4}$$

We have included an ancillary space storing the total length of the code-words generated so far.<sup>5</sup> This space requires  $\log(Nl_{\max})$  qubits.

Even though the encoding of signals themselves are done one by one, the shifting operation can be sped up by parallel computation. Indeed, as before, the required controlled-shifting operation can be performed in  $O(\log N + \log l_{\max})$  depth. As before, if a sequential implementation is used instead, the complete encoding of one signal still requires only  $O(N(\log N)^a)$  gates.

Now the encoding of the  $N$  signals in quantum communication is done sequentially, implying  $O(N)$  applications of the shifting operation. Therefore, with a parallel implementation of the shifting operation, the whole process has depth  $O(N(\log N)^a)$ . With a sequential implementation, it takes  $O(N^2(\log N)^a)$  steps.

##### B. Transmission

Notice that the message is written on the message tape from left to right. Moreover, starting from left to right, the state of each qubit once written remains unchanged throughout the encoding process. This decoupling effect suggests that rather than waiting for the completion of the whole encoding process, the sender, Alice, can start the transmission immediately after the encoding. For instance, after encoding the first  $r$  signals, Alice is absolutely sure that at least the first  $rl_{\min}$  (where  $l_{\min}$  is the minimal length of each codeword) qubits on the tape have already been written. She is free to send those qubits to Bob immediately. There is no penalty for such a transmission because it is easy to see that the remaining encoding process requires no help from Bob at all. (Note that in the asymptotic limit of large  $r$ , after encoding  $r$  signals, Alice can even send  $r(\bar{L} - \epsilon)$  qubits for any  $\epsilon > 0$  to Bob without worrying about fidelity.)

In addition, Alice can send the first  $r$  length variables  $l_1, \dots, l_r$ , but she must retain the total-length variable for continued encoding. Since the total-length variable is entangled with each branch of the encoded state, decoding cannot be completed by Bob without use of this information. In other words, Alice must disentangle her system from the encoded message before decoding may be completed.

##### C. Decoding

With the length information of each signal and the received qubits, Bob can *start* the decoding process before the whole transmission is complete *provided* that he does not perform any measurement at this moment. For instance, having received  $rl_{\min}$  qubits in the message tape from Alice, Bob is sure that at least  $s = \lfloor rl_{\min}/l_{\max} \rfloor$  signals have already arrived. He can separate those  $s$  signals immediately using the length information of each signal. This part of the decoding process is rather straightforward and we will skip its description here.

The important observation is, however, the following: If Bob were to perform a measurement on his signals now, he would find that they are

<sup>5</sup>As in (2), we do not include the position of the head.

of poor fidelity. The reason behind this has already been noted in Section II. Even though the subsequent encoding process does not involve Bob's system, there is still entanglement between Alice and Bob's systems. More specifically, the shifting operations in the remaining encoding process by Alice require explicitly the information on the total length of decoded signals. Before Bob performs any measurement on his signals, it is, therefore, crucial for Alice to disentangle her system first, as mentioned above.

Suppose in the middle of their communication in which Bob has already received  $K\bar{L}$  qubits from Alice, Bob suddenly would like to perform a measurement on his signals. He shall first inform Alice of his intention. Afterwards, one way to proceed is the following: They choose some convenient point, say the  $m$ th signal, to stop and consider quantum Huffman coding for only the first  $m$  signals and complete the encoding and decoding processes.

We shall consider two subcases. In the first subcase, the number  $m$  is chosen such that the  $m$ th signal is most likely still in the sender (Alice)'s hands (e.g.,  $m > K + O(\sqrt{K})$  in the asymptotic limit). The sender Alice now disentangles the remaining signal from the first  $m$  quantum signals by applying a quantum shifting operation. She can now complete the encoding process for quantum Huffman coding of the  $m$  signals and send Bob any untransmitted qubits on the tape. In the asymptotic limit of large  $K$ ,  $O(\sqrt{m})$  qubits of forward transmission (from Alice to Bob) are needed. (The required depth of the network is polynomial in  $\log m$  if a parallel implementation of a quantum gate array is used.) In addition, Alice must send her record of the total length of the signals. However, this requires only an additional  $\lceil \log(m l_{\max}) \rceil$  qubits, so the total number which must be transmitted for disentanglement is still  $O(\sqrt{m})$ .

In the second subcase, the number  $m$  is chosen such that the  $m$ th signal is most likely already in the receiver (Bob)'s hands (e.g.,  $m < K - O(\sqrt{K})$  in the asymptotic limit). The receiver Bob now attempts to disentangle the remaining signals from the first  $m$  quantum signals by applying a quantum shifting operation. Of course, he needs to shift some of his qubits back to Alice. This asymptotically amounts to  $O(\sqrt{m})$  qubits of *backward* communication. This is a penalty that one must pay for this method. After this is done, Alice must again send her length register to Bob (after subtracting the lengths of the signals returned to her). This requires an additional  $O(\log m)$  qubits.

If  $m$  is chosen between  $K - O(\sqrt{K})$  and  $K + O(\sqrt{K})$ , neither sending signals forward or backward will suffice to properly disentangle the varying lengths of the signals. One possible solution is to choose  $m' > K + O(\sqrt{K})$  and perform the above procedure, sending  $m'$  total signals to Bob. Then Bob decodes and returns the  $m' - m$  extra signals to Alice. This method requires  $O(\sqrt{K})$  qubits transmitted forward and  $O(\sqrt{K})$  qubits transmitted backward to disentangle.

We remark that the shifting operation can be done rather easily in distributed quantum computation between Alice and Bob. This is a nontrivial observation because the number of qubits to be shifted from Alice to Bob is itself a quantum-mechanical variable. This, however, does not create much problem. Bob can always communicate with Alice using a bus of fixed length. For example, he applies local operations to swap the desired quantum superposition of various numbers of qubits from his tape to the bus, sends such a bus to Alice, etc.

The result is the following theorem.

*Theorem 6:* Alice and Bob may truncate a communication session after the transmission of  $m$  encoded signals, retaining high fidelity with the cost of  $O(\sqrt{m})$  additional qubits transmitted.

In the above discussion, we have focused on the simple case when Bob would like to perform a measurement on the whole set of the first  $m$  signals. Suppose Bob is interested only in a particular signal, say the

$m$ th one, but not the others. There exists a more efficient scheme for doing it. We shall skip the discussion here.

## V. CONCLUDING REMARKS

We have successfully constructed a Huffman-coding-inspired scheme for the storage of quantum information. Our scheme is highly efficient. The encoding and decoding processes of  $N$  quantum signals can be done *in parallel* with depth polynomial in  $\log N$ . (If parallel machines are unavailable, as shown in Section IV-A our encoding scheme will still take only  $O(N(\log N)^a)$  computational steps for a sequential implementation. In contrast, a naive implementation of Schumacher's scheme will require  $O(N^3)$  computational steps.) This massive parallelism is possible because we explicitly use another tape to store the length information of the individual signals. The storage space needed is asymptotically  $N(\bar{L} + \delta + \lceil \log l_{\max} \rceil)$  where  $\bar{L}$  is the average length of the corresponding classical Huffman coding problem for the density matrix in the diagonal form,  $\delta$  is an arbitrary small positive number, and  $l_{\max}$  is the length of the longest Huffman codeword.

We also considered the problem of using quantum Huffman coding for communication in which case Alice encodes the signals one by one.  $N(\bar{L} + \delta + \lceil \log l_{\max} \rceil) + O(\log N)$  qubits are needed. With a parallel implementation of the shifting operation, depth of  $O(N(\log N)^a)$  is needed. On the other hand, with a sequential implementation,  $O(N^2(\log N)^a)$  computational steps are needed. In either case, the code is not instantaneous, but, by paying a small penalty in terms of communication and computational costs, Alice and Bob have the option of stopping the transmission and Bob may then start measuring his signals.

More specifically, while the receiver Bob is free to separate the signals from one another, he is not allowed to measure them until the sender Alice has completed the encoding process. This is because Alice's encoder head generally contains the information of the total length of the signals. In other words, its state is entangled with Bob's signals. Therefore, whenever Bob would like to perform a measurement, he should first inform Alice and the two should proceed with disentanglement. We present two alternative methods of achieving such disentanglement one of which involves forward communication and the other of which involves both forward and backward.

Since real communication channels are always noisy, in actual implementation source coding is always followed by encoding into an error-correcting code. Following the pioneering work by Shor [11] and independently by Steane [12], various quantum error-correcting codes have been constructed. We remark that quantum Huffman coding algorithm (even the version for communication) can be immediately combined with the encoding process of a quantum error-correcting code for efficient communication through a noisy channel.

As quantum information is fragile against noises in the environment, it may be useful to work out a fault-tolerant procedure for quantum source coding. The generalizations of other classical coding schemes to the quantum case are also interesting [6]. Moreover, there exist universal quantum data compression schemes motivated by the Lempel-Ziv compression algorithm for classical information [8].

## ACKNOWLEDGMENT

H.-K. Lo would like to thank D. P. DiVincenzo, J. Preskill, and T. Spiller for helpful discussions.

## REFERENCES

- [1] H. Barnum, C. A. Fuchs, R. Jozsa, and B. Schumacher, "General fidelity limit for quantum channels," *Phys. Rev.*, vol. A54, p. 4707, 1996.

- [2] R. Cleve and D. P. DiVincenzo, "Schumacher's quantum data compression as a quantum computation," *Phys. Rev.*, vol. A54, p. 2636, 1996.
- [3] T. M. Cover and J. A. Thomas, *Elements of Information Theory*. New York: Wiley, 1991.
- [4] R. Jozsa, "Fidelity for mixed quantum states," *J. Mod. Opt.*, vol. 41, p. 2315, 1994.
- [5] R. Jozsa and B. Schumacher, "A new proof of the quantum noiseless coding theorem," *J. Mod. Opt.*, vol. 41, p. 2343, 1994.
- [6] R. Jozsa, M. Horodecki, P. Horodecki, and R. Horodecki, "Universal quantum information compression," *Phys. Rev. Lett.*, vol. 81, p. 1714, 1998.
- [7] C. Moore and M. Nilsson. Parallel Quantum Computation and Quantum Codes (Los Alamos e-print archive). [Online] Available: <http://xxx.lanl.gov/abs/quant-ph/9808027>
- [8] M. A. Nielsen, "Quantum information theory," Ph.D. dissertation, Univ. New Mexico, Albuquerque, 1998.
- [9] B. Schumacher, "Quantum Kraft Inequality," presented at the Santa Fe Institute, 1994.
- [10] —, "Quantum coding," *Phys. Rev.*, vol. A51, p. 2738, 1995.
- [11] P. W. Shor, "Scheme for reducing decoherence in quantum computer memory," *Phys. Rev.*, vol. A52, p. R2493, 1995.
- [12] A. M. Steane, "Error correcting codes in quantum theory," *Phys. Rev. Lett.*, vol. 77, p. 793, 1996.

## Optimization of Distributed Detection Systems Under the Minimum Average Misclassification Risk Criterion

Maurizio Magarini and Arnaldo Spalvieri

**Abstract**—A common model for distributed detection systems is that of several separated sensors each of which measures some observable, quantizes it, and communicates to a fusion center the quantized observation. The fusion center collects the quantized observations and takes the decision. The present correspondence deals with the design of the quantizers and of the fusion center under a rate constraint. The system of interest allows soft nonbreakpoint quantizers and nonindependent observations. Our finding is that locally optimal design of the distributed detection system is feasible via alternate minimization of the average misclassification risk.

**Index Terms**—Alternate optimization, average misclassification risk, distributed detection.

### I. INTRODUCTION

Distributed detection systems have received a lot of attention in the past two decades, as documented in the special issue of the PROCEEDINGS OF THE IEEE [1]. A common model for these systems involves several separated sensors, each of which measures some observable, quantizes it, and communicates to a fusion center the quantized observation. The fusion center collects the quantized observations and takes the decision. Since the rate of transmission between the sensors and the fusion center is a cost, fine quantization of data may be not allowed. A crucial problem is therefore the design of coarse quantizers that satisfy a rate constraint and that introduce low degradation in the detection capability of the system. Tsitsiklis and Athans have shown in [2] that, when conditional independence of

the observation given the hypothesis cannot be assumed, the design problem is NP complete. Hence one is lead to renounce to global optimality and to study suboptimal strategies. Several design strategies have been studied in the past, most of which were tailored to hard (one-bit) quantizers. Tenney and Sandell optimized the decentralized quantizers with a fixed fusion rule [3], while Chair and Varshney considered the design of the fusion rule for fixed quantizers [4]. Joint design of soft (multibit) quantizers has been studied by Longo *et al.* in [5], where an alternate optimization technique is proposed. Specifically, the approach in [5] is to maximize the Bhattacharyya distance between the multivariate conditional probabilities of quantized data given the hypotheses. The potential weakness of this approach is that the Bhattacharyya distance is not the natural measure of performance of detection systems. Therefore, one wonders whether joint design of quantizers and the fusion rule under the natural criterion of performance is feasible. Our answer is that locally optimal design, that is, minimization of the average misclassification risk, is feasible by alternate optimization. A similar method was adopted in [6] in the framework of decentralized parameter estimation. Also, in [7] the alternate optimization technique is considered as a method to minimize a general distortion measure. Like [5], [7], our method applies to nonindependent observations and to soft (multibit) nonbreakpoint quantizers.

### II. SYSTEM MODEL AND PROBLEM STATEMENT

For the sake of simplicity, consider two scalar observations and binary detection. Extensions are straightforward. Let  $x_1, x_2$  denote the observations, and assume that they are drawn from the continuous spaces  $\mathcal{X}_1, \mathcal{X}_2$ . In the classical formulation of the detection problem, a hidden discrete random variable (the *class*, or the *hypothesis*) is drawn together with the observation vector according to some known joint probability distribution. We call such a discrete random variable  $c \in \mathcal{C} = \{c_1, c_2\}$ . The goal of the detection system is to guess the hidden class given the observation vector.

#### A. System Description

The decentralized detection system we are concerned with is modeled as a decision rule made by two scalar quantizers and a fusion center. Each scalar quantizer is allowed here to be a nonbreakpoint one. Quantizer  $Q_n(x_n), n = 1, 2$ , is modeled as a mapping from  $\mathcal{X}_n$  to  $\mathcal{I}_n$ , where  $\mathcal{I}_n = \{0, 1, \dots, I_n - 1\}$ . Of course, the rate  $R_n$  of the  $n$ th quantizer is  $R_n = \log_2 I_n$ . Inversion of  $Q_n(x)$  is hereafter intended as

$$Q_n^{-1}(i) = \{x_n \in \mathcal{X}_n: Q_n(x_n) = i\}.$$

The decision function performed by the fusion center, denoted  $\Phi(i_1, i_2)$ , is a mapping from  $\mathcal{I}_1 \times \mathcal{I}_2$  to  $\mathcal{C}$ . The decision rule of the decentralized detection system, denoted  $\Phi(Q_1(x_1), Q_2(x_2))$ , is a mapping from  $\mathcal{X}_1 \times \mathcal{X}_2$  to  $\mathcal{C}$ . As in [5], we assume that the processing to be performed at the fusion center is unlimited in complexity. In practice, this means that the fusion center is a lookup table with  $2^{R_1+R_2}$  entries. A pictorial example of the decision rule for a specific two-dimensional decentralized detection system is later illustrated in Fig. 6.

#### B. Statement of the Problem

The Bayesian risk (or cost) in deciding in favor of class  $\hat{c} \in \mathcal{C}$  when  $x_1, x_2$  is observed is

$$R(\hat{c}|x_1, x_2) = \sum_{i=1}^2 b(c_i \mapsto \hat{c})P(c_i|x_1, x_2) \quad (1)$$

Manuscript received November 19, 1998; revised December 22, 1999.  
The authors are with the Dipartimento di Elettronica e Informazione, Politecnico di Milano, 20133 Milano, Italy (e-mail: magarini@elet.polimi.it; spalvier@elet.polimi.it).

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

Publisher Item Identifier S 0018-9448(00)05016-1.

where  $b(c_i \mapsto \hat{c}) \geq 0$  is the risk of deciding in favor of class  $\hat{c}$  when  $c = c_i$ , and the familiar notation is adopted for the conditional probability of the class given the observation. Let  $\Phi(x_1, x_2)$  be a decision rule. The classification performance of  $\Phi$  is measured by the expectation of the local risk (1) over the observation space

$$R(\Phi) = \int_{\mathcal{X}_1} \int_{\mathcal{X}_2} R(\Phi(x_1, x_2)|x_1, x_2)p(x_1, x_2) dx_1 dx_2 \quad (2)$$

where  $p(x_1, x_2)$  is the (bivariate) probability density function of the observation. If we set  $b(c_i \mapsto c_j) = 1$  for  $i \neq j$  and  $b(c_i \mapsto c_j) = 0$  for  $i = j$ , then the average misclassification risk (or, in short, average risk) is equal to the average error probability. The optimal decision rule, that is the rule minimizing the risk, is the Bayes test

$$\Phi_B(x_1, x_2) = \arg \min_{\hat{c} \in \mathcal{C}} R(\hat{c}|x_1, x_2). \quad (3)$$

Before enunciating the statement of the problem, we recall that the Neyman–Pearson approach minimizes  $\alpha$  subject to  $\beta$  being equal to a prefixed constant, with

$$\alpha = \int_{D(c_2)} p(x_1, x_2|c_1) dV_x \quad (\text{probability of type I error})$$

$$\beta = \int_{D(c_1)} p(x_1, x_2|c_2) dV_x \quad (\text{probability of type II error})$$

where  $D(c)$  is the decision region of class  $c$  and  $dV_x$  is the differential volume in the observation space. The test resulting from the constrained minimization has the form

$$\Phi_{NP}(x_1, x_2) = \arg \max_{\{c_1, c_2\}} \{p(x_1, x_2|c_1), \lambda p(x_1, x_2|c_2)\} \quad (4)$$

where  $\lambda$  is the decision threshold of the Neyman–Pearson test and  $p(x_1, x_2|c)$  is the conditional probability density function of the observation given the class. In the practice, the constrained minimization is worked out by finding experimentally or numerically the value of  $\lambda$  that makes  $\beta$  equal to the prefixed constant. Note that, with

$$\lambda = P(c_2)b(c_2 \mapsto c_1)/P(c_1)b(c_1 \mapsto c_2)$$

$P(c_2)b(c_2 \mapsto c_1) \neq 0$ ,  $b(c_i \mapsto c_j) = 0$  for  $i = j$ , one readily gets from (3), (4)  $\Phi_B = \Phi_{NP}$ . Therefore, both from the Bayesian approach and from the Neyman–Pearson approach, one comes to the following statement of the problem:

*Fix  $P(c_i)b(c_i \mapsto c_j) \forall i, j$ , and find  $\Phi(i_1, i_2)$ ,  $Q_1(x_1)$ , and  $Q_2(x_2)$  that minimize  $R(\Phi(Q_1(x_1), Q_2(x_2)))$ .*

### III. THE ALTERNATE OPTIMIZATION

The optimization algorithm is based on alternate optimization of the three functions (two quantizers and the decision rule). The optimality condition for  $Q_1$  given  $Q_2$  and  $\Phi$  is obtained from (1). Specifically, for all the points in  $\mathcal{X}_1$  one finds the best index by taking the expectation of the local risk over  $\mathcal{X}_2$

$$Q_1^{opt}(x_1) = \arg \min_{i_1 \in \mathcal{I}_1} \sum_{i_2=0}^{I_2-1} \int_{Q_2^{-1}(i_2)} R(\Phi(i_1, Q_2(x_2))|x_1, x_2) \cdot p(x_2) dx_2, \quad \forall x_1 \in \mathcal{X}_1. \quad (5)$$

A similar condition holds for  $Q_2$  given  $Q_1$  and  $\Phi$ .

The optimality condition for  $\Phi(i_1, i_2)$  given  $Q_1$  and  $Q_2$  is obtained by taking the expectation of the local risk over the region indexed  $i_1, i_2$

$$\Phi^{opt}(i_1, i_2) = \arg \min_{c \in \mathcal{C}} \int_{Q_1^{-1}(i_1)} \int_{Q_2^{-1}(i_2)} R(c|x_1, x_2) \cdot p(x_1, x_2) dx_1 dx_2, \quad \forall i_1, i_2 \in \mathcal{I}_1 \times \mathcal{I}_2. \quad (6)$$

In the alternate optimization, we look for a new  $\Phi$  only when both quantizers are optimal for the old  $\Phi$ . Hence, after a proper initialization to be discussed in the next section, the alternate optimization proceeds as follows:

- 1) determine  $Q_1$  by (5),
- 2) determine  $Q_2$  by (5), if the new  $Q_2$  is different from the old  $Q_2$  then go to 1),
- 3) determine  $\Phi$  by (6), if the new  $\Phi$  is different from the old  $\Phi$  then go to 1), else STOP.

Since each of the optimization steps does not increase the average risk, the procedure will terminate in a local optimum.

To compare the computational complexity of the proposed method with the method [5], it should be noted that the method [5] aims to approximate the conditional probabilities of the observation given the hypothesis, independently of the decision threshold. The decision rule is then obtained by applying the decision threshold to the conditional probabilities. Note that one optimization leads to a set of decision rules when a set of the decision thresholds is considered. Conversely, our method aims to approximate the Bayesian decision rule; hence the decision threshold is embedded in the alternate optimization. If the Bayesian approach is considered, where only one value of the threshold is of interest, then the optimization method that we propose has the same computational complexity as the method proposed in [5], because the proposed alternate optimization must run only one time. Elsewhere, when the Neyman–Pearson approach is considered, then, as noted in the previous section, one has to find experimentally the value of  $\lambda$  that makes  $\beta$  equal to the prefixed constant. This task is performed by running the proposed alternate optimization for a set of values of  $\lambda$ , and then selecting that value of  $\lambda$  that yields the closest  $\beta$  to the desired value of  $\beta$ . However, it should be noted that the practice of both procedures is limited by the complexity of calculation of the optimality conditions and by the size of the lookup table: both are proportional to  $2^{R_1+R_2}$ . Therefore, these methods can be actually applied only to coarse quantization of low-dimensional spaces.

### IV. COMPUTER IMPLEMENTATION AND EXPERIMENTAL RESULTS

In the computer implementation, as well as in many actual detection systems, where continuous signals are digitally processed after analog-to-digital conversion, the continuous spaces  $\mathcal{X}_1$  and  $\mathcal{X}_2$  are discretized. In our experiments  $\mathcal{X}_1$  and  $\mathcal{X}_2$  are discretized in 256 slices of equal size. We have verified that such a slicing has negligible impact on system performance. Dealing with discrete signals, the integrals appearing in (5) and (6) are replaced by sums in an obvious way.

The major trouble that we encountered in the experimental part of this work was that of local minima. It is well known that, when dealing with a cost function having local minima, the result of an alternate minimization is strongly influenced by the initial guess, and, to our knowledge, methods that guarantee a good initialization do not exist. We have tested two initial guesses. The first were the quantizers found by [5]. The second was to initialize the quantizers from breakpoint quantizers with equally sized cells. We observed that often one has the chance of improving the minimum found by the alternate optimization. Specifically, it sometimes happens that, after the optimization, the region assigned to one or more indices vanishes, and the indices remain inactive. Some other times two (or more) rows (and/or columns) of the decision matrix  $\Phi(i_1, i_2)$ , say rows 0 and 1, are identical. Then the regions  $Q_1^{-1}(0)$  and  $Q_1^{-1}(1)$  can be grouped in a unique region without losing performance. After grouping, only one of the two (or more) indices is still active. In some trivial cases, the global minimum can be obtained with a reduced number of indices; therefore, the inactive indices can be discarded. However, often the presence of inactive indices

indicates that the minimization procedure has found a nonglobal minimum. Hence, aiming to improve performance, one can try to re-initialize the inactive indices. Again, neither method can guarantee successful re-initialization. Note, however, that now one has to initialize only a few indices, not the whole system. Hence extensive search of a good re-initialization may become feasible. Assume that there is only one inactive index, and, without losing generality, assume that it belongs to  $Q_1$ . We assign one slice to the inactive index, apply (6), and check the new  $\Phi$  against the old  $\Phi$ . If  $\Phi$  has been modified by the re-initialization, then the average risk has been improved, and one is certain that, by repeating the alternate optimization from the new guess, the procedure will eventually find some new minimum better than the old one. Or else, we sequentially examine the remaining slices, until some slice can be conveniently assigned to the inactive index. If there are more than one inactive indices of  $Q_1$ , then we repeat the above procedure to re-initialize the alternate optimization. If none of the slices can be conveniently re-assigned, then this search fails. Assume now that there is at least one inactive index per dimension. Actually this often happens, at least in the studied examples. With a procedure similar to the one above, one can try to jointly reassign two slices. Note that, if the minimum is nonglobal, then it is guaranteed that proper joint re-initialization of the two indices improves the average risk. To see this, assume that the decision rule found by the alternate optimization differs from the Bayes rule at point  $(i, j)$ . Then assigning the inactive indices to slices  $i$  and  $j$  leads to a convenient relabeling of the point, hence to an improved risk. Conversely, it is not guaranteed that, adding only one index, one can modify the label of the above mentioned point, because of the constraints imposed by quantization in the second dimension.

The results of the alternate optimization with re-initialization seem to be insensitive to the initial guess (breakpoint or [5]), at least in the studied examples. This fact makes us confident of having found good minima. To summarize, the results to be presented have been worked out by the following pseudocode:

- 1) initialize  $Q_1$  and  $Q_2$ ;
- 2) determine  $\Phi$  by (6);
- 3) apply the alternate optimization;
- 4) check the minimum: if there are no inactive indices then STOP, else try to reactivate them;
- 5) if reactivation fails then STOP, or else go to 3).

The examples of interest are the same considered in [5], and our results are compared to the results of [5]. We assume  $P(c_1) = P(c_2) = 0.5$ ,  $b(c_1 \mapsto c_1) = b(c_2 \mapsto c_2) = 0$ , and repeat the optimization with several values of  $\lambda = b(c_1 \mapsto c_2)/b(c_2 \mapsto c_1)$ ,  $b(c_1 \mapsto c_2) = \min\{\lambda, 1\}$ ,  $b(c_2 \mapsto c_1) = \min\{\lambda^{-1}, 1\}$ . The results are presented as average risk against  $\lambda$ .

*Example 1—Known Signal in Spatially Correlated Noise:* The observation model is

$$\begin{aligned}
 c_1: \mathbf{X} &= \mathbf{N} \\
 c_2: \mathbf{X} &= \mathbf{A} + \mathbf{N}
 \end{aligned}$$

where  $\mathbf{A}$  is the known signal vector and  $\mathbf{N}$  is a zero-mean Gaussian noise vector with covariance matrix (let  $r$  be the spatial correlation coefficient)

$$\Sigma = \begin{pmatrix} \sigma^2 & r\sigma^2 \\ r\sigma^2 & \sigma^2 \end{pmatrix}, \quad |r| < 1. \quad (7)$$

As in [5], the experiment concerns signals having equal energy, say  $|a_1|^2 = |a_2|^2 = \mathcal{E}$ , and per-channel signal-to-noise ratio  $\text{SNR} = \mathcal{E}/\sigma^2$ . Figs. 1–3 report the performance of the initial

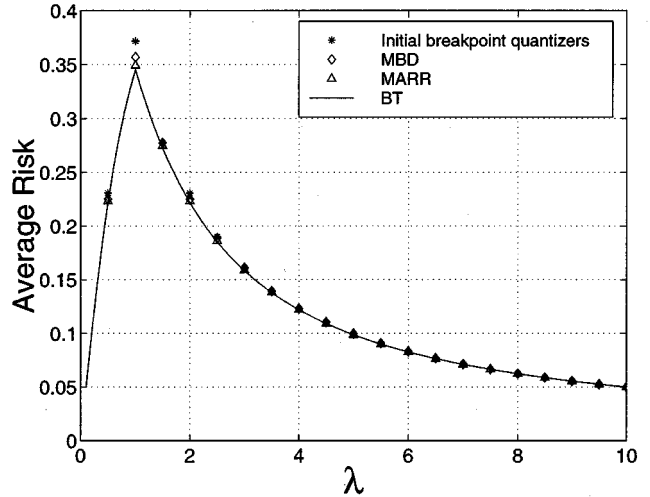


Fig. 1. Example 1. SNR = -5 dB, correlation coefficient  $r = 0$ ,  $R = 2$  bits/sample for each quantizer.

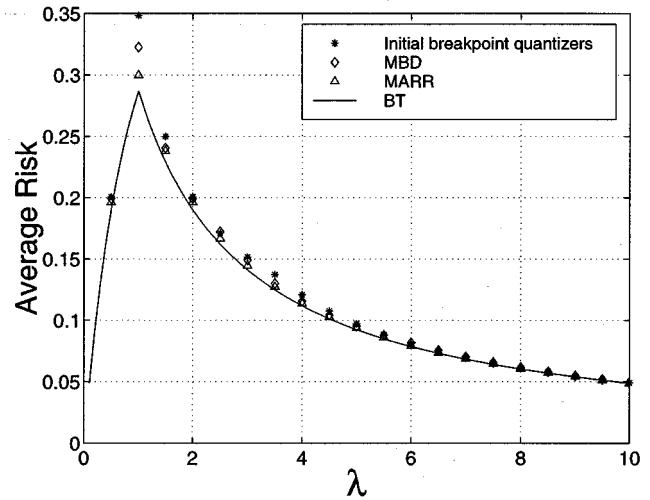


Fig. 2. Example 1. SNR = -5 dB, correlation coefficient  $r = 0.5$ ,  $R = 2$  bits/sample for each quantizer.

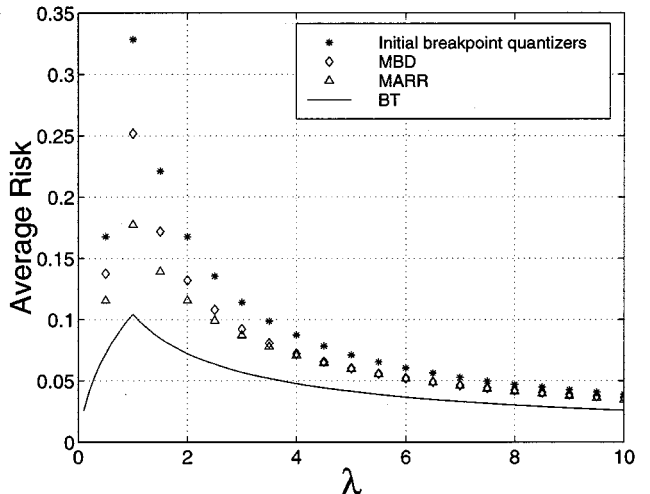


Fig. 3. Example 1. SNR = -5 dB, correlation coefficient  $r = 0.9$ ,  $R = 2$  bits/sample for each quantizer.

TABLE I  
AVERAGE RISK WITH  $\lambda = 1$  FOR EXAMPLE 1. MAR: MINIMUM AVERAGE RISK; MARR: MINIMUM AVERAGE RISK WITH RE-INITIALIZATION;  
MBD: MINIMUM BHATTACHARYYA DISTANCE; BT: BAYES TEST

Correlation $r$	SNR (dB)	Number of Cells	Rate R (bits/quant.)	Average Risk			
				MAR	MARR	MBD	BT
0.9	-5	256	2	0.1956	0.1774	0.2574	0.1015
-0.9	-5	256	2	0.3890	0.3865	0.3886	0.3853
0	-5	256	2	0.3503	0.3490	0.3568	0.3470
0.5	-5	256	2	0.3041	0.2999	0.3226	0.2844
0.99	-5	256	2	0.0645	0.0575	0.2291	3.5e-5
0.9	-10	256	2	0.3127	0.2994	0.3557	0.2383
0.9	0	256	2	0.0675	0.0532	0.1282	0.0127
0.9	5	256	2	0.0044	0.0023	0.0323	3.5e-5
0.9	-5	64	2	0.1960	0.1803	0.2575	0.1015
0.9	-5	256	1	0.3879	0.3879	0.3879	0.1015
0.9	-5	256	3	0.1276	0.1252	0.1701	0.1015
0.9	-5	256	4	0.1105	0.1104	0.1377	0.1015

TABLE II  
AVERAGE RISK WITH  $\lambda = 1$  FOR EXAMPLE 2. MAR: MINIMUM AVERAGE RISK; MARR: MINIMUM AVERAGE RISK WITH RE-INITIALIZATION;  
MBD: MINIMUM BHATTACHARYYA DISTANCE; BT: BAYES TEST

Correlation $r$	SNR (dB)	Number of Cells	Rate R (bits/quant.)	Average Risk			
				MAR	MARR	MBD	BT
0.9	10	256	2	0.2025	0.1994	0.2055	0.1829
0	10	256	2	0.1470	0.1431	0.1526	0.1424
0.5	10	256	2	0.1580	0.1542	0.1602	0.1518
0.99	10	256	2	0.2303	0.2290	0.2297	0.1889
0.9	-5	256	2	0.4536	0.4507	0.4542	0.4432
0.9	0	256	2	0.3933	0.3899	0.3922	0.3736
0.9	5	256	2	0.3046	0.3021	0.3062	0.2788

breakpoint quantizers, of our method initialized from the breakpoint quantizers with re-initialization of inactive indices (MARR, minimum average risk with re-initialization), of the method of [5] initialized from the breakpoint quantizers (MBD, minimum Bhattacharyya distance), and of the Bayes test (BT), with rate  $2 \text{ bits/sample}$ , SNR =  $-5 \text{ dB}$ , and correlation coefficients  $r = 0$ ,  $r = 0.5$ , and  $r = 0.9$ , respectively. We see that our proposed method gives substantial benefits for  $r = 0.9$ . Figs. 4 and 5 report the performance of the above methods for  $r = 0.9$ , and for  $R = 3, 4 \text{ bits/sample}$ . Regarding Fig. 5, it should be noted that, with  $\lambda \approx 3$ , the performance of the method of [5] is slightly worse than the performance of the initial quantizers. This fact is not completely unexpected, since the method of [5] does not optimize the average misclassification risk. The benefits obtained with the re-initialization procedure are reported in Table I. From Table I one also appreciates that, with  $r = 0.9$ ,  $R = 2$ , and SNR =  $5 \text{ dB}$  (row 8), the proposed method outperforms method of [5] by one order of magnitude. The CPU time spent to perform the alternate optimization by a Matlab program processed by Intel Pentium 133 MHz is as follows: MBD: 8.0 s, MARR: 1161.9 s (Fig. 1); MBD: 9.1 s, MARR: 1560.0 s (Fig. 2); MBD: 15.8 s, MARR: 366.0 s (Fig. 3); MBD: 65.3 s, MARR: 1238.8 s (Fig. 4); MBD: 209.4 s, MARR: 6333.4 s (Fig. 5). Note that, with the MARR method, the alternate optimization runs 20 times for each figure, while with the MBD method the alternate optimization runs only once. For Figs. 3 and 4, the time for MARR is about 20 times the time for MBD. In the other figures, the MARR method incurs some penalty, which is mainly due to the re-initialization. For instance, in row 8 of Table I, the CPU time for MBD is 7.3 s, for MAR 5.7 s (no re-initialization), for MARR 143.2 s.

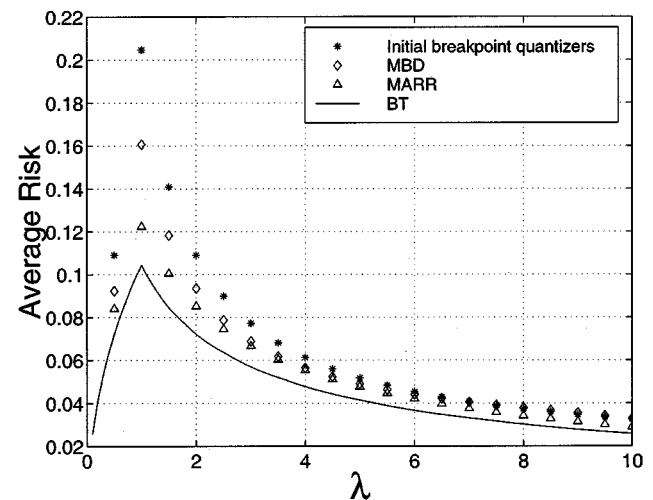


Fig. 4. Example 1. SNR =  $-5 \text{ dB}$ , correlation coefficient  $r = 0.9$ ,  $R = 3$  bits/sample for each quantizer.

Example 2—Spatially Correlated Unknown Signal in Uncorrelated Noise: The observation model is

$$c_1: \mathbf{X} = \mathbf{W}$$

$$c_2: \mathbf{X} = \mathbf{S} + \mathbf{W}$$

where  $\mathbf{W}$  is a zero-mean Gaussian noise vector with independent, unitary variance components and  $\mathbf{S}$  is a Gaussian signal vector with covari-

TABLE III  
 AVERAGE RISK WITH  $\lambda = 10$  FOR EXAMPLE 2. MAR: MINIMUM AVERAGE RISK; MARR: MINIMUM AVERAGE RISK WITH RE-INITIALIZATION;  
 MBD: MINIMUM BHATTACHARYYA DISTANCE; BT: BAYES TEST

Correlation $r$	SNR (dB)	Number of Cells	Rate R (bits/quant.)	Average Risk			
				MAR	MARR	MBD	BT
0.9	10	256	2	0.0306	0.0298	0.0313	0.0255
0	10	256	2	0.0228	0.0218	0.0235	0.0216
0.5	10	256	2	0.0245	0.0235	0.0248	0.0228
0.99	10	256	2	0.0326	0.0318	0.0328	0.0255
0.9	-5	256	2	0.0500	0.0500	0.0500	0.0499
0.9	0	256	2	0.0488	0.0487	0.0492	0.0474
0.9	5	256	2	0.0431	0.0424	0.0452	0.0379

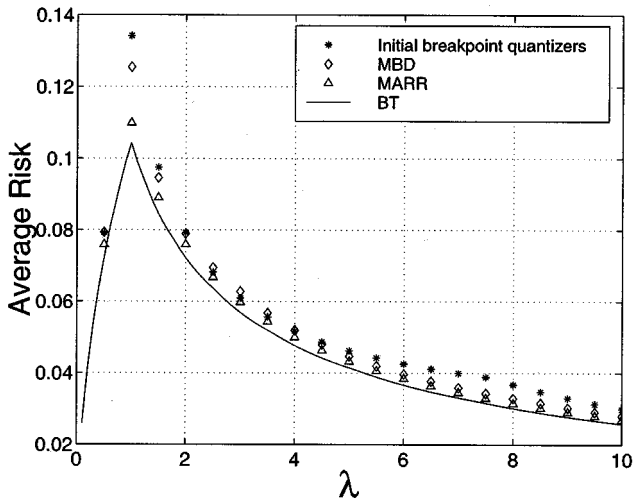


Fig. 5. Example 1. SNR = -5 dB, correlation coefficient  $r = 0.9$ ,  $R = 4$  bits/sample for each quantizer.

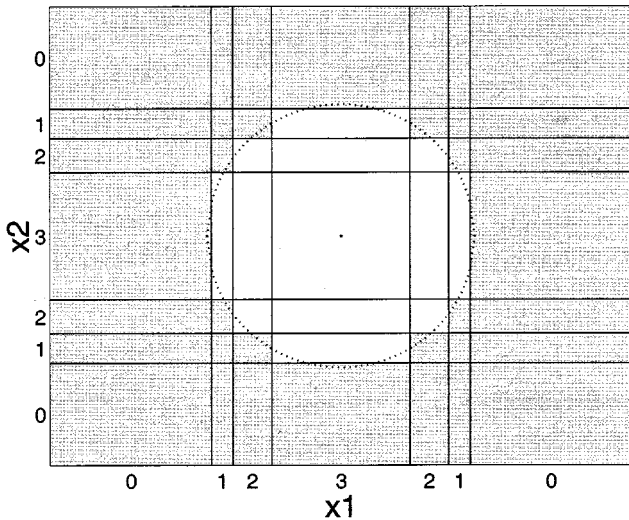


Fig. 6. Example 2. Two-bit quantization ( $\{0, 1, 2, 3\}$ ) and decision function (white = noise; gray = signal + noise) for white signal. Dotted line: Bayes border.

ance matrix (7) where, now,  $r$  represents the spatial correlation coefficient of the signal and  $\sigma^2$  is the per-channel SNR. In Tables II and III the performance of the same methods considered in Table I are reported for  $R = 2$  bits/sample. The quantization and the decision rule obtained

by our method for white signal and SNR = 10 dB are reported in Fig. 6. In this example, minor benefits are obtained by our proposed method.

V. CONCLUSIONS

We have presented a design method for rate-constrained distributed detection systems. It is based on an alternate optimization procedure that minimizes the average misclassification risk. Our method is inspired to the cooperative design of [5], the main difference being the cost function. Also, it is similar to the design of [6], where continuous parameter estimation is a concern, and can be seen as a special case of Scheme A of [7], where a general cost function is considered. Since the proposed optimization technique often gets trapped in poor minima, a repeated initialization strategy that allows, at least in the studied examples, to find good minima, has been worked out. The strength and the novelty of our proposed method is that it guarantees a local optimum of the average misclassification risk, which is the actual measure of performance of detection systems.

REFERENCES

- [1] P. K. Varshney, Guest Editor, "Special issue on data fusion," *Proc. IEEE*, vol. 85, pp. 1-199, Jan. 1997.
- [2] J. N. Tsitsiklis and M. Athans, "On the complexity of decentralized detection making and detection problems," *IEEE Trans. Automat. Contr.*, vol. AC-30, no. 5, pp. 440-446, May 1985.
- [3] R. R. Tenney and N. R. Sandell Jr., "Detection with distributed sensors," *IEEE Trans. Aerosp. Electron. Syst.*, vol. AES-17, pp. 501-510, July 1981.
- [4] Z. Chair and P. K. Varshney, "Optimal data fusion in multiple sensor detection systems," *IEEE Trans. Aerosp. Electron. Syst.*, vol. AES-22, pp. 98-101, Jan. 1986.
- [5] M. Longo, T. D. Lookbaugh, and R. M. Gray, "Quantization for decentralized hypothesis testing under communication constraints," *IEEE Trans. Inform. Theory*, vol. 36, pp. 241-255, Mar. 1990.
- [6] W. Lam and A. M. Reibman, "Design of quantizers for decentralized estimation systems," *IEEE Trans. Commun.*, vol. 41, pp. 1602-1605, Nov. 1993.
- [7] M. Di Bisceglie and M. Longo, "Decentralized encoding of a remote source," *Signal Processing*, vol. 55, pp. 15-29, 1996.