UNIVERSITY OF CALIFORNIA, SAN DIEGO

Quantization Over Noisy Channels and Bit Allocation

A dissertation submitted in partial satisfaction of the

requirements for the degree Doctor of Philosophy

in

Electrical Engineering (Communication Theory and Systems)

by

Benjamin Farber

Committee in charge:

Professor Kenneth Zeger, Chair
Professor Larry Carter
Professor Larry Milstein
Professor Alon Orlitsky
Professor Jack Wolf

2005

The dissertation of Benjamin Farber is approved, and it is acceptable in quality and form for publication on microfilm:

_____

_____

_____

_____

_____

_____

Chair

University of California, San Diego

2005

TABLE OF CONTENTS

LIST OF FIGURES

ACKNOWLEDGMENTS

I would have not been able to complete this work without help. I would like to thank my advisor Professor Ken Zeger for his guidance, his patient and persistent efforts to help me become a better writer and researcher, and for his many helpful suggestions and insights. I want to thank Professors Larry Carter, Larry Milstein, Alon Orlitsky, and Jack Wolf for serving on my committee. I must also thank my old roommate Richard Flynn for countless hours of useful advice and for being the best roommate I have ever had. From the Information Coding Laboratory alumni I am indebted to Tamás Frajka, Zsigmond Nagy, Greg Sherwood, Song Cen, Zsolt Kukorelly, and Qinghua Zhao for the help they gave me in many different forms, as well as their sympathy and companionship. Also in the Electrical and Computer Engineering department, I would like to thank Karol Previte, Charmaine Samahin, and M'lissa Michelson for their kind and efficient support of all my efforts at UCSD.

In addition, I would like to thank my family. Even from a distance, they were always there for me. I'm grateful for their support, love, and patience. Finally, I must thank my wife for her unlimited support and understanding. Without it, I would not have finished this work.

The text of Chapter 2, in full, is a reprint of the material as it appears in: Benjamin Farber and Kenneth Zeger, "Quantizers with Uniform Encoders and Channel Optimized Decoders," *IEEE Transactions on Information Theory*, vol. 50, no. 1, pp. 62–77, January 2004. The text of Chapter 3, in full, has been submitted for publication as:

Benjamin Farber and Kenneth Zeger, "Quantizers with Uniform Decoders and Channel Optimized Encoders," *IEEE Transactions on Information Theory*, April 14, 2004. The text of Chapter 4, in full, has been submitted for publication as: Benjamin Farber and Kenneth Zeger, "Quantization of Multiple Sources Using Nonnegative Integer Bit Allocation," *IEEE Transactions on Information Theory*, May 2005. I was the primary researcher and the co-author Kenneth Zeger listed in these publications directed and supervised the research which forms the basis for this dissertation.

VITA

1999	B.S. in Electrical Engineering, Cornell University
2001	M.S. in Electrical and Computer Engineering (Communication Theory and Systems), University of California, San Diego
1999-2004	Research Assistant, University of California, San Diego
2004-2005	Analytic Science Lead, Fair Isaac Corporation
2005	Ph.D. in Electrical and Computer Engineering (Communication Theory and Systems), University of California, San Diego

PUBLICATIONS

B. Farber, K. Zeger, "Quantizers with Uniform Encoders and Channel Optimized Decoders," *Proceedings, Data Compression Conference (DCC)*, pp. 292–301, Snowbird, Utah, March 2002.

B. Farber, K. Zeger, "Optimality of the Natural Binary Code for Quantizers with Channel Optimized Decoders," *Proceedings, 2003 IEEE International Symposium on Information Theory (ISIT)*, pp. 483, Yokohama, Japan, June 2003.

B. Farber, K. Zeger, "Quantizers with Uniform Encoders and Channel Optimized Decoders," *IEEE Transactions on Information Theory*, vol. 50, no. 1, pp. 62–77, January 2004.

B. Farber, K. Zeger, "Quantizers with Uniform Decoders and Channel Optimized Encoders," *IEEE Transactions on Information Theory* (submitted April 2004).

B. Farber, K. Zeger, "Cell Density Functions and Effective Channel Code Rates for Quantizers with Uniform Decoders and Channel Optimized Encoders," *Proceedings, 2004 IEEE International Symposium on Information Theory (ISIT)*, pp. 429, Chicago, Illinois, July 2004.

B. Farber, K. Zeger, "Quantization of Multiple Sources Using Integer Bit Allocation" *Proceedings, Data Compression Conference (DCC)*, pp. 368–377, Snowbird, Utah, March 2005.

B. Farber and K. Zeger, "Quantization of Multiple Sources Using Nonnegative Integer Bit Allocation," *IEEE Transactions on Information Theory* (submitted May 2005).

ABSTRACT OF THE DISSERTATION

Quantization Over Noisy Channels and Bit Allocation

by

Benjamin Farber

Doctor of Philosophy in Electrical Engineering

(Communication Theory and Systems)

University of California, San Diego, 2005

Professor Kenneth Zeger, Chair

In this dissertation we study two problems related to scalar quantization, namely quantization over a noisy channel and bit allocation. Scalar quantizers have been extensively studied for the case of a noiseless channel. However, their structure and performance is not well understood when operating over a noisy channel. The bit allocation problem is how to allocate a limited number of bits to a set of scalar quantizers so as to minimize the sum of their mean squared errors.

We first examine scalar quantizers with uniform encoders and channel optimized decoders for uniform sources and binary symmetric channels. We calculate the point density functions and the mean squared errors for several different index assignments. We also show that the Natural Binary Code is mean squared optimal among all possible index assignments, for all bit error rates, and all quantizer transmission rates. In contrast, we find that almost all index assignments perform poorly and have degenerate

codebooks.

Next, we study scalar quantizers with uniform decoders and channel optimized encoders for uniform sources and binary symmetric channels. We compute the number of empty cells in the quantizer encoder, the asymptotic cell distribution, and the effective channel code rates for two families of index assignments. Also, we demonstrate that the Natural Binary Code is sub-optimal for a large range of transmission rates and bit error probabilities. This contrasts with its known optimality when either both the encoder and decoder are not channel optimized, or when only the decoder is channel optimized.

Lastly, we consider bit allocation. The problem of asymptotically optimal bit allocation among a set of quantizers for a finite collection of sources was analytically solved in 1963 by Huang and Schultheiss. Their solution gives a real-valued bit allocation, however in practice, integer-valued bit allocations are needed. In 1966, Fox gave an algorithm for finding optimal nonnegative integer bit allocations. We prove that Fox's solution is equivalent to finding a nonnegative integer-valued vector closest in the Euclidean sense to the Huang-Schultheiss solution. Additionally, we derive upper and lower bounds on the deviation of the mean squared error using integer bit allocation from the mean squared error using optimal real-valued bit allocation.

Chapter 1

Introduction

To motivate this dissertation, consider the following toy problem. By transmitting only two bits, how can one most accurately convey an observation of a random variable X uniformly distributed on the interval $[0, 1]$ to a remote receiver? That is, the receiver forms an estimate Y of the observation X based on the received two bits from the transmitter. This is the problem of scalar quantization over a communications channel where performance is measured by the distortion between the original sample from the random variable and its reconstruction. Typically the distortion is measured by the mean squared error, $E[(X - Y)^2]$.

One approach to solving this example is the following. Evenly partition the interval $[0, 1]$ into four encoding cells: $[0, 1/4)$, $[1/4, 1/2)$, $[1/2, 3/4)$, and $[3/4, 1]$ and assign them indices 00, 01, 10, and 11, respectively. If the random variable X falls in a particular cell, then transmit its corresponding 2-bit index. This is an example of a quantizer encoder. At the receiver, take the two bits from the encoder and pick the corresponding encoding cell midpoint as the estimate Y of the original source random variable. Suppose, for example, the encoder transmitted 10 and these bits are observed error free by the receiver. Then the receiver picks the point $5/8$ as its estimate Y. This is an example of a quantizer decoder; its estimate Y is called a codepoint.

The combination of a quantizer encoder and a quantizer decoder works well when the communication between the encoder and the decoder is error free. It is easy to show that in this example the encoder and decoder described minimize the mean squared error for an error free channel.

What if the channel between the encoder and decoder is not perfect? For example, suppose the encoder transmits one bit at a time to the decoder, and each transmission has a fixed probability ϵ of being in error, independent of the previous transmissions. This is the problem of scalar quantization over a discrete noisy channel. For example, if $\epsilon = 0.1$, then a known numerical quantizer design algorithm yields the following encoding cells: $[0, 0.37)$, $[0.37, 0.5)$, $[0.5, 0.63)$, and $[0.63, 1]$ with indices 00, 01, 10, and 11, respectively, and corresponding (non-midpoint) codepoints equal to: 0.21, 0.46, 0.54, and 0.79, respectively.

In general, it is not known how to algorithmically find optimal quantizer encoders and quantizer decoders for transmission over a noisy channel. Also, almost nothing is known analytically about optimal quantizer encoders or quantizer decoders for noisy channels except certain necessary (but not sufficient) conditions they must satisfy (e.g. [4]).

As an approach to understanding quantization in the presence of channel noise, one can fix the quantizer encoder and study the resulting quantizer decoder or fix the quantizer decoder and study the resulting quantizer encoder. While this approach is sub-optimal, it can yield valuable insight into the problem of quantization over a noisy channel.

In the toy example we have been considering, if we fix the quantizer encoder to be uniform, i.e. encoding cells $[0, 1/4)$, $[1/4, 1/2)$, $[1/2, 3/4)$, and $[3/4, 1]$ with indices 00, 01, 10, and 11, respectively, it can be shown that the optimal decoder for $\epsilon = 0.1$ has corresponding codepoints of $1/5$, $2/5$, $3/5$, and $4/5$, respectively. Similarly, if we fix the quantizer decoder to be uniform, i.e. codepoints of $1/8$, $3/8$, $5/8$, and $7/8$, it can

be shown that the optimal encoder for $\epsilon = 0.1$ has encoding cells of $[0, 0.3)$, $[0.3, 0.5)$, $[0.5, 0.7)$, $[0.7, 1]$ with indices 00, 01, 10, and 11, respectively.

This toy example illustrates some of the main ideas considered in the first part of this thesis (Chapters 2 and 3). The mean squared error ($E[(X - Y)^2]$) in the toy example illustrates the idea of "loss." The receiver cannot perfectly determine the observed quantity at the source.

In general, there are two types of data compression, lossy and lossless. Each strives to reduces the number of bits required to describe a random source. An observation from a random source can be perfectly reconstructed from the output of a lossless data compression scheme, where as it cannot from a lossy data compression scheme. Lossy data compression is used to represent analog signals digitally (commonly known as analog-to-digital conversion). In this dissertation, we study only lossy data compression.

There are two main approaches to studying lossy data compression. One is to fix the transmission rate (i.e. the number of bits transmitted by the encoder to the decoder) and let the dimension or block length grow without bound. This approach was started by Shannon [5, 6] who laid the information theoretic foundation for the study of both lossy and lossless data compression with his ground breaking works in the 1948 and 1959. By considering asymptotically large block lengths, i.e. the number of samples from a random source described by one use of a quantizer encoder, he was able to derive bounds on the achievable performance of lossless and lossy compression schemes. He also showed that with infinite time and computing resources, there was nothing to be lost by separating the two problems of source coding, i.e. for a given random source and a perfect channel finding the best quantizer encoder and decoder, and channel coding, i.e. for a given random channel, finding the best quantizer encoder and decoder to combat channel noise.

Another approach to studying lossy data compression was started by Bennett [1]

in 1948. This approach is to consider the performance and structure of quantizers as their rate becomes asymptotically large. The assumption of large rate is often called *high resolution*. We use this approach since the structure and performance of high resolution quantizers can be a good guide for the design, analysis, and expected performance of more practical quantizers, i.e. quantizers with relatively small rate which are easy to implement. In particular, Chapters 2 and 3 examine the high resolution structure of families of quantizers under different assumptions on their quantizer encoders and decoders and in Chapter 4 we assume the performance of a scalar quantizer is given by a formula for the mean squared error of a generic high resolution quantizer.

In Chapters 2 and 3 we attempt to gain a better understanding of high resolution quantization theory over noisy channels. In both Chapters 2 and 3 we assume the source random variable to be quantized is uniformly distributed on the interval $[0, 1]$ and the quantizer encoder and quantizer decoder must communicate over a binary symmetric channel using index assignments. In Chapter 2, as the title suggests, we examine quantizers with uniform encoders and channel optimized decoders. This means we consider quantizer decoders whose choice of codepoint location has been optimized to the statistics of the channel. For such quantizers we determine the high resolution structure of the quantizer decoder and the mean squared error achieved by several different families of index assignments. Chapter 2 is a copy of a paper published in the IEEE Transactions on Information Theory. In Chapter 3, as the title suggests, we examine quantizers with uniform decoders and channel optimized encoders. This means we consider quantizer encoders whose choice of encoding cell boundaries has been optimized to the statistics of the channel. For such quantizers we determine the high resolution structure of the quantizer encoder and the mean squared error achieved by two different families of index assignments. Chapter 3 is a copy of a paper in revision for the IEEE Transactions on Information Theory.

In Chapter 4 we consider the bit allocation problem. This problem concerns how

to allocate a constrained number of bits among a set of quantizers to as to minimize the sum of their distortions.

Huang and Schultheiss [3] provided an optimal real-valued solution to this problem. However, applications generally impose integer-value constraints on the rates used. Unfortunately, it has been shown that finding optimal integer bit allocations is NP-hard (as the number of sources grows), via reduction to the multiple choice knapsack problem. In practice, authors have suggested using using combinatorial optimization methods such as integer linear programming or dynamic programming [2] or optimizing with respect to the convex hull of the quantizers' rate-versus-distortion curves to find bit allocations. There are also many algorithmic techniques in the literature for obtaining integer-valued bit allocations.

Despite the wealth of knowledge about finding integer bit allocations, there has been no published theoretical analysis comparing the performance of optimal bit allocations with integer constraints to the performance obtained using the real-valued allocations due to Huang and Schultheiss. In this thesis, we characterize optimal integer bit allocations as those that minimize the Euclidean distance to the solution proposed by Huang and Schulthiess. Also, we derive upper and lower bounds on the deviation of the sum of the component mean squared error's using integer bit allocation from the sum of the component mean squared error's using optimal real-valued bit allocation. Chapter 4 has been submitted for publication to the IEEE Transactions on Information Theory.

References

[1] W.R. Bennett, “Spectra of quantized signals,” *Bell System Technical Journal*, vol. 27, pp. 446–472, July 1948.

[2] B. Fox, “Discrete optimization via marginal analysis,” *Management Science*, vol. 13, no. 3, pp. 210–216, November, 1966.

[3] J.J. Huang and P.M. Schultheiss, “Block quantization of correlated Gaussian random variables,” *IEEE Transactions on Communications Systems*, vol. 11, pp. 289–296, September 1963.

[4] H. Kumazawa, M. Kasahara, and T. Namekawa, “A Construction of Vector Quantizers for Noisy Channels,” *Electronics and Engineering in Japan*, vol. 67-B, no. 4, pp. 39–47, 1984.

[5] C.E. Shannon, “A mathematical theory of communication,” *Bell System Technical Journal*, vol. 27, pp. 379–423, 623–656, July and October, 1948.

[6] C.E. Shannon, “Coding Theorems for a discrete source with a fidelity criterion,” *IRE National Convention Record*, Part 4, pp. 142–163, 1959.

Chapter 2

Quantizers with Uniform Encoders and Channel Optimized Decoders

Abstract

Scalar quantizers with uniform encoders and channel optimized decoders are studied for uniform sources and binary symmetric channels. It is shown that the Natural Binary Code and Folded Binary Code induce point density functions that are uniform on proper subintervals of the source support, whereas the Gray Code does not induce a point density function. The mean squared errors for the Natural Binary Code, Folded Binary Code, Gray Code, and for randomly chosen index assignments are calculated and the Natural Binary Code is shown to be mean squared optimal among all possible index assignments, for all bit error rates and all quantizer transmission rates. In contrast, it is shown that almost all index assignments perform poorly and have degenerate codebooks.

2.1 Introduction

The most basic source and quantizer are the uniform scalar source and the uniform scalar quantizer. If the source is uniform on $[0, 1]$, for example, then an n-bit

uniform quantizer has equally spaced encoding cells of size 2^{-n} and has equally spaced output points which are the centers of the encoding cells. For this source, the mean squared distortion of this quantizer is known exactly when there is no channel noise, and is known to be minimal among all quantizers.

In the presence of channel noise, one approach to improving system performance is to add explicit error control coding, so that some of the transmission rate is devoted towards source coding and some towards channel coding. Drawbacks of this include the added complexity and delay of channel decoding.

An alternative low-complexity approach in the presence of channel noise is to add to the quantizer an index assignment, which permutes the binary words associated with each encoding cell prior to transmission over the channel, and then unpermutes the binary words at the receiver prior to assigning a reproduction point at the output. The cells are assumed to be labeled in increasing order from left to right, before the index assignment. Examples of index assignments include the Natural Binary Code, the Folded Binary Code, and the Gray Code. The benefit of an index assignment is derived from the fact that reproduction codepoints that are relatively close on the real line can be assigned binary words which are close in the Hamming sense (i.e. in the number of same bits) on average. Thus when channel errors occur, the mean squared error impact on the quantizer is reduced.

Yamaguchi and Huang [8] and Huang [9] derived formulas for the mean squared error of uniform scalar quantizers and uniform sources for the Natural Binary Code, the Gray Code, and for a randomly chosen index assignment on a binary symmetric channel. They also asserted (without a published proof) the optimality of the Natural Binary Code for the binary symmetric channel. Crimmins et al. [1] studied the uniform scalar quantizer for the uniform source and proved the Yamaguchi-Huang assertion, that the Natural Binary Code is the best possible index assignment in the mean squared sense for the binary symmetric channel, for all bit error probabilities and all quantizer rates.

McLaughlin, Neuhoff, and Ashley [3] generalized this result for certain uniform vector quantizers and uniform vector sources. Other than these papers, there are no others presently known in the literature giving index assignment optimality results.

There have been some analytic studies on the performance of various index assignments. Hagen and Hedelin [7] used Hadamard transforms to study certain lattice-type quantizers with index assignments on noisy channels. Knagenhjelm and Agrell [10] introduced an analytic method of approximating the quality of an index assignment using Hadamard transforms. Skoglund [12] provided index assignment analysis for more general channels and sources. In [4] explicit mean squared error formulas were computed for uniform sources on binary asymmetric channels with various structured classes of index assignments. In [5] it was shown that for the uniform source and uniform quantizer the mean squared error resulting from a randomly chosen index assignment was, on average, equal in the limit of large n to that of the worst possible index assignment. In this sense the result showed that randomly chosen index assignments are asymptotically bad. A number of papers have also studied algorithmic techniques for designing good index assignments for particular sources and channels (see the citations in [6, p. 2372]).

While index assignments can improve the robustness of quantizers designed for noiseless channels to the degradation caused by channel noise, another low-complexity approach is to use quantizers whose encoders and/or decoders are designed for the channel's statistical behavior. It is known that an optimal quantizer for a noiseless channel must satisfy what are known as "nearest neighbor" and "centroid" conditions on its encoder and decoder, respectively [2]. For discrete memoryless channels it is known that an optimal quantizer must satisfy what we call "weighted nearest neighbor" and "weighted centroid" conditions on its encoder and decoder, respectively (see [11] for example). Even for uniform scalar sources, the resulting quantizers in general do not have uniform encoding cells nor equally spaced reproduction codepoints. In fact very little is presently understood analytically about quantizers for noisy channels beyond the

Natural Binary Code optimality results previously mentioned for uniform quantizers.

In the present paper, we attempt to move a step closer towards understanding optimal quantization for noisy channels by examining the structure of quantizers with uniform encoders and channel optimized decoders (i.e. that satisfy the weighted centroid condition), for uniform sources on $[0, 1]$ and for certain previously studied index assignments. In particular, we study the high resolution distribution of codepoints for such quantizers and the resulting distortions. Slightly more general, but notationally cumbersome results could also be easily obtained from our results by allowing the source to be confined to any bounded interval instead of just $[0, 1]$.

An important tool in analyzing the performance of quantizers is the concept of point density functions. Point density functions characterize the high resolution distribution of scalar quantizer codepoints. As a result, they provide insight about the asymptotic behavior of scalar quantizer codebooks and encoding cells. Point density functions also are useful in analyzing the distortion of quantizers. For example, Bennett's integral [2, p.163] gives the average distortion in the high resolution case for a nonuniform quantizer in terms of a point density function, source distribution, and size of the quantizer codebook (see [6] for more details). For uniform quantizers, the computation of a point density function is trivial. For nonuniform quantizers however, point density functions are not always guaranteed to exist, and when they do, their computation can be difficult.

Point density functions depend on the quantizer decoders. Channel optimized quantizer decoders, in turn, depend on the source, the quantizer encoder, the channel, and the index assignment. For this paper, we assume a uniform source on [0,1], a uniform quantizer encoder, a channel optimized quantizer decoder, and a binary symmetric channel with bit error probability ϵ. An index assignment maps source codewords to channel codewords. The quantizer has 2^n encoding cells, and index assignments are one-to-one maps from the index of an encoding cell to a binary word of length n. These

words are transmitted across the channel and decoded according to the weighted centroid condition.

Certain results we obtain are somewhat counter-intuitive. For example, we show that for a binary symmetric channel with bit error probability ϵ, quantizers using the Natural Binary Code index assignment and n bits of resolution have codepoints uniformly distributed on the interval $[\epsilon + \delta, 1 - \epsilon - \delta]$ where $\delta = (1 - 2\epsilon)/2^{n+1}$. This is peculiar in light of the fact that the source is uniformly distributed on the interval $[0, 1]$, and yet asymptotically as $n \to \infty$ no codepoints are located within a distance of ϵ from 0 or 1. The lack of codepoints in regions of positive source probability is due to the reduction in average distortion that results from moving codepoints closer to the source mean (by the weighted centroid condition), to avoid large jumps in Euclidean distance from channel errors. The weighted centroid condition dictates this movement of codepoints to minimize average distortion for a given quantizer encoder. A similar result occurs for the Folded Binary Code.

For the Gray Code index assignment, we show that in fact no point density function exists. In other words, the location of codepoints cannot be described according to a point density function as $n \to \infty$. The structure of the Gray Code simply does not allow the histogram of codepoint locations to converge to a smooth function in the limit of high resolution.

We also show that asymptotically, almost all index assignments give rise to quantizers which have almost all of their codepoints clustered very close to the source's mean value (i.e. $1/2$). Thus almost all index assignments are bad. As n grows, the clustering of codepoints becomes tighter and tighter. This contrasts with the Natural Binary Code and the Folded Binary Code cases where the codepoints remain uniformly distributed on proper subsets of $[0, 1]$ no matter how large n becomes. An additional curiosity we show is that among all possible index assignments, the Natural Binary Code is optimal despite its lack of codepoints within ϵ of 0 or 1.

Our main results for quantizers with uniform encoders and channel optimized decoders are the following. First, we show that the Natural Binary Code index assignment yields a uniform point density function on the interval $(\epsilon, 1-\epsilon)$ (Theorem 2.4), the Folded Binary Code index assignment yields a uniform point density function on a union of two proper subintervals of $[0,1]$ (Theorem 2.6), the Gray Code index assignment does not yield a point density function (Theorem 2.16), and an arbitrarily large fraction of all index assignments have an arbitrarily large fraction of codepoints arbitrarily close to the source mean as $n \to \infty$ (Theorem 2.20). Then we extend a result in [5] by showing that most index assignments are asymptotically bad (Theorem 2.22), and we extend results in [4], [8], and [9] by computing the mean squared error resulting from the Natural Binary Code (Theorem 2.24), the Folded Binary Code (Theorem 2.26), the Gray Code (Theorem 2.28), and a randomly chosen index assignment (Theorem 2.30). As comparisons, we state previously known mean squared error formulas for channel unoptimized decoders (i.e. that satisfy the centroid condition), for the Natural Binary Code (Theorem 2.23), the Folded Binary Code (Theorem 2.25), the Gray Code (Theorem 2.27), and for a randomly chosen index assignment (Theorem 2.29). Finally we extend the (uniform scalar quantizer) proof in [3] by showing that the Natural Binary Code is an optimal index assignment (Theorem 2.32).

The paper is organized as follows. Section 2.2 gives definitions and notation. Section 2.3 gives Natural Binary Code results, Section 2.4 gives Folded Binary Code results, Section 2.5 gives Gray Code results, Section 2.6 considers arbitrarily selected index assignments, and Section 2.7 gives distortion analysis.

2.2 Preliminaries

A *rate* n *quantizer* on $[0,1]$ is a mapping

$$\mathcal{Q} : [0,1] \longrightarrow \{y_n(0), y_n(1), \ldots, y_n(2^n-1)\}.$$

The real-valued quantities $y_n(i)$ are called *codepoints* and the set $\{y_n(0), \ldots, y_n(2^n-1)\}$ is called a *codebook*. For a noiseless channel, the quantizer $\mathcal{Q}$ is the composition of a *quantizer encoder* and a *quantizer decoder*. These are respectively mappings

$$\begin{aligned} \mathcal{Q}_e &: [0,1] \longrightarrow \{0,1,\ldots,2^n-1\} \\ \mathcal{Q}_d &: \{0,1,\ldots,2^n-1\} \longrightarrow \{y_n(0), y_n(1), \ldots, y_n(2^n-1)\} \end{aligned}$$

such that $\mathcal{Q}_d(i) = y_n(i)$ for all i. For each i the set $\mathcal{Q}^{-1}(y_n(i)) = \mathcal{Q}_e^{-1}(\mathcal{Q}_d^{-1}(y_n(i)))$ is called the ith encoding *cell*. The quantizer encoder is said to be *uniform* if for each i,

$$\mathcal{Q}^{-1}(y_n(i)) \supseteq (i2^{-n}, (i+1)2^{-n}).$$

The *nearest neighbor* cells of a rate n quantizer are the sets

$$R_n(i) = \{x : |y_n(i) - x| < |y_n(j) - x|, \ \ \forall j \neq i\}$$

for $0 \leq i \leq 2^n - 1$. Let m denote Lebesgue measure and for each i let

$$\mu_n(i) = m(R_n(i)).$$

A quantizer's encoder is said to satisfy the *nearest neighbor condition* if for each i,

$$\mathcal{Q}^{-1}(y_n(i)) \supseteq R_n(i).$$

That is, its encoding cells are essentially nearest neighbor cells (boundary points can be assigned arbitrarily).

For a given n, i, and source random variable X, the *centroid* of the ith cell of the quantizer $\mathcal{Q}$ is the conditional mean

$$c_n(i) = E[X|\mathcal{Q}(X) = y_n(i)].$$

The quantizer decoder is said to satisfy the *centroid condition* if the codepoints satisfy

$$y_n(i) = c_n(i)$$

for all i. A quantizer is *uniform* if the encoder is uniform and for each i the decoder codepoint $y_n(i)$ is the midpoint of the cell $\mathcal{Q}^{-1}(y_n(i))$. It is known that if a quantizer minimizes the mean squared error for a given source and a noiseless channel, then it satisfies the nearest neighbor and centroid conditions [2]. In particular, if the source is uniform, then a uniform quantizer satisfies the nearest neighbor and centroid conditions.

For a rate n quantizer, an *index assignment* π_n is a permutation of the set $\{0, 1, \ldots, 2^n - 1\}$. Let S_{2^n} denote the set of all $2^n!$ such permutations. For a noisy channel, a random variable $X \in [0, 1]$ is quantized by transmitting the index $I = \pi_n(\mathcal{Q}_e(X))$ across the channel, receiving index J from the channel, and then decoding the codepoint $y_n(\pi_n^{-1}(J)) = \mathcal{Q}_d(\pi_n^{-1}(J))$. We impose the following monotonicity constraint on quantizer encoders in order to be able to unambiguously refer to certain index assignments: For all $s, t \in [0, 1]$, if $s < t$, then $\mathcal{Q}_e(s) \leq \mathcal{Q}_e(t)$. The *mean squared error* (MSE) is defined as

$$D = E\left[(X - \mathcal{Q}_d(\pi_n^{-1}(J)))^2\right].$$

The random index J is a function of the source random variable X, the randomness in the channel, and the deterministic functions $\mathcal{Q}_e$ and π_n.

An alternative approach would be to view the quantizer encoder as the composition $\pi_n \cdot \mathcal{Q}_e$ and the quantizer decoder as the composition $\mathcal{Q}_d \cdot \pi_n^{-1}$, by relaxing the monotonicity assumption made above. This would remove the role of index assignments from the study of quantizers for noisy channels. However, we retain these encoder and decoder decompositions, as a convenient way to isolate the effects of index assignments, given known quantizer encoders and decoders.

Assume a binary symmetric channel with bit error probability ϵ. Denote the probability that index j was received, given that index i was sent by $p(j|i) = \epsilon^{H_n(i,j)}(1-\epsilon)^{n-H_n(i,j)}$ for $0 \leq \epsilon \leq 1/2$, where $H_n(i,j)$ is the Hamming distance between n-bit binary words i and j. Let $q(i|j)$ denote the probability that index i was sent, given that index j was received.

For a given source X, channel $p(\cdot|\cdot)$, index assignment π_n, and quantizer encoder, the quantizer decoder is said to satisfy the *weighted centroid condition* if the codepoints satisfy

$$y_n(j) \quad = \quad \sum_{i=0}^{2^n-1} c_n(i) q(\pi_n(i)|\pi_n(j)).$$

Throughout this paper we assume a uniform quantizer encoder, so the centroids of the encoder cells are given by

$$c_n(i) = (i + (1/2))2^{-n}$$

for $0 \leq i \leq 2^n - 1$. Since the source is uniform and the encoder cells are each of length 2^{-n}, we know that $p(j|i) = q(i|j)$ for all i and j. Hence the weighted centroid condition implies that

$$y_n(j) \quad = \quad \sum_{i=0}^{2^n-1} c_n(i) p(\pi_n(j)|\pi_n(i))$$

$$= \sum_{i=0}^{2^n-1} \frac{i+(1/2)}{2^n} \epsilon^{H_n(\pi_n(i),\pi_n(j))} (1-\epsilon)^{n-H_n(\pi_n(i),\pi_n(j))}.$$

For a given quantizer encoder and index assignment, we say the quantizer decoder is *channel optimized* if it satisfies the weighted centroid condition.

Notice that if the centroid condition is assumed, then the quantizer decoder $\mathcal{Q}_d$ does not depend on the index assignment, even though the mean squared error does. In contrast, if the weighted centroid condition is assumed, then the quantizer decoder $\mathcal{Q}_d$ does depend on the index assignment, as does the mean squared error. Thus, under the centroid condition, minimizing the mean squared error over all possible index assignments is carried out for a fixed quantizer decoder. However, under the weighted centroid condition, minimizing the mean squared error over all possible index assignments involves altering the quantizer decoder for each new index assignment.

For any set A, let the indicator function $\mathcal{I}_A(x)$ of A be

$$\mathcal{I}_A(x) = \begin{cases} 1 & \text{if } x \in A \\ 0 & \text{if } x \notin A \end{cases}.$$

For each n and each index assignment $\pi_n \in S_{2^n}$, define the function $\lambda_{\pi_n}^{(n)} : [0,1] \to [0,\infty)$ by

$$\lambda_{\pi_n}^{(n)}(x) = \sum_{i=0}^{2^n-1} \frac{1}{2^n \mu_n(i)} \mathcal{I}_{R_n(i)}(x).$$

For a sequence $\pi_n \in S_{2^n}$ (for $n = 1, 2, \ldots$) of index assignments, if there exists a function λ such that

$$\lambda(x) = \lim_{n\to\infty} \lambda_{\pi_n}^{(n)}(x)$$

for almost all $x \in [0,1]$ and $\int_0^1 \lambda(x)\, dx = 1$, then λ is said to be a *point density function* with respect to $\{\pi_n\}$.

The following lemma is a result of the fact that index assignments are permutations.

Lemma 2.1. *For any n and any index assignment $\pi_n \in S_{2^n}$,*

$$\sum_{j=0}^{2^n-1}(1-\epsilon)^{n-H_n(\pi_n(i),\pi_n(j))}\epsilon^{H_n(\pi_n(i),\pi_n(j))} = 1$$

for $0 \le j \le 2^n - 1$.

Let a *decoder optimized uniform quantizer* denote a rate n quantizer with a uniform encoder on $[0,1]$ and a channel optimized decoder, along with a uniform source on $[0,1]$, and a binary symmetric channel with bit error probability ϵ. Let a *channel unoptimized uniform quantizer* denote a rate n uniform quantizer on $[0,1]$, along with a uniform source on $[0,1]$, and a binary symmetric channel with bit error probability ϵ.

2.3 Natural Binary Code Index Assignment

For each n, the *Natural Binary Code* (NBC) is the index assignment defined by

$$\pi_n^{(NBC)}(i) = i \quad \text{for } 0 \le i \le 2^n - 1.$$

The following lemma is easy to prove and is used in the proof of Proposition 2.3.

Lemma 2.2.

$$H_{n+1}(i,j) = H_n(i,j) \quad \text{if} \quad 0 \le i,j \le 2^n - 1 \tag{2.1}$$

$$H_{n+1}(i+2^n,j) = H_n(i,j) + 1 \quad \text{if} \quad 0 \le i,j \le 2^n - 1 \tag{2.2}$$

$$H_{n+1}(i,j) = H_n(i,j-2^n) + 1 \quad \text{if} \quad 0 \le i \le 2^n - 1,\ 2^n \le j \le 2^{n+1} - 1 \tag{2.3}$$

$$H_{n+1}(i,j) = H_n(i-2^n,j-2^n) \quad \text{if} \quad 2^n \le i,j \le 2^{n+1} - 1. \tag{2.4}$$

Proposition 2.3. *The codepoints of a decoder optimized uniform quantizer with the Natural Binary Code index assignment are, for* $0 \leq j \leq 2^n - 1$

$$y_n(j) = \epsilon + (1 - 2\epsilon)c_n(j). \tag{2.5}$$

Proof. We use induction on n. The weighted centroid condition implies that

$$y_n(j) = 2^{-n-1} \sum_{j=0}^{2^n-1} (1-\epsilon)^{n-H_n(i,j)} \epsilon^{H_n(i,j)} (2i+1). \tag{2.6}$$

In particular, (2.6) gives

$$y_0(0) \;=\; 1/2$$

which satisfies (2.5). Now assume (2.5) is true for n and consider two cases for $n+1$.

If $0 \leq j \leq 2^n - 1$, then

$$\begin{aligned}
& y_{n+1}(j) \\
&= 2^{-n-2} \sum_{j=0}^{2^{n+1}-1} (1-\epsilon)^{n+1-H_{n+1}(i,j)} \epsilon^{H_{n+1}(i,j)} (2i+1) \\
&= (1-\epsilon) 2^{-n-2} \sum_{j=0}^{2^n-1} (1-\epsilon)^{n-H_n(i,j)} \epsilon^{H_n(i,j)} (2i+1) \\
&\quad + 2^{-n-2} \sum_{j=2^n}^{2^{n+1}-1} (1-\epsilon)^{n+1-H_{n+1}(i,j)} \epsilon^{H_{n+1}(i,j)} (2i+1) && (2.7) \\
&= \frac{(1-\epsilon) y_n(j)}{2} + 2^{-n-2} \sum_{j=0}^{2^n-1} (1-\epsilon)^{n-H_n(i,j)} \epsilon^{H_n(i,j)+1} (2i+1+2^{n+1}) && (2.8) \\
&= \frac{(1-\epsilon) y_n(j)}{2} + \frac{\epsilon y_n(j)}{2} + \frac{\epsilon}{2} && (2.9) \\
&= \epsilon + (1-2\epsilon) c_{n+1}(j) && (2.10)
\end{aligned}$$

where the first sum in (2.7) and the second sum in (2.8) follow from (2.1) and (2.2), respectively, (2.9) follows from Lemma 2.1, and (2.10) follows from the induction hypothesis.

If $2^n \leq j \leq 2^{n+1} - 1$, then

$$\begin{aligned}
y_{n+1}(j) &= 2^{-n-2} \sum_{j=0}^{2^{n+1}-1} (1-\epsilon)^{n+1-H_{n+1}(i,j)} \epsilon^{H_{n+1}(i,j)} (2i+1) \\
&= \epsilon\, 2^{-n-2} \sum_{j=0}^{2^n-1} (1-\epsilon)^{n-H_n(i,j-2^n)} \epsilon^{H_n(i,j-2^n)} (2i+1) \\
&\quad + 2^{-n-2} \sum_{j=2^n}^{2^{n+1}-1} (1-\epsilon)^{n+1-H_n(i-2^n,j-2^n)} \epsilon^{H_n(i-2^n,j-2^n)} (2i+1) \qquad (2.11) \\
&= \frac{y_n(j-2^n)\epsilon}{2} + 2^{-n-2} \sum_{j=0}^{2^n-1} (1-\epsilon)^{n+1-H_n(i,j-2^n)} \epsilon^{H_n(i,j-2^n)} (2i+1+2^{n+1}) \\
&= \frac{y_n(j-2^n)\epsilon}{2} + \frac{y_n(j-2^n)(1-\epsilon)}{2} + \frac{(1-\epsilon)}{2} \qquad (2.12) \\
&= \epsilon + (1-2\epsilon) c_{n+1}(j) \qquad (2.13)
\end{aligned}$$

where the sums in (2.11) follow from (2.3) and (2.4), respectively, (2.12) follows from Lemma 2.1, and (2.13) follows from the induction hypothesis. □

The following theorem shows that with the Natural Binary Code, the quantizer codepoints are uniformly distributed on a proper subinterval in the source's support region, in the limit of high resolution. As the channel improves (i.e. as $\epsilon \to 0$), the point density function approaches a uniform distribution on $[0, 1]$.

Theorem 2.4. *A sequence of decoder optimized uniform quantizers with the Natural Binary Code index assignment has a point density function given by*

$$\lambda(x) = \begin{cases} \frac{1}{1-2\epsilon} & \text{if } \epsilon < x < 1-\epsilon \\ 0 & \text{otherwise} \end{cases}.$$

Proof. Let

$$\lambda(x) \;=\; \begin{cases} \frac{1}{1-2\epsilon} & \text{if} \quad \epsilon < x < 1-\epsilon \\ 0 & \text{if} \quad 0 \le x \le \epsilon \;\text{ or }\; (1-\epsilon) \le x \le 1. \end{cases}$$

From (2.5), the codepoints satisfy $y_n(j+1) - y_n(j) = (1-2\epsilon)2^{-n}$ and thus are equally spaced apart. Also,

$$\begin{aligned} y_n(0) &= \epsilon + (1-2\epsilon)\, 2^{-n-1} \\ y_n(2^n-1) &= \epsilon + (1-2\epsilon)\left(1 - 2^{-n-1}\right). \end{aligned}$$

Thus,

$$\mu_n(i) = \begin{cases} (1-2\epsilon)\, 2^{-n} & \text{if} \quad 1 \le i \le 2^n - 2 \\ \epsilon + (1-2\epsilon)\, 2^{-n} & \text{if} \quad i = 0 \;\text{ or }\; i = 2^n - 1 \end{cases}$$

and therefore

$$\begin{aligned} \lambda^{(n)}_{\pi_n^{(NBC)}}(x) \quad &= \quad \begin{cases} \frac{1}{1-2\epsilon} & \text{if} \quad \epsilon + (1-2\epsilon)2^{-n} \le x < (1-\epsilon) - (1-2\epsilon)2^{-n} \\ \frac{1}{2^n\epsilon + (1-2\epsilon)} & \text{if} \quad 0 \le x < \epsilon + (1-2\epsilon)2^{-n} \\ & \quad \text{or} \;\; (1-\epsilon) - (1-2\epsilon)2^{-n} \le x \le 1 \end{cases} \\ &\longrightarrow \quad \lambda(x) \quad \text{as } n \to \infty. \end{aligned}$$

□

2.4 Folded Binary Code Index Assignment

For each n, the *Folded Binary Code* (FBC) is the index assignment defined by

$$\pi_n^{(FBC)}(i) = \begin{cases} 2^{n-1} - 1 - i & \text{if} \quad 0 \leq i \leq 2^{n-1} - 1 \\ i & \text{if} \quad 2^{n-1} \leq i \leq 2^n - 1. \end{cases}$$

The FBC is closely related to the NBC and has somewhat similar properties for decoder optimized uniform quantizers, as shown by Proposition 2.5 and Theorem 2.6. The proofs of Proposition 2.5 and Theorem 2.6 are similar to those of Proposition 2.3 and Theorem 2.4, respectively, and are therefore omitted for brevity.

Proposition 2.5. *The codepoints of a decoder optimized uniform quantizer with the Folded Binary Code index assignment are*

$$y_n(j) = \begin{cases} \frac{3\epsilon - 2\epsilon^2}{2} + (1-2\epsilon)^2 c_n(j) & \text{if} \quad 0 \leq j \leq 2^{n-1} - 1 \\ \frac{5\epsilon - 6\epsilon^2}{2} + (1-2\epsilon)^2 c_n(j) & \text{if} \quad 2^{n-1} \leq j \leq 2^n - 1. \end{cases}$$

The following theorem shows that with the Folded Binary Code, the quantizer codepoints are uniformly distributed on two proper subintervals of the source's support region, in the limit of high resolution. As the channel improves (i.e. as $\epsilon \to 0$), the point density function approaches a uniform distribution on $[0, 1]$.

Theorem 2.6. *A sequence of decoder optimized uniform quantizers with the Folded Binary Code index assignment has a point density function given by*

$$\lambda(x) = \begin{cases} \frac{1}{(1-2\epsilon)^2} & \text{if } \frac{3\epsilon - 2\epsilon^2}{2} < x < \frac{1 - \epsilon + 2\epsilon^2}{2} \text{ or } \frac{1 + \epsilon - 2\epsilon^2}{2} < x < 1 - \left(\frac{3\epsilon - 2\epsilon^2}{2}\right) \\ 0 & \text{otherwise} \end{cases}.$$

2.5 Gray Code Index Assignment

For each n, let $\pi_n^{(GC)}$ denote the Gray Code (GC) index assignment, recursively defined by

$$\begin{aligned}
\pi_1^{(GC)}(0) &= 0 \\
\pi_1^{(GC)}(1) &= 1 \\
\pi_{n+1}^{(GC)}(i) &= \begin{cases} \pi_n^{(GC)}(i) & \text{if } 0 \le i \le 2^n - 1 \\ \pi_n^{(GC)}(2^{n+1} - 1 - i) + 2^n & \text{if } 2^n \le i \le 2^{n+1} - 1. \end{cases}
\end{aligned}$$

Define the quantity

$$\hat{H}_n(i,j) = H(\pi_n^{(GC)}(i), \pi_n^{(GC)}(j)).$$

The definition of the Gray Code directly implies the following lemma.

Lemma 2.7.

$$\hat{H}_{n+1}(i,j) = \hat{H}_n(i,j) \quad \text{if} \quad 0 \le i,j \le 2^n - 1 \tag{2.14}$$

$$\hat{H}_{n+1}(i+2^n,j) = \hat{H}_n(2^n - 1 - i, j) + 1 \quad \text{if} \quad 0 \le i,j \le 2^n - 1 \tag{2.15}$$

$$\hat{H}_n(i,j) = \hat{H}_n(2^n - 1 - i, 2^n - 1 - j) \quad \text{if} \quad \begin{cases} 2^{n-1} \le j \le 2^n - 1 \\ 0 \le i \le 2^n - 1. \end{cases} \tag{2.16}$$

Lemma 2.8. *The codepoints of a decoder optimized uniform quantizer with the Gray Code index assignment satisfy*

$$y_n(j) = 1 - 2^{-n-1} \sum_{j=0}^{2^n-1} (1-\epsilon)^{n - \hat{H}_n(2^n-1-i,j)} \epsilon^{\hat{H}_n(2^n-1-i,j)} (2i+1)$$

for $0 \le j \le 2^n - 1$.

Proof.

$$
\begin{aligned}
& 2^{-n-1}\sum_{j=0}^{2^n-1}(1-\epsilon)^{n-\hat{H}_n(2^n-1-i,j)}\epsilon^{\hat{H}_n(2^n-1-i,j)}(2i+1) \\
&= \frac{1}{2}+2^{-n-1}\sum_{j=0}^{2^n-1}(1-\epsilon)^{n-\hat{H}_n(2^n-1-i,j)}\epsilon^{\hat{H}_n(2^n-1-i,j)}(2i-(2^n-1)) && (2.17)\\
&= 1-\left[\frac{1}{2}-2^{-n-1}\sum_{i=0}^{2^n-1}(1-\epsilon)^{n-\hat{H}_n(i,j)}\epsilon^{\hat{H}_n(i,j)}\left(2(2^n-1-i)-(2^n-1)\right)\right] \\
&= 1-\left[2^{-n-1}\sum_{i=0}^{2^n-1}(1-\epsilon)^{n-\hat{H}_n(i,j)}\epsilon^{\hat{H}_n(i,j)}(2i+1)\right] && (2.18)\\
&= 1-y_n(j) && (2.19)
\end{aligned}
$$

where (2.17) and (2.18) follow from Lemma 2.1. □

Corollary 2.9. *The codepoints of a decoder optimized uniform quantizer with the Gray Code index assignment satisfy*

$$y_n(j) = 1 - y_n(2^n - 1 - j)$$

for $2^{n-1} \leq j \leq 2^n - 1$.

Proof.

$$
\begin{aligned}
y_n(j) &= 2^{-n-1}\sum_{j=0}^{2^n-1}(1-\epsilon)^{n-\hat{H}_n(i,j)}\epsilon^{\hat{H}_n(i,j)}(2i+1) \\
&= 2^{-n-1}\sum_{j=0}^{2^n-1}(1-\epsilon)^{n-\hat{H}_n(2^n-1-i,2^n-1-j)}\epsilon^{\hat{H}_n(2^n-1-i,2^n-1-j)}(2i+1) && (2.20)\\
&= 1-y_n(2^n-1-j) && (2.21)
\end{aligned}
$$

where (2.20) follows from (2.16), and (2.21) follows from Lemma 2.8. □

For $0 \leq j \leq 2^n - 1$ and $1 \leq i \leq n$, let $b_n(j,i)$ be the i^{th} most significant bit of the n-bit binary representation of j. Then

$$j = \sum_{i=1}^{n} b_n(j,i) 2^{n-i}$$

and it follows that for $0 \leq j \leq 2^n - 1$,

$$b_n(j,i) = b_{n+1}(j, i+1) = 1 - b_n(2^n - 1 - j, i). \tag{2.22}$$

Proposition 2.10. *The codepoints of a decoder optimized uniform quantizer with the Gray Code index assignment are*

$$y_n(j) = \frac{1}{2} + \frac{1}{2} \sum_{i=1}^{n} (-1)^{b_n(j,i)+1} \left(\frac{1}{2} - \epsilon \right)^i \tag{2.23}$$

for $0 \leq j \leq 2^n - 1$.

Proof. We use induction on n. The weighted centroid condition implies that for all j,

$$y_n(j) = 2^{-n-1} \sum_{j=0}^{2^n-1} (1-\epsilon)^{n - \hat{H}_n(i,j)} \epsilon^{\hat{H}_n(i,j)} (2i+1).$$

For $n = 0$ this reduces to

$$y_0(0) \;\; = \;\; 1/2$$

which satisfies (2.23). Now assume Proposition 2.10 is true for n and consider two cases for $n+1$.

If $0 \leq j \leq 2^n - 1$, then

$y_{n+1}(j)$

$$
\begin{aligned}
&= 2^{-n-2}\sum_{j=0}^{2^{n+1}-1}(1-\epsilon)^{n+1-\hat{H}_{n+1}(i,j)}\epsilon^{\hat{H}_{n+1}(i,j)}(2i+1) \\
&= (1-\epsilon)2^{-n-2}\sum_{j=0}^{2^n-1}(1-\epsilon)^{n-\hat{H}_n(i,j)}\epsilon^{\hat{H}_n(i,j)}(2i+1) \\
&\quad + 2^{-n-2}\sum_{j=0}^{2^n-1}(1-\epsilon)^{n-\hat{H}_n(2^n-1-i,j)}\epsilon^{\hat{H}_n(2^n-1-i,j)+1}(2i+1+2^{n+1}) \qquad (2.24) \\
&= \frac{(1-\epsilon)y_n(j)}{2} + \frac{\epsilon[2^{n+1}+2^{n+1}(1-y_n(j))]}{2^{n+2}} \qquad (2.25) \\
&= \frac{1}{2}\left(\frac{1}{2}+\epsilon\right) + \left(\frac{1}{2}-\epsilon\right)y_n(j) - \frac{1}{2}\left(\frac{1}{2}-\epsilon\right) \\
&= \frac{1}{2}\left(\frac{1}{2}+\epsilon\right) + \left(\frac{1}{2}-\epsilon\right)\left[\frac{1}{2}+\frac{1}{2}\sum_{i=1}^{n}(-1)^{b_n(j,i)+1}\left(\frac{1}{2}-\epsilon\right)^i\right] \\
&\quad - \frac{1}{2}\left(\frac{1}{2}-\epsilon\right) \qquad (2.26) \\
&= \frac{1}{2}+\frac{1}{2}\left[-\left(\frac{1}{2}-\epsilon\right)+\sum_{i=2}^{n+1}(-1)^{b_{n+1}(j,i)+1}\left(\frac{1}{2}-\epsilon\right)^i\right] \qquad (2.27) \\
&= \frac{1}{2}+\frac{1}{2}\sum_{i=1}^{n+1}(-1)^{b_{n+1}(j,i)+1}\left(\frac{1}{2}-\epsilon\right)^i \qquad (2.28)
\end{aligned}
$$

where the sums in (2.24) follow from (2.14) and (2.15) respectively, (2.25) follows from Lemmas 2.1 and 2.8, (2.26) follows from the induction hypothesis, (2.27) follows from (2.22), and (2.28) follows from the fact that $b_{n+1}(j,1)=0$ whenever $0 \le j \le 2^n-1$.

If $2^n \le j \le 2^{n+1}-1$, then

$$
\begin{aligned}
y_{n+1}(j) &= 1-\left(\frac{1}{2}+\frac{1}{2}\sum_{i=1}^{n+1}(-1)^{b_{n+1}(2^{n+1}-1-j,i)+1}\left(\frac{1}{2}-\epsilon\right)^i\right) \qquad (2.29) \\
&= \frac{1}{2}+\frac{1}{2}\sum_{i=1}^{n+1}(-1)^{b_{n+1}(j,i)+1}\left(\frac{1}{2}-\epsilon\right)^i \qquad (2.30)
\end{aligned}
$$

where (2.29) follows from Corollary 2.9 and (2.28), and (2.30) follows from (2.22). □

To show that no point density function arises from the Gray Code index assign-

ment, we will show that $\lambda(x) = \lim_{n\to\infty} \lambda^{(n)}_{\pi_n^{(GC)}}(x) = 0$ almost everywhere on $[0,1]$, and hence $\int_0^1 \lambda(x)dx \neq 1$. To simplify notation, let $\lambda^{(n)}_{\pi_n^{(GC)}}$ be denoted by λ_n.

First, several preliminary results are necessary. In order to determine the asymptotic behavior of λ_n we examine the values of $\mu_n(i)$ and the relationship of $R_n(i)$ to $R_{n-1}(\lfloor i/2 \rfloor)$. For any fixed value of n there are groups of nearest neighbor cells with the same length. These groups and the properties of the cells in them are key to the subsequent results.

Lemma 2.12 describes each of these groups by the number of cells in the group and their common length. This is done by identifying a cell in each group whose index is of the form $i = 2^{n-k} - 1$ and considering its length. Lemma 2.11 shows that the codepoints are indexed in increasing order, and is used in the proof of Lemma 2.12.

Lemma 2.11. *The codepoints of a decoder optimized uniform quantizer with the Gray Code index assignment satisfy* $y_n(j+1) > y_n(j)$ *whenever* $0 \leq j \leq 2^n - 2$.

Proof. Let $k = \min\{k' : b_n(j,i) = 1, \forall i \geq k'\}$. Then the binary representation of j ends in exactly $n-k+1$ ones, and therefore

$$\begin{aligned} b_n(j,k-1) &= 0 \\ b_n(j,i) &= 1 \quad \text{for} \quad i \geq k \\ b_n(j+1,i) &= b_n(j,i) \quad \text{for} \quad 1 \leq i \leq k-2 \\ b_n(j+1,k-1) &= 1 \\ b_n(j+1,i) &= 0 \quad \text{for} \quad i \geq k. \end{aligned}$$

Thus, from (2.23), we have

$$\begin{aligned} & y_n(j+1) - y_n(j) \\ &= \frac{1}{2}\sum_{i=1}^{n} \left[(-1)^{b_n(j,i)} - (-1)^{b_n(j+1,i)}\right] \left(\frac{1}{2} - \epsilon\right)^i \end{aligned}$$

$$
\begin{aligned}
&= \frac{1}{2}\sum_{i=k}^{n}\left[(-1)^1-(-1)^0\right]\left(\frac{1}{2}-\epsilon\right)^i+\frac{1}{2}\left[(-1)^0-(-1)^1\right]\left(\frac{1}{2}-\epsilon\right)^{k-1}\\
&= \left(\frac{1}{2}-\epsilon\right)^{k-1}-\sum_{i=k}^{n}\left(\frac{1}{2}-\epsilon\right)^i\\
&> 0.
\end{aligned}
$$

□

Lemma 2.12. *For $1 \le k \le n-1$, a decoder optimized uniform quantizer with the Gray Code index assignment has 2^k nearest neighbor cells whose lengths equal $\mu_n(2^{n-k}-1)$.*

Proof. By Lemma 2.11, the codepoints $y_n(j)$ are increasing in j. Thus, for $1 \le i \le 2^n-2$,

$$\mu_n(i) = \frac{1}{2}\left(y_n(i+1)-y_n(i-1)\right).$$

Note that for $1 \le k \le n-1$, the binary representation of 2^{n-k} is $\underbrace{00\ldots01}_{k}\underbrace{00\ldots00}_{n-k}$ and the binary representation of $2^{n-k}-2$ is $\underbrace{00\ldots00}_{k}\underbrace{11\ldots10}_{n-k}$, which agree on the first $k-1$ digits and on the last digit. By (2.23), the difference between the ith and jth codepoints depends only on the locations in the binary representations of i and j where they differ. For all $w \in \{0,\ldots,2^{k-1}-1\}$ and $b \in \{0,1\}$, the binary representations of $2^{n-k}+w2^{n-k+1}+b$ and $2^{n-k}-2+w2^{n-k+1}+b$ agree in exactly the same locations that 2^{n-k} and $2^{n-k}-2$ agree in, and hence

$$
\begin{aligned}
\mu_n(2^{n-k}-1) &= \frac{1}{2}\left(y_n(2^{n-k})-y_n(2^{n-k}-2)\right)\\
&= \frac{1}{2}\left(y_n(2^{n-k}+w2^{n-k+1}+b)-y_n(2^{n-k}-2+w2^{n-k+1}+b)\right).
\end{aligned}
$$

The claimed 2^k nearest neighbor cells are thus $R(2^{n-k}-1+w2^{n-k+1}+b)$. □

The next lemma computes $\mu_n(i)$ for $0 \leq i \leq 2^n - 1$. By Lemma 2.12, it suffices to consider the lengths of $R_n(0)$, $R_n(2^n - 1)$, and $R_n(2^{n-k} - 1)$ for $1 \leq k \leq n - 1$.

Lemma 2.13. *For a decoder optimized uniform quantizer with the Gray Code index assignment,*

$$\mu_n(0) = \mu_n(2^n - 1) = \frac{\epsilon + \frac{1}{2}(\frac{1}{2} - \epsilon)^n}{\frac{1}{2} + \epsilon} \tag{2.31}$$

and for $1 \leq k \leq n - 1$,

$$\mu_n(2^{n-k} - 1) = \frac{\epsilon(\frac{1}{2} - \epsilon)^k + \frac{1}{2}(\frac{1}{2} - \epsilon)^n}{\frac{1}{2} + \epsilon}.$$

Proof. By Corollary 2.9, Lemma 2.11, and the definitions of $R_n(0)$ and $R_n(2^n - 1)$,

$$\begin{aligned} \mu_n(2^n - 1) &= 1 - \frac{1}{2}(y_n(2^n - 2) + y_n(2^n - 1)) \\ &= 1 - \frac{1}{2}(1 - y_n(1) + 1 - y_n(0)) \\ &= \mu_n(0). \end{aligned}$$

Since the n-bit binary representations of 0 and 1 differ only in the least significant bit,

$$\begin{aligned} \mu_n(0) &= \frac{1}{2}(y_n(0) + y_n(1)) \\ &= \frac{1}{2} + \frac{1}{2}\sum_{i=1}^{n-1}(-1)^{b_n(0,i)+1}\left(\frac{1}{2} - \epsilon\right)^i \qquad (2.32) \\ &= \frac{1}{2} - \frac{1}{2}\left(\frac{1 - (\frac{1}{2} - \epsilon)^n}{\frac{1}{2} + \epsilon} - 1\right) \\ &= \frac{\epsilon + \frac{1}{2}(\frac{1}{2} - \epsilon)^n}{\frac{1}{2} + \epsilon} \end{aligned}$$

where (2.32) follows from (2.23).

Recall from the proof of Lemma 2.12 that $\mu_n(2^{n-k} - 1) = \frac{1}{2}(y_n(2^{n-k}) - y_n(2^{n-k} - 2))$ and that the binary representations of 2^{n-k} and $2^{n-k} - 2$ are

$\underbrace{00\ldots01}_{k}\underbrace{00\ldots00}_{n-k}$ and $\underbrace{00\ldots00}_{k}\underbrace{11\ldots10}_{n-k}$ respectively. Combining this information with (2.23) gives

$$\begin{aligned}
&\mu_n(2^{n-k}-1)\\
&= \frac{1}{2}(y_n(2^{n-k}) - y_n(2^{n-k}-2))\\
&= \frac{1}{4}\sum_{i=1}^{n}(-1)^{b_n(2^{n-k},i)+1}\left(\frac{1}{2}-\epsilon\right)^i - \frac{1}{4}\sum_{i=1}^{n}(-1)^{b_n(2^{n-k}-2,i)+1}\left(\frac{1}{2}-\epsilon\right)^i\\
&= \frac{1}{2}\left(\frac{1}{2}-\epsilon\right)^k - \frac{1}{2}\sum_{i=k+1}^{n-1}\left(\frac{1}{2}-\epsilon\right)^i\\
&= \frac{\epsilon(\frac{1}{2}-\epsilon)^k + \frac{1}{2}(\frac{1}{2}-\epsilon)^n}{\frac{1}{2}+\epsilon}.
\end{aligned}$$

□

The next result follows directly from Lemma 2.13 and will be important in determining the behavior of λ_n as $n \to \infty$.

Corollary 2.14. *For a decoder optimized uniform quantizer with the Gray Code index assignment,*

$$\lim_{n\to\infty}\frac{1}{2^n\mu_n(0)} = \lim_{n\to\infty}\frac{1}{2^n\mu_n(2^n-1)} = 0$$

and for each fixed $k \geq 1$,

$$\lim_{n\to\infty}\frac{1}{2^n\mu_n(2^{n-k}-1)} = 0. \tag{2.33}$$

Define the sets

$$\begin{aligned}
E_{n,k} &= \bigcup_{i:\mu_n(i)=\mu_n(2^{n-k}-1)} R_n(i) \qquad \text{for } 1 \leq k \leq n-1\\
F_n &= R_n(0) \cup R_n(2^n-1) \qquad \text{for } n \geq 1
\end{aligned}$$

and note that $E_{n,k}$ and F_n are disjoint for all k and n.

Lemma 2.15. *For a decoder optimized uniform quantizer with the Gray Code index assignment,*

$$
\begin{aligned}
&(i) \quad R_1(0) \supset R_2(0) \supset R_3(0) \supset \cdots \\
&(ii) \quad R_1(2^1-1) \supset R_2(2^2-1) \supset R_3(2^3-1) \supset \cdots \\
&(iii) \quad F_1 \supset F_2 \supset F_3 \supset \cdots \\
&(iv) \quad \forall\, k \geq 1\,, E_{k+1,k} \supset E_{k+2,k} \supset E_{k+3,k} \supset \cdots \\
&(v) \quad m\left(\bigcap_{n=k+1}^{\infty} E_{n,k}\right) = \frac{\epsilon(1-2\epsilon)^k}{\frac{1}{2}+\epsilon}.
\end{aligned}
$$

Proof. By Lemma 2.13,

$$\frac{y_n(0)+y_n(1)}{2} = \frac{\epsilon + \frac{1}{2}(\frac{1}{2}-\epsilon)^n}{\frac{1}{2}+\epsilon}$$

which is decreasing in n. This proves part (i) and also shows that (using Corollary 2.9)

$$\frac{y_n(2^n-2)+y_n(2^n-1)}{2} = \frac{1-y_n(1)+1-y_n(0)}{2}$$

is increasing in n, thus proving part (ii). Part (iii) follows directly from parts (i) and (ii).

To prove part (iv), first note that (2.23) implies that for $0 \leq i \leq 2^{n-1}-1$,

$$
\begin{aligned}
y_n(2i) &= y_{n-1}(i) - \frac{1}{2}\left(\frac{1}{2}-\epsilon\right)^n \\
y_n(2i+1) &= y_{n-1}(i) + \frac{1}{2}\left(\frac{1}{2}-\epsilon\right)^n.
\end{aligned}
$$

Also assume without loss of generality that

$$\{x : |y_n(i)-x| = |y_n(i+1)-x|\} \subseteq R_n(i+1).$$

Suppose $1 \leq i \leq 2^n - 2$ and $n \geq 2$.

If i is even (say $i = 2j$), then

$$\begin{aligned} R_n(i) = R_n(2j) &= \left[\frac{y_n(2(j-1)+1) + y_n(2j)}{2}, \frac{y_n(2j) + y_n(2j+1)}{2}\right) \\ &= \left[\frac{y_{n-1}(j-1) + y_{n-1}(j)}{2}, y_{n-1}(j)\right) \\ &\subset R_{n-1}(j) = R_{n-1}(i/2) \end{aligned} \tag{2.34}$$

where (2.34) follows from the definition of $R_{n-1}(i)$.

If i is odd (say $i = 2j + 1$), then

$$\begin{aligned} R_n(i) = R_n(2j+1) &= \left[\frac{y_n(2j) + y_n(2j+1)}{2}, \frac{y_n(2j+1) + y_n(2j+2)}{2}\right) \\ &= \left[y_{n-1}(j), \frac{y_{n-1}(j) + y_{n-1}(j+1)}{2}\right) \\ &\subset R_{n-1}(j) = R_{n-1}((i-1)/2) \end{aligned} \tag{2.35}$$

where (2.35) follows from the definition of $R_{n-1}(i)$.

For each cell $R_n(i)$ in $E_{n,k}$ with $1 \leq k \leq n-1$, the proof of Lemma 2.12 shows that i is of the form $i = 2^{n-k} - 1 + w2^{n-k+1} + b$ where $w \in \{0, \ldots, 2^{k-1} - 1\}$ and $b \in \{0, 1\}$. If $R_n(i) \subset E_{n,k}$ and i is even, then $b = 1$ and

$$i = 2^{n-k} + w2^{n-k+1}$$

or equivalently

$$\frac{i}{2} = 2^{(n-1)-k} + w2^{(n-1)-k+1}$$

which implies $R_{n-1}(i/2) \subset E_{n-1,k}$. (2.34) shows that $R_n(i) \subset R_{n-1}(i/2)$, and hence $R_n(i) \subset E_{n-1,k}$.

Likewise if $R_n(i) \subset E_{n,k}$ and i is odd, then $b = 0$ and

$$i = 2^{n-k} - 1 + w2^{n-k+1}$$

or equivalently

$$\frac{(i-1)}{2} = 2^{(n-1)-k} - 1 + w2^{(n-1)-k+1}$$

which implies $R_{n-1}((i-1)/2) \subset E_{n-1,k}$. (2.35) shows that $R_n(i) \subset R_{n-1}((i-1)/2)$, and hence $R_n(i) \subset E_{n-1,k}$. Therefore,

$$E_{n,k} = \bigcup_{i:\mu_n(i)=\mu_n(2^{n-k}-1)} R_n(i) \quad \subset E_{n-1,k}$$

proving part (iv).

Since $\{E_{n,k}\}_{n=k+1}^{\infty}$ is a decreasing sequence of bounded sets (for each fixed k) by part (iv),

$$\begin{aligned} m\left(\bigcap_{n=k+1}^{\infty} E_{n,k}\right) &= \lim_{n\to\infty} m(E_{n,k}) \\ &= \lim_{n\to\infty} m\left(\bigcup_{i:\mu_n(i)=\mu_n(2^{n-k}-1)} R_n(i)\right) \\ &= \lim_{n\to\infty} \sum_{i:\mu_n(i)=\mu_n(2^{n-k}-1)} m(R_n(i)) \\ &= \lim_{n\to\infty} 2^k \left(\frac{\epsilon(\frac{1}{2}-\epsilon)^k + \frac{1}{2}(\frac{1}{2}-\epsilon)^n}{\frac{1}{2}+\epsilon}\right) \qquad (2.36) \\ &= \frac{\epsilon(1-2\epsilon)^k}{\frac{1}{2}+\epsilon}, \end{aligned}$$

where (2.36) follows from Lemmas 2.12 and 2.13. This proves part (v). □

The following theorem shows that the sequence of functions $\lambda^{(n)}_{\pi_n^{(GC)}}$ does not converge to a point density function as $n \to \infty$.

Theorem 2.16. *A sequence of decoder optimized uniform quantizers with the Gray Code index assignment does not have a point density function.*

Proof. We construct disjoint sets $E_k \subset [0,1]$ whose union has measure 1 and for which $\lim_{n\to\infty} \lambda_n(x) = 0$ for all $x \in E_k$ and for all k.

Let $E_0 = \bigcap_{n=1}^{\infty} F_n$. Then for any n and any $x \in E_0$, either $x \in R_n(0)$ or $x \in R_n(2^n - 1)$, and therefore $\lambda_n(x) = 1/(2^n \mu_n(0)) = 1/(2^n \mu_n(2^n - 1))$ by Lemma 2.13. Hence for any $x \in E_0$, $\lim_{n\to\infty} \lambda_n(x) = \lim_{n\to\infty} 1/(2^n \mu_n(0)) = 0$, by Corollary 2.14.

Let $E_k = \bigcap_{n=k+1}^{\infty} E_{n,k}$ for $k \geq 1$. Then for any n and k such that $n \geq k+1$ and for any $x \in E_k$, there exists an i such that $x \in R_n(i)$ and $\mu_n(i) = \mu_n(2^{n-k} - 1)$, which implies $\lambda_n(x) = 1/(2^n \mu_n(2^{n-k} - 1))$. Hence for any $x \in E_k$, $\lim_{n\to\infty} \lambda_n(x) = \lim_{n\to\infty} 1/(2^n \mu_n(2^{n-k} - 1)) = 0$, by Corollary 2.14.

Lemma 2.15(v) shows that E_k is nonempty for all $k \geq 1$. It will be shown below that E_0 is nonempty.

E_0 and E_k are disjoint for all $k \geq 1$, since $E_{n,k}$ and F_n are disjoint for all k and n. The sets E_k are disjoint for $k \geq 1$, for otherwise $E_{n,i}$ and $E_{n,j}$ would intersect for some n and some $i \neq j$. Therefore,

$$\begin{aligned}
m\left(\bigcup_{k=0}^{\infty} E_k\right) &= \sum_{k=0}^{\infty} m(E_k) \\
&= m\left(\bigcap_{n=1}^{\infty} F_n\right) + \sum_{k=1}^{\infty} m\left(\bigcap_{n=k+1}^{\infty} E_{n,k}\right) \\
&= \lim_{n\to\infty} m(F_n) + \sum_{k=1}^{\infty} \frac{\epsilon(1-2\epsilon)^k}{\frac{1}{2}+\epsilon} \qquad (2.37) \\
&= \frac{2\epsilon}{\frac{1}{2}+\epsilon} + \frac{\epsilon}{\frac{1}{2}+\epsilon}\left(\frac{1}{2\epsilon} - 1\right) \qquad (2.38) \\
&= 1
\end{aligned}$$

where the first term in (2.37) follows from Lemma 2.15(iii) and the boundedness of

$R_n(0)$ and $R_n(2^n-1)$, the second term in (2.37) follows from Lemma 2.15(v), and the first term in (2.38) follows from (2.31). Thus the set $\{x \in [0,1] : \lim_{n\to\infty} \lambda_n(x) \neq 0\}$ has measure 0 since it is a subset of $(\cup_{k=0}^{\infty} E_k)^c \cap [0,1]$. □

2.6 Randomly Chosen Index Assignments

Suppose for each $n \geq 1$ an index assignment Π_n is chosen uniformly at random from the set of all $2^n!$ index assignments. Then λ does not exist in a deterministic sense as the limit of $\lambda_{\Pi_n}^{(n)}$. However, the distribution of codepoints can still be characterized probabilistically.

Proposition 2.17. *Suppose an index assignment is chosen uniformly at random for a decoder optimized uniform quantizer. Then for all j, the expected value of the jth codepoint is*

$$E[y_n(j)] = \frac{1}{2} + \left(c_n(j) - \frac{1}{2}\right)\left(\frac{1}{1-2^{-n}}\right)\left((1-\epsilon)^n - 2^{-n}\right).$$

Proof. Let $\delta = \epsilon/(1-\epsilon)$ and note that $(1-\epsilon)(1+\delta) = 1$. Then,

$$\begin{aligned}
&E[y_n(j)] \\
&= \sum_{i=0}^{2^n-1} c_n(i) E[p(\Pi_n(j)|\Pi_n(i))] \\
&= \sum_{i=0}^{2^n-1} c_n(i) \cdot \frac{1}{2^n!} \sum_{\pi_n \in S_{2^n}} \epsilon^{H_n(\pi_n(i),\pi_n(j))} (1-\epsilon)^{n - H_n(\pi_n(i),\pi_n(j))} \\
&= \frac{(1-\epsilon)^n}{2^n!} \sum_{i=0}^{2^n-1} c_n(i) \sum_{\pi_n \in S_{2^n}} \delta^{H_n(\pi_n(i),\pi_n(j))} \\
&= \frac{(1-\epsilon)^n}{2^n!} \left(2^n! c_n(j) + \sum_{i \neq j} c_n(i) \sum_{\pi_n \in S_{2^n}} \delta^{H_n(\pi_n(i),\pi_n(j))} \right)
\end{aligned}$$

$$
\begin{aligned}
&= (1-\epsilon)^n \left(c_n(j) + \left(\sum_{i=0}^{2^n-1} c_n(i) - c_n(j) \right) \left(\frac{2^n(2^n-2)!}{2^n!} \sum_{k=1}^{n} \binom{n}{k} \delta^k \right) \right) && (2.39)\\
&= (1-\epsilon)^n \left(c_n(j) + \left(2^{n-1} - c_n(j)\right) \frac{(1+\delta)^n - 1}{(2^n-1)} \right) && (2.40)\\
&= \frac{1}{2} + \left(c_n(j) - \frac{1}{2} \right) \left(\frac{1}{1-2^{-n}} \right) \left((1-\epsilon)^n - 2^{-n} \right).
\end{aligned}
$$

To justify (2.39), consider the following observations. Suppose $i \neq j$. There are 2^n possible values $\pi_n(j)$ can have, and for each one there are $2^n - 1$ values $\pi_n(i)$ can take, $\binom{n}{k}$ of which must have Hamming distance k from $\pi_n(j)$. Given any of the $2^n(2^n-1)$ possible choices of $\pi_n(j)$ and $\pi_n(i)$, there are $(2^n-2)!$ ways to assign the remaining index assignment words. (2.40) follows from the fact that $\sum_{i=0}^{2^n-1} c_n(i) = 2^{-n} \sum_{i=0}^{2^n-1} (i + (1/2)) = 2^{n-1}$. □

With Proposition 2.17 the variance of the jth codepoint is

$$
\begin{aligned}
&\mathrm{Var}(y_n(j))\\
&= \mathrm{Var}\left(y_n(j) - \frac{1}{2} \right)\\
&= E\left[\left(y_n(j) - \frac{1}{2} \right)^2 \right] - \left[\left(c_n(j) - \frac{1}{2} \right) \left(\frac{1}{1-2^{-n}} \right) \left((1-\epsilon)^n - 2^{-n} \right) \right]^2. && (2.41)
\end{aligned}
$$

The motivation for the form of (2.41) will become clear in the proof of Theorem 2.20. Evaluation of the expectation in (2.41) yields Proposition 2.19 below.

Lemma 2.18.

$$
\sum_{i=0}^{2^n-1} \left(c_n(i) - \frac{1}{2} \right)^2 = \frac{1}{12} \left(2^n - 2^{-n} \right).
$$

Proof.

$$\begin{aligned}
&\sum_{i=0}^{2^n-1}\left(c_n(i)-\frac{1}{2}\right)^2 \\
&= \sum_{i=0}^{2^n-1}\left(2^{-n}\left(i+\frac{1}{2}\right)-\frac{1}{2}\right)^2 \\
&= 2^{-2n}\left[\sum_{i=0}^{2^n-1} i^2-(2^n-1)\sum_{i=0}^{2^n-1} i+2^n\left(\frac{2^n-1}{2}\right)^2\right] \\
&= 2^{-2n}\left[\frac{2^n(2^n-1)(2^{n+1}-1)}{6}-(2^n-1)\frac{2^n(2^n-1)}{2}+2^n\left(\frac{2^n-1}{2}\right)^2\right] \\
&= \frac{1}{12}\left(2^n-2^{-n}\right).
\end{aligned}$$

□

Proposition 2.19. *Suppose for each n, an index assignment is chosen uniformly at random for the nth quantizer (of rate n) in a sequence of decoder optimized uniform quantizers. Then for all j, the variance of the jth codepoint decays to zero at the rate* $\mathrm{Var}(y_n(j)) = O(2^{-\beta n})$ *as* $n \to \infty$, *where* $\beta = -\log_2(1-2\epsilon+2\epsilon^2)$.

Proof. Recall from (2.41) that the variance of $y_n(j)$ is

$$\begin{aligned}
&\mathrm{Var}(y_n(j)) \\
&= \mathrm{Var}\left(y_n(j)-\frac{1}{2}\right) \\
&= E\left[\left(y_n(j)-\frac{1}{2}\right)^2\right]-\left[\left(c_n(j)-\frac{1}{2}\right)\left(\frac{1}{1-2^{-n}}\right)\left((1-\epsilon)^n-2^{-n}\right)\right]^2 \quad (2.42)
\end{aligned}$$

whose second term goes to zero as $O((1-\epsilon)^{2n})$ when $n \to \infty$. Expanding the first term of (2.42) yields

$$E\left[\left(y_n(j)-\frac{1}{2}\right)^2\right]$$

$$
\begin{aligned}
&= E\left[\left(\sum_{i=0}^{2^n-1}\left(c_n(i)-\frac{1}{2}\right)p(\Pi_n(j)|\Pi_n(i))\right)^2\right] \\
&= \sum_{i=0}^{2^n-1}\sum_{l=0}^{2^n-1}\left(c_n(i)-\frac{1}{2}\right)\left(c_n(l)-\frac{1}{2}\right)E[p(\Pi_n(j)|\Pi_n(i))p(\Pi_n(j)|\Pi_n(l))] \\
&= (1-\epsilon)^{2n}\cdot \\
&\quad \sum_{i=0}^{2^n-1}\sum_{l=0}^{2^n-1}\left(c_n(i)-\frac{1}{2}\right)\left(c_n(l)-\frac{1}{2}\right)E\left[\delta^{H_n(\Pi_n(i),\Pi_n(j))+H_n(\Pi_n(l),\Pi_n(j))}\right]. \quad (2.43)
\end{aligned}
$$

We consider four cases. The computation in the last three cases is justified by an argument similar to the one used to justify (2.39).

(1) If $i = l = j$, then

$$E\left[\delta^{H_n(\Pi_n(i),\Pi_n(j))+H_n(\Pi_n(l),\Pi_n(j))}\right] = 1.$$

(2) If $i = l \neq j$, then

$$
\begin{aligned}
E\left[\delta^{H_n(\Pi_n(i),\Pi_n(j))+H_n(\Pi_n(l),\Pi_n(j))}\right] &= \frac{1}{2^n!}\sum_{\pi_n\in S_{2^n}}\delta^{2H_n(\pi_n(i),\pi_n(j))} \\
&= \frac{2^n(2^n-2)!}{2^n!}\sum_{r=1}^{n}\delta^{2r}\binom{n}{r} \\
&= \frac{(1+\delta^2)^n-1}{2^n-1}.
\end{aligned}
$$

(3) If $i \neq l$, $i \neq j$, and $l \neq j$, then

$$
\begin{aligned}
&E\left[\delta^{H_n(\Pi_n(i),\Pi_n(j))+H_n(\Pi_n(l),\Pi_n(j))}\right] \\
&= \frac{1}{2^n!}\sum_{\pi_n\in S_{2^n}}\delta^{H_n(\pi_n(i),\pi_n(j))+H_n(\pi_n(l),\pi_n(j))} \\
&= \frac{2^n(2^n-3)!}{2^n!}\sum_{k=1}^{n}\sum_{m=1}^{n}\delta^{k+m}\binom{n}{k}\cdot\begin{cases}\binom{n}{m} & \text{if } m\neq k\\ \binom{n}{m}-1 & \text{if } m=k\end{cases}
\end{aligned}
$$

$$
\begin{aligned}
&= \frac{1}{(2^n-1)(2^n-2)}\left\{\left[\sum_{k=1}^{n}\binom{n}{k}\delta^k\right]^2-\sum_{k=1}^{n}\binom{n}{k}\delta^{2k}\right\}\\
&= \frac{((1+\delta)^n-1)^2-((1+\delta^2)^n-1)}{(2^n-1)(2^n-2)}\\
&= \frac{2+(1+\delta)^{2n}-2(1+\delta)^n-(1+\delta^2)^n}{(2^n-1)(2^n-2)}.
\end{aligned}
$$

(4) If $j=i\neq l$ (or $j=l\neq i$), then

$$
\begin{aligned}
E\left[\delta^{H_n(\Pi_n(i),\Pi_n(j))+H_n(\Pi_n(l),\Pi_n(j))}\right] &= \frac{1}{2^n!}\sum_{\pi_n\in S_{2^n}}\delta^{H_n(\pi_n(l),\pi_n(j))}\\
&= \frac{2^n(2^n-2)}{2^n!}\sum_{r=1}^{n}\delta^r\binom{n}{r}\\
&= \frac{(1+\delta)^n-1}{2^n-1}.
\end{aligned}
$$

Thus (2.43) can be written in terms of the four cases as

$$
\begin{aligned}
&E\left[\left(y_n(j)-\frac{1}{2}\right)^2\right]\\
&= (1-\epsilon)^{2n}\left(c_n(j)-\frac{1}{2}\right)^2\\
&\quad+(1-\epsilon)^{2n}\left(\frac{(1+\delta^2)^n-1}{2^n-1}\right)\left[\left(\sum_{i=0}^{2^n-1}\left(c_n(i)-\frac{1}{2}\right)^2\right)-\left(c_n(j)-\frac{1}{2}\right)^2\right]\\
&\quad+(1-\epsilon)^{2n}\left(\frac{2+(1+\delta)^{2n}-2(1+\delta)^n-(1+\delta^2)^n}{(2^n-1)(2^n-2)}\right)\\
&\quad\cdot\sum_{i\neq j}\sum_{l\neq i,l\neq j}\left(c_n(i)-\frac{1}{2}\right)\left(c_n(l)-\frac{1}{2}\right)\\
&\quad+(1-\epsilon)^{2n}\cdot 2\left(\frac{(1+\delta)^n-1}{2^n-1}\right)\sum_{l\neq j}\left(c_n(j)-\frac{1}{2}\right)\left(c_n(l)-\frac{1}{2}\right). \qquad (2.44)
\end{aligned}
$$

The first term in (2.44) decays to 0 as $O((1-\epsilon)^{2n})$ as $n\to\infty$. The second term in

(2.44) is

$$\begin{aligned}
&(1-\epsilon)^{2n}\left(\frac{(1+\delta^2)^n-1}{2^n-1}\right)^n\left[\left(\sum_{i=0}^{2^n-1}\left(c_n(i)-\frac{1}{2}\right)^2\right)-\left(c_n(j)-\frac{1}{2}\right)^2\right]\\
&= (1-\epsilon)^{2n}\left(\frac{(1+\delta^2)^n-1}{2^n-1}\right)\left[\frac{2^{2n}-1}{12\cdot 2^n}-\left(c_n(j)-\frac{1}{2}\right)^2\right]\\
&= \left[(1-2\epsilon+2\epsilon^2)^n-(1-\epsilon)^{2n}\right]\cdot\left[\frac{2^n+1}{12\cdot 2^n}+\frac{j^2}{2^{2n}(2^n-1)}+\frac{2^n-1-4j}{2^{2n+2}}\right]\\
&= O\left((1-2\epsilon+2\epsilon^2)^n\right)
\end{aligned}$$

as $n\to\infty$, since $0<\epsilon<1$. To evaluate the third term in (2.44) note that

$$\begin{aligned}
&\sum_{i\neq j}\sum_{l\neq i,l\neq j}\left(c_n(i)-\frac{1}{2}\right)\left(c_n(l)-\frac{1}{2}\right)\\
&= \left[\sum_{i=0}^{2^n-1}\left(c_n(i)-\frac{1}{2}\right)\right]^2-\left[\sum_{i=0}^{2^n-1}\left(c_n(i)-\frac{1}{2}\right)^2\right]\\
&\quad -2\left(c_n(j)-\frac{1}{2}\right)\left[-\left(c_n(j)-\frac{1}{2}\right)+\sum_{i=0}^{2^n-1}\left(c_n(i)-\frac{1}{2}\right)\right]\\
&= 0-\frac{1}{12}\left(2^n-2^{-n}\right)+2\left(c_n(j)-\frac{1}{2}\right)^2.
\end{aligned}$$

Thus, since $(1-\epsilon)(1+\delta)=1$, the third term in (2.44) is

$$\left(\frac{2(1-\epsilon)^{2n}+1-2(1-\epsilon)^n-(1-2\epsilon+2\epsilon^2)^n}{(2^n-1)(2^n-2)}\right)\left(2\left(c_n(j)-\frac{1}{2}\right)^2-\frac{2^n}{12}+\frac{1}{12\cdot 2^n}\right)$$

which tends to 0 as $O(2^{-n})$ as $n\to\infty$. To evaluate the forth term in (2.44) note that

$$\begin{aligned}
\sum_{l\neq j}\left(c_n(j)-\frac{1}{2}\right)\left(c_n(l)-\frac{1}{2}\right) &= \left(c_n(j)-\frac{1}{2}\right)\sum_{l\neq j}\left(c_n(l)-\frac{1}{2}\right)\\
&= -\left(c_n(j)-\frac{1}{2}\right)^2.
\end{aligned}$$

Thus the forth term in (2.44) is

$$\begin{aligned}
&(1-\epsilon)^{2n} \cdot 2\left(\frac{(1+\delta)^n - 1}{2^n - 1}\right) \sum_{l \neq j} \left(c_n(j) - \frac{1}{2}\right)\left(c_n(l) - \frac{1}{2}\right) \\
&= \quad -2\left(\frac{(1-\epsilon)^n - (1-\epsilon)^{2n}}{2^n - 1}\right)\left(c_n(j) - \frac{1}{2}\right)^2 \\
&= \quad O\left(2^{-n}\right)
\end{aligned}$$

as $n \to \infty$. Thus $\mathrm{Var}(y_n(j)) = O\left((1 - 2\epsilon + 2\epsilon^2)^n\right) = O(2^{-\beta n})$ as $n \to \infty$, where $\beta = -\log_2(1 - 2\epsilon + 2\epsilon^2) > 0$. □

Proposition 2.19 is key to the proof of the next result. The theorem below shows that asymptotically, an arbitrarily large fraction of index assignments induce an arbitrarily large fraction of codepoints to be arbitrarily close to $1/2$. This result is in contrast to the fact that the Natural Binary Code index assignment has an arbitrarily small fraction of codepoints arbitrarily close to $1/2$.

Theorem 2.20. *For a decoder optimized uniform quantizer, arbitrarily small $r, s, t > 0$, and n sufficiently large, at least $(1-r)2^n!$ index assignments each have at least $(1-s)2^n$ codepoints within a distance of t from $1/2$.*

Proof. Assume Π_n is chosen uniformly at random from the set S_{2^n} of all $2^n!$ index assignments. Let $\delta = \epsilon/(1-\epsilon)$ and note that $(1-\epsilon)(1+\delta) = 1$. Also, let

$$a_n = \left(c_n(j) - \frac{1}{2}\right)\left(\frac{1}{1 - 2^{-n}}\right)\left((1-\epsilon)^n - 2^{-n}\right).$$

By the Chebychev inequality, for any $t > 0$,

$$\begin{aligned}
P\left[\left|y_n(j) - \frac{1}{2}\right| > t\right] &= P\left[\left|y_n(j) - \frac{1}{2} - a_n + a_n\right| > t\right] \\
&\leq P\left[\left|y_n(j) - \frac{1}{2} - a_n\right| > t - |a_n|\right]
\end{aligned}$$

$$\begin{aligned} &= P\left[|y_n(j) - E[y_n(j)]| > t - |a_n|\right] \\ &< \frac{\mathrm{Var}(y_n(j))}{(t-|a_n|)^2} \end{aligned}$$

which means that

$$\frac{1}{2^n!}\left|\left\{\pi_n \in S_{2^n} : \left|y_n(j) - \frac{1}{2}\right| > t\right\}\right| < \frac{\mathrm{Var}(y_n(j))}{(t-|a_n|)^2}.$$

Thus, for any $A > 0$ there are at most $\frac{2^n!2^n}{A} \cdot \frac{\mathrm{Var}(y_n(j))}{(t-|a_n|)^2}$ index assignments $\pi_n \in S_{2^n}$, such that for each such π_n, there exist at least A codepoints $y_n(j)$ satisfying $\left|y_n(j) - \frac{1}{2}\right| > t$. Taking $A = \alpha 2^n$ we get the following equivalent conclusion. For any $\alpha \in (0,1)$, there are at most $\frac{2^n!}{\alpha} \cdot \frac{\mathrm{Var}(y_n(j))}{(t-|a_n|)^2}$ index assignments $\pi_n \in S_{2^n}$, such that for each such π_n, there exist at least $\alpha 2^n$ codepoints $y_n(j)$ satisfying $\left|y_n(j) - \frac{1}{2}\right| > t$. This implies that for any $\alpha \in (0,1)$, there are at least $2^n!\left(1 - \frac{\mathrm{Var}(y_n(j))}{\alpha(t-|a_n|)^2}\right)$ index assignments $\pi_n \in S_{2^n}$ such that for each such π_n, there exist at most $\alpha 2^n$ codepoints $y_n(j)$ satisfying $\left|y_n(j) - \frac{1}{2}\right| > t$.

A careful look at the variance shows a dependency on j but we can easily make a uniform upper bound on the variance which goes to zero at the speed $O(2^{-\beta n})$, where $\beta = -\log_2(1 - 2\epsilon + 2\epsilon^2) > 0$. We choose $t = \alpha = 2^{-\beta n/4}$. This implies that for any n, a fraction of at least $1 - O(2^{-\beta n/4})$ of all index assignments have the property that the fraction of codepoints $y_n(j)$ farther from $1/2$ than $2^{-\beta n/4}$, is at most $2^{-\beta n/4}$. In other words, as $n \to \infty$, an arbitrarily large fraction of all index assignments give rise to codebooks with an arbitrarily large fraction of codepoints arbitrarily close to $1/2$. □

Note that the proof of Theorem 2.20 demonstrates that the random mapping $\lambda_{\Pi_n}^{(n)}$ converges to zero in probability.

2.7 Distortion Analysis

Let π_n be the index assignment for a rate n quantizer with a uniform encoder on $[0,1]$ for a uniform source on $[0,1]$ and a binary symmetric channel with bit error probability ϵ. Then the end-to-end MSE can be written as

$$\begin{aligned} D^{(\pi_n)} &= \sum_{i=0}^{2^n-1}\sum_{j=0}^{2^n-1} p(\pi_n(j)|\pi_n(i)) \int_{i/2^n}^{(i+1)/2^n} (x-y_n(j))^2\,dx \\ &= \frac{1}{3} + 2^{-n}\sum_{i=0}^{2^n-1}\sum_{j=0}^{2^n-1} p(\pi_n(j)|\pi_n(i)) \left[y_n^2(j) - 2c_n(i)y_n(j)\right]. \end{aligned}$$

For any index assignment $\pi_n \in S_{2^n}$, let $D_{CU}^{(\pi_n)}$ denote the MSE of a channel unoptimized uniform quantizer and let $D_{CO}^{(\pi_n)}$ denote the MSE of a decoder optimized uniform quantizer. For given ϵ and n, an index assignment $\pi_n \in S_{2^n}$ is said to be *optimal for a channel unoptimized uniform quantizer* if for all $\pi_n' \in S_{2^n}$,

$$D_{CU}^{(\pi_n)} \le D_{CU}^{(\pi_n')}$$

and π_n is said to be *optimal for a decoder optimized uniform quantizer* if for all $\pi_n' \in S_{2^n}$,

$$D_{CO}^{(\pi_n)} \le D_{CO}^{(\pi_n')}.$$

Lemma 2.21. *The mean squared error of a decoder optimized uniform quantizer with index assignment $\pi_n \in S_{2^n}$ is*

$$D_{CO}^{(\pi_n)} = \frac{1}{3} - 2^{-n}\sum_{j=0}^{2^n-1} y_n^2(j).$$

Proof.

$$\begin{aligned} D_{CO}^{(\pi_n)} &= \frac{1}{3} + 2^{-n} \sum_{j=0}^{2^n-1} \sum_{i=0}^{2^n-1} p(\pi_n(j)|\pi_n(i)) \left[y_n^2(j) - 2c_n(i)y_n(j) \right] \\ &= \frac{1}{3} + 2^{-n} \sum_{j=0}^{2^n-1} \left[y_n^2(j) - 2y_n(j) \sum_{i=0}^{2^n-1} p(\pi_n(j)|\pi_n(i))c_n(i) \right] \\ &= \frac{1}{3} - 2^{-n} \sum_{j=0}^{2^n-1} y_n^2(j) \qquad (2.45) \end{aligned}$$

where (2.45) follows from the weighted centroid condition. □

In [5] it was shown that randomly chosen index assignments for a channel unoptimized uniform quantizer are asymptotically bad in the sense that their MSE approaches that of the worst possible index assignment in the limit as $n \to \infty$. The proof involved an explicit construction of a worst index assignment. The following theorem extends the result to a decoder optimized uniform quantizer and its proof does not require the construction of a worst case index assignment. In Theorem 2.22 the term $1/12$ is in fact the variance of the source.

Theorem 2.22. *The mean squared error of a decoder optimized uniform quantizer is at most* $1/12$*, and for* n *sufficiently large, an arbitrarily large fraction of index assignments achieve a mean squared error arbitrarily close to* $1/12$*.*

Proof. For any index assignment π_n, the average of the codepoints is

$$\begin{aligned} 2^{-n} \sum_{j=0}^{2^n-1} y_n(j) &= 2^{-n} \sum_{j=0}^{2^n-1} \sum_{i=0}^{2^n-1} c_n(i) p(\pi_n(j)|\pi_n(i)) \\ &= 2^{-n} \sum_{i=0}^{2^n-1} c_n(i) \\ &= \frac{1}{2}. \end{aligned}$$

Thus,

$$\begin{aligned} D_{CO}^{(\pi_n)} &= \frac{1}{3} - 2^{-n} \sum_{j=0}^{2^n-1} y_n^2(j) \\ &\leq \frac{1}{3} - \left(2^{-n} \sum_{j=0}^{2^n-1} y_n(j) \right)^2 \qquad (2.46) \\ &= \frac{1}{3} - \frac{1}{4} = \frac{1}{12} \end{aligned}$$

where (2.46) follows from Jensen's inequality. The second assertion follows from Theorem 2.20 and Lemma 2.21. □

Although Theorem 2.22 indicates that asymptotically most index assignments yield mean squared errors close to $1/12$, in the following it will be shown that the Natural Binary Code, the Folded Binary Code, and the Gray Code perform substantially better asymptotically.

The next two theorems give the mean squared errors for the Natural Binary Code with a channel unoptimized decoder and with a channel optimized decoder. Theorem 2.23 was stated in [8] (see, e.g. [4] for a proof). The results are given as a function of the quantizer rate n and the channel bit error probability ϵ. Analogous results are then given for the Folded Binary Code, the Gray Code, and the average for an index assignment chosen uniformly at random.

Theorem 2.23. *The mean squared error of a channel unoptimized uniform quantizer with the Natural Binary Code index assignment is*

$$D_{CU}^{(NBC)} = \frac{2^{-2n}}{12} + \frac{\epsilon}{3} \left(1 - 2^{-2n} \right).$$

Theorem 2.24. *The mean squared error of a decoder optimized uniform quantizer with*

the Natural Binary Code index assignment is

$$D_{CO}^{(NBC)} = \frac{2^{-2n}}{12} + \frac{\epsilon(1-\epsilon)}{3}\left(1 - 2^{-2n}\right).$$

Proof. Combining Proposition 2.3 and Lemma 2.21 gives

$$\begin{aligned}
&D_{CO}^{(NBC)} \\
&= \frac{1}{3} - 2^{-n}\sum_{j=0}^{2^n-1}\left[\epsilon^2 + 2\epsilon 2^{-n}(1-2\epsilon)\left(j+\frac{1}{2}\right) + 2^{-2n}(1-2\epsilon)^2\left(j^2+j+\frac{1}{4}\right)\right] \\
&= \frac{1}{3} - \Bigg[\epsilon^2 + 2^{-n}\epsilon(1-2\epsilon) + 2^{-n}\epsilon(1-2\epsilon)(2^n-1) \\
&\qquad +2^{-2n}(1-2\epsilon)^2\left(\frac{(2^n-1)(2^{n+1}-1)}{6} + \frac{2^n-1}{2} + \frac{1}{4}\right)\Bigg] \\
&= \frac{2^{-2n}}{12} + \frac{\epsilon(1-\epsilon)}{3}\left(1-2^{-2n}\right).
\end{aligned}$$

□

The next two theorems give the mean squared errors for the Folded Binary Code with a channel unoptimized decoder and with a channel optimized decoder. Theorem 2.25 was given in [4]. The proof of Theorem 2.26 is similar to that of Theorem 2.24 and is omitted for brevity.

Theorem 2.25. *The mean squared error of a channel unoptimized uniform quantizer with the Folded Binary Code index assignment is*

$$D_{CU}^{(FBC)} = \frac{1}{12}\left(5\epsilon - 2\epsilon^2 + 2^{-2n}(1 - 8\epsilon + 8\epsilon^2)\right).$$

Theorem 2.26. *The mean squared error of a decoder optimized uniform quantizer with*

the Folded Binary Code index assignment is

$$D_{CO}^{(FBC)} = \frac{1}{12}\left(5\epsilon - 9\epsilon^2 + 8\epsilon^3 - 4\epsilon^4 - 2^{-2n}(1-2\epsilon)^4\right).$$

The next two theorems give the mean squared errors for the Gray Code with a channel unoptimized decoder and with a channel optimized decoder. Theorem 2.27 was stated in [9] (see, e.g. [4] for a proof).

Theorem 2.27. *The mean squared error of a channel unoptimized uniform quantizer with the Gray Code index assignment is*

$$D_{CU}^{(GC)} = \frac{1}{6} - \frac{2^{-2n}}{12} - \frac{\left(\frac{1}{4}-\frac{\epsilon}{2}\right)\left(1-\left(\frac{1}{4}-\frac{\epsilon}{2}\right)^n\right)}{\frac{3}{2}+\epsilon}.$$

Theorem 2.28. *The mean squared error of a decoder optimized uniform quantizer with the Gray Code index assignment is*

$$D_{CO}^{(GC)} = \frac{1}{12} - \frac{1}{4}\cdot\frac{1-\left(\frac{1}{2}-\epsilon\right)^{2n}}{\left(\frac{1}{2}-\epsilon\right)^{-2}-1}.$$

Proof. Combining Proposition 2.10 and Lemma 2.21 gives

$$\begin{aligned}
&D_{CO}^{(GC)} \\
&= \frac{1}{3} - 2^{-n}\sum_{j=0}^{2^n-1}\left[\frac{1}{4} + \frac{1}{2}\sum_{i=1}^{n}(-1)^{b_n(j,i)+1}\left(\frac{1}{2}-\epsilon\right)^i\right. \\
&\quad \left. + \frac{1}{4}\sum_{i=1}^{n}\sum_{k=1}^{n}(-1)^{b_n(j,i)+b_n(j,k)+2}\left(\frac{1}{2}-\epsilon\right)^{i+k}\right] \\
&= \frac{1}{12} - 2^{-n}\sum_{j=0}^{n}\left(y_n(j)-\frac{1}{2}\right) \\
&\quad - 2^{-n-2}\sum_{i=1}^{n}\sum_{k=1}^{n}\sum_{j=0}^{2^n-1}(-1)^{b_n(j,i)+b_n(j,k)+2}\left(\frac{1}{2}-\epsilon\right)^{i+k}
\end{aligned}$$

$$
\begin{aligned}
&= \frac{1}{12} - 2^{-n-2} \sum_{i=1}^{n} 2^n \left(\frac{1}{2} - \epsilon\right)^{2i} && (2.47)\\
&= \frac{1}{12} - \frac{\left(\frac{1}{2} - \epsilon\right)^2 \left(1 - \left(\frac{1}{2} - \epsilon\right)^{2n}\right)}{4 - (1-2\epsilon)^2}
\end{aligned}
$$

where (2.47) follows from the fact that the average of the codepoints for any index assignment is $1/2$ (see the proof of Theorem 2.22) and that for $i \neq k$, the sum $b_n(j,i) + b_n(j,k)$ is even 2^{n-1} times and odd 2^{n-1} times as j ranges between 0 and $2^n - 1$. □

It can be seen from Theorem 2.23 and Theorem 2.24 that for the NBC, the reduction in MSE obtained by using a channel optimized quantizer decoder instead of one obeying the centroid condition, is $\epsilon^2(1 - 2^{-2n})/3$. For small ϵ, the MSE reduction is thus small. For a randomly chosen index assignment however, Theorem 2.29 and Theorem 2.30 show that channel optimized decoders reduce the average distortion by a factor of two over decoders obeying the centroid condition, independent of ϵ, in the limit as $n \to \infty$. Theorem 2.29 was stated in [8], and [5] contains a concise proof. Let $D_{CU}^{(RAN)}$ be a random variable denoting the MSE of a channel unoptimized uniform quantizer with a randomly chosen index assignment.

Theorem 2.29. *The average mean squared error of a channel unoptimized uniform quantizer with an index assignment chosen uniformly at random is*

$$E[D_{CU}^{(RAN)}] = \frac{2^{-2n}}{12} + \frac{1}{6} + \frac{1 - (2^n + 1)(1-\epsilon)^n}{6 \cdot 2^n}.$$

Since most index assignments are asymptotically bad, their average is bad as well. More precisely, the next theorem shows that the asymptotic average MSE of a decoder optimized uniform quantizer with an arbitrary index assignment converges to $1/12$, consistent with Theorem 2.22. Let $D_{CO}^{(RAN)}$ be a random variable denoting the MSE of a decoder optimized uniform quantizer with a randomly chosen index assign-

ment.

Theorem 2.30. *The average mean squared error of a decoder optimized uniform quantizer with an index assignment chosen uniformly at random is*

$$E[D_{CO}^{(RAN)}] = \frac{2^{-2n}}{12} + \frac{1}{12} + \frac{1-(2^n+1)(1-2\epsilon+2\epsilon^2)^n}{12\cdot 2^n}.$$

Proof. Let

$$a_n = \left(c_n(j) - \frac{1}{2}\right)\left(\frac{1}{1-2^{-n}}\right)\left((1-\epsilon)^n - 2^{-n}\right).$$

By Lemma 2.21, the expected value of $D_{CO}^{(RAN)}$ (over all index assignments) is

$$\begin{aligned}
&E[D_{CO}^{(RAN)}] \\
&= \frac{1}{3} - 2^{-n}\sum_{j=0}^{2^n-1} E[y_n^2(j)] \\
&= \frac{1}{3} - 2^{-n}\sum_{j=0}^{2^n-1} (\mathrm{Var}(y_n(j)) + E[y_n(j)]^2) \\
&= \frac{1}{3} - 2^{-n}\sum_{j=0}^{2^n-1} \left(E\left[\left(y_n(j)-\frac{1}{2}\right)^2\right] + a_n + \frac{1}{4}\right) && (2.48) \\
&= \frac{1}{12} - 2^{-n}\sum_{j=0}^{2^n-1} E\left[\left(y_n(j)-\frac{1}{2}\right)^2\right] && (2.49) \\
&= \frac{1}{12} - 2^{-n}\sum_{j=0}^{2^n-1} \Bigg[(1-\epsilon)^{2n}\left(c_n(j)-\frac{1}{2}\right)^2 \\
&\quad + (1-\epsilon)^{2n}\left(\frac{(1+\delta^2)^n-1}{2^n-1}\right)\left[\left(\sum_{i=0}^{2^n-1}\left(c_n(i)-\frac{1}{2}\right)^2\right) - \left(c_n(j)-\frac{1}{2}\right)^2\right] \\
&\quad + (1-\epsilon)^{2n}\left(\frac{2+(1+\delta)^{2n}-2(1+\delta)^n-(1+\delta^2)^n}{(2^n-1)(2^n-2)}\right) \\
&\qquad \cdot \sum_{i\neq j}\sum_{l\neq i, l\neq j}\left(c_n(i)-\frac{1}{2}\right)\left(c_n(l)-\frac{1}{2}\right) \\
&\quad + (1-\epsilon)^{2n}\cdot 2\left(\frac{(1+\delta)^n-1}{2^n-1}\right)\sum_{l\neq j}\left(c_n(j)-\frac{1}{2}\right)\left(c_n(l)-\frac{1}{2}\right)\Bigg] && (2.50)
\end{aligned}$$

$$
\begin{aligned}
= \quad & \frac{1}{12} - 2^{-n} \sum_{j=0}^{2^n-1} \Bigg[(1-\epsilon)^{2n} \left(c_n(j) - \frac{1}{2} \right)^2 \\
& + (1-\epsilon)^{2n} \left(\frac{(1+\delta^2)^n - 1}{2^n - 1} \right) \left[\left(\frac{2^{2n}-1}{12 \cdot 2^n} \right) - \left(c_n(j) - \frac{1}{2} \right)^2 \right] \\
& + (1-\epsilon)^{2n} \left(\frac{2 + (1+\delta)^{2n} - 2(1+\delta)^n - (1+\delta^2)^n}{(2^n-1)(2^n-2)} \right) \\
& \quad \cdot \left(2 \left(c_n(j) - \frac{1}{2} \right)^2 - \frac{2^n}{12} + \frac{2^{-n}}{12} \right) \\
& - (1-\epsilon)^{2n} \cdot 2 \left(\frac{(1+\delta)^n - 1}{2^n - 1} \right) \left(c_n(j) - \frac{1}{2} \right)^2 \Bigg]
\end{aligned} \tag{2.51}
$$

where (2.48) follows from Proposition 2.17 and (2.41), (2.49) follows from the fact that $\sum_{j=0}^{2^n-1} \left(c_n(j) - \frac{1}{2} \right) = 0$, (2.50) follows from (2.44), and (2.51) results from the computations following (2.44). Passing the sum over j inside, distributing the factor of 2^{-n} over all terms, applying Lemma 2.18, and multiplying the $(1-\epsilon)^{2n}$ term through gives

$$
\begin{aligned}
& E[D_{CO}^{(RAN)}] \\
& \quad = \quad \frac{1}{12} - \Bigg\{ \frac{(2^{2n}-1)(1-\epsilon)^{2n}}{12 \cdot 2^{2n}} \\
& \qquad + \frac{(1-2\epsilon+2\epsilon^2)^n - (1-\epsilon)^{2n}}{2^n - 1} \cdot \left(\frac{2^{2n}-1}{12 \cdot 2^n} - \frac{2^{2n}-1}{12 \cdot 2^{2n}} \right) \\
& \qquad + \left[\frac{2(1-\epsilon)^{2n} + 1 - 2(1-\epsilon)^n - (1-2\epsilon+2\epsilon^2)^n}{(2^n-1)(2^n-2)} \right] \\
& \qquad \cdot \left[2 \left(\frac{2^{2n}-1}{12 \cdot 2^{2n}} \right) - \left(\frac{2^{2n}-1}{12 \cdot 2^n} \right) \right] \\
& \qquad -2 \left(\frac{(1-\epsilon)^n - (1-\epsilon)^{2n}}{2^n - 1} \right) \cdot \left(\frac{2^{2n}-1}{12 \cdot 2^{2n}} \right) \Bigg\} \\
& \quad = \quad \frac{1}{12} - \left[\frac{2^{2n}-1}{12 \cdot 2^{2n}} \right] \cdot \left[\frac{2^n (1-2\epsilon+2\epsilon^2)^n - 1}{2^n - 1} \right] \\
& \quad = \quad \frac{2^{-2n}}{12} + \frac{1}{12} + \frac{1 - (2^n+1)(1-2\epsilon+2\epsilon^2)^n}{12 \cdot 2^n}
\end{aligned} \tag{2.52}
$$

where (2.52) makes use of the computations following (2.44). □

Crimmins et al. [1] and McLaughlin, Neuhoff, and Ashley [3] showed that for every ϵ and every n the Natural Binary Code is optimal for a channel unoptimized uniform quantizer. We next extend the proof in [3] to show that for every ϵ and every n the Natural Binary Code is also optimal for a decoder optimized uniform quantizer.

Lemma 2.31. *Let Q_{π_n} denote the $2^n \times 2^n$ matrix whose $(i,j)^{th}$ elements are $q(\pi_n(i)|\pi_n(j))$. For any index assignment π_n, there exists a $2^n \times 2^n$ permutation matrix P such that $Q^2_{\pi_n} = PQ^2_{\pi_n^{(NBC)}}P^t$.*

Proof. Let P be the permutation matrix whose elements are

$$p_{i,j} = \begin{cases} 1 & \text{if } \pi_n(i) = j \\ 0 & \text{otherwise} \end{cases}$$

for $0 \leq i,j \leq 2^n - 1$. Let $a_{i,j}$ and $b_{i,j}$ respectively denote the $(i,j)^{th}$ elements of $Q_{\pi_n^{(NBC)}}$ and $PQ_{\pi_n^{(NBC)}}P^t$. Then

$$q(i|j) = a_{i,j} = b_{\pi_n^{-1}(i),\pi_n^{-1}(j)}$$

or equivalently

$$q(\pi_n(i)|\pi_n(j)) = a_{\pi_n(i),\pi_n(j)} = b_{i,j}$$

which implies $Q_{\pi_n} = PQ_{\pi_n^{(NBC)}}P^t$. Thus $Q^2_{\pi_n} = PQ^2_{\pi_n^{(NBC)}}P^t$ since P is orthogonal. □

Theorem 2.32. *The Natural Binary Code index assignment is optimal for a decoder optimized uniform quantizer, for every bit error probability $\epsilon \geq 0$ and every quantizer rate $n \geq 1$.*

Proof. Let $\underline{c} = [c_n(0), c_n(1), \ldots, c_n(2^n - 1)]^t$ and $\underline{y} = [y_n(0), y_n(1), \ldots, y_n(2^n - 1)]^t$ denote the column vectors of cell centroids and codepoints, respectively. Then Lemma

2.21, Lemma 2.31, and the weighted centroid condition imply that

$$\begin{aligned} D_{CO}^{(\pi_n)} &= \frac{1}{3} - 2^{-n}\|\underline{y}\|^2 \\ &= \frac{1}{3} - 2^{-n}\underline{c}^t Q_{\pi_n}^2 \underline{c} \\ &= \frac{1}{3} - 2^{-n}\underline{c}^t P Q_{\pi_n^{(NBC)}}^2 P^t \underline{c} \\ &= \frac{1}{3} - 2^{-n}\underline{z}^t Q_{\pi_n^{(NBC)}}^2 \underline{z} \\ &= \frac{1}{3} - 2^{-n}\underline{z}^t \hat{Q}_{\pi_n^{(NBC)}} \underline{z} \end{aligned} \tag{2.53}$$

where $\underline{z} = P^t\underline{c}$, and where $\hat{Q}_{\pi_n^{(NBC)}}$ is the same as $Q_{\pi_n^{(NBC)}}$ but with ϵ replaced by $2\epsilon(1-\epsilon) \in (0, 1/2)$. McLaughlin, Neuhoff, and Ashley [3] showed that for every $\epsilon \in (0, 1/2)$, the quadratic form $\underline{z}^t Q_{\pi_n^{(NBC)}} \underline{z}$ (and thus in particular $\underline{z}^t \hat{Q}_{\pi_n^{(NBC)}} \underline{z}$) is maximized for uniform sources and uniform quantizers satisfying $\sum_i c_n(i) = 0$, when $\pi_n = \pi_n^{(NBC)}$. Shifting the support of a uniform source from $[0, 1]$ to $[-1/2, 1/2]$ changes each term in (2.53) by a constant term, independent of the index assignment. Thus $D_{CO}^{(\pi_n)}$ is minimized when $\pi_n = \pi_n^{(NBC)}$, and therefore the Natural Binary Code is optimal for decoder optimized uniform quantizers for all ϵ and n. □

This chapter, in full, is a reprint of the material as it appears in: Benjamin Farber and Kenneth Zeger, "Quantizers with Uniform Encoders and Channel Optimized Decoders," *IEEE Transactions on Information Theory*, vol. 50, no. 1, pp. 62–77, January 2004. The dissertation author was the primary investigator of this paper.

References

[1] T. R. Crimmins, H. M. Horwitz, C. J. Palermo, and R. V. Palermo, "Minimization of Mean-Square Error for Data Transmitted Via Group Codes," *IEEE Transactions on Information Theory*, vol. IT-15, pp. 72–78, January 1969.

[2] A. Gersho and R. Gray, *Vector Quantization and Signal Compression*, Kluwer Academic Publishers, 1991.

[3] S. W. McLaughlin, D. L. Neuhoff, and J. J. Ashley, "Optimal Binary Index Assignments for a Class of Equiprobable Scalar and Vector Quantizers," *IEEE Transactions on Information Theory*, vol. 41, pp. 2031-2037, November 1995.

[4] A. Méhes and K. Zeger, "Binary Lattice Vector Quantization with Linear Block Codes and Affine Index Assignments," *IEEE Transactions on Information Theory*, vol. 44, no. 1, pp. 79-94, January 1998.

[5] A. Méhes and K. Zeger, "Randomly Chosen Index Assignments Are Asymptotically Bad for Uniform Sources," *IEEE Transactions on Information Theory*, vol. 45. pp.788-794, March 1999.

[6] R. M. Gray and D. L. Neuhoff, "Quantization," *IEEE Transactions on Information Theory*, vol. 44. no. 6, pp. 2355-2383, October 1998.

[7] R. Hagen and P. Hedelin, "Robust Vector Quantization by a Linear Mapping of a Block Code," *IEEE Transactions on Information Theory*, vol. 45. no. 1, pp. 200-218, January 1999.

[8] Y. Yamaguchi and T. S. Huang, "Optimum Binary Fixed-Length Block Codes," Quarterly Progress Report 78, M.I.T. Research Lab. of Electronics, Cambridge, Mass., pp. 231-233, July 1965.

[9] T. S. Huang, "Optimum binary code," Quarterly Progress Report 82, M.I.T. Research Lab. of Electronics, Cambridge, Mass., pp. 223-225, July 15, 1966.

[10] P. Knagenhjelm and E. Agrell, "The Hadamard transform-a tool for index assignment ," *IEEE Transactions on Information Theory*, vol. 42. no. 4, pp. 1139-1151, July 1996.

[11] H. Kumazawa, M. Kasahara, and T. Namekawa, "A Construction of Vector Quantizers for Noisy Channels," *Electronics and Engineering in Japan*, vol. 67-B, no. 4, pp. 39–47, 1984.

[12] M. Skoglund, "On channel-constrained vector quantization and index assignment for discrete memoryless channels," *IEEE Transactions on Information Theory*, vol. 45. no. 7, pp. 2615-2622, November 1999.

Chapter 3

Quantizers with Uniform Decoders and Channel Optimized Encoders

Abstract

Scalar quantizers with uniform decoders and channel-optimized encoders are studied for a uniform source on $[0, 1]$ and binary symmetric channels. Two families of affine index assignments are considered: the complemented natural code (CNC), introduced here, and the natural binary code (NBC). It is shown that the NBC never induces empty cells in the quantizer encoder, whereas the CNC can. Nevertheless, we show that the asymptotic distributions of quantizer encoder cells for the NBC and the CNC are equal and are uniform over a proper subset of the source's support region. Empty cells act as a form of implicit channel coding. An effective channel code rate associated with a quantizer designed for a noisy channel is defined and computed for the codes studied. By explicitly showing that the mean squared error of the CNC can be strictly smaller than that of the NBC, we also demonstrate that the NBC is sub-optimal for a large range of transmission rates and bit error probabilities. This contrasts with the known optimality of the NBC when either both the encoder and decoder are not channel optimized, or when only the decoder is channel optimized.

3.1 Introduction

One approach to improving the performance of a quantizer that transmits across a noisy channel is to design the quantizer's encoder and/or decoder to specifically take into account the statistics of the transmission channel. Necessary optimality conditions for such channel-optimized encoders and decoders were given, for example, in [2, 11, 12]. Alternatively, an explicit error control code can be cascaded with the quantizer, at the expense of added transmission rate. Additionally, the choice of index assignment in mapping source code words to channel code words can increase the performance of a quantization system with a noisy channel. Examples of index assignments include the natural binary code (NBC), the folded binary code, and the Gray code.

Ideally, one seeks a complete theoretical understanding of the structure and performance of a quantizer that transmits across a noisy channel, and whose encoder and decoder are channel optimized. Unfortunately, other than the optimality conditions given in [11], virtually no other analytical results are known regarding such quantizers. Quantizer design and performance with index assignments for general encoders and decoders (i.e. not necessarily channel optimized) was considered in [7, 16]. Experimentally, it was observed in [4] and [5] that quantizers with both channel-optimized encoders and decoders can have empty cells, which serve as a form of implicit channel coding. Some theoretical results are known, however, when the quantizer has no channel optimization, or when only the quantizer decoder is channel optimized.

For uniform scalar quantizers with neither channel-optimized encoders nor decoders and with no explicit error control coding, formulas for the mean squared error with uniform sources were given in [8, 9] for the NBC, the Gray code, and for randomly chosen index assignments on a binary symmetric channel. They also asserted (without a published proof) the optimality of the NBC for the binary symmetric channel. Crimmins et al. [1] proved the optimality of the NBC as asserted in [8, 9], and McLaughlin,

Neuhoff, and Ashley [15] generalized this result to uniform vector quantizers. Various other analytical results on index assignments without channel-optimized encoders or decoders have been given in [10, 13, 14].

Quantizers with uniform encoders and channel-optimized decoders on binary symmetric channels were studied in [3]. For such quantizers, exact descriptions of the decoders were computed, and the asymptotic distributions of codepoints were determined for various index assignments. Distortions were calculated and compared to those of quantizers without channel optimization. The proof in [15] of the optimality of the NBC for quantizers with no channel optimization was extended in [3] to show that the NBC is also optimal for quantizers with uniform encoders and channel-optimized decoders.

In the present paper, we examine quantizers with uniform decoders and channel-optimized encoders operating over binary symmetric channels. In particular, we investigate a previously studied index assignment, namely the NBC. In addition, we introduce a new affine index assignment which we call the complemented natural code (CNC) and which turns out to have a number of interesting properties. We specifically analyze the entropy of the encoder output in such quantizers, the high resolution distribution of their encoding cells (i.e. the cell density function), and the mean squared errors the quantizers achieve. We calculate a quantity we call the "effective channel code rate", which describes implicit channel coding, viewed in terms of the entropy of the encoder output. We also show that the NBC optimality results of [1, 3, 15] do not extend to quantizers with uniform decoders and channel-optimized encoders. In fact, the CNC is shown to perform better than the NBC.

Our main results for quantizers with uniform decoders and channel-optimized encoders are the following. For a uniform source on $[0, 1]$ and a binary symmetric channel with bit error probability $\epsilon \in [0, 1/2)$, we compute the effective channel code rates and cell densities for the NBC and CNC. It is shown that the NBC index assignment

never induces empty cells (Corollary 3.3), and the cell density function generated by the NBC is the same as the point density function for quantizers with uniform encoders and channel-optimized decoders with the NBC (Theorem 3.5). In contrast it is shown that the CNC can induce many empty cells (Corollary 3.8). However, the cell density functions generated by the CNC and the NBC are both uniform over the same interval (Theorem 3.10). We also show that the cell density function generated by the CNC is the same as the point density function for quantizers with uniform encoders and channel-optimized decoders with both the CNC and the NBC (Theorem 3.11). Then we extend a result in [8] by computing the mean squared error resulting from the NBC (Theorem 3.14). As a comparison, we state the previously known mean squared error formula for channel unoptimized encoders with the NBC (Theorem 3.13). Finally we show that the NBC is sub-optimal for quantizers with uniform decoders and channel-optimized encoders for many bit error probabilities (Theorem 3.17).

We restrict attention in this paper to a uniform source on $[0, 1]$. However, it will be apparent that the results can be generalized to any bounded interval on the real line.

The paper is organized as follows. Section 3.2 gives definitions and notation. Sections 3.3 and 3.4, respectively, give results for the NBC and CNC. Section 3.5 gives distortion analysis. Appendices 3.7-3.10 contain the proofs of all lemmas, and selected theorems as well as various lemma statements.

3.2 Preliminaries

For any set S of reals, let $\overline{S}$ denote its closure. If S is an interval, let $l(S)$ denote its length. Let $\emptyset$ denote the empty set. Throughout this paper "$\log$" will mean logarithm base two.

A *rate n quantizer* on $[0,1]$ is a mapping

$$\mathcal{Q} : [0,1] \longrightarrow \{y_n(0), y_n(1), \ldots, y_n(2^n - 1)\}.$$

Throughout this paper, all quantizers will be on the interval $[0,1]$ and we will assume $n \geq 2$. The real-valued quantities $y_n(i)$ are called *codepoints* and the set $\{y_n(0), \ldots, y_n(2^n - 1)\}$ is called a *codebook*. For a noiseless channel, the quantizer $\mathcal{Q}$ is the composition of a *quantizer encoder* and a *quantizer decoder*. These are respectively mappings

$$\begin{aligned} \mathcal{Q}_e &: [0,1] \longrightarrow \{0, 1, \ldots, 2^n - 1\} \\ \mathcal{Q}_d &: \{0, 1, \ldots, 2^n - 1\} \longrightarrow \{y_n(0), y_n(1), \ldots, y_n(2^n - 1)\} \end{aligned}$$

such that $\mathcal{Q}_d(i) = y_n(i)$ for all i. On a discrete, memoryless, noisy channel a quantizer is a composition of the quantizer encoder, the channel, and the quantizer decoder.

Without channel noise it is known that for an optimal quantizer, the encoder $\mathcal{Q}_e$ is a surjective mapping. However, in the presence of channel noise, it is possible that in an optimal quantizer the range of $\mathcal{Q}_e$ may contain fewer than 2^n points.

For each i the ith encoding *cell* is the set

$$R_n(i) = \mathcal{Q}_e^{-1}(i).$$

If $R_n(i) = \emptyset$ we say $R_n(i)$ is an *empty cell*.

A quantizer with empty cells can be thought of as implicitly using channel coding to protect against channel noise. For example, if one half of the cells of a quantizer were empty, and the other half were equal size, this could be thought of as effectively using one bit of error protection. More generally, the cascade of a rate k quantizer having 2^k equal size cells with a (n, k) block channel code can equivalently be viewed as a

rate n quantizer with 2^n cells, 2^k of which are nonempty. That is, for any input lying in one of the 2^k nonempty cells, the k-bit index produced by the original quantizer encoder is expanded to n bits, which is then used for transmission. A quantizer can also introduce redundancy by making some encoding cells smaller than others. This reduces the entropy of the encoder output while maintaining the same transmission rate. To quantify the amount of natural error protection embedded in quantizers designed for noisy channels, we define the *effective channel code rate* of a quantizer as

$$r_c = \frac{H(\mathcal{Q}_e(X))}{n}$$

where X is a real-valued source random variable and H denotes the Shannon entropy. Then

$$0 \leq r_c \leq \frac{\log \left|\{i : R_n(i) \neq \emptyset\}\right|}{n} \leq 1.$$

In particular, the effective channel code rate of a rate k quantizer, having no empty cells, cascaded with an (n, k) block channel code (viewed as a rate n quantizer) is at most k/n, i.e., the rate of the channel code. For such a cascaded system, if r_b denotes the rate of the block channel code and if cell sizes are equal, then

$$r_c = r_b.$$

In this paper, we compute the effective channel code rates of certain quantizers that cannot be decomposed as cascades of (lower transmission rate) quantizers with block channel codes.

A quantizer encoder is said to be *uniform* if for each i, the ith cell satisfies

$$R_n(i) \supseteq (i2^{-n}, (i+1)2^{-n}).$$

We say the quantizer decoder is uniform, if for each i, the ith codepoint satisfies

$$y_n(i) = \left(i + \frac{1}{2}\right) 2^{-n}.$$

The *nearest neighbor* cells of a rate n quantizer are the sets

$$T_n(i) = \{x : |y_n(i) - x| < |y_n(j) - x|, \ \ \forall j \neq i\}$$

for $0 \leq i \leq 2^n - 1$. A quantizer's encoder is said to satisfy the *nearest neighbor condition* if for each i,

$$T_n(i) \subset R_n(i) \subset \overline{T}_n(i).$$

That is, its encoding cells are the nearest neighbor cells together with some boundary points (which can be assigned arbitrarily).

For given n, i, and real-valued source random variable X, the *centroid* of the ith cell of the quantizer $\mathcal{Q}$ is the conditional mean

$$c_n(i) = E[X|X \in R_n(i)].$$

The quantizer decoder is said to satisfy the *centroid condition* if the codepoints satisfy

$$y_n(i) = c_n(i)$$

for all i. A quantizer is *uniform* if both the encoder and decoder are uniform. It is known that if a quantizer minimizes the mean squared error for a given source and a noiseless channel, then it satisfies the nearest neighbor and centroid conditions [6]. In particular, if the source is uniform, then a uniform quantizer satisfies the nearest neighbor and centroid conditions.

For a rate n quantizer, an *index assignment* π_n is a permutation of the set

$\{0, 1, \ldots, 2^n - 1\}$. Let S_n denote the set of all $2^n!$ such permutations. For a noisy channel, a random variable $X \in [0, 1]$ is quantized by transmitting the index $I = \pi_n(\mathcal{Q}_e(X))$ across the channel, receiving index J from the channel, and then decoding the codepoint $y_n(\pi_n^{-1}(J)) = \mathcal{Q}_d(\pi_n^{-1}(J))$. The *mean squared error* (MSE) is defined as

$$D = E\left[(X - \mathcal{Q}_d(\pi_n^{-1}(J)))^2\right]. \tag{3.1}$$

The random index J is a function of the source random variable X, the randomness in the channel, and the deterministic functions $\mathcal{Q}_e$ and π_n.

Assume a binary symmetric channel with bit error probability ϵ. Throughout this paper we use the notation:

$$\begin{aligned} \delta &= 1 - 2\epsilon \\ \omega &= \frac{\epsilon}{1 - \epsilon}. \end{aligned}$$

Denote the probability that index j was received, given that index i was sent, by $p_n(j|i) = \epsilon^{H_n(i,j)}(1 - \epsilon)^{n - H_n(i,j)}$ for $0 \leq \epsilon \leq 1/2$, where $H_n(i, j)$ is the Hamming distance between n-bit binary words i and j. Let $q_n(i|j)$ denote the probability that index i was sent, given that index j was received.

For a given source X, channel $p_n(\cdot|\cdot)$, index assignment π_n, and quantizer encoder, the quantizer decoder is said to satisfy the *weighted centroid condition* if the codepoints satisfy

$$y_n(j) = \sum_{i=0}^{2^n - 1} c_n(i) q_n(\pi_n(i)|\pi_n(j)).$$

For a given source X, channel $p_n(\cdot|\cdot)$, index assignment π_n, and quantizer decoder, the quantizer encoder is said to satisfy the *weighted nearest neighbor condition* if the

encoding cells satisfy

$$W_i \subset R_n(i) \subset \overline{W}_i \tag{3.2}$$

where

$$W_i = \left\{ x : \sum_{j=0}^{2^n-1} (x - y_n(j))^2 p_n(\pi_n(j)|\pi_n(i)) \right. \\ \left. < \sum_{j=0}^{2^n-1} (x - y_n(j))^2 p_n(\pi_n(j)|\pi_n(k)), \ \ \forall k \neq i \right\}.$$

For a given quantizer encoder and index assignment, we say the quantizer has a *channel-optimized decoder* if it satisfies the weighted centroid condition. Similarly, for a given quantizer decoder and index assignment, we say the quantizer has a *channel-optimized encoder* if it satisfies the weighted nearest neighbor condition. It is known that a minimum mean-squared error quantizer for a noisy channel must have both a channel-optimized encoder and decoder [11].

Lemma 3.1. *A quantizer with a uniform decoder and channel-optimized encoder satisfies, for all* i*,*

$$\overline{R}_n(i) = \{x \in [0,1] : \alpha_n(i,k)\, x \geq \beta_n(i,k), \ \ \forall\, k \neq i\} \tag{3.3}$$

where

$$\alpha_n(i,k) = \sum_{j=0}^{2^n-1} j[p_n(\pi_n(j)|\pi_n(i)) - p_n(\pi_n(j)|\pi_n(k))] \tag{3.4}$$

$$\beta_n(i,k) = 2^{-n-1} \left(\alpha_n(i,k) + \sum_{j=0}^{2^n-1} j^2 [p_n(\pi_n(j)|\pi_n(i)) - p_n(\pi_n(j)|\pi_n(k))] \right) \tag{3.5}$$

Lemma 3.1 implies that each $R_n(i)$ is a (possibly empty) interval. Therefore, in this

paper, when we describe quantizer encoding cells it suffices to describe their closures.

For any set A, denote the indicator function of A by

$$\chi_A(x) = \begin{cases} 1 & for\ \ x \in A \\ 0 & for\ \ x \notin A. \end{cases}$$

For a given quantizer encoder, let

$$\Lambda = \{i : R_n(i) \neq \emptyset\}.$$

These are the indices of non-empty cells.

For each n and each index assignment $\pi_n \in S_n$, define the function $\gamma_{\pi_n}^{(n)} : [0,1] \to [0,\infty)$ by

$$\gamma_{\pi_n}^{(n)}(x) = \sum_{i \in \Lambda} \frac{1}{|\Lambda| \cdot l(R_n(i))} \chi_{R_n(i)}(x).$$

For a sequence $\pi_n \in S_n$ (for $n = 1, 2, \ldots$) of index assignments, if there exists a measurable function γ such that

$$\gamma(x) = \lim_{n \to \infty} \gamma_{\pi_n}^{(n)}(x)$$

for almost all $x \in [0,1]$ and $\int_0^1 \gamma(x)\, dx = 1$, then we say γ is a *cell density function* with respect to $\{\pi_n\}$.

For each n and each index assignment $\pi_n \in S_n$, define the function $\lambda_{\pi_n}^{(n)} : [0,1] \to [0,\infty)$ by

$$\lambda_{\pi_n}^{(n)}(x) = \sum_{i=0}^{2^n - 1} \frac{1}{2^n \cdot l(T_n(i))} \chi_{T_n(i)}(x).$$

For a sequence $\pi_n \in S_n$ (for $n = 1, 2, \ldots$) of index assignments, if there exists a

measurable function λ such that

$$\lambda(x) = \lim_{n\to\infty} \lambda_{\pi_n}^{(n)}(x)$$

for almost all $x \in [0,1]$ and $\int_0^1 \lambda(x)\,dx = 1$, then we say λ is a *point density function* with respect to $\{\pi_n\}$.

The integrals $\int_a^b \gamma$ and $\int_a^b \lambda$ give the asymptotic fraction of encoding cells and decoder codepoints, respectively, that appear in the interval $[a,b]$ as $n \to \infty$.

Let a *decoder-optimized uniform quantizer* (DOUQ) denote a rate n quantizer with a uniform encoder on $[0,1]$ and a channel-optimized decoder, along with a uniform source on $[0,1]$, and a binary symmetric channel with bit error probability ϵ. When considering DOUQs, we impose the following monotonicity constraint on the quantizer encoder in order to be able to unambiguously refer to particular index assignments: For all $s,t \in [0,1]$, if $s < t$, then $\mathcal{Q}_e(s) \leq \mathcal{Q}_e(t)$. In other words, the encoding cells are labeled from left to right.

Let an *encoder-optimized uniform quantizer* (EOUQ) denote a rate n quantizer with a uniform decoder and a channel-optimized encoder, along with a uniform source on $[0,1]$, and a binary symmetric channel with bit error probability ϵ. When considering EOUQs, we impose the following monotonicity constraint on the quantizer decoder in order to be able to unambiguously refer to particular index assignments: For any $y_n(i)$ and $y_n(j)$, if $y_n(i) < y_n(j)$, then $\mathcal{Q}_d^{-1}(y_n(i)) < \mathcal{Q}_d^{-1}(y_n(j))$. In other words, the codepoints are labeled in increasing order.

An alternative approach would be to view the quantizer encoder as the composition $\pi_n \cdot \mathcal{Q}_e$ and the quantizer decoder as the composition $\mathcal{Q}_d \cdot \pi_n^{-1}$, by relaxing the monotonicity assumption made above. This would remove the role of index assignments from the study of quantizers for noisy channels. However, we retain these encoder and decoder decompositions, as a convenient way to isolate the effects of index assignments,

given known quantizer encoders and decoders.

Let a *channel unoptimized uniform quantizer* denote a rate n uniform quantizer on $[0,1]$, along with a uniform source on $[0,1]$, and a binary symmetric channel with bit error probability ϵ.

3.3 Natural Binary Code Index Assignment

For each n, the natural binary code (NBC) is the index assignment defined by

$$\pi_n^{(NBC)}(i) = i \ \ for \ \ 0 \le i \le 2^n - 1.$$

Theorem 3.2. *An EOUQ with the NBC index assignment has encoding cells given by*

$$\overline{R}_n(i) = \begin{cases} \left[0, \epsilon + \delta 2^{-n}\right] & for \quad i = 0 \\ \left[\epsilon + i\delta 2^{-n}, \epsilon + \delta(i+1)2^{-n}\right] & for \quad 1 \le i \le 2^n - 2 \\ \left[1 - \epsilon - \delta 2^{-n}, 1\right] & for \quad i = 2^n - 1. \end{cases}$$

Proof. The encoding cells satisfy (3.3) in Lemma 3.1, with

$$\begin{aligned} &\alpha_n(i,k) \\ &\quad = \sum_{j=0}^{2^n-1} j \left[p_n(\pi_n^{(NBC)}(j)|\pi_n^{(NBC)}(i)) - p_n(\pi_n^{(NBC)}(j)|\pi_n^{(NBC)}(k))\right] \\ &\quad = (i-k)\delta \end{aligned} \tag{3.6}$$

$$\begin{aligned} &\beta_n(i,k) \\ &\quad = 2^{-n-1}\Bigg(\alpha_n(i,k) \\ &\qquad + \sum_{j=0}^{2^n-1} j^2 \left[p_n(\pi_n^{(NBC)}(j)|\pi_n^{(NBC)}(i)) - p_n(\pi_n^{(NBC)}(j)|\pi_n^{(NBC)}(k))\right]\Bigg) \\ &\quad = 2^{-n-1}\Big[(i-k)\delta[1 + 2\epsilon(2^n - 1)] + (i^2 - k^2)\delta^2\Big] \end{aligned} \tag{3.7}$$

where (3.6) follows from Lemma 3.20; and (3.7) follows from (3.6) and Lemma 3.21. Thus,

$$\frac{\beta_n(i,k)}{\alpha_n(i,k)} = \epsilon + \delta(i+k+1)2^{-n-1} \quad 0 \le i,k \le 2^n - 1. \tag{3.8}$$

From (3.6), we have that $\alpha_n(i,k) > 0$ if and only if $i > k$, and $\alpha_n(i,k) < 0$ if and only if $i < k$. Therefore, (3.3) can be rewritten as

$$\overline{R}_n(i) = \left\{ x \in [0,1] : x \ge \frac{\beta_n(i,k)}{\alpha_n(i,k)},\ \forall\ k < i \ and\ x \le \frac{\beta_n(i,k)}{\alpha_n(i,k)},\ \forall\ k > i \right\}. \tag{3.9}$$

By (3.8), the quantity $\frac{\beta_n(i,k)}{\alpha_n(i,k)}$ is increasing in both i and k. Hence, if $1 \le i \le 2^n - 1$, then (taking $k = i-1$ in (3.9)) $x \in \overline{R}_n(i)$ if and only if

$$\begin{aligned} x &\ge \epsilon + \delta(i + (i-1) + 1)2^{-n-1} \\ &= \epsilon + i\delta 2^{-n}. \end{aligned}$$

Similarly, if $0 \le i \le 2^n - 2$, then (taking $k = i+1$ in (3.9)) $x \in \overline{R}_n(i)$ if and only if

$$\begin{aligned} x &\le \epsilon + \delta(i + (i+1) + 1)2^{-n-1} \\ &= \epsilon + \delta(i+1)2^{-n}. \end{aligned} \tag{3.10}$$

□

A consequence of the preceding theorem is that the NBC produces no empty cells when the weighted nearest neighbor condition is used together with uniformly spaced codepoints. This fact is stated as the following result.

Corollary 3.3. *For all n and for all $\epsilon \in [0, 1/2)$, an EOUQ with the NBC index assignment has no empty cells.*

Figures 3.1 and 3.2 illustrate the encoding cells of a rate 3 EOUQ with the NBC

index assignment for bit error rates 0.05 and 0.25 respectively. Figure 3.3 plots the encoding cell boundaries of a rate 3 EOUQ with the NBC index assignment as a function of bit error rate.

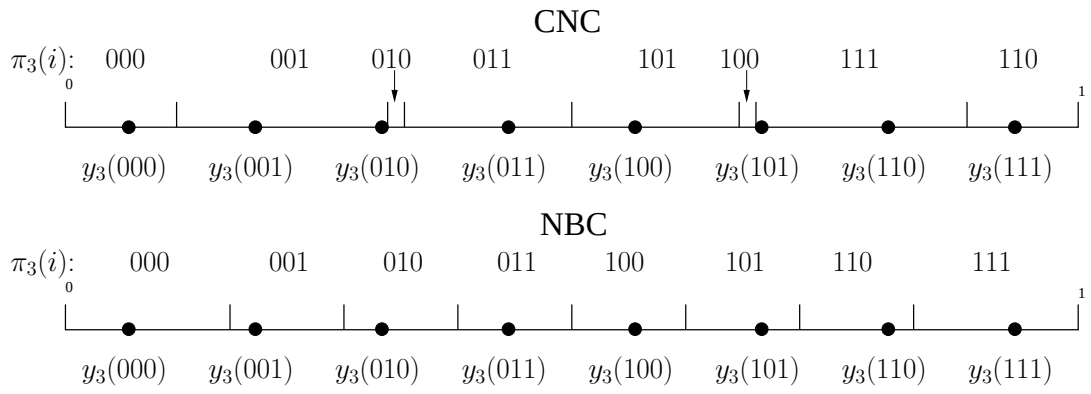

Figure 3.1: Plot of the encoding cells of rate 3 EOUQs with the CNC and NBC index assignments and a bit error rate 0.05.

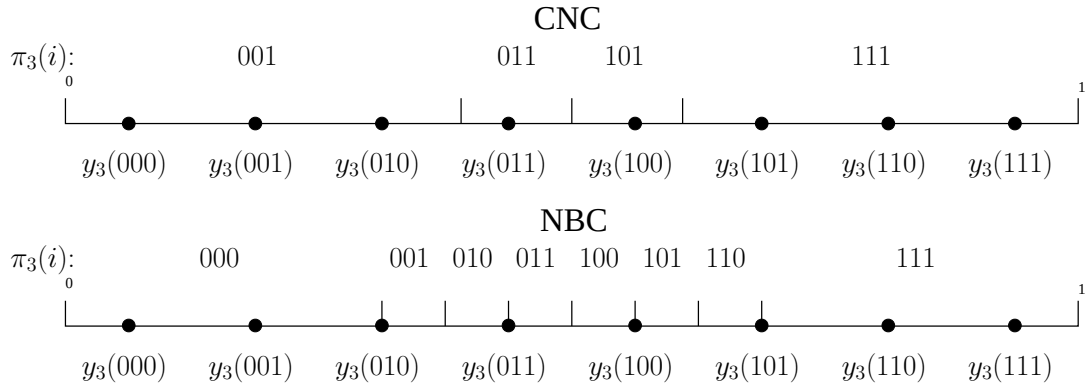

Figure 3.2: Plot of the encoding cells of rate 3 EOUQs with the CNC and NBC index assignments and a bit error rate 0.25.

Theorem 3.4. *An EOUQ with the NBC index assignment has an effective channel code rate given by*

$$r_c = (1-2^{1-n})(1-2\epsilon)\left(1-\frac{\log(1-2\epsilon)}{n}\right)+\frac{2\epsilon+(1-2\epsilon)2^{1-n}}{n}\log\left(\frac{1}{\epsilon+(1-2\epsilon)2^{-n}}\right).$$

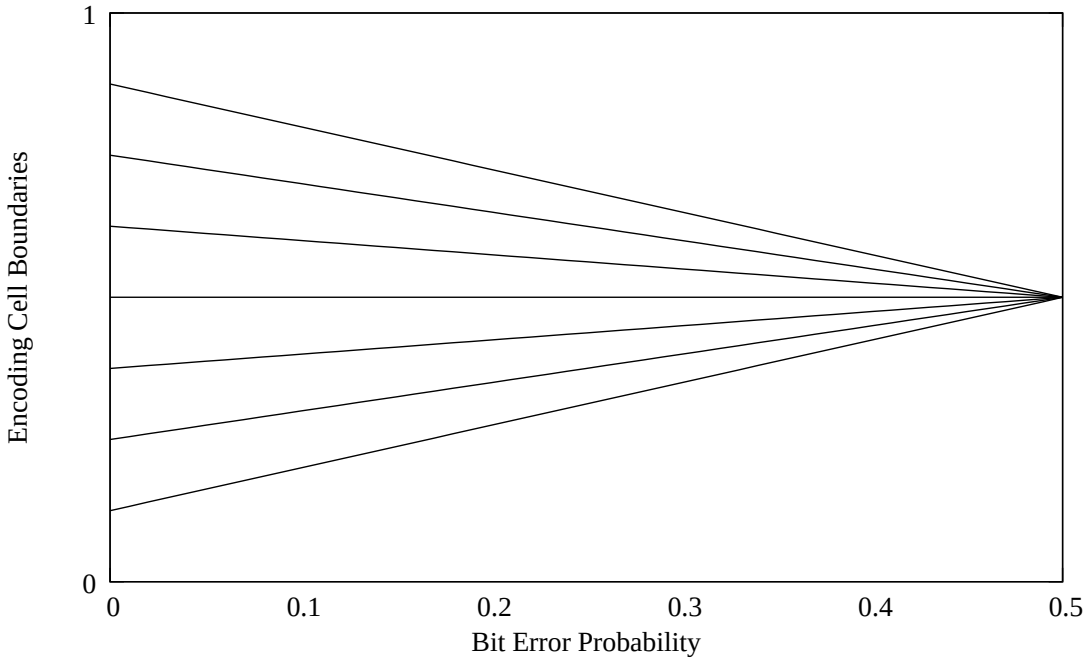

Figure 3.3: Plot of the encoding cells boundaries of a rate 3 EOUQ with the NBC index assignment as a function of bit error rate.

Proof. The definition of r_c implies

$$r_c = \frac{1}{n} \sum_{i \in \Lambda} l(R_n(i)) \log \frac{1}{l(R_n(i))}.$$

From Theorem 3.2,

$$l(R_n(i)) = \begin{cases} \epsilon + \delta 2^{-n} & for \quad i = 0 \\ \delta 2^{-n} & for \quad 1 \leq i \leq 2^n - 2 \\ \epsilon + \delta 2^{-n} & for \quad i = 2^n - 1. \end{cases}$$

Therefore,

$$r_c = (2^n - 2)\frac{\delta 2^{-n}}{n} \log \left(\frac{1}{\delta 2^{-n}} \right) + 2\frac{(\epsilon + \delta 2^{-n})}{n} \log \left(\frac{1}{\epsilon + \delta 2^{-n}} \right)$$

$$= (1 - 2^{1-n})\delta \left(1 - \frac{\log \delta}{n}\right) + \frac{2\epsilon + \delta 2^{1-n}}{n} \log \left(\frac{1}{\epsilon + \delta 2^{-n}}\right).$$

□

As $n \to \infty$, the effective channel code rate given by Theorem 3.4 converges to $1 - 2\epsilon$. Figure 3.5 plots the quantity r_c from Theorem 3.4 for rate $n = 4$.

The following theorem shows that the cell density function for a sequence of EOUQs with the NBC is the same as the point density function found in [3] for a sequence of DOUQs with the NBC.

Theorem 3.5. *A sequence of EOUQs with the NBC index assignment has a cell density function given by*

$$\gamma(x) = \begin{cases} \frac{1}{1-2\epsilon} & for\ \epsilon < x < 1 - \epsilon \\ 0 & else. \end{cases}$$

Proof. From Theorem 3.2,

$$l(R_n(i)) = \begin{cases} \epsilon + \delta 2^{-n} & for \quad i = 0 \\ \delta 2^{-n} & for \quad 1 \le i \le 2^n - 2 \\ \epsilon + \delta 2^{-n} & for \quad i = 2^n - 1. \end{cases}$$

Therefore, since $|\Lambda| = 2^n$ by Corollary 3.3,

$$\gamma^{(n)}_{\pi_n^{(NBC)}}(x) \quad = \quad \begin{cases} \frac{1}{\delta} & for \quad \epsilon + \delta 2^{-n} \le x < 1 - \epsilon - \delta 2^{-n} \\ \frac{1}{\epsilon 2^n + \delta} & for \quad 0 \le x < \epsilon + \delta 2^{-n} \ \ or \\ & \qquad 1 - \epsilon - \delta 2^{-n} \le x \le 1 \end{cases}$$

$$\longrightarrow \quad \begin{cases} \frac{1}{\delta} & for \quad \epsilon < x < 1 - \epsilon \\ 0 & for \quad 0 \le x \le \epsilon \ \ or \ \ 1 - \epsilon \le x \le 1 \end{cases}$$

$$as \ \ n \to \infty.$$

□

3.4 Complemented Natural Code Index Assignment

Let the complemented natural code (CNC) be the index assignment defined by

$$\pi_n^{(CNC)}(i) = \begin{cases} i & for \quad 0 \leq i \leq 2^{n-1} - 1 \\ i+1 & for \quad 2^{n-1} \leq i \leq 2^n - 2 \qquad and \;\; i \;\; even \\ i-1 & for \quad 2^{n-1} + 1 \leq i \leq 2^n - 1 \quad and \;\; i \;\; odd. \end{cases}$$

Note that the CNC is a linear index assignment[1], since

$$\pi_n^{(CNC)}(i) = iG_n$$

where i is an n-bit binary word, G_n is the $n \times n$ identity matrix with an additional 1 in the upper right hand corner, and arithmetic is performed modulo 2 in the product iG_n. The CNC is closely related to the NBC. However, it induces very different encoding cell boundaries for EOUQs, as shown by Theorem 3.7.

Lemma 3.6. *For each n, the polynomial $\phi_n(\epsilon) = -8\epsilon^3 + (4 - 2^{n+1})\epsilon^2 + (2 + 2^{n+1})\epsilon - 1$ restricted to $\epsilon \in (0, 1/2)$ has a unique root ϵ_n^*. The polynomial is negative if and only if $\epsilon < \epsilon_n^*$. Furthermore, ϵ_n^* is monotonic decreasing and $\epsilon_n^* < (2^{n/2} + 2)^{-1}$.*

The quantity ϵ_n^* defined in Lemma 3.6 will be frequently referenced throughout the remainder of the paper. ϵ_n^* plays an important role as a threshold value for the bit error probability of a binary symmetric channel, beyond which the encoding regions and empty cells of an EOUQ with the CNC index assignment change in behavior. It can be

[1] Affine index assignments were studied in [13]. The NBC and Gray code are linear, and the folded binary code is affine.

shown, using the general solution to a cubic, that

$$\epsilon_n^* = \frac{2^n+4}{12} \cdot \left[\sqrt{3} \sin\left(\frac{\arctan(\tau/\sigma)+\pi}{3} \right) - \cos\left(\frac{\arctan(\tau/\sigma)+\pi}{3} \right) - 1 \right] + \frac{1}{2}$$

where

$$\sigma = 2^{n-5} - \frac{1}{27}\left(2^{n-2}+1\right)^3 \quad and \quad \tau = \sqrt{2^{n-4}(2^{n-6}-\sigma)}.$$

Theorem 3.7. *The encoding cells of an EOUQ with the CNC index assignment are given as follows.*

If $n = 2$ *and* $\epsilon \in [0, 1/4)$*, or if* $n \geq 3$ *and* $\epsilon \in [0, \epsilon_n^*)$*, then*

$$\overline{R}_n(i) =$$

$$\begin{cases}
\left[0, \left(\delta - \dfrac{2^n\epsilon^2}{\delta}\right)2^{-n}\right] & \text{for } i = 0 \\
\left[\left(i\delta - \dfrac{2^n\epsilon^2}{\delta}\right)2^{-n}, \left((i+1)\delta + \dfrac{2^n\epsilon(2+\epsilon)}{1+2\epsilon}\right)2^{-n}\right] & \\
\qquad \text{for } 1 \le i \le 2^{n-1} - 3,\ i \text{ odd} & \\
\left[\left(i\delta + \dfrac{2^n\epsilon(2+\epsilon)}{1+2\epsilon}\right)2^{-n}, \left((i+1)\delta - \dfrac{2^n\epsilon^2}{\delta}\right)2^{-n}\right] & \\
\qquad \text{for } 2 \le i \le 2^{n-1} - 2,\ i \text{ even} & \\
\left[\left((2^{n-1}-1)\delta - \dfrac{2^n\epsilon^2}{\delta}\right)2^{-n}, 1/2\right] & \text{for } i = 2^{n-1} - 1 \\
\left[1/2, \left((2^{n-1}+1)\delta + \dfrac{2^n\epsilon(2-3\epsilon)}{\delta}\right)2^{-n}\right] & \text{for } i = 2^{n-1} \\
\left[\left(i\delta + \dfrac{2^n\epsilon(2-3\epsilon)}{\delta}\right)2^{-n}, \left((i+1)\delta + \dfrac{3\cdot 2^n\epsilon^2}{1+2\epsilon}\right)2^{-n}\right] & \\
\qquad \text{for } 2^{n-1} + 1 \le i \le 2^n - 3,\ i \text{ odd} & \\
\left[\left(i\delta + \dfrac{3\cdot 2^n\epsilon^2}{1+2\epsilon}\right)2^{-n}, \left((i+1)\delta + \dfrac{2^n\epsilon(2-3\epsilon)}{\delta}\right)2^{-n}\right] & \\
\qquad \text{for } 2^{n-1} + 2 \le i \le 2^n - 2, i \text{ even} & \\
\left[\left((2^n-1)\delta + \dfrac{2^n\epsilon(2-3\epsilon)}{\delta}\right)2^{-n}, 1\right] & \text{for } i = 2^n - 1.
\end{cases}$$

If $n = 2$ *and* $\epsilon \in [1/4, 1/2)$*, then*

$$\overline{R}_n(i) = \begin{cases} \emptyset & \text{for } i = 0 \\ [0, 1/2] & \text{for } i = 1 \\ [1/2, 1] & \text{for } i = 2 \\ \emptyset & \text{for } i = 3. \end{cases}$$

If $n \geq 3$ and $\epsilon \in [\epsilon_n^*, 1/2)$, then

$$\overline{R}_n(i) =$$

$$\begin{cases} \left[0, \left(\delta - \dfrac{2^n\epsilon^2}{\delta}\right)2^{-n}\right] \\ \qquad\qquad \textit{for } i = 0 \textit{ and } \epsilon < 1/(2^{n/2}+2) \\ \left[\left(\delta - \dfrac{2^n\epsilon^2}{\delta}\right)2^{-n}, \big(4\delta + \delta^2 + 2^{n+1}\epsilon\big)2^{-n-1}\right] \cap [0,1] \quad \textit{for } i = 1 \\ \left[\big((2i-2)\delta + \delta^2 + 2^{n+1}\epsilon\big)2^{-n-1}, \big((2i+2)\delta + \delta^2 + 2^{n+1}\epsilon\big)2^{-n-1}\right] \\ \qquad\qquad \textit{for } 3 \leq i \leq 2^{n-1}-3,\ i \textit{ odd} \\ \left[\big((2^n-4)\delta + \delta^2 + 2^{n+1}\epsilon\big)2^{-n-1}, 1/2\right] \qquad \textit{for } i = 2^{n-1}-1 \\ \left[1/2, \big((2^n+2)\delta + 1 - 4\epsilon^2 + 2^{n+1}\epsilon\big)2^{-n-1}\right] \qquad \textit{for } i = 2^{n-1} \\ \left[\big((2i-2)\delta + 1 - 4\epsilon^2 + 2^{n+1}\epsilon\big)2^{-n-1}, \big((2i+2)\delta + 1 - 4\epsilon^2 + 2^{n+1}\epsilon\big)2^{-n-1}\right] \\ \qquad\qquad \textit{for } 2^{n-1}+2 \leq i \leq 2^n-4,\ i \textit{ even} \\ \left[\big((2^{n+1}-6)\delta + 1 - 4\epsilon^2 + 2^{n+1}\epsilon\big)2^{-n-1}, \left((2^n-1)\delta + \dfrac{2^n\epsilon(2-3\epsilon)}{\delta}\right)2^{-n}\right] \cap [0,1] \\ \qquad\qquad\qquad\qquad \textit{for } i = 2^n-2 \\ \left[\left((2^n-1)\delta + \dfrac{2^n\epsilon(2-3\epsilon)}{\delta}\right)2^{-n}, 1\right] \\ \qquad\qquad \textit{for } i = 2^n-1 \textit{ and } \epsilon < 1/(2^{n/2}+2) \\ \emptyset \qquad\qquad\qquad\qquad\qquad\qquad \textit{else.} \end{cases}$$

Corollary 3.8. *For an EOUQ with the CNC index assignment, the number of nonempty cells is*

$$|\Lambda| = \begin{cases} 2^n & \textit{for } \epsilon \in [0, \epsilon_n^*) \\ 2^{n-1}+2 & \textit{for } \epsilon \in [\epsilon_n^*, 1/(2^{n/2}+2)) \\ 2^{n-1} & \textit{for } \epsilon \in [1/(2^{n/2}+2), 1/2)\,. \end{cases}$$

If $\epsilon \in [\epsilon_n^*, 1/(2^{n/2}+2))$*, then the indices of the empty cells are*

$$\{i : 2 \leq i \leq 2^{n-1}-2,\ i\ even\} \cup \{i : 2^{n-1}+1 \leq i \leq 2^n-3,\ i\ odd\}.$$

If $\epsilon \in [1/(2^{n/2}+2), 1/2)$*, then the indices of the empty cells are*

$$\{i : 0 \leq i \leq 2^{n-1}-2,\ i\ even\} \cup \{i : 2^{n-1}+1 \leq i \leq 2^n-1,\ i\ odd\}.$$

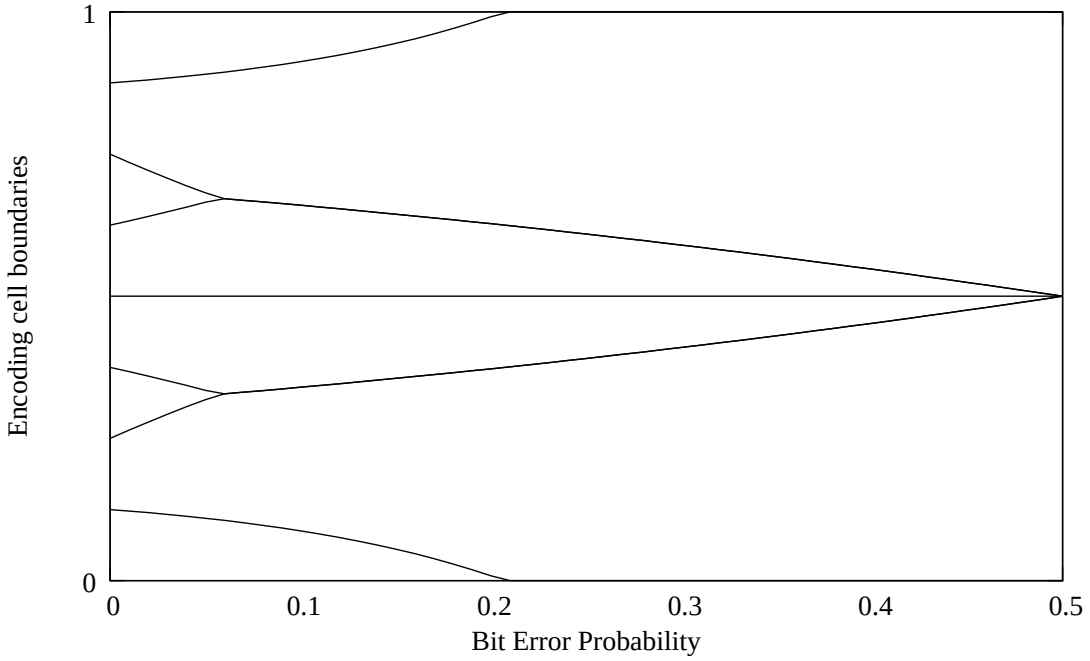

Figure 3.4: Plot of the encoding cells boundaries of a rate 3 EOUQ with the CNC index assignment as a function of bit error rate.

Figures 3.1 and 3.2 illustrate the encoding cells of a rate 3 EOUQ with the CNC index assignment for bit error rates 0.05 and 0.25 respectively. Figure 3.4 plots the encoding cell boundaries of a rate 3 EOUQ with the CNC index assignment as a function of bit error rate.

Theorem 3.9. *An EOUQ with the CNC index assignment has an effective channel code rate given as follows. Let h be the binary entropy function and let*

$$
\begin{aligned}
p_1 &= (1-2\epsilon)2^{-n} - \frac{\epsilon^2}{1-2\epsilon} \\
p_2 &= (1-2\epsilon)2^{-n} + \frac{\epsilon^2}{1-2\epsilon} + \epsilon \\
p_3 &= (1-2\epsilon)2^{-n} + \frac{2\epsilon(1-\epsilon)}{1-4\epsilon^2} \\
p_4 &= (1-2\epsilon)2^{-n} - \frac{2\epsilon(1-\epsilon)}{1-4\epsilon^2} \\
p_5 &= (1-2\epsilon)2^{-n} + 2^{-n-1}(1-2\epsilon)^2 + \frac{\epsilon^2}{1-2\epsilon} \\
p_6 &= (1-2\epsilon)2^{1-n} - 2^{-n-1}(1-2\epsilon)^2 \\
p_7 &= (1-2\epsilon)2^{1-n} + 2^{-n-1}(1-2\epsilon)^2 + \epsilon \\
p_8 &= \frac{1}{2}(1-2^{3-n})(1-2\epsilon)\left(n-1-\log\left(1-2\epsilon\right)\right).
\end{aligned}
$$

If $n = 2$ and $\epsilon \in [0, 1/4)$, then

$$
r_c = \frac{1}{2}\left(1 + h\left(\frac{1-2\epsilon}{2^{n-1}} - \frac{2\epsilon^2}{1-2\epsilon}\right)\right).
$$

If $n = 2$ and $\epsilon \in [1/4, 1/2)$, then $r_c = 1/2$.

If $n \geq 3$ and $\epsilon \in [0, \epsilon_n^*)$, then

$$
r_c = -\frac{2}{n}\left(p_1 \log p_1 + p_2 \log p_2 + (2^{n-2}-1)(p_3 \log p_3 + p_4 \log p_4)\right).
$$

If $n \geq 3$ and $\epsilon \in [\epsilon_n^*, 1/(2^{n/2}+2))$, then

$$
r_c = -\frac{2}{n}\left(p_1 \log p_1 + p_5 \log p_5 + p_6 \log p_6 - p_8\right).
$$

If $n \geq 3$ *and* $\epsilon \in [1/(2^{n/2}+2), 1/2)$*, then*

$$r_c = -\frac{2}{n}\left(p_7 \log p_7 - p_8 + p_6 \log p_6\right).$$

As $n \to \infty$ the effective channel code rate given by Theorem 3.9 converges to $1 - 2\epsilon$, for all $\epsilon \in [0, 1/2)$. Figure 3.5 plots the quantity r_c from Theorem 3.9 for rate $n = 4$.

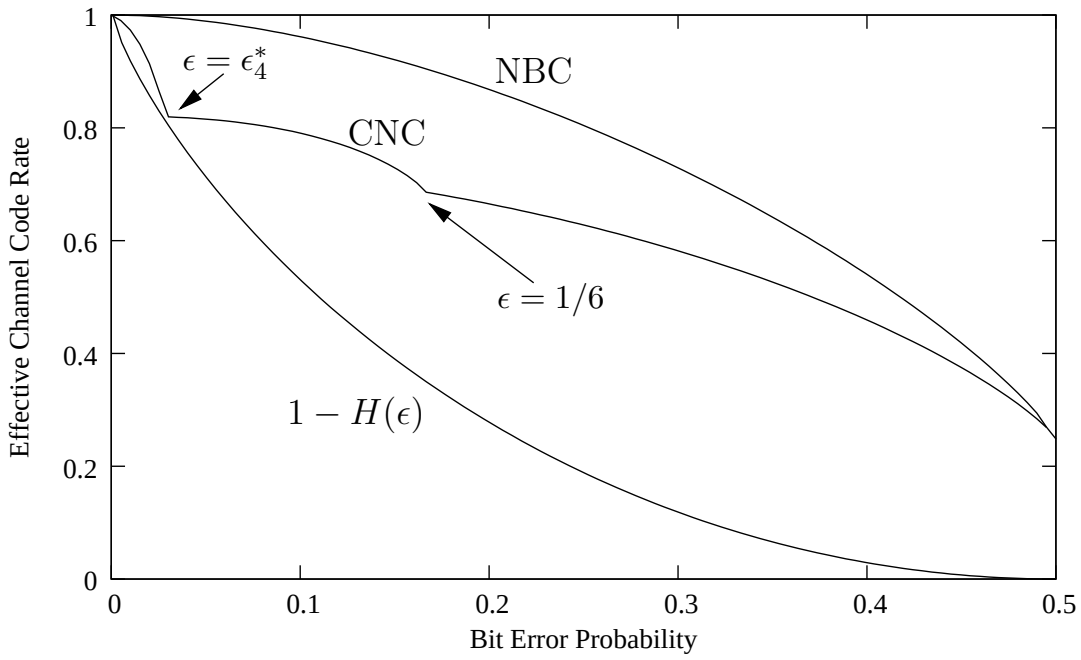

Figure 3.5: Plot of the effective channel code rate r_c of EOUQs with the NBC and the CNC index assignments for rate $n = 4$. The horizontal axis is the bit error probability ϵ of a binary symmetric channel. Also shown for comparison is the channel's capacity $1 - H(\epsilon)$.

Corollary 3.8 shows that given a bit error probability $\epsilon > 0$, for n sufficiently large an EOUQ with the CNC has half the number of nonempty encoding cells as one with the NBC. The following theorem shows that despite this fact, for a sequence of EOUQs, the CNC and the NBC induce the same cell density function (via Theorem 3.5).

Theorem 3.10. *A sequence of EOUQs with the CNC index assignment has a cell density function given by*

$$\gamma(x) = \begin{cases} \frac{1}{1-2\epsilon} & for\ \epsilon < x < 1-\epsilon \\ 0 & else. \end{cases}$$

Proof. For each $\epsilon > 0$ and n sufficiently large,

$$\frac{1}{2^{n/2}+2} < \epsilon$$

and therefore, by Corollary 3.8, the indices of the nonempty cells are

$$\{i : 1 \leq i \leq 2^{n-1}-1,\ i\ odd\} \cup \{i : 2^{n-1} \leq i \leq 2^n - 2,\ i\ even\}.$$

As n grows, the encoding cells $R_n(i)$ in Theorem 3.7 corresponding to $i = 2^{n-1}-1, 2^{n-1}$ do not affect the cell density function. At the same time, the right endpoint of the encoding cell in Theorem 3.7 corresponding to $i = 1$ converges to ϵ and the left endpoint of the encoding cell in Theorem 3.7 corresponding to $i = 2^n - 2$ converges to $1-\epsilon$. All other encoding cells have length $\delta 2^{1-n}$. Hence, in the limit as $n \to \infty$ they uniformly partition the interval $[\epsilon, 1-\epsilon]$. □

For completeness we derive the point density function of a DOUQ with the CNC. Analogous to the NBC, the cell density function in Theorem 3.10 is equal to the point density function for a sequence of DOUQs with the CNC.

Theorem 3.11. *A sequence of DOUQs with the CNC index assignment has a point density function given by*

$$\lambda(x) = \begin{cases} \frac{1}{1-2\epsilon} & for\ \epsilon < x < 1-\epsilon \\ 0 & else. \end{cases}$$

Proof. From [3], the codepoints of a DOUQ with the CNC index assignment satisfy

$$\begin{aligned} y_n(i) &= \sum_{j=0}^{2^n-1} \left(\frac{j+1/2}{2^n} \right) p_n\left(\pi_n^{(CNC)}(j) | \pi_n^{(CNC)}(i)\right) \\ &= 2^{-n} \cdot \begin{cases} (2^n-1)\epsilon + \delta(i+\epsilon) + \frac{1}{2} & for\ \ i\ \ even \\ (2^n-1)\epsilon + \delta(i-\epsilon) + \frac{1}{2} & for\ \ i\ \ odd \end{cases} \end{aligned} \tag{3.11}$$

where (3.11) follows from Lemma 3.24. Thus,

$$y_n(i+1) - y_n(i) \;\; = \;\; \begin{cases} 2^{-n}(1-4\epsilon^2) & for\ \ i\ \ odd \\ 2^{-n}(1-2\epsilon)^2 & for\ \ i\ \ even \end{cases}$$

which implies the codepoints are uniformly distributed in the limit as $n \to \infty$. Since $y_n(0) = \epsilon + 2^{-(n+1)}(1-4\epsilon^2) \to \epsilon$ as $n \to \infty$, and $y_n(2^n-1) = 1-\epsilon-2^{-(n+1)}(1-4\epsilon^2) \to 1-\epsilon$ as $n \to \infty$, the point density function is uniform on $(\epsilon, 1-\epsilon)$. □

3.5 Distortion Analysis

Let $D_{EO}^{(\pi_n)}$ denote the end-to-end MSE of an EOUQ with index assignment π_n. Recall that $\Lambda = \{i : R_n(i) \neq \emptyset\}$. For $i \in \Lambda$, define the quantities

$$\begin{aligned} I_r(i) &= \underset{\substack{j \in \Lambda \\ c_n(j) > c_n(i)}}{\operatorname{argmin}}\ c_n(j) \\ I_l(i) &= \underset{\substack{j \in \Lambda \\ c_n(j) < c_n(i)}}{\operatorname{argmax}}\ c_n(j) \\ z_n(i) &= \sup R_n(i) \end{aligned}$$

(I_r and I_l are defined when the argmin and argmax, respectively, exist). Also, define

$$V \;\; = \;\; \{i : 1 \notin \overline{R}_n(i)\} \cap \Lambda$$

$$I_1 \;=\; V^c \cap \Lambda.$$

$I_r(i)$ and $I_l(i)$ are the indices of cells immediately to the right and left, respectively, of the cell with index i; V is the set of indices of nonempty cells that don't contain 1; and I_1 is the index of the nonempty cell containing 1.

Lemma 3.12. *The mean squared error of a EOUQ with index assignment π_n is*

$$\begin{aligned} & D_{EO}^{(\pi_n)} \\ &= \frac{1}{3} - 2^{-n-1} + 2^{-2n-2} + 2^{-n}\left[\sum_{i\in V} z_n^2(i)\cdot\alpha_n(i, I_r(i)) - \sum_{j=0}^{2^n-1} j p_n(\pi_n(j)|\pi_n(I_1))\right] \\ &\quad + 2^{-2n}\sum_{j=0}^{2^n-1}(j+j^2)p_n(\pi_n(j)|\pi_n(I_1)). \end{aligned}$$

The next two theorems give the mean squared errors for the NBC with a channel unoptimized uniform quantizer and with an EOUQ. Theorem 3.13 was stated in [8] (see, e.g. [13] for a proof). The results are given as a function of the quantizer rate n and the channel bit error probability ϵ. Let $D_{CU}^{(\pi_n)}$ denote the end-to-end MSE of an channel unoptimized uniform quantizer with index assignment π_n.

With no channel noise the MSE is $2^{-2n}/12$. If a quantizer with the NBC is designed for a noiseless channel but used on a noisy channel, then Theorem 3.13 shows that (for large n) roughly $\epsilon/3$ is added to the MSE. If a quantizer with the NBC and a channel-optimized encoder is used on a noisy channel, then Theorem 3.14 shows that (for large n) the MSE is reduced by roughly $\epsilon^2/3$ from the channel unoptimized case.

Theorem 3.13. *The mean squared error of a channel unoptimized uniform quantizer with the NBC index assignment is*

$$D_{CU}^{(NBC)} = \frac{2^{-2n}}{12} + \frac{\epsilon}{3}(1 - 2^{-2n}).$$

Theorem 3.14. *The mean squared error of an EOUQ with the NBC index assignment is*

$$D_{EO}^{(NBC)} = D_{CU}^{(NBC)} - \frac{\epsilon^2}{3}(1-2\epsilon)(1-2^{-n})(1-2^{-n+1}).$$

Proof. For the NBC, $p_n(\pi_n(j)|\pi_n(i)) = p_n(j|i)$ and Theorem 3.2 and Corollary 3.3 imply that $V = \{0, 1, \ldots, 2^n - 2\}$, $I_r(i) = i + 1$, and $I_1 = 2^n - 1$. Hence, Lemma 3.12 gives

$$\begin{aligned}
&D_{EO}^{(NBC)} - \left(\frac{1}{3} - 2^{-n-1} + 2^{-2n-2}\right) \\
&= 2^{-2n}\sum_{j=0}^{2^n-1} j^2 p_n(j|2^n-1) + (2^{-2n} - 2^{-n})\sum_{j=0}^{2^n-1} j p_n(j|2^n-1) \\
&\quad + 2^{-n}\sum_{i=0}^{2^n-2} z_n^2(i)\cdot\alpha_n(i,i+1) \\
&= 2^{-2n}\left(\frac{\epsilon}{3}(4^n-1) + \frac{2\epsilon^2}{3}(2^n-1)(2^n-2) + \delta(2^n-1)^2\right) \\
&\quad + (2^{-2n} - 2^{-n})(2^n-1)(1-\epsilon) - \delta 2^{-n}\sum_{i=0}^{2^n-2}[\epsilon + \delta(i+1)2^{-n}]^2 \qquad (3.12) \\
&= 2^{-2n}\left(\frac{\epsilon}{3}(4^n-1) + \frac{2\epsilon^2}{3}(2^n-1)(2^n-2) - \epsilon(2^n-1)^2\right) \\
&\quad - \delta 2^{-n}\Bigg[(2^n-1)\epsilon^2 + (2^n-1)2\epsilon\delta 2^{-n} + (2^n-1)(2^n-2)\epsilon\delta 2^{-n} \\
&\quad + \frac{(2^n-1)(2^n-2)(2^{n+1}-3)\delta^2 2^{-2n}}{6} + (2^n-1)^2\delta^2 2^{-2n}\Bigg] \qquad (3.13) \\
&D_{EO}^{(NBC)} = \frac{2^{-2n}}{12} + \frac{(2^n-1)(2^n-2)(2\epsilon^3-\epsilon^2) + (2^{2n}-1)\epsilon}{3\cdot 2^{2n}} \qquad (3.14)
\end{aligned}$$

where the last three terms in (3.12) follow from Lemma 3.21, Lemma 3.20, and (3.6) and (3.10) respectively; and where (3.14) follows from (3.13) after arithmetic. □

Let $D_{DO}^{(\pi_n)}$ denote the end-to-end MSE of a DOUQ with index assignment π_n. For a given n and ϵ, an index assignment $\pi_n \in S_n$ is said to be *optimal for an EOUQ* if

for all $\pi'_n \in S_n$,

$$D_{EO}^{(\pi_n)} \leq D_{EO}^{(\pi'_n)}$$

and is said to be *optimal for a DOUQ* if for all $\pi'_n \in S_n$,

$$D_{DO}^{(\pi_n)} \leq D_{DO}^{(\pi'_n)}.$$

In [3] it was shown that for all n and all ϵ the NBC is optimal for a DOUQ. Theorems 3.13 and 3.14 show that with the NBC, the reduction in MSE obtained by using a channel-optimized quantizer encoder instead of one obeying the nearest neighbor condition is

$$\frac{(\epsilon^2 - 2\epsilon^3)(2^n - 1)(2^n - 2)}{3 \cdot 2^{2n}}.$$

The next two theorems show, however, that the NBC is not optimal for an EOUQ for all n and all ϵ.

Theorem 3.15. *The mean squared error of an EOUQ with the CNC index assignment is*

$$D_{EO}^{(CNC)} = \begin{cases} D_1(n,\epsilon) & \textit{for } \ 0 \leq \epsilon < \epsilon_n^* \\ D_2(n,\epsilon) & \textit{for } \ \epsilon_n^* \leq \epsilon < \frac{1}{2^{n/2}+2} \\ D_3(n,\epsilon) & \textit{for } \ \frac{1}{2^{n/2}+2} \leq \epsilon < 1/2 \end{cases}$$

where

$$\begin{aligned} &D_1(n,\epsilon) \\ &= \frac{2^{-2n}}{3(1+2\epsilon)}\Big((1/4) + (2^{2n} + (5/2))\epsilon - (2^{2n+1} - 15 \cdot 2^n + 4)\epsilon^2 \\ &\quad + 6(2^{2n} - 2^{n+2} - 4)\epsilon^3 + (2^n - 4)(2^n - 2)\epsilon^4 - 12(2^n - 4)\epsilon^5\Big) \\ &D_2(n,\epsilon) \\ &= \frac{2^{-3n}}{3}\Big(2^n - 3 + [(2^n - 3)(2^{2n} + 10) - 2^{n-1} + 48]\epsilon \end{aligned}$$

$$
\begin{aligned}
&- [(2^n-6)(2^n-5)(2^n-4) - 3(2^{2n}+2^{n+2}-48)]\epsilon^2 \\
&+ 2(2^n-4)(2^{2n}-11\cdot 2^n+6)\epsilon^3 + 6(2^n-6)(2^n-4)\epsilon^4 + 24(2^n-4)\epsilon^5\Big)
\end{aligned}
$$

$$
\begin{aligned}
&D_3(n,\epsilon) \\
&= \frac{2^{-3n}}{3}\Big(2^n+3+[(2^n-3)(2^{2n}+10)-2^{n-1}]\epsilon - [(2^n-6)(2^n-5)(2^n-4) \\
&\quad - 3\cdot 2^{2n}]\epsilon^2 + 2(2^n-6)(2^n-5)(2^n-4)\epsilon^3 + 12(2^n-5)(2^n-4)\epsilon^4 \\
&\quad + 24(2^n-4)\epsilon^5\Big).
\end{aligned}
$$

Lemma 3.16. *On the interval* $[0, 1/2]$*, the polynomial*

$$g_n(\epsilon) = 4(2^n-4)\epsilon^4 + 2^n(2^n-2)\epsilon^3 - 2(2^{2n}-2^{n+2}-4)\epsilon^2 + 2^n(2^n-4)\epsilon - 1$$

has exactly one root $\hat{\epsilon}_n$*, and* $g_n(\epsilon) < 0$ *if and only if* $\epsilon < \hat{\epsilon}_n$*. Furthermore,* $2^{-2n} < \hat{\epsilon}_n < 2^{-2n+1}$ *when* $n \geq 4$*.*

Note that $g_n(2^{-2n+a}) \to 2^a - 1 > 0$ as $n \to \infty$, for any $a > 0$. Hence, the bound on $\hat{\epsilon}_n$ can be strengthened to $2^{-2n} < \hat{\epsilon}_n < 2^{-2n+a}$, for arbitrarily small $a > 0$ and sufficiently large n. Thus, $\hat{\epsilon}_n \sim 2^{-2n}$, for asymptotically large n.

The following theorem shows that the quantity $\hat{\epsilon}_n$ defined in Lemma 3.16 is a threshold value for the bit error probability of a binary symmetric channel, beyond which the MSE of an EOUQ with the CNC index assignment is smaller than with the NBC, for $n \geq 3$. Lemma 3.16 then implies that the NBC is sub-optimal for a large range of transmission rates and bit error probabilities (i.e., for all ϵ and n satisfying $\epsilon > 2^{-2n+o(1)}$, where $o(1) \to 0$ as $n \to \infty$). In particular, for every $\epsilon > 0$, the CNC index assignment eventually outperforms the NBC for a large enough transmission rate. Figure 3.6 plots the quantity $D_{EO}^{(NBC)} - D_{EO}^{(CNC)}$ as a function of ϵ for rate $n = 3$.

Theorem 3.17. $D_{EO}^{(CNC)} < D_{EO}^{(NBC)}$ *if and only if* $n \geq 3$ *and* $\epsilon > \hat{\epsilon}_n$*.*

Some intuition for why EOUQs with the CNC achieve lower MSEs than those

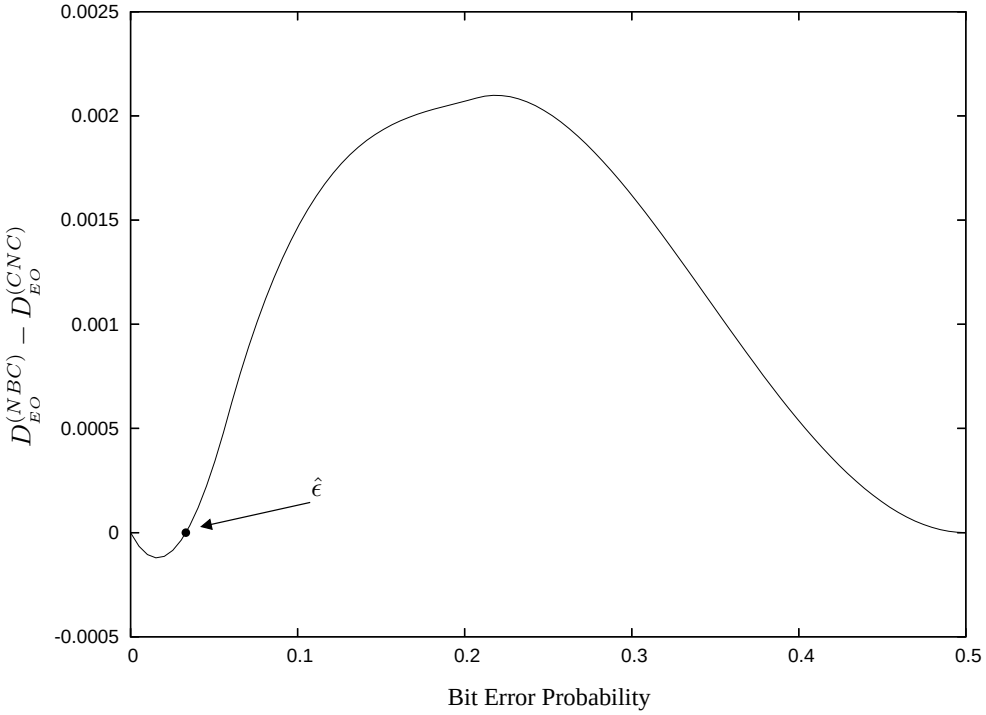

Figure 3.6: Plot of the difference in MSE achieved by EOUQs with the NBC index assignment and CNC index assignment for rate $n = 3$. The horizontal axis is the bit error probability ϵ of a binary symmetric channel. The quantity $\hat{\epsilon}_n$ from Lemma 3.16 is also shown.

with the NBC can be gained by examining the index generated by the CNC. For every $\epsilon > 0$ and for n sufficiently large, we have

$$\frac{1}{2^{n/2}+2} < \epsilon$$

which, by Corollary 3.8, implies the indices of the nonempty cells in an EOUQ with the CNC are

$$\{i : 1 \leq i \leq 2^{n-1} - 1,\ i\ odd\} \cup \{i : 2^{n-1} \leq i \leq 2^n - 2,\ i\ even\}.$$

Corresponding to such nonempty cells, the encoder transmits (by the definition of CNC) only the odd integers $1, 3, \ldots, 2^n - 1$. Hence, the encoder of an EOUQ with the CNC

emulates the encoder of a rate $n-1$ EOUQ with the NBC, and then adds an extra bit (carrying no information) before transmission over the channel. Since the CNC uses longer codewords than the NBC, the CNC codewords are exposed to fewer channel errors on average, while being penalized with a lower level of quantizer resolution. This tradeoff makes the CNC superior to the NBC, except for very small bit error rates.

3.6 Acknowledgment

The authors would like to thank two anonymous reviewers for their excellent suggestions and careful reading of the manuscript.

Appendix

3.7 Lemmas and Proofs for Section 3.2

Lemma 3.18. *For any index assignment $\pi_n \in S_n$ and for $0 \leq i \leq 2^n - 1$,*

$$\sum_{j=0}^{2^n-1} (1-\epsilon)^{n-H_n(\pi_n(i),\pi_n(j))} \epsilon^{H_n(\pi_n(i),\pi_n(j))} = 1.$$

Proof of Lemma 3.18. It follows immediately since index assignments are permutations. □

Proof of Lemma 3.1. Let i and k be two distinct integers between 0 and $2^n - 1$. Then the inequality in (3.2) can be rewritten as

$$\sum_{j=0}^{2^n-1} [x^2 - 2xy_n(j) + y_n^2(j)] p_n(\pi_n(j)|\pi_n(i)) \leq \sum_{j=0}^{2^n-1} [x^2 - 2xy_n(j) + y_n^2(j)] p_n(\pi_n(j)|\pi_n(k)).$$

Since π_n is bijective and $\sum_j p_n(j|i) = 1, \ \forall j$, cancellation of terms gives

$$\sum_{j=0}^{2^n-1} \left[\frac{-2xj}{2^n} + \frac{(j^2+j)}{2^{2n}} \right] p_n(\pi_n(j)|\pi_n(i)) \leq \sum_{j=0}^{2^n-1} \left[\frac{-2xj}{2^n} + \frac{(j^2+j)}{2^{2n}} \right] p_n(\pi_n(j)|\pi_n(k))$$

or equivalently,

$$x \sum_{j=0}^{2^n-1} j[p_n(\pi_n(j)|\pi_n(i)) - p_n(\pi_n(j)|\pi_n(k))]$$

$$\geq 2^{-n-1} \sum_{j=0}^{2^n-1} (j^2+j)[p_n(\pi_n(j)|\pi_n(i)) - p_n(\pi_n(j)|\pi_n(k))].$$

□

3.8 Lemmas and Proofs for Section 3.3

The following lemma is easy to prove and is used in the proofs of Lemmas 3.20 and 3.21.

Lemma 3.19.

$$H_{n+1}(i,j) = H_n(i,j) \quad for \quad 0 \leq i,j \leq 2^n - 1 \tag{3.15}$$

$$H_{n+1}(i,j+2^n) = H_n(i,j) + 1 \quad for \quad 0 \leq i,j \leq 2^n - 1 \tag{3.16}$$

$$H_{n+1}(i,j) = H_n(i-2^n,j) + 1 \quad for \quad 0 \leq j \leq 2^n - 1, 2^n \leq i \leq 2^{n+1} - 1 \tag{3.17}$$

$$H_{n+1}(i,j) = H_n(i-2^n,j-2^n) \quad for \quad 2^n \leq i,j \leq 2^{n+1} - 1. \tag{3.18}$$

Lemma 3.20. *If* $0 \leq i \leq 2^n - 1$*, then*

$$\sum_{j=0}^{2^n-1} j p_n\left(\pi_n^{(NBC)}(j)|\pi_n^{(NBC)}(i)\right) = (2^n-1)\epsilon + i\delta. \tag{3.19}$$

Proof of Lemma 3.20. We use induction on n. The case of $n = 1$ is true since

$$\begin{aligned} p_1\left(\pi_n^{(NBC)}(j)|\pi_n^{(NBC)}(i)\right) &= p_1(j|i) \\ \sum_{j=0}^{1} j p_1(j|0) &= \epsilon \\ \sum_{j=0}^{1} j p_1(j|1) &= 1-\epsilon. \end{aligned}$$

Now assume (3.19) is true for n and consider two cases for $n+1$.

If $0 \le i \le 2^n - 1$, then using (3.15) and (3.16) to express $p_{n+1}(j|i)$ in terms of $p_n(j|i)$ and simplifying with Lemma 3.18 gives

$$\begin{aligned} \sum_{j=0}^{2^{n+1}-1} j p_{n+1}(j|i) &= (1-\epsilon)\sum_{j=0}^{2^n-1} j p_n(j|i) + \epsilon \sum_{j=0}^{2^n-1} j p_n(j|i) + 2^n\epsilon \\ &= (2^{n+1}-1)\epsilon + i\delta \end{aligned} \tag{3.20}$$

where (3.20) follows from the induction hypothesis.

If $2^n \le i \le 2^{n+1} - 1$, then using (3.17) and (3.18) to express $p_{n+1}(j|i)$ in terms of $p_n(j|i)$ and simplifying with Lemma 3.18 gives

$$\begin{aligned} \sum_{j=0}^{2^{n+1}-1} j p_{n+1}(j|i) &= \epsilon\sum_{j=0}^{2^n-1} j p_n(j|i-2^n) + (1-\epsilon) \sum_{j=0}^{2^n-1} j p_n(j|i-2^n) + 2^n(1-\epsilon) \\ &= (2^{n+1}-1)\epsilon + i\delta \end{aligned} \tag{3.21}$$

where (3.21) follows from the induction hypothesis. □

Lemma 3.21. *If* $0 \le i \le 2^n - 1$, *then*

$$\sum_{j=0}^{2^n-1} j^2 p_n\left(\pi_n^{(NBC)}(j)|\pi_n^{(NBC)}(i)\right) = \frac{\epsilon}{3}(4^n-1)+\frac{2\epsilon^2}{3}(2^n-1)(2^n-2)+i2\epsilon\delta(2^n-1)+i^2\delta^2. \tag{3.22}$$

Proof of Lemma 3.21. We use induction on n. The case of $n = 1$ is true since

$$\begin{aligned} p_1\left(\pi_n^{(NBC)}(j)|\pi_n^{(NBC)}(i)\right) &= p_1(j|i) \\ \sum_{j=0}^{1} j^2 p_1(j|0) &= \epsilon \\ \sum_{j=0}^{1} j^2 p_1(j|1) &= 1-\epsilon \end{aligned}$$

which satisfies the right hand side of (3.22). Now assume (3.22) is true for n and consider two cases for $n+1$.

If $0 \leq i \leq 2^n - 1$, then using (3.15) and (3.16) to express $p_{n+1}(j|i)$ in terms of $p_n(j|i)$ and simplifying with Lemma 3.18 gives

$$\begin{aligned}
&\sum_{j=0}^{2^{n+1}-1} j^2 p_{n+1}(j|i) \\
&\quad = (1-\epsilon)\sum_{j=0}^{2^n-1} j^2 p_n(j|i) + \epsilon \sum_{j=0}^{2^n-1} j^2 p_n(j|i) + \epsilon\, 2^{n+1} \sum_{j=0}^{2^n-1} j p_n(j|i) + 2^{2n}\epsilon \\
&\quad = \frac{\epsilon}{3}(4^{n+1}-1) + \frac{2\epsilon^2}{3}(2^{n+1}-1)(2^{n+1}-2) + i2\epsilon\delta(2^{n+1}-1) + i^2\delta^2 \qquad (3.23)
\end{aligned}$$

where (3.23) follows from the induction hypothesis and Lemma 3.20.

If $2^n \leq i \leq 2^{n+1} - 1$, then using (3.17) and (3.18) to express $p_{n+1}(j|i)$ in terms of $p_n(j|i)$ and simplifying with Lemma 3.18 gives

$$\begin{aligned}
&\sum_{j=0}^{2^{n+1}-1} j^2 p_{n+1}(j|i) \\
&\quad = \epsilon \sum_{j=0}^{2^n-1} j^2 p_n(j|i-2^n) + (1-\epsilon)\sum_{j=0}^{2^n-1} j^2 p_n(j|i-2^n) \\
&\quad + (1-\epsilon)2^{n+1} \sum_{j=0}^{2^n-1} j p_n(j|i-2^n) + 2^{2n}(1-\epsilon) \\
&\quad = \frac{\epsilon}{3}(4^{n+1}-1) + \frac{2\epsilon^2}{3}(2^{n+1}-1)(2^{n+1}-2) + i2\epsilon\delta(2^{n+1}-1) + i^2\delta^2 \qquad (3.24)
\end{aligned}$$

where (3.24) follows from the induction hypothesis and Lemma 3.20. □

3.9 Lemmas and Proofs for Section 3.4

The following two lemmas are used in the proofs of Lemmas 3.24 and 3.26. Let

$$\hat{H}_n(i,j) = H(\pi_n^{(CNC)}(i), \pi_n^{(CNC)}(j)).$$

Lemma 3.22.

$$\hat{H}_n(i,j) = H_n(i,j) \qquad for\ \ 0 \le i,j \le 2^{n-1}-1\ \ or\ \ 2^{n-1} \le i,j \le 2^n - 1 \tag{3.25}$$

$$\hat{H}_n(i,j) = H_n(i,j+1)\ for\ \ 0 \le i \le 2^{n-1}-1,\ 2^{n-1} \le j \le 2^n - 2,\ and\ \ j\ even \tag{3.26}$$

$$\hat{H}_n(i,j) = H_n(i,j-1)\ for\ \ 0 \le i \le 2^{n-1}-1,\ 2^{n-1}+1 \le j \le 2^n - 1,\ and\ \ j\ odd \tag{3.27}$$

$$\hat{H}_n(i,j) = H_n(i,j+1)\ for\ \ 2^{n-1} \le i \le 2^n-1,\ 0 \le j \le 2^{n-1} - 2,\ and\ j\ even \tag{3.28}$$

$$\hat{H}_n(i,j) = H_n(i,j-1)\ for\ \ 2^{n-1} \le i \le 2^n-1,\ 1 \le j \le 2^{n-1} - 1,\ and\ j\ odd. \tag{3.29}$$

Proof of Lemma 3.22. It follows from the definition of the CNC. □

Lemma 3.23. *If* $0 \le i \le 2^{n-1}-1$, *then*

$$\sum_{\substack{j=2^{n-1} \\ j\ even}}^{2^n-2} (1-\epsilon)^{n-H_n(i,j)} \epsilon^{H_n(i,j)} = \begin{cases} \epsilon(1-\epsilon) & for\ i\ even \\ \epsilon^2 & for\ i\ odd \end{cases} \tag{3.30}$$

$$\sum_{\substack{j=2^{n-1}+1 \\ j\ odd}}^{2^n-1} (1-\epsilon)^{n-H_n(i,j)} \epsilon^{H_n(i,j)} = \begin{cases} \epsilon^2 & for\ i\ even \\ \epsilon(1-\epsilon) & for\ i\ odd \end{cases} \tag{3.31}$$

and if $2^{n-1} \leq i \leq 2^n - 1$, *then*

$$\sum_{\substack{j=0 \\ j\ even}}^{2^{n-1}-2} (1-\epsilon)^{n-H_n(i,j)} \epsilon^{H_n(i,j)} = \begin{cases} \epsilon(1-\epsilon) & for\ i\ even \\ \epsilon^2 & for\ i\ odd \end{cases} \tag{3.32}$$

$$\sum_{\substack{j=1 \\ j\ odd}}^{2^{n-1}-1} (1-\epsilon)^{n-H_n(i,j)} \epsilon^{H_n(i,j)} = \begin{cases} \epsilon^2 & for\ i\ even \\ \epsilon(1-\epsilon) & for\ i\ odd. \end{cases} \tag{3.33}$$

Proof of Lemma 3.23. It follows from the definition of the NBC. □

Lemma 3.24. *If* $0 \leq i \leq 2^n - 1$, *then*

$$\sum_{j=0}^{2^n-1} j p_n\left(\pi_n^{(CNC)}(j) | \pi_n^{(CNC)}(i)\right) = \begin{cases} (2^n-1)\epsilon + \delta(i+\epsilon) & for \quad i \ \ even \\ (2^n-1)\epsilon + \delta(i-\epsilon) & for \quad i \ \ odd. \end{cases}$$

Proof of Lemma 3.24. If $0 \leq i \leq 2^{n-1} - 1$, then using (3.25), (3.26), and (3.27) in Lemma 3.22 to express $p_n\left(\pi_n^{(CNC)}(j) | \pi_n^{(CNC)}(i)\right)$ in terms of ϵ, n, and $H_n(i,j)$ gives

$$\begin{aligned} &\sum_{j=0}^{2^n-1} j p_n\left(\pi_n^{(CNC)}(j) | \pi_n^{(CNC)}(i)\right) \\ &= \sum_{j=0}^{2^{n-1}-1} j(1-\epsilon)^{n-H_n(i,j)} \epsilon^{H_n(i,j)} + \sum_{\substack{j=2^{n-1}+1 \\ j\ odd}}^{2^n-1} (j-1)(1-\epsilon)^{n-H_n(i,j)} \epsilon^{H_n(i,j)} \\ &\quad + \sum_{\substack{j=2^{n-1} \\ j\ even}}^{2^n-2} (j+1)(1-\epsilon)^{n-H_n(i,j)} \epsilon^{H_n(i,j)} \\ &= \begin{cases} (2^n-1)\epsilon + \delta(i+\epsilon) & for \quad i \ \ even \\ (2^n-1)\epsilon + \delta(i-\epsilon) & for \quad i \ \ odd \end{cases} \qquad (3.34) \\ & \qquad (3.35) \end{aligned}$$

where (3.34) follows from Lemma 3.20 and (3.30) and (3.31) in Lemma 3.23.

If $2^{n-1} \leq i \leq 2^n - 1$, then using (3.28), (3.29), and (3.25) in Lemma 3.22 to express

$p_n\left(\pi_n^{(CNC)}(j)|\pi_n^{(CNC)}(i)\right)$ in terms of ϵ, n, and $H_n(i,j)$ gives

$$\begin{aligned}
&\sum_{j=0}^{2^n-1} j p_n\left(\pi_n^{(CNC)}(j)|\pi_n^{(CNC)}(i)\right) \\
&= \sum_{\substack{j=1 \\ j\ odd}}^{2^{n-1}-1} (j-1)(1-\epsilon)^{n-H_n(i,j)}\epsilon^{H_n(i,j)} + \sum_{\substack{j=0 \\ j\ even}}^{2^{n-1}-2} (j+1)(1-\epsilon)^{n-H_n(i,j)}\epsilon^{H_n(i,j)} \\
&\quad + \sum_{j=2^{n-1}}^{2^n-1} j(1-\epsilon)^{n-H_n(i,j)}\epsilon^{H_n(i,j)} \\
&= \begin{cases} (2^n-1)\epsilon + \delta(i+\epsilon) & for \quad i \ \ even \\ (2^n-1)\epsilon + \delta(i-\epsilon) & for \quad i \ \ odd \end{cases}
\end{aligned} \tag{3.36}$$

where (3.36) follows from Lemma 3.20 and (3.32) and (3.33) in Lemma 3.23. □

The following lemma is used in the proof of Lemma 3.26.

Lemma 3.25. *If* $0 \leq i \leq 2^{n-1} - 1$, *then*

$$\begin{aligned}
&\sum_{\substack{j=2^{n-1} \\ j\ even}}^{2^n-2} j(1-\epsilon)^{n-H_n(i,j)}\epsilon^{H_n(i,j)} \\
&\qquad = \begin{cases} 2^{n-1}\epsilon(1-\epsilon^2) - 2\epsilon^2(1-\epsilon) + i\delta(1-\epsilon)\epsilon & for\ i\ even \\ (2^{n-1}-1)\epsilon^2 + i\delta\epsilon^2 + 2^{n-1}\epsilon^3 & for\ i\ odd \end{cases}
\end{aligned} \tag{3.37}$$

$$\begin{aligned}
&\sum_{\substack{j=2^{n-1}+1 \\ j\ odd}}^{2^n-1} j(1-\epsilon)^{n-H_n(i,j)}\epsilon^{H_n(i,j)} \\
&\qquad = \begin{cases} (2^{n-1}+1)\epsilon^2 + i\delta\epsilon^2 + (2^{n-1}-2)\epsilon^3 & for\ i\ even \\ 2^{n-1}\epsilon(1-\epsilon^2) + i\delta(1-\epsilon)\epsilon & for\ i\ odd \end{cases}
\end{aligned} \tag{3.38}$$

and if $2^{n-1} \leq i \leq 2^n - 1$, *then*

$$\sum_{\substack{j=0 \\ j \text{ even}}}^{2^{n-1}-2} j(1-\epsilon)^{n-H_n(i,j)}\epsilon^{H_n(i,j)}$$

$$= \begin{cases} (2^{n-1}-1)\epsilon^2(1-\epsilon) - \epsilon^2(1-\epsilon) + (i-2^{n-1})\delta(1-\epsilon)\epsilon & \textit{for } i \textit{ even} \\ (2^{n-1}-1)\epsilon^3 - \epsilon^2(1-\epsilon) + (i-2^{n-1})\delta\epsilon^2 & \textit{for } i \textit{ odd} \end{cases} \tag{3.39}$$

$$\sum_{\substack{j=1 \\ j \text{ odd}}}^{2^{n-1}-1} j(1-\epsilon)^{n-H_n(i,j)}\epsilon^{H_n(i,j)}$$

$$= \begin{cases} (2^{n-1}-1)\epsilon^3 + \epsilon^2(1-\epsilon) + (i-2^{n-1})\delta\epsilon^2 & \textit{for } i \textit{ even} \\ (2^{n-1}-1)\epsilon^2(1-\epsilon) + \epsilon^2(1-\epsilon) + (i-2^{n-1})\delta(1-\epsilon)\epsilon & \textit{for } i \textit{ odd}. \end{cases} \tag{3.40}$$

Proof of Lemma 3.25. For each sum in (3.37)-(3.40) the first and last digits of the binary expansions of i and j are constant over all terms in the sum. Therefore, their contribution to the Hamming distance $H_n(i,j)$ is the same for each term in the sum. Hence, by summing over the middle $n-2$ bits of j, the left hand sides of (3.37)-(3.40) are, respectively,

$$\begin{cases} \displaystyle\sum_{j=0}^{2^{n-2}-1} (2j+2^{n-1})(1-\epsilon)^{n-(H_{n-2}(i/2,j)+1)}\epsilon^{H_{n-2}(i/2,j)+1} & \textit{for } i \textit{ even} \\ \displaystyle\sum_{j=0}^{2^{n-2}-1} (2j+2^{n-1})(1-\epsilon)^{n-(H_{n-2}((i-1)/2,j)+2)}\epsilon^{H_{n-2}((i-1)/2,j)+2} & \textit{for } i \textit{ odd}, \end{cases}$$

$$\begin{cases} \displaystyle\sum_{j=0}^{2^{n-2}-1} (2j+1+2^{n-1})(1-\epsilon)^{n-(H_{n-2}(i/2,j)+2)}\epsilon^{H_{n-2}(i/2,j)+2} & \textit{for } i \textit{ even} \\ \displaystyle\sum_{j=0}^{2^{n-2}-1} (2j+1+2^{n-1})(1-\epsilon)^{n-(H_{n-2}((i-1)/2,j)+1)}\epsilon^{H_{n-2}((i-1)/2,j)+1} & \textit{for } i \textit{ odd}, \end{cases}$$

$$\begin{cases} \sum_{j=0}^{2^{n-2}-1} 2j(1-\epsilon)^{n-(H_{n-2}((i-2^{n-1})/2,j)+1)}\epsilon^{H_{n-2}((i-2^{n-1})/2,j)+1} & \text{for } i \text{ even} \\ \sum_{j=0}^{2^{n-2}-1} 2j(1-\epsilon)^{n-(H_{n-2}((i-1-2^{n-1})/2,j)+2)}\epsilon^{H_{n-2}((i-1-2^{n-1})/2,j)+2} & \text{for } i \text{ odd}, \end{cases}$$

$$\begin{cases} \sum_{j=0}^{2^{n-2}-1} (2j+1)(1-\epsilon)^{n-(H_{n-2}((i-2^{n-1})/2,j)+2)}\epsilon^{H_{n-2}((i-2^{n-1})/2,j)+2} & \text{for } i \text{ even} \\ \sum_{j=0}^{2^{n-2}-1} (2j+1)(1-\epsilon)^{n-(H_{n-2}((i-1-2^{n-1})/2,j)+1)}\epsilon^{H_{n-2}((i-1-2^{n-1})/2,j)+1} & \text{for } i \text{ odd}. \end{cases}$$

The right hand sides of (3.37)-(3.40) then follow from Lemma 3.20. □

Lemma 3.26. *If* $0 \le i \le 2^{n-1}-1$, *then*

$$\sum_{j=0}^{2^n-1} j^2 p_n\left(\pi_n^{(CNC)}(j)|\pi_n^{(CNC)}(i)\right)$$
$$= \begin{cases} \epsilon\left[\left(\frac{2^{2n}-1}{3}\right)+2^n+1\right]+\frac{\epsilon^2}{3}(2^{2n+1}-9\cdot 2^n-14)-\epsilon^3(2^{n+1}-8) & \\ \quad +i\delta[2\epsilon(2^n-1)+2\epsilon\delta]+i^2\delta^2 & \text{for } i \text{ even} \\ \epsilon\left[\left(\frac{2^{2n}-1}{3}\right)-2^n+1\right]+\frac{\epsilon^2}{3}(2^{2n+1}-3\cdot 2^n-2)+2^{n+1}\epsilon^3 & \\ \quad +i\delta[2\epsilon(2^n-1)-2\epsilon\delta]+i^2\delta^2 & \text{for } i \text{ odd} \end{cases}$$

and if $2^{n-1} \le i \le 2^n-1$, *then*

$$\sum_{j=0}^{2^n-1} j^2 p_n\left(\pi_n^{(CNC)}(j)|\pi_n^{(CNC)}(i)\right)$$
$$= \begin{cases} \epsilon\left[\left(\frac{2^{2n}-1}{3}\right)-2^n+1\right]+\frac{\epsilon^2}{3}(2^{2n+1}+9\cdot 2^n-14)-\epsilon^3(3\cdot 2^{n+1}-8) & \\ \quad +i\delta[2\epsilon(2^n-1)+2\epsilon\delta]+i^2\delta^2 & \text{for } i \text{ even} \\ \epsilon\left[\left(\frac{2^{2n}-1}{3}\right)+2^n+1\right]+\frac{\epsilon^2}{3}(2^{2n+1}-21\cdot 2^n-2)+3\cdot 2^{n+1}\epsilon^3 & \\ \quad +i\delta[2\epsilon(2^n-1)-2\epsilon\delta]+i^2\delta^2 & \text{for } i \text{ odd}. \end{cases}$$

Proof of Lemma 3.26. If $0 \leq i \leq 2^{n-1}-1$, then using (3.25), (3.26), and (3.27) in Lemma 3.22 to express $p_n\left(\pi_n^{(CNC)}(j)|\pi_n^{(CNC)}(i)\right)$ in terms of ϵ, n, and $H_n(i,j)$ and simplifying with (3.30) and (3.31) from Lemma 3.23 gives

$$\begin{aligned}
&\sum_{j=0}^{2^n-1} j^2 p_n\left(\pi_n^{(CNC)}(j)|\pi_n^{(CNC)}(i)\right) \\
&= \sum_{j=0}^{2^n-1} j^2(1-\epsilon)^{n-H_n(i,j)}\epsilon^{H_n(i,j)} + \epsilon \\
&\quad - 2\left[\sum_{\substack{j=2^{n-1}+1 \\ j \text{ odd}}}^{2^n-1} j(1-\epsilon)^{n-H_n(i,j)}\epsilon^{H_n(i,j)} - \sum_{\substack{j=2^{n-1} \\ j \text{ even}}}^{2^n-2} j(1-\epsilon)^{n-H_n(i,j)}\epsilon^{H_n(i,j)}\right] \\
&= \begin{cases}
\epsilon\left[\left(\frac{2^{2n}-1}{3}\right)+2^n+1\right] + \frac{\epsilon^2}{3}(2^{2n+1}-9\cdot 2^n-14) - \epsilon^3(2^{n+1}-8) & \\
\quad + i\delta[2\epsilon(2^n-1)+2\epsilon\delta] + i^2\delta^2 & \textit{for } i \textit{ even} \\
\epsilon\left[\left(\frac{2^{2n}-1}{3}\right)-2^n+1\right] + \frac{\epsilon^2}{3}(2^{2n+1}-3\cdot 2^n-2) + 2^{n+1}\epsilon^3 & \\
\quad + i\delta[2\epsilon(2^n-1)-2\epsilon\delta] + i^2\delta^2 & \textit{for } i \textit{ odd}
\end{cases}
\end{aligned} \tag{3.41}$$

where (3.41) follows from Lemma 3.21 and (3.37) and (3.38) in Lemma 3.25.

If $2^{n-1} \leq i \leq 2^n-1$, then using (3.25), (3.28), and (3.29) in Lemma 3.22 to express
$p_n\left(\pi_n^{(CNC)}(j)|\pi_n^{(CNC)}(i)\right)$ in terms of ϵ, n, and $H_n(i,j)$ and simplifying with (3.32) and (3.33) from Lemma 3.23 gives

$$\begin{aligned}
&\sum_{j=0}^{2^n-1} j^2 p_n\left(\pi_n^{(CNC)}(j)|\pi_n^{(CNC)}(i)\right) \\
&= \sum_{j=0}^{2^n-1} j^2(1-\epsilon)^{n-H_n(i,j)}\epsilon^{H_n(i,j)} + \epsilon
\end{aligned}$$

$$
\begin{aligned}
&- 2\left[\sum_{\substack{j=1\\ j\ odd}}^{2^{n-1}-1} j(1-\epsilon)^{n-H_n(i,j)}\epsilon^{H_n(i,j)} - \sum_{\substack{j=0\\ j\ even}}^{2^{n-1}-2} j(1-\epsilon)^{n-H_n(i,j)}\epsilon^{H_n(i,j)}\right] \\
&= \begin{cases}
\epsilon\left[\left(\frac{2^{2n}-1}{3}\right) - 2^n + 1\right] + \frac{\epsilon^2}{3}(2^{2n+1} + 9\cdot 2^n - 14) - \epsilon^3(3\cdot 2^{n+1} - 8) & \\
\quad + i\delta[2\epsilon(2^n - 1) + 2\epsilon\delta] + i^2\delta^2 & \textit{for } i \textit{ even} \\
\epsilon\left[\left(\frac{2^{2n}-1}{3}\right) + 2^n + 1\right] + \frac{\epsilon^2}{3}(2^{2n+1} - 21\cdot 2^n - 2) + 3\cdot 2^{n+1}\epsilon^3 & \\
\quad + i\delta[2\epsilon(2^n - 1) - 2\epsilon\delta] + i^2\delta^2 & \textit{for } i \textit{ odd}
\end{cases}
\end{aligned}
\tag{3.42}
$$

where (3.42) follows from Lemma 3.21 and (3.39) and (3.40) in Lemma 3.25. □

Proof of Lemma 3.6. Since $\phi_n''' < 0$, $\phi_n(0) = -1$, $\phi_n(1/2) = 2^{n-1}$, and $\phi_n(1) = -3$, the cubic function ϕ_n has exactly one root (i.e. ϵ_n^*) in $(0, 1/2)$, $\phi_n < 0$ on $[0, \epsilon_n^*)$, and $\phi_n > 0$ on $(\epsilon_n^*, 1/2]$. Furthermore, $\epsilon_n^* > \epsilon_{n+1}^*$ since

$$\phi_{n+1}(\epsilon) - \phi_n(\epsilon) = 2^{n+1}\epsilon(1-\epsilon), \ \ \forall \epsilon.$$

The fact that

$$\epsilon_n^* < \frac{1}{2^{n/2}+2}$$

follows from the fact that

$$\phi_n\left(\frac{1}{2^{n/2}+2}\right) = \frac{2^{2n+1} + 5\cdot 2^{3n/2}}{(2^{n/2}+2)^3} > 0.$$

□

Proof of Theorem 3.7. The encoding cells satisfy (3.3) in Lemma 3.1, with

$$\alpha_n(i,k) \ = \ \sum_{j=0}^{2^n-1} j\left[p_n(\pi_n^{(CNC)}(j)|\pi_n^{(CNC)}(i)) - p_n(\pi_n^{(CNC)}(j)|\pi_n^{(CNC)}(k))\right]$$

$$= \begin{cases} \delta(i-k) & for\ i,k\ even\ \ or\ \ i,k\ odd \\ \delta(i-k-2\epsilon) & for\ k\ even,\ i\ odd \\ \delta(i-k+2\epsilon) & for\ i\ even,\ k\ odd \end{cases} \tag{3.43}$$

where (3.43) follows from Lemma 3.24. Let $\rho_n(i,k) = \beta_n(i,k)/\alpha_n(i,k)$ for $i \neq k$. Note that $\rho_n(i,k)$ is well defined because $\alpha_n(i,k) \neq 0$ whenever $k \neq i$, by (3.43). Also, from (3.43), we have that $\alpha_n(i,k) > 0$ if and only if $i > k$, and $\alpha_n(i,k) < 0$ if and only if $i < k$.

Thus (3.3) can be rewritten as

$$\begin{aligned} \overline{R}_n(i) &= \{x \in [0,1] : x \geq \rho_n(i,k),\ \forall\ k < i\ \ and\ \ x \leq \rho_n(i,k),\ \forall\ k > i\} \\ &= \left\{ x \in [0,1] : \max_{k<i} \rho_n(i,k) \leq x \leq \min_{k>i} \rho_n(i,k) \right\}. \end{aligned} \tag{3.44}$$

Therefore, the encoding cell with index i is empty if and only if at least one of the following conditions holds

$$\max_{k<i} \rho_n(i,k) \geq \min_{k>i} \rho_n(i,k) \tag{3.45}$$

$$\min_{k>i} \rho_n(i,k) \leq 0 \tag{3.46}$$

$$\max_{k<i} \rho_n(i,k) \geq 1. \tag{3.47}$$

For notational convenience, assume $\max_{k<i} \rho_n(0,k) = 0$ and $\min_{k>i} \rho_n(2^n - 1, k) = 1$.

We will examine four cases, corresponding to the parity and size of a cell's index i.

Case 1 : i even , $0 \leq i \leq 2^{n-1} - 2$

Equations (3.5) and (3.43) and Lemma 3.26 imply

$$\rho_n(i,k)$$
$$= \begin{cases} \epsilon + 2^{-n-1}(i+k+1+2\epsilon)\delta & for \quad 0 \le k \le 2^{n-1}-2,\ k\ even \\ \epsilon + 2^{-n-1}(i+k+1+2\epsilon)\delta + \dfrac{\epsilon(1-\epsilon)}{i-k} & for \quad 2^{n-1} \le k \le 2^n-2,\ k\ even \\ \epsilon + 2^{-n-1}(i+k+1)\delta + \dfrac{\epsilon(1-\epsilon)}{i-k+2\epsilon} & for \quad 1 \le k \le 2^{n-1}-1,\ k\ odd \\ \epsilon + 2^{-n-1}(i+k+1)\delta & for \quad 2^{n-1}+1 \le k \le 2^n-1,\ k\ odd. \end{cases} \tag{3.48}$$

Equations (3.49) and (3.50) below follow from (3.48) and the fact that $\rho_n(i,k)$ is increasing in k for $k < i$ and $k > i$.

$$\max_{\substack{k<i \\ i\neq 0}} \rho_n(i,k) = \left(i\delta + \frac{2^n\epsilon(2+\epsilon)}{1+2\epsilon}\right)2^{-n} \tag{3.49}$$

$$\min_{k>i} \rho_n(i,k) = \left((i+1)\delta - \frac{2^n\epsilon^2}{\delta}\right)2^{-n}. \tag{3.50}$$

For $i \neq 0$ the ith encoding cell $R_n(i)$ is nonempty if and only if the conditions in (3.45)–(3.47) are each false. (3.49), (3.50), and Lemma 3.6 imply (3.45) is false if and only if

$$\left(i\delta + \frac{2^n\epsilon(2+\epsilon)}{1+2\epsilon}\right)2^{-n} < \left((i+1)\delta - \frac{2^n\epsilon^2}{\delta}\right)2^{-n}$$

or equivalently, if and only if

$$\epsilon < \epsilon_n^*. \tag{3.51}$$

(3.46) is false if and only if (3.50) is positive, or equivalently,

$$\epsilon < \frac{1}{2^{(n/2)-(1/2)\log(i+1)} + 2} \,. \tag{3.52}$$

Similarly, (3.47) is false if and only if (3.49) is less than 1, or equivalently

$$i < \frac{2^n(1-\epsilon^2)}{1-4\epsilon^2} \tag{3.53}$$

which is always true, since $\frac{1-\epsilon^2}{1-4\epsilon^2} > 1$ and $i < 2^n$. Lemma 3.6 implies that

$$\epsilon_n^* < \frac{1}{2^{(n/2)-(1/2)\log(i+1)} + 2}$$

for $i \geq 0$. Hence, if $\epsilon < \epsilon_n^*$, then (3.52) holds, and therefore $R_n(i)$ is nonempty for $i \neq 0$ if and only if $\epsilon < \epsilon_n^*$.

For $i = 0$ the conditions in (3.45) and (3.46) are equivalent and the condition in (3.47) is always false. Therefore, the encoding cell $R_n(0)$ is nonempty (from (3.46) and (3.50)) if and only if

$$\epsilon < \frac{1}{2^{n/2} + 2}. \tag{3.54}$$

Case 2 : i odd , $1 \leq i \leq 2^{n-1} - 1$

Equations (3.5) and (3.43) and Lemma 3.26 imply

$\rho_n(i,k)$

$$= \begin{cases} \epsilon + 2^{-n-1}(i+k+1)\delta - \dfrac{\epsilon(1-\epsilon)}{i-k-2\epsilon} & for \quad 0 \le k \le 2^{n-1}-2,\ k\ even \\ \epsilon + 2^{-n-1}(i+k+1)\delta & for \quad 2^{n-1} \le k \le 2^n-2,\ k\ even \\ \epsilon + 2^{-n-1}(i+k+\delta)\delta & for \quad 1 \le k \le 2^{n-1}-1,\ k\ odd \\ \epsilon + 2^{-n-1}(i+k+\delta)\delta - \dfrac{\epsilon(1-\epsilon)}{i-k} & for \quad 2^{n-1}+1 \le k \le 2^n-1,\ k\ odd. \end{cases} \tag{3.55}$$

If $i \neq 1$, then from (3.55),

$$\begin{aligned} \max_{k<i} \rho_n(i,k) &= \max \left\{ \max_{\substack{k<i \\ k\ even}} \epsilon + 2^{-n-1}(i+k+1)\delta - \frac{\epsilon(1-\epsilon)}{i-k-2\epsilon}, \right. \\ &\qquad \left. \epsilon + 2^{-n-1}(i+k+\delta)\delta\Big|_{k=i-2} \right\} \qquad (3.56) \\ &= \begin{cases} \left(i\delta - \frac{2^n\epsilon^2}{\delta}\right)2^{-n} & for\ \epsilon < \epsilon_n^* \\ \left((2i-2)\delta + \delta^2 + 2^{n+1}\epsilon\right)2^{-n-1} & for\ \epsilon \ge \epsilon_n^*. \end{cases} \qquad (3.57) \end{aligned}$$

Equation (3.57) was obtained by noting that in (3.56) the first term is greater than the second term if and only if both $k = i-1$ (since k is even) and (after some algebra) $\phi(\epsilon) < 0$ (i.e. $\epsilon < \epsilon_n^*$ via Lemma 3.6). If $i = 1$, then from (3.55),

$$\max_{k<1} \rho_n(i,k) = \rho_n(1,0) = \left(\delta - \frac{2^n\epsilon^2}{\delta}\right)2^{-n}. \tag{3.58}$$

For $i \neq 2^{n-1}-1$,

$$\begin{aligned} &\min_{k>i} \rho_n(i,k) \\ &= \min \left\{ \min_{\substack{k>i \\ 2\le k<2^{n-1} \\ k\ even}} \epsilon + 2^{-n-1}(i+k+1)\delta - \frac{\epsilon(1-\epsilon)}{i-k-2\epsilon}, \right. \\ &\qquad \left. \epsilon + 2^{-n-1}(i+k+1)\delta\Big|_{k=2^{n-1}}, \right. \end{aligned}$$

$$\epsilon + 2^{-n-1}(i+k+\delta)\delta\Big|_{k=i+2}\,,$$

$$\min_{\substack{i<k \\ 2^{n-1}+1<k<2^n-1 \\ k\ odd}} \epsilon + 2^{-n-1}(i+k+\delta)\delta - \frac{\epsilon(1-\epsilon)}{i-k}\Bigg\} \tag{3.59}$$

$$= \begin{cases} \left((i+1)\delta + \frac{2^n\epsilon(2+\epsilon)}{1+2\epsilon}\right)2^{-n} & \text{for } \epsilon < \epsilon_n^* \\ \big((2i+2)\delta + \delta^2 + 2^{n+1}\epsilon\big)2^{-n-1} & \text{for } \epsilon \geq \epsilon_n^*. \end{cases} \tag{3.60}$$

Equation (3.60) was obtained by noting that in (3.59) the third term is less than the forth term evaluated at any odd k between $2^{n-1}+1$ and 2^n-1; and the first term is less than the third term if and only if both $k = i+1$ (since k is even) and (after some algebra) $\phi(\epsilon) < 0$ (i.e. $\epsilon < \epsilon_n^*$ via Lemma 3.6). If $i = 2^{n-1}-1$, then

$$\min_{k>i} \rho_n(i,k)$$

$$= \min\Bigg\{ \epsilon + 2^{-n-1}(i+k+1)\delta\Big|_{k=2^{n-1}}\,,$$

$$\min_{\substack{i<k \\ 2^{n-1}+1<k<2^n-1 \\ k\ odd}} \epsilon + 2^{-n-1}(i+k+\delta)\delta - \frac{\epsilon(1-\epsilon)}{i-k}\Bigg\} \tag{3.61}$$

$$= \frac{1}{2}\,. \tag{3.62}$$

Equation (3.62) was obtained by noting that in (3.61) the second term evaluated at any odd k between $2^{n-1}+1$ and 2^n-1 is always greater than the first term.

The ith encoding cell $R_n(i)$ is nonempty if and only if the conditions in (3.45)-(3.47) are each false. Suppose $0 \leq \epsilon < \epsilon_n^*$. Then (3.57), (3.58), and (3.60) imply (3.45) is false for $i \neq 2^{n-1}-1$ if and only if

$$-\frac{2^n\epsilon^2}{\delta} < \delta + \frac{2^n\epsilon(2+\epsilon)}{1+2\epsilon} \tag{3.63}$$

which is always true. If $i = 2^{n-1} - 1$, then (3.57) and (3.62) imply (3.45) is false if and only if

$$-\delta - \frac{2^n \epsilon^2}{\delta} < 2^{n-1}(1-\delta) = 2^n \epsilon \tag{3.64}$$

which is always true. (3.60) and (3.62) imply (3.46) is always false since $\min_{k>i} \rho_n(i,k) > 0$, by inspection. (3.57) and (3.58) imply (3.47) is false if and only if

$$i\delta^2 < 2^n(1-\epsilon)^2 \tag{3.65}$$

which is always true. Hence, if $\epsilon \in [0, \epsilon_n^*)$, then $R_n(i)$ is nonempty.

Suppose $\epsilon_n^* \leq \epsilon < 1/2$. (3.57) and (3.60) imply (3.45) is false (assuming $i \neq 1$ and $i \neq 2^{n-1} - 1$) if and only if

$$\big((2i-2)\delta + \delta^2 + 2^{n+1}\epsilon\big)2^{-n-1} < \big((2i+2)\delta + \delta^2 + 2^{n+1}\epsilon\big)2^{-n-1} \tag{3.66}$$

which is always true. If $i = 2^{n-1} - 1$, then (3.57) and (3.62) imply (3.45) is false if and only if

$$2^n - \delta(4-\delta) < 2^n \tag{3.67}$$

which is always true. If $i = 1$, then (3.58) and (3.60) imply (3.45) is false if and only if

$$-2^{n+1}\epsilon^2 < 2\delta^2 + \delta^3 + 2^{n+1}\epsilon\delta \tag{3.68}$$

which is always true. (3.60) and (3.62) imply (3.46) is always false since $\min_{k>i} \rho_n(i,k) > 0$, by inspection. (3.57) implies (3.47) is false for $i \neq 1$ if and only if

$$(2i-2)\delta + \delta^2 < 2^{n+1}(1-\epsilon) \tag{3.69}$$

which is always true. If $i = 1$, then (3.65) implies that $\max_{k<i} \rho_n(1,k) < 1$ and hence (3.47) is false. Therefore, if $\epsilon \in [\epsilon_n^*, 1/2)$, then $R_n(i)$ is nonempty.

Case 3 : i even , $2^{n-1} \le i \le 2^n - 2$

Equations (3.5), (3.43), and (3.55) and Lemma 3.26 imply

$$\rho_n(i,k)$$
$$= \begin{cases} \epsilon + 2^{-n-1}(i+k+1+2\epsilon)\delta - \dfrac{\epsilon(1-\epsilon)}{i-k} & for \quad 0 \le k \le 2^{n-1}-2,\ k\ even \\ \epsilon + 2^{-n-1}(i+k+1+2\epsilon)\delta & for \quad 2^{n-1} \le k \le 2^n - 2,\ k\ even \\ \epsilon + 2^{-n-1}(i+k+1)\delta & for \quad 1 \le k \le 2^{n-1}-1,\ k\ odd \\ \epsilon + 2^{-n-1}(i+k+1)\delta - \dfrac{\epsilon(1-\epsilon)}{i-k+2\epsilon} & for \quad 2^{n-1}+1 \le k \le 2^n - 1,\ k\ odd \end{cases}$$
$$= 1 - \rho_n(2^n - 1 - i, 2^n - 1 - k). \tag{3.70}$$

Equation (3.70) implies that

$$\max_{k<i} \rho_n(i,k) = 1 - \min_{k<i} \rho_n(2^n - 1 - i, 2^n - 1 - k)$$
$$= \begin{cases} \left(i\delta + \dfrac{3 \cdot 2^n \epsilon^2}{1+2\epsilon}\right) 2^{-n} & for \quad i \ne 2^{n-1}\ and\ \epsilon < \epsilon_n^* \\ \big((2i-2)\delta + 1 - 4\epsilon^2 + 2^{n+1}\epsilon\big) 2^{-n-1} & for \quad i \ne 2^{n-1}\ and\ \epsilon \ge \epsilon_n^* \\ \dfrac{1}{2} & for \quad i = 2^{n-1} \end{cases} \tag{3.71}$$

where (3.71) follows from (3.60) and (3.62), and

$$\min_{k>i} \rho_n(i,k) = 1 - \max_{k>i} \rho_n(2^n - 1 - i, 2^n - 1 - k)$$

$$= \begin{cases} \left((i+1)\delta + \dfrac{2^n\epsilon(2-3\epsilon)}{\delta}\right)2^{-n} & for \quad \epsilon < \epsilon_n^* \\ \big((2i+2)\delta + 1 - 4\epsilon^2 + 2^{n+1}\epsilon\big)2^{-n-1} & for \quad i \neq 2^n - 2 \ and \ \epsilon \geq \epsilon_n^* \\ \left((2^n-1)\delta + \dfrac{2^n\epsilon(2-3\epsilon)}{\delta}\right)2^{-n} & for \quad i = 2^n - 2 \ and \ \epsilon \geq \epsilon_n^* \end{cases} \tag{3.72}$$

where (3.72) follows from (3.57) and (3.58).

The ith encoding cell $R_n(i)$ is nonempty if and only if the conditions in (3.45)-(3.47) are each false. (3.70) implies (3.45) is false if and only if

$$\min_{k<i} \rho_n(2^n - 1 - i, 2^n - 1 - k) > \max_{k>i} \rho_n(2^n - 1 - i, 2^n - 1 - k)$$
$$\iff \min_{k>j} \rho_n(j,k) > \max_{k<j} \rho_n(j,k) \quad for \ \ 1 \leq j \leq 2^{n-1} - 1, \ j \ odd$$

which is always true, as shown by (3.63), (3.64), (3.66), (3.67), and (3.68). (3.70) implies (3.46) is false if and only if

$$\max_{k>i} \rho_n(2^n - 1 - i, 2^n - 1 - k) < 1$$
$$\iff \max_{k<j} \rho_n(j,k) < 1 \quad for \ \ 1 \leq j \leq 2^{n-1} - 1, \ j \ odd$$

which is always true, as shown by (3.65) and (3.69). (3.70) implies (3.47) is false if and only if

$$\min_{k<i} \rho_n(2^n - 1 - i, 2^n - 1 - k) > 0$$
$$\iff \min_{k>j} \rho_n(j,k) > 0 \quad for \ \ 1 \leq j \leq 2^{n-1} - 1, \ j \ odd$$

which is always true, as shown by inspection of (3.60) and (3.62). Hence, $R_n(i)$ is nonempty.

Case 4 : i odd , $2^{n-1}+1 \le i \le 2^n - 1$

Equations (3.5), (3.43), and (3.48) and Lemma 3.26 imply

$$\rho_n(i,k)$$
$$= \begin{cases} \epsilon + 2^{-n-1}(i+k+1)\delta & for \quad 0 \le k \le 2^{n-1}-2,\ k\ even \\ \epsilon + 2^{-n-1}(i+k+1)\delta + \dfrac{\epsilon(1-\epsilon)}{i-k-2\epsilon} & for \quad 2^{n-1} \le k \le 2^n - 2,\ k\ even \\ \epsilon + 2^{-n-1}(i+k+\delta)\delta + \dfrac{\epsilon(1-\epsilon)}{i-k} & for \quad 1 \le k \le 2^{n-1}-1,\ k\ odd \\ \epsilon + 2^{-n-1}(i+k+\delta)\delta & for \quad 2^{n-1}+1 \le k \le 2^n - 1,\ k\ odd \end{cases}$$
$$= 1 - \rho_n(2^n - 1 - i, 2^n - 1 - k). \tag{3.73}$$

Equation (3.73) implies that

$$\max_{k<i} \rho_n(i,k) = 1 - \min_{k<i} \rho_n(2^n - 1 - i, 2^n - 1 - k) \tag{3.74}$$
$$= \left(i\delta + \frac{2^n \epsilon(2-3\epsilon)}{\delta} \right) 2^{-n} \tag{3.75}$$

where (3.75) follows from (3.50), and (assuming $i \neq 2^n - 1$)

$$\min_{k>i} \rho_n(i,k) = 1 - \max_{k>i} \rho_n(2^n - 1 - i, 2^n - 1 - k) \tag{3.76}$$
$$= \left((i+1)\delta + \frac{3 \cdot 2^n \epsilon^2}{1+2\epsilon} \right) 2^{-n} \tag{3.77}$$

where (3.77) follows from (3.49).

For $i \neq 2^n - 1$ the ith encoding cell, $R_n(i)$, is nonempty if and only if the conditions in (3.45)-(3.47) are each false. (3.74) and (3.76) implies (3.45) is false if and

only if

$$\begin{aligned}\min_{k<i} \rho_n(2^n-1-i, 2^n-1-k) &> \max_{k>i} \rho_n(2^n-1-i, 2^n-1-k) \\ &\iff \min_{k>j} \rho_n(j,k) > \max_{k<j} \rho_n(j,k) \quad for\ \ 2 \le j \le 2^{n-1}-2,\ j\ even \\ &\iff \epsilon < \epsilon_n^* \end{aligned} \tag{3.78}$$

where (3.78) follows from (3.51). (3.76) implies (3.46) is false if and only if

$$\begin{aligned}\max_{k>i} \rho_n(2^n-1-i, 2^n-1-k) &< 1 \\ &\iff \max_{k<j} \rho_n(j,k) < 1 \quad for\ \ 2 \le j \le 2^{n-1}-2,\ j\ even. \end{aligned} \tag{3.79}$$

Equation (3.53) implies (3.79) is always true. (3.74) implies (3.47) is false if and only if

$$\begin{aligned}\min_{k<i} \rho_n(2^n-1-i, 2^n-1-k) &> 0 \\ &\iff \min_{k>j} \rho_n(j,k) > 0 \quad for\ \ 2 \le j \le 2^{n-1}-2,\ j\ even. \end{aligned} \tag{3.80}$$

Equation (3.52) implies that (3.80) holds if and only if

$$\epsilon < \frac{1}{2^{(n/2)-(1/2)\log(2^n-1-i+1)}+2}\,. \tag{3.81}$$

Lemma 3.6 implies that ϵ_n^* is smaller than the right hand side of (3.81) for $i \le 2^n-1$. Hence, if $\epsilon < \epsilon_n^*$, then (3.81) holds and therefore, $R_n(i)$ is nonempty for $i \ne 2^n-1$ if and only if $\epsilon < \epsilon_n^*$.

For $i = 2^n-1$ the conditions in (3.45) and (3.47) are equivalent and the condition in (3.46) is always false. Therefore, the encoding cell $R_n(2^n-1)$ is nonempty (from

(3.47) and (3.74)) if and only if

$$\begin{aligned} \min_{k<2^n-1} \rho_n(2^n-1-(2^n-1), 2^n-1-k) &> 0 \\ \iff \min_{k>0} \rho_n(0,k) &> 0 \\ \iff \epsilon &< \frac{1}{2^{n/2}+2} \end{aligned} \tag{3.82}$$

where (3.82) follows from (3.54). □

Proof of Theorem 3.9. The definition of r_c implies

$$r_c = \frac{1}{n} \sum_{i \in \Lambda} l(R_n(i)) \log \frac{1}{l(R_n(i))}. \tag{3.83}$$

For $i \in \Lambda$, Theorem 3.7 and Corollary 3.8 give $l(R_n(i))$ as follows. If $n = 2$ and $\epsilon \in [0, 1/4)$, then

$$l(R_n(i)) = \begin{cases} \delta 2^{-n} - \frac{\epsilon^2}{\delta} & for \quad i = 0, 3 \\ \frac{1}{2} - \delta 2^{-n} + \frac{\epsilon^2}{\delta} & for \quad i = 1, 2. \end{cases}$$

If $n = 2$ and $\epsilon \in [1/4, 1/2)$, then

$$l(R_n(i)) = \frac{1}{2} \ for \ \ i = 1, 2.$$

If $n \geq 3$ and $\epsilon \in [0, \epsilon_n^*)$, then

$$l(R_n(i)) = \begin{cases} \delta 2^{-n} - \frac{\epsilon^2}{\delta} & for \quad i = 0,\ 2^n - 1 \\ \delta 2^{-n} + \frac{2\epsilon(1-\epsilon)}{1-4\epsilon^2} & for \quad 1 \leq i \leq 2^{n-1} - 3,\ i\ odd; and \\ & \qquad 2^{n-1} + 2 \leq i \leq 2^n - 2,\ i\ even \\ \delta 2^{-n} - \frac{2\epsilon(1-\epsilon)}{1-4\epsilon^2} & for \quad 2 \leq i \leq 2^{n-1} - 2,\ i\ even; and \\ & \qquad 2^{n-1} + 1 \leq i \leq 2^n - 3,\ i\ odd \\ \delta 2^{-n} + \frac{\epsilon^2}{\delta} + \epsilon & for \quad i = 2^{n-1} - 1,\ 2^{n-1}. \end{cases}$$

If $n \geq 3$ and $\epsilon \in [\epsilon_n^*, 1/(2^{n/2} + 2))$, then

$$l(R_n(i)) = \begin{cases} \delta 2^{-n} - \frac{\epsilon^2}{\delta} & for \quad i = 0,\ 2^n - 1 \\ \delta 2^{-n} + 2^{-n-1}\delta^2 + \epsilon + \frac{\epsilon^2}{\delta} & for \quad i = 1,\ 2^n - 2 \\ \delta 2^{1-n} & for \quad 3 \leq i \leq 2^{n-1} - 3,\ i\ odd; and \\ & \qquad 2^{n-1} + 2 \leq i \leq 2^n - 4,\ i\ even \\ \delta 2^{1-n} - 2^{-n-1}\delta^2 & for \quad i = 2^{n-1} - 1,\ 2^{n-1}. \end{cases}$$

If $n \geq 3$ and $\epsilon \in [1/(2^{n/2} + 2), 1/2)$, then

$$l(R_n(i)) = \begin{cases} \delta 2^{1-n} + 2^{-n-1}\delta^2 + \epsilon & for \quad i = 1,\ 2^n - 2 \\ \delta 2^{1-n} & for \quad 3 \leq i \leq 2^{n-1} - 3,\ i\ odd; and \\ & \qquad 2^{n-1} + 2 \leq i \leq 2^n - 4,\ i\ even \\ \delta 2^{1-n} - 2^{-n-1}\delta^2 & for \quad i = 2^{n-1} - 1,\ 2^{n-1}. \end{cases}$$

The result follows from (3.83) and routine algebra. □

3.10 Lemmas and Proofs for Section 3.5

Lemma 3.27. $z_n(i) \cdot \alpha_n(i, I_r(i)) = \beta_n(i, I_r(i))$.

Proof of Lemma 3.27. Let i and j denote the indices of two adjacent, nonempty encoding cells. Then for all $x \in \overline{R}_n(i)$, the weighted nearest neighbor condition implies that $\alpha_n(i,j)x \geq \beta_n(i,j)$. Assume, without loss of generality, that $\alpha_n(i,j) < 0$. Then $x \leq \frac{\beta_n(i,j)}{\alpha_n(i,j)}$ for all $x \in \overline{R}_n(i)$. The weighted nearest neighbor condition also implies that $\alpha_n(j,i)x \geq \beta_n(j,i)$ for all $x \in \overline{R}_n(j)$, or equivalently that $x \geq \frac{\beta_n(j,i)}{\alpha_n(j,i)}$ for all $x \in \overline{R}_n(j)$ because $\alpha_n(j,i) = -\alpha_n(i,j) > 0$. Note, however, that $\frac{\beta_n(i,j)}{\alpha_n(i,j)} = \frac{\beta_n(j,i)}{\alpha_n(j,i)}$. Hence, $\frac{\beta_n(i,j)}{\alpha_n(i,j)}$ must be the boundary between $R_n(i)$ and $R_n(j)$, for otherwise they cannot be adjacent. The lemma now follows from the definition of $z_n(i)$. □

Proof of Lemma 3.12. From (3.1), we have

$$D_{EO}^{(\pi_n)} = \sum_{i \in \Lambda} \sum_{j=0}^{2^n-1} p_n(\pi_n(j)|\pi_n(i)) \int_{R_n(i)} (x - y_n(j))^2 \, dx. \tag{3.84}$$

Substituting $y_n(j) = (j + 1/2)2^{-n}$ into (3.84), expanding the squared term, integrating and then summing over constant terms, and expressing the result in terms of $z_n(i)$ and $I_l(i)$ gives

$$\begin{aligned} D_{EO}^{(\pi_n)} &= \frac{1}{3} - 2^{-n-1} + 2^{-2n-2} - 2^{-n} \sum_{i \in \Lambda} [z_n^2(i) - z_n^2(I_l(i))] \sum_{j=0}^{2^n-1} j p_n(\pi_n(j)|\pi_n(i)) \\ &\quad + 2^{-2n} \sum_{i \in \Lambda} [z_n(i) - z_n(I_l(i))] \sum_{j=0}^{2^n-1} (j + j^2) p_n(\pi(j)|\pi(i)) \end{aligned} \tag{3.85}$$

where (3.85) follows since each $R_n(i)$ is an interval. Re-expressing the elements of (3.85) which include $I_l(i)$ in terms of $I_r(i)$, collecting terms using the definitions of $\alpha_n(i,k)$ and $\beta_n(i,k)$ in (3.4) and (3.5), respectively, and simplifying with Lemma 3.27

gives

$$
\begin{aligned}
&D_{EO}^{(\pi_n)} \\
&= \frac{1}{3} - 2^{-n-1} + 2^{-2n-2} + 2^{-n}\left[\sum_{i\in V} z_n^2(i)\cdot\alpha_n(i, I_r(i)) - \sum_{j=0}^{2^n-1} j p_n(\pi_n(j)|\pi_n(I_1))\right] \\
&\quad + 2^{-2n}\sum_{j=0}^{2^n-1}(j+j^2)p_n(\pi_n(j)|\pi_n(I_1)).
\end{aligned}
$$

□

Proof of Theorem 3.15. Let $\hat{p}_n(j|i) = p_n(\pi_n^{(CNC)}(j)|\pi_n^{(CNC)}(i))$.

Case 1: $0 \le \epsilon < \epsilon_n^*$

Theorem 3.7 and Corollary 3.8 imply that $V = \{1, 2, \ldots, 2^n - 2\}$, $I_r(i) = i + 1$, and $I_1 = 2^n - 1$. Hence, using Lemmas 3.24 and 3.26 to evaluate the last two sums in Lemma 3.12 and (3.43) to simplify the first sum in Lemma 3.12 gives

$$
\begin{aligned}
&D_{EO}^{(CNC)} \\
&= \frac{1}{3} - 2^{-n-1} + 2^{-2n-2} + (2^{-2n} - 2^{-n})\left[(2^n-1)\epsilon + \delta(2^n - 1 - \epsilon)\right] \\
&\quad + 2^{-2n}\left(\epsilon\left[\left(\frac{2^{2n}-1}{3}\right) + 2^n + 1\right] + \frac{\epsilon^2}{3}(2^{2n+1} - 21\cdot 2^n - 2) + 3\cdot 2^{n+1}\epsilon^3\right. \\
&\quad \left. + (2^n-1)\delta[2\epsilon(2^n-1) - 2\epsilon\delta] + (2^n-1)^2\delta^2\right) \\
&\quad - 2^{-n}\left(\sum_{\substack{i\in V \\ i \text{ even}}} z_n^2(i)\cdot\delta^2 + \sum_{\substack{i\in V \\ i \text{ odd}}} z_n^2(i)\cdot\delta(1+2\epsilon)\right) \\
&= \frac{2^{-2n}}{3(1+2\epsilon)}\left((1/4) + (2^{2n} + (5/2))\epsilon - (2^{2n+1} - 15\cdot 2^n + 4)\epsilon^2\right. \\
&\quad \left. + 6(2^{2n} - 2^{n+2} - 4)\epsilon^3 + (2^n-4)(2^n-2)\epsilon^4 - 12(2^n-4)\epsilon^5\right) \qquad (3.86) \\
&= D_1(n, \epsilon)
\end{aligned}
$$

where (3.86) follows from considerable arithmetic and using (from Theorem 3.7)

$$z_n(i) = \begin{cases} \left((i+1)\delta - \dfrac{2^n\epsilon^2}{\delta}\right)2^{-n} & for \quad 0 \le i \le 2^{n-1}-2,\ i\ even \\ \left((i+1)\delta + \dfrac{2^n\epsilon(2-3\epsilon)}{\delta}\right)2^{-n} & for \quad 2^{n-1} \le i \le 2^n-2,\ i\ even \\ \left((i+1)\delta + \dfrac{2^n\epsilon(2+\epsilon)}{1+2\epsilon}\right)2^{-n} & for \quad 1 \le i \le 2^{n-1}-3,\ i\ odd \\ \dfrac{1}{2} & for \quad i = 2^{n-1}-1 \\ \left((i+1)\delta + \dfrac{3\cdot 2^n\epsilon^2}{1+2\epsilon}\right)2^{-n} & for \quad 2^{n-1}+1 \le i \le 2^n-3,\ i\ odd. \end{cases}$$

Case 2: $\frac{1}{2^{n/2}+2} \le \epsilon < 1/2$

Theorem 3.7 and Corollary 3.8 imply that

$$V = \{1,3,5,\ldots,2^{n-1}-1\} \cup \{2^{n-1}, 2^{n-1}+2, 2^{n-1}+4, \ldots, 2^n-4\}$$

$$I_r(i) = i+2 \ \ for \ \ i \in \{i : i \in V, i \ne 2^{n-1}-1\}$$

$$I_r(2^{n-1}-1) = 2^{n-1}$$

$$I_1 = 2^n - 2.$$

Hence, using Lemmas 3.24 and 3.26 to evaluate the last two sums in Lemma 3.12; using (3.43) and Theorem 3.7 to simplify the first sum in Lemma 3.12; and collecting terms according to which power of ϵ they contain gives

$$\begin{aligned} D_{EO}^{(CNC)} &= \frac{1}{3} - 3\cdot 2^{-n-1} + 9\cdot 2^{-2n-2} + \left(-\frac{2}{3} + 6\cdot 2^{-n} - 34\cdot\frac{2^{-2n}}{3}\right)\epsilon \\ &\quad + \left(\frac{2}{3} - 7\cdot 2^{-n} + 52\cdot\frac{2^{-2n}}{3}\right)\epsilon^2 + (2^{-n+1} - 2^{-2n+3})\epsilon^3 \\ &\quad - 2^{-n}\left[\frac{1-4\epsilon^2}{4} + 2\delta\sum_{\substack{i\in V \\ i\ne 2^{n-1}-1}} z_n^2(i)\right]. \end{aligned} \tag{3.87}$$

Theorem 3.7 shows that $z_n(i) = 1 - z_n(2^n - 3 - i)$ for $2^{n-1} \leq i \leq 2^n - 4$ and i even when $\frac{1}{2^{n/2}+2} \leq \epsilon < 1/2$. Therefore, using Theorem 3.7 to evaluate $z_n(i)$, the last term in (3.87) can be rewritten as

$$
\begin{aligned}
&-2^{-n}\left[\frac{1-4\epsilon^2}{4} + 2\delta\left(\sum_{\substack{i=1\\ i\ odd}}^{2^{n-1}-3} z_n^2(i) + \sum_{\substack{i=2^{n-1}\\ i\ even}}^{2^n-4} z_n^2(i)\right)\right] \\
&= \left(-\frac{1}{3} + 3\cdot 2^{-n-1} - \frac{23}{3}\cdot 2^{-2n-2} + 2^{-3n}\right) \\
&\quad + \left(1 - 7\cdot 2^{-n} + 29\cdot 2^{-2n-1} - 5\cdot 2^{-3n+1}\right)\epsilon \\
&\quad + \left(-1 + 13\cdot 2^{-n} - 21\cdot 2^{-2n+1} + 5\cdot 2^{-3n+3}\right)\epsilon^2 \\
&\quad + \left(\frac{2}{3} - 3\cdot 2^{-n+2} + \frac{43}{3}\cdot 2^{-2n+2} - 5\cdot 2^{-3n+4}\right)\epsilon^3 \\
&\quad + (2^{-n+2} - 9\cdot 2^{-2n+2} + 5\cdot 2^{-3n+4})\epsilon^4 + (2^{-2n+3} - 2^{-3n+5})\epsilon^5 \qquad (3.88)
\end{aligned}
$$

where (3.88) follows after considerable arithmetic. Substituting (3.88) for the last term in (3.87) and collecting terms gives

$$
\begin{aligned}
&D_{EO}^{(CNC)} \\
&= \frac{2^{-3n}}{3}\Big(2^n + 3 + [(2^n-3)(2^{2n}+10) - 2^{n-1}]\epsilon \\
&\quad - [(2^n-6)(2^n-5)(2^n-4)\delta - 3\cdot 2^{2n}]\epsilon^2 + 12(2^n-5)(2^n-4)\epsilon^4 + 24(2^n-4)\epsilon^5\Big) \qquad (3.89) \\
&= D_3(n,\epsilon).
\end{aligned}
$$

Case 3: $\epsilon_n^* \leq \epsilon < \frac{1}{2^{n/2}+2}$

Theorem 3.7 and Corollary 3.8 imply that

$$V = \{0, 1, 3, 5, \ldots, 2^{n-1} - 1\} \cup \{2^{n-1}, 2^{n-1} + 2, 2^{n-1} + 4, \ldots, 2^n - 2\}$$

$$I_r(0) = 1$$
$$I_r(2^n - 2) = 2^n - 1$$
$$I_1 = 2^n - 1.$$

Theorem 3.7 also shows that if $i \in V - \{0, 2^n - 2\}$, then the expressions for $z_n(i)$ and $I_r(i)$ are the same as the expressions for $z_n(i)$ and $I_r(i)$ in Case 2. Hence, Lemma 3.12 gives

$$\begin{aligned} D_{EO}^{(CNC)} = D_3(n,\epsilon) &+ 2^{-n}\big[z_n^2(0)\alpha_n\big(0, I_r(0)\big) + z_n^2(2^n-2)\alpha_n\big(2^n-2, I_r(2^n-2)\big)\big] \\ &+ (2^{-2n} - 2^{-n}) \sum_{j=0}^{2^n-1} j\big[\hat{p}_n(j|2^n-1) - \hat{p}_n(j|2^n-2)\big] \\ &+ 2^{-2n} \sum_{j=0}^{2^n-1} j^2\big[\hat{p}_n(j|2^n-1) - \hat{p}_n(j|2^n-2)\big]. \end{aligned} \tag{3.90}$$

Simplifying (3.90) with (3.4) and Theorem 3.7, using (3.43) to evaluate α_n, and using Lemma 3.26 to calculate the sum over j^2 gives

$$\begin{aligned} D_{EO}^{(CNC)} = D_3(n,\epsilon) &- 2^{-n}\big[z_n^2(0)\delta^2 + (1 - z_n(0))^2\delta^2\big] + (2^{-2n} - 2^{-n})\delta^2 \\ &+ 2^{-2n}\left[2^{n+1}\epsilon + \frac{\epsilon^2}{3}(-30 \cdot 2^n + 12) + \epsilon^3(6 \cdot 2^{n+1} - 8)\right. \\ &\left. + \delta[2\epsilon(2^n - 1) - 2(2^{n+1} - 3)\epsilon\delta] + (2^{n+1} - 3)\delta^2\right]. \end{aligned} \tag{3.91}$$

Theorem 3.7 implies

$$z_n(0) = 1 - \frac{2^{-n}}{\delta^2}\big(2^n - 1 + (-2^{n+2} + 6)\epsilon + (5 \cdot 2^n - 12)\epsilon^2 + (-2^{n+1} + 8)\epsilon^3\big). \tag{3.92}$$

Substituting (3.89) and (3.92) into (3.91) and performing considerable arithmetic gives

$$D_{EO}^{(CNC)} = D_2(n, \epsilon).$$

□

Proof of Lemma 3.16. Let $N = 2^n$. The proof is straightforward for the case $N = 4$, so assume $N \geq 8$. Note that

$$\begin{aligned} g_n(N^{-2}) &= -4N^{-1} - 2N^{-2} + 8N^{-3} + 9N^{-4} - 2N^{-5} + 4N^{-7} - 16N^{-8} \\ g_n(2N^{-2}) &= 1 - 8N^{-1} - 8N^{-2} + 32N^{-3} + 40N^{-4} - 16N^{-5} + 64N^{-7} - 256N^{-8}. \end{aligned}$$

We have $g_n(N^{-2}) < 0$ since $-4N^{-1}+8N^{-3}+9N^{-4} < 0$ and $-2N^{-5}+4N^{-7} < 0$, and we have $g_n(2N^{-2}) > 0$ for $N > 8$ since $64N^{-7} - 256N^{-8} > 0$, $40N^{-4} - 16N^{-5} > 0$, and $1 - 8N^{-1} - 8N^{-2} > 0$.

Thus, the function g_n has a root in $(N^{-2}, 2N^{-2}) \subset (0, 1/2)$ for $N > 8$, and it has a root in $(0, 1/2)$ for $N = 8$ since $g_3(0) = -1 < 0$ and $g_3(1/2) = 8$. The first three derivatives of g_n are:

$$\begin{aligned} g_n'(\epsilon) &= 16(N-4)\epsilon^3 + 3N(N-2)\epsilon^2 - 4(N^2 - 4N - 4)\epsilon + N(N-4) \\ g_n''(\epsilon) &= 48(N-4)\epsilon^2 + 6N(N-2)\epsilon - 4(N^2 - 4N - 4) \\ g_n'''(\epsilon) &= 96(N-4)\epsilon + 6N(N-2). \end{aligned}$$

Since $g_n''' > 0$ on $[0, 1/2]$ and $g_n''(1/2) = -5N^2 + 38N - 16 < 0$, we must have $g_n'' < 0$ on $[0, 1/2]$, which implies $g_n' = 0$ at most once on $[0, 1/2]$. Therefore, since $g_n(0) = -1 < 0$ and $g_n(1/2) = N^2/8 > 0$, the function g_n has exactly one root on $[0, 1/2]$, which implies that $g_n(\epsilon) < 0$ on $[0, 1/2)$ if and only if $\epsilon < \hat{\epsilon}_n$. □

Note that the root $\hat{\epsilon}_n$ of g_n could be found explicitly using the formula for the

general solution to a quartic polynomial equation.

Proof of Theorem 3.17. If $n = 2$, then Theorem 3.15 implies that the value of $D_{EO}^{(CNC)}$ is the same for $0 \leq \epsilon < \epsilon_n^*$ and $\epsilon_n^* \leq \epsilon < \frac{1}{2^{n/2}+2}$, which gives

$$D_{EO}^{(CNC)} = \begin{cases} \dfrac{1+72\epsilon-48\epsilon^2}{192} & for \quad 0 \leq \epsilon < \dfrac{1}{4} \\ \dfrac{7+24\epsilon+48\epsilon^2}{192} & for \quad \dfrac{1}{4} \leq \epsilon < \dfrac{1}{2} \end{cases}.$$

Theorem 3.14 gives

$$D_{EO}^{(NBC)} = \frac{1+60\epsilon-24\epsilon^2+48\epsilon^3}{192}.$$

Therefore, for $n = 2$,

$$\begin{aligned} D_{EO}^{(CNC)} - D_{EO}^{(NBC)} &= \begin{cases} \dfrac{\epsilon(1-2\epsilon-4\epsilon^2)}{16} & for \quad 0 \leq \epsilon < \dfrac{1}{4} \\ \dfrac{(1-2\epsilon)^3}{32} & for \quad \dfrac{1}{4} \leq \epsilon < \dfrac{1}{2} \end{cases} \\ &> 0. \end{aligned}$$

Now let $n \geq 3$.

Case 1: $0 \leq \epsilon < \epsilon_n^*$

Theorem 3.14 and Theorem 3.15 imply that

$$\begin{aligned} & D_{EO}^{(NBC)} - D_{EO}^{(CNC)} \\ &= \frac{2^{-2n}\epsilon}{1+2\epsilon}\Big[-1+(2^{2n}-2^{n+2})\epsilon-(2^{2n+1}-2^{n+3}-8)\epsilon^2+(2^{2n}-2^{n+1})\epsilon^3 \\ &\quad +(2^{n+2}-16)\epsilon^4\Big] \end{aligned}$$

which (by Lemma 3.16) is positive if and only if $\epsilon > \hat{\epsilon}_n$.

Case 2: $\epsilon_n^* \leq \epsilon < \frac{1}{2^{n/2}+2}$

Theorem 3.14 and Theorem 3.15 imply that

$$\frac{D_{EO}^{(NBC)} - D_{EO}^{(CNC)}}{2^{-n}\delta^2} = (1-\epsilon)\epsilon + \frac{2\epsilon^4}{\delta^2} + 2^{-n-2}\left[\delta^3 - 2(1+6\epsilon)\right] + 2^{-2n}(1+2\epsilon)\delta^2. \tag{3.93}$$

For $n = 3$, the right hand side of (3.93) is

$$\frac{-1+26\epsilon-44\epsilon^2-8\epsilon^3}{64} + \frac{2\epsilon^4}{\delta^2} = 2\epsilon\left(\frac{\epsilon^3}{(1-2\epsilon)^2} + \frac{1-4\epsilon}{16}\right) + \frac{\phi_3(\epsilon)}{64} > 0 \tag{3.94}$$

where (3.94) follows from $\phi_3(\epsilon) \geq 0$ and $\epsilon < 1/(2+\sqrt{8}) < 1/4$. For $n \geq 4$, the right hand side of (3.93) can be lower bounded as:

$$\begin{aligned}
&\epsilon(1-\epsilon) + \frac{2\epsilon^4}{\delta^2} + 2^{-n-2}\left[\delta^3 - 2(1+6\epsilon)\right] + 2^{-2n}(1+2\epsilon)\delta^2 \\
&\geq \epsilon - \epsilon^2 + 2\epsilon^4 + 2^{-n-2}\left(-1-18\epsilon+12\epsilon^2-8\epsilon^3\right) + 2^{-2n}\left(1-2\epsilon-4\epsilon^2+8\epsilon^3\right) && (3.95)\\
&\geq \epsilon 2^{-n-1}\left[\frac{\phi_n(\epsilon)}{2\epsilon} + 2^n(1-\epsilon) - 10 - (1+2\epsilon)2^{-n+2} - 2\epsilon\right] && (3.96)\\
&\geq \epsilon 2^{-n-1}\left[2^4(1-(1/6)) - 10 - (1+2(1/6))2^{-2} - 2(1/6)\right] && (3.97)\\
&> 0
\end{aligned}$$

where (3.95) follows from $\frac{2\epsilon^4}{\delta^2} \geq 2\epsilon^4$ and simplifying; (3.96) follows by eliminating all positive terms except ϵ, and then simplifying; and (3.97) follows from the fact that $\phi_n(\epsilon) \geq 0$ when $\epsilon \geq \epsilon_n^*$ (by Lemma 3.6), and the fact that $\epsilon < \frac{1}{2^{n/2}+2} \leq \frac{1}{2^{4/2}+2} = 1/6$, for all $n \geq 4$.

Case 3: $\frac{1}{2^{n/2}+2} \leq \epsilon < 1/2$

Theorem 3.14 and Theorem 3.15 imply that

$$\begin{aligned} D_{EO}^{(NBC)} - D_{EO}^{(CNC)} &= 2^{-n}\delta^2\Big[\epsilon(1-\epsilon) - 2^{-n-2}(1+18\epsilon-28\epsilon^2+8\epsilon^3) - 2^{-2n}\delta^3\Big] \\ &> 2^{-n}\delta^2\Big[\epsilon(1-\epsilon) - 2^{-n-2}\cdot 2^{2.1} - 2^{-2n}\Big] \qquad (3.98) \\ &> 0 \qquad (3.99) \end{aligned}$$

where (3.98) follows from the fact that $\delta^3 < 1$ and $\log(1+18\epsilon-28\epsilon^2+8\epsilon^3) < 2.1$ for $0 \le \epsilon \le 1/2$; and (3.99) follows from the facts that $\epsilon(1-\epsilon)$ is monotone increasing with ϵ and $\epsilon(1-\epsilon) - 2^{-n+0.1} - 2^{-2n} > 0$ for $\epsilon = \frac{1}{2^{n/2}+2}$ and $n \ge 3$. □

This chapter, in full, has been submitted for publication as: Benjamin Farber and Kenneth Zeger, "Quantizers with Uniform Decoders and Channel Optimized Encoders," *IEEE Transactions on Information Theory*, April 14, 2004. The dissertation author was the primary investigator of this paper.

References

[1] T. R. Crimmins, H. M. Horwitz, C. J. Palermo, and R. V. Palermo, "Minimization of mean-square error for data transmitted via group codes," *IEEE Transactions on Information Theory*, vol. IT-15, pp. 72–78, January 1969.

[2] J. Dunham and R. M. Gray, "Joint source and noisy channel trellis encoding," *IEEE Transactions on Information Theory*, vol. 27, pp. 516–519, July 1981.

[3] B. Farber and K. Zeger, "Quantizers with uniform encoders and channel optimized decoders," *IEEE Transactions on Information Theory*, vol. 50, no. 1, pp. 62–77, January 2004.

[4] N. Farvardin and V. A. Vaishampayan, "Optimal quantizer design for noisy channels: An approach to combined source - channel coding," *IEEE Transactions on Information Theory*, vol. IT-33, pp. 827–838, November 1987.

[5] N. Farvardin and V. A. Vaishampayan, "On the performance and complexity of channel-optimized vector quantizers," *IEEE Transactions on Information Theory*, vol. IT-37, pp. 155–160, January 1991.

[6] A. Gersho and R. Gray, *Vector Quantization and Signal Compression*, Kluwer Academic Publishers, 1991.

[7] R. Hagen and P. Hedelin, "Robust vector quantization by a linear mapping of a block code," *IEEE Transactions on Information Theory*, vol. 45. no. 1, pp. 200–218, January 1999.

[8] Y. Yamaguchi and T. S. Huang, "Optimum binary fixed-length block codes," Quarterly Progress Report 78, M.I.T. Research Lab. of Electronics, Cambridge, MA, pp. 231–233, July 1965.

[9] T. S. Huang, "Optimum binary code," Quarterly Progress Report 82, M.I.T. Research Lab. of Electronics, Cambridge, MA, pp. 223–225, July 15, 1966.

[10] P. Knagenhjelm and E. Agrell, "The Hadamard transform-a tool for index assignment ," *IEEE Transactions on Information Theory*, vol. 42. no. 4, pp. 1139–1151, July 1996.

[11] H. Kumazawa, M. Kasahara, and T. Namekawa, "A construction of vector quantizers for noisy channels," *Electronics and Engineering in Japan*, vol. 67-B, no. 4, pp. 39–47, 1984.

[12] A. Kurtenbach and P. Wintz, "Quantizing for noisy channels," *IEEE Transactions on Communication Technology*, vol. 17, pp. 291–302, April 1969.

[13] A. Méhes and K. Zeger, "Binary lattice vector quantization with linear block codes and affine index assignments," *IEEE Transactions on Information Theory,* vol. 44, no. 1, pp. 79–94, January 1998.

[14] A. Méhes and K. Zeger, "Randomly chosen index assignments are asymptotically bad for uniform sources," *IEEE Transactions on Information Theory*, vol. 45. pp.788–794, March 1999.

[15] S. W. McLaughlin, D. L. Neuhoff, and J. J. Ashley, "Optimal binary index assignments for a class of equiprobable scalar and vector quantizers," *IEEE Transactions on Information Theory*, vol. 41, pp. 2031–2037, November 1995.

[16] M. Skoglund, "On channel-constrained vector quantization and index assignment for discrete memoryless channels," *IEEE Transactions on Information Theory*, vol. 45. no. 7, pp. 2615–2622, November 1999.

Chapter 4

Quantization of Multiple Sources Using Nonnegative Integer Bit Allocation

Abstract

Asymptotically optimal real-valued bit allocation among a set of quantizers for a finite collection of sources was derived in 1963 by Huang and Schultheiss, and an algorithm for obtaining an optimal nonnegative integer-valued bit allocation was given by Fox in 1966. We prove that, for a given bit budget, the set of optimal nonnegative integer-valued bit allocations is equal to the set of nonnegative integer-valued bit allocation vectors which minimize the Euclidean distance to the optimal real-valued bit-allocation vector of Huang and Schultheiss. We also give an algorithm for finding optimal nonnegative integer-valued bit allocations. The algorithm has lower computational complexity than Fox's algorithm, as the bit budget grows. Finally, we compare the performance of the Huang-Schultheiss solution to that of an optimal integer-valued bit allocation. Specifically, we derive upper and lower bounds on the deviation of the mean-squared error using optimal integer-valued bit allocation from the mean-squared error using optimal real-valued bit allocation. It is shown that, for asymptotically large transmission rates, optimal integer-valued bit allocations do not necessarily

achieve the same performance as that predicted by Huang-Schultheiss for optimal real-valued bit allocations.

4.1 Introduction

The classical bit allocation problem for lossy source coding is to determine the individual rates of a finite collection of scalar quantizers so as to minimize the sum of their distortions, subject to a constraint on the sum of the quantizer rates. Bit allocation arises in applications such as speech, image, and video coding. It has been shown [1,20] that finding optimal integer bit allocations is NP-hard (as the number of sources grows), via reduction to the multiple choice knapsack problem.

Huang and Schultheiss [19] analytically solved the bit allocation problem when the mean-squared error of each quantizer decreases exponentially as its rate grows. The results in [19] were generalized in [25] by finding optimal real-valued bit allocations when the mean-squared error of each quantizer is a convex function of its rate. Other generalizations were given in [16] and [23]. Bit allocation was studied in [3], in the context of trading off the total bit budget and the quantization error, a generalization of the Lagrangian approach.

The formulaic solution given in [19] allows arbitrary real-valued bit allocations. However, applications generally impose integer-value constraints on the rates used. In practice, bit allocations may be obtained by using some combinatorial optimization method such as integer linear programming or dynamic programming [10, 14, 15, 18, 29, 30, 33] or by optimizing with respect to the convex hull of the quantizers' rate-versus-distortion curves [6, 7, 24, 28, 31]. These techniques generally ignore the Huang-Schultheiss solution. Alternatively, a widely-used technique is to explicitly use an optimal real-valued bit allocation as a starting point and then home in on an integer-valued bit allocation that is close by. As noted in the textbook by Gersho and Gray [13, p. 230-

231]:

> "In practice, ... if an integer valued allocation is needed, then each non-integer allocation b_i is adjusted to the nearest integer. These modifications can lead to a violation of the allocation quota, B, so that some incremental adjustment is needed to achieve an allocation satisfying the quota. The final integer valued selection can be made heuristically. Alternatively, a local optimization of a few candidate allocations that are close to the initial solution obtained from [the Huang-Schultheiss solution] can be performed by simply computing the overall distortion for each candidate and selecting the minimum. ... Any simple heuristic procedure, however, can be used to perform this modification."

In 1966, Fox [11] gave an algorithm for finding nonnegative integer-valued bit allocations. His algorithm is greedy in that at each step it allocates one bit to the quantizer whose distortion will be reduced the most by receiving an extra bit. Fox proved this intuitive approach is optimal for any convex decreasing quantizer distortion function. There are many other algorithmic techniques in the literature for obtaining integer-valued bit allocations. Some examples of these include [1, 4, 5, 12, 21, 22, 26, 34].

In this paper we first prove that, for a given bit budget, the set of optimal nonnegative integer-valued bit allocations is equal to the set of nonnegative integer-valued bit allocation vectors which minimize the Euclidean distance to the optimal real-valued bit-allocation vector of Huang and Schultheiss. The proof of this result yields an alternate algorithm to that given by Fox for finding optimal nonnegative integer-valued bit allocations. This algorithm uses asymptotically (as the bit budget grows) less computational complexity than Fox's algorithm.

Despite the wealth of knowledge about bit allocation algorithms, there has been no published theoretical analysis comparing the performance of optimal bit allocations

with integer constraints to the performance obtained using the real-valued allocations due to Huang and Schultheiss.

We provide some such theoretical analysis. Specifically, we derive upper and lower bounds on the deviation of the mean-squared error using optimal integer-valued bit allocation from the mean-squared error using optimal real-valued bit allocation. Informally speaking, we show that no matter what bit budget is chosen, optimal integer-valued bit allocation might be as much as 6% worse than optimal real-valued bit allocation, but never more than 26% worse.

Our main results are summarized in the following (for $k \geq 2$):

(i) For any k scalar sources and any bit budget, the set of optimal nonnegative integer-valued bit allocations is the same as the set of nonnegative integer-valued bit allocation vectors (with the same bit budget) which are closest to the optimal real-valued bit-allocation vector of Huang and Schultheiss. (Theorem 4.17).

(ii) An algorithm is given for finding the set of optimal nonnegative integer-valued bit allocations from the Huang-Schultheiss optimal real-valued bit allocation (Algorithm 4.18).

(iii)
- For any k scalar sources, suppose the optimal real-valued bit allocation is nonnegative integer valued for at least *some* bit budget. Then there is a fixed number $n < k$ such that for every bit budget B, an optimal integer-valued bit allocation achieves the same performance as the optimal real-valued bit allocation if and only if $B \bmod k = n$ (Theorem 4.22).
- For any k scalar sources, suppose the optimal real-valued bit allocation is *never* integer-valued for any bit budget. Then the ratio of the mean-squared error due to optimal integer-valued bit allocation and the mean-squared error due to optimal real-valued bit allocation is bounded away from 1 over all bit budgets (Theorem 4.22).

(iv) There exist k scalar sources, such that for all bit budgets, the mean-squared error due to optimal integer-valued bit allocation is at least 6% greater than the mean-squared error due to optimal real-valued bit allocation (Theorem 4.24).

(v) For any k scalar sources and for all bit budgets, the mean-squared error due to optimal integer-valued bit allocation is at most 26% greater than the mean-squared error due to optimal real-valued bit allocation (Theorem 4.25).

Cases (i) and (ii) are first established for integer-valued bit allocations and then extended to such allocations with nonnegative components. In case (ii), the problem of finding an optimal nonnegative integer-valued bit allocation is reduced to first computing a particular real-valued bit allocation for the same bit budget, and then performing a (low complexity) nearest neighbor search in a certain lattice using the real-valued bit allocation vector as the input to the search procedure. In each of the cases (iii), (iv), and (v) we derive explicit bounds on the mean-squared error penalty paid for using integer-valued bit allocation rather than real-valued bit allocation.

This paper is organized as follows. Section 4.2 gives definitions, notation, and some lemmas. Section 4.3 shows the equivalence of closest nonnegative integer-valued bit allocation and optimal nonnegative integer-valued bit allocation. Section 4.4 characterizes, for a given set of sources, the set of bit budgets for which no penalty occurs when using integer-valued bit allocation instead of real-valued bit allocation. Also, a lower bound is given on the ratio of the mean-squared errors achieved by using optimal integer-valued bit allocation and optimal real-valued bit allocation. Section 4.5 presents an upper bound on the ratio of the mean-squared errors achieved by using optimal integer-valued bit allocation and optimal real-valued bit allocation. The Appendix contains proofs of lemmas.

4.2 Preliminaries

Let $X_1, \ldots, X_k$ be real-valued random variables (i.e. scalar sources) with variances $\sigma_1^2, \ldots, \sigma_k^2$. Throughout this paper, we assume $k \geq 2$ and $0 < \sigma_1^2, \ldots, \sigma_k^2 < \infty$. The sources $X_1, \ldots, X_k$ are scalar quantized with resolutions $b_1, \ldots, b_k$, respectively, measured in bits. The goal in bit allocation is to determine the k quantizer resolutions, subject to a constraint on their sum, so as to minimize the sum of the resulting mean-squared errors.

Let $\mathbb{R}$ denote the reals and $\mathbb{Z}$ denote the integers. We will use the following notation:

$$
\begin{aligned}
b &= (b_1, \ldots, b_k) \\
|u| &= \sum_{i=1}^{k} u_i \quad \forall u \in \mathbb{R}^k \\
g &= \left(\prod_{i=1}^{k} \sigma_i^2 \right)^{1/k} \\
\mathcal{A}_R(B) &= \{u \in \mathbb{R}^k : |u| = B\} \\
\mathcal{A}_I(B) &= \{u \in \mathbb{Z}^k : |u| = B\} \\
\mathcal{A}_I^+(B) &= \{u \in \mathbb{Z}^k : u_i \geq 0 \, \forall i, \ |u| = B\}.
\end{aligned}
$$

The vector b will be called a *bit allocation* and the integer $B \geq 1$ a *bit budget*. We say that b is a *nonnegative bit allocation* if $b_i \geq 0$ for all i. $\mathcal{A}_R(B)$, $\mathcal{A}_I(B)$, and $\mathcal{A}_I^+(B)$ are, respectively, the sets of all real-valued, integer-valued, and nonnegative integer-valued bit allocations b with bit budgets B. Bit allocations in $\mathcal{A}_I(B)$ and $\mathcal{A}_I^+(B)$ are said to be *integer bit allocations*. We use the notation $B \bmod k$ to represent the unique integer x satisfying $k \mid (B - x)$ and $0 \leq x \leq k - 1$. If the components of two vectors are the same but ordered differently, then each vector is said to be a *permutation* of the other vector.

We will assume the mean-squared error of the ith quantizer is equal to

$$d_i = h_i \sigma_i^2 4^{-b_i} \tag{4.1}$$

where h_i is a quantity dependent on the distribution of X_i, but independent of b_i. It is known that (4.1) is satisfied for asymptotically optimal scalar quantization [13], in which case

$$h_i = (1/12) \left(\int |f_{X_i/\sigma_i}|^{1/3} \right)^3$$

where f_{X_i} denotes the probability density function of X_i. Also, uniform quantizers satisfy (4.1), but with a different constant h_i. Many useful quantizers have distortions of the form in (4.1), as the distortion d_i in (4.1) often represents a reasonable approximation even for non-asymptotic bit rates.

The total mean-squared error (MSE) resulting from the bit allocation b is

$$d = \sum_{i=1}^{k} d_i.$$

We will also assume that $h_i = h$ for all i. It is straightforward to generalize our results to the case where d is a weighted combination of the d_i's and where not all the h_i's are equal.

For any k scalar sources and for each bit budget B, let

$$a_{or}(B) = \operatorname*{argmin}_{b \in \mathcal{A}_R(B)} \sum_{i=1}^{k} h \sigma_i^2 4^{-b_i}$$
$$d_{or} = \sum_{i=1}^{k} h \sigma_i^2 4^{-a_{or}(B)_i}.$$

We call $a_{or}(B)$ the *optimal real-valued bit allocation* and d_{or} the *MSE achieved by*

$a_{or}(B)$. In 1963, Huang and Schultheiss [19] derived the optimal high resolution real-valued bit allocation for the multiple source quantization problem. Their result, stated in the following lemma, shows that $a_{or}(B)$ is unique.

Lemma 4.1. *For any k scalar sources and for each bit budget B,*

$$a_{or}(B) = \frac{B}{k}(\underbrace{1,\ldots,1}_{k}) + \frac{1}{2}\left(\log_2 \frac{\sigma_1^2}{g},\ldots,\log_2 \frac{\sigma_k^2}{g}\right)$$
$$d_{or} = khg4^{-B/k}.$$

Lemma 4.1 implies that the components of the bit allocation $a_{or}(B)$ are positive for a sufficiently large bit budget B; however, $a_{or}(B)$ need not be an integer bit allocation for any particular bit budget. The next lemma follows immediately from Lemma 4.1.

Lemma 4.2. *For any k scalar sources, for each bit budget B, and for any bit allocation $b \in \mathcal{A}_R(B)$, the mean-squared error resulting from b is*

$$d = hg4^{-B/k} \cdot \sum_{i=1}^{k} 4^{(a_{or}(B)-b)_i}.$$

For any k scalar sources and for each bit budget B, let

$$d_{oi} = \min_{b\in\mathcal{A}_I(B)} \sum_{i=1}^{k} h\sigma_i^2 4^{-b_i}$$
$$\mathcal{A}_{oi}(B) = \left\{ b \in \mathcal{A}_I(B) : \sum_{i=1}^{k} h\sigma_i^2 4^{-b_i} = d_{oi} \right\}$$
$$d_{oi}^{+} = \min_{b\in\mathcal{A}_I^{+}(B)} \sum_{i=1}^{k} h\sigma_i^2 4^{-b_i}$$
$$\mathcal{A}_{oi}^{+}(B) = \left\{ b \in \mathcal{A}_I^{+}(B) : \sum_{i=1}^{k} h\sigma_i^2 4^{-b_i} = d_{oi}^{+} \right\}.$$

By Lemma 4.1, these equations are equivalent to

$$\begin{aligned} d_{oi} &= \min_{b \in \mathcal{A}_I(B)} hg 4^{-B/k} \sum_{i=1}^{k} 4^{(a_{or}(B)-b)_i} \\ \mathcal{A}_{oi}(B) &= \left\{ b \in \mathcal{A}_I(B) : hg 4^{-B/k} \sum_{i=1}^{k} 4^{(a_{or}(B)-b)_i} = d_{oi} \right\} \\ d_{oi}^{+} &= \min_{b \in \mathcal{A}_I^{+}(B)} hg 4^{-B/k} \sum_{i=1}^{k} 4^{(a_{or}(B)-b)_i} \\ \mathcal{A}_{oi}^{+}(B) &= \left\{ b \in \mathcal{A}_I^{+}(B) : hg 4^{-B/k} \sum_{i=1}^{k} 4^{(a_{or}(B)-b)_i} = d_{oi}^{+} \right\}. \end{aligned} \tag{4.2}$$

We call $\mathcal{A}_{oi}(B)$ the set of *optimal integer bit allocations* and d_{oi} the *MSE achieved by any bit allocation in* $\mathcal{A}_{oi}(B)$. The set $\mathcal{A}_{oi}^{+}(B)$ and the scalar d_{oi}^{+} are the analogous quantities for nonnegative bit allocations. In order to analyze $\mathcal{A}_{oi}^{+}(B)$ and d_{oi}^{+}, we will first obtain results about $\mathcal{A}_{oi}(B)$ and d_{oi}.

4.2.1 Lattice Tools

We next introduce some notation and terminology related to lattices that will be useful throughout the paper. We exploit certain facts from lattice theory to establish bit allocation results, specifically Theorems 4.24 and 4.25. Most of the following definitions and notation are adapted from [8].

For any $w \in \mathbb{R}^m$, denote a set $\Gamma \subset \mathbb{R}^m$ translated by the vector w by

$$\Gamma + w = \{u + w : u \in \Gamma\}.$$

For any $k \geq 1$, define[1] the following lattice:

$$\Lambda_k = \{u \in \mathbb{Z}^{k+1} : |u| = 0\}.$$

The lattice Λ_{k-1} is useful for analyzing bit allocations for k scalar sources since it consists of points with k integer coordinates which sum to zero. For $0 \leq j \leq k$, define the $(k+1)$-dimensional vector

$$c(k,j) = \frac{1}{k+1}(\underbrace{-j,\ldots,-j}_{k+1-j}, \underbrace{k+1-j,\ldots,k+1-j}_{j}). \tag{4.3}$$

Note that $|c(k,j)| = 0$ for all j and k.

Let $\|w\|$ denote the Euclidean norm of w. For any $k \geq 1$ and $w \in \mathbb{R}^{k+1}$, define

$$\Phi_k(w) = \left\{u \in \Lambda_k : \|w-u\| = \min_{v\in\Lambda_k} \|w-v\|\right\}$$

i.e., the closest lattice points in Λ_k to w.

Lemma 4.3. *For any* $w, y \in \mathbb{R}^{k+1}$,

$$\left\{u \in \Lambda_k + y : \|w-u\| = \min_{v\in\Lambda_k+y} \|w-v\|\right\} = \Phi_k(w-y) + y.$$

Denote the *Voronoi cell* associated with any point y in the lattice Λ_k by

$$V(y) = \{u \in \mathbb{R}^{k+1} : \|u-y\| \leq \|u-w\|, \ \ \forall w \in \Lambda_k\}.$$

Let

$$\mathcal{H}^k = \left\{u \in \mathbb{R}^{k+1} : |u| = 0\right\}.$$

[1]Usually denoted A_k in the literature. We use alternate notation to avoid confusion with sets of bit allocations.

The lattice Λ_k is a subset of $\mathbb{R}^{k+1}$ and also a subset of the k-dimensional hyperplane $\mathcal{H}^k$. Define the quantity

$$V_k(y) = V(y) \cap \mathcal{H}^k.$$

4.3 Closest Integer Bit Allocation

In this section, we first demonstrate the equivalence of closest integer bit allocation and optimal integer bit allocation. Then, we extend this equivalence to the case where the bit allocations must have nonnegative integer components. Finally, we obtain an algorithm for finding optimal nonnegative integer bit allocations.

For any k scalar sources and for each bit budget B, let

$$\begin{aligned}
\mathcal{A}_{ci}(B) &= \left\{ b \in \mathcal{A}_I(B) : \|b - a_{or}(B)\| = \min_{\hat{b} \in \mathcal{A}_I(B)} \|\hat{b} - a_{or}(B)\| \right\} \\
\mathcal{D}_{ci} &= \left\{ \sum_{i=1}^{k} h\sigma_i^2 4^{-b_i} : b \in \mathcal{A}_{ci}(B) \right\} \\
\Delta &= \mathcal{A}_{ci}(B) - a_{or}(B) \\
\mathcal{A}^+_{ci}(B) &= \left\{ b \in \mathcal{A}^+_I(B) : \|b - a_{or}(B)\| = \min_{\hat{b} \in \mathcal{A}^+_I(B)} \|\hat{b} - a_{or}(B)\| \right\} \\
\mathcal{D}^+_{ci} &= \left\{ \sum_{i=1}^{k} h\sigma_i^2 4^{-b_i} : b \in \mathcal{A}^+_{ci}(B) \right\}.
\end{aligned}$$

For a given bit budget B, $\mathcal{A}_{ci}(B)$ is the set of closest integer bit allocations, with respect to Euclidean distance, to the optimal real-valued bit allocation. Note that each $b \in \mathcal{A}_{ci}(B)$ is, in general, different from a bit allocation obtained by finding the closest integer to each component of $a_{or}(B)$, since such a component-wise closest bit allocation might result in using either more or less than B bits. The set Δ is a translate of $\mathcal{A}_{ci}(B)$ and is a function of $\sigma_1^2, \ldots, \sigma_k^2$ and B, although we will notationally omit these depen-

dencies. $\mathcal{A}_{ci}^{+}(B)$ and $\mathcal{D}_{ci}^{+}$ are the analogous quantities to $\mathcal{A}_{ci}(B)$ and $\mathcal{D}_{ci}$, respectively, for nonnegative bit allocations.

The following lemma will be used to prove Lemmas 4.5 and 4.23, Corollary 4.11, and Theorem 4.25. Define the quantities

$$\mu = \frac{1}{2}\left(\log_2 \frac{\sigma_1^2}{g}, \ldots, \log_2 \frac{\sigma_k^2}{g}\right) - c(k-1, B \bmod k)$$
$$M_B = \bigcup_{\sigma_1^2,\ldots,\sigma_k^2} \Delta$$

and note that $\mu \in \mathcal{H}^{k-1}$.

Lemma 4.4. *For any k scalar sources with variances $\sigma_1^2, \ldots, \sigma_k^2$ and for each bit budget B,*

$$\Delta = \Phi_{k-1}(\mu) - \mu.$$

Furthermore, $M_B = V_{k-1}(0)$ for all B.

The next lemma states that the smallest distance (in the Euclidean sense) that a closest integer bit allocation can be to the optimal real-valued bit allocation vector must occur when the bit budget is at most the number of sources.

Lemma 4.5. *For any k scalar sources,*

$$\inf_{\substack{w\in\Delta \\ B\geq 1}} \|w\| = \min_{\substack{w\in\Delta \\ 1\leq B\leq k}} \|w\|.$$

4.3.1 An Algorithm for Finding $\mathcal{A}_{ci}(B)$

The following theorem is adapted from [9, p. 230-231] and immediately yields an algorithm for finding closest integer bit allocation vectors (the components of the

resulting bit allocation vectors need not all be nonnegative). For all $u \in \mathbb{R}$, define

$$r(u) = \lfloor u + (1/2) \rfloor$$
$$\rho(u) = u - r(u).$$

The quantity $r(u)$ is a closest integer to u.

Theorem 4.6. *Let B be a bit budget, $\hat{b} = (r(a_{or}(B)_1), \ldots, r(a_{or}(B)_k))$, and $t = |\hat{b}| - B \in \mathbb{Z}$. Let $\mathcal{I}_k$ denote the set of all permutations $(i_1, \ldots, i_k)$ of $\{1, \ldots, k\}$ such that*

$$-\frac{1}{2} \leq \rho(a_{or}(B)_{i_1}) \leq \cdots \leq \rho(a_{or}(B)_{i_k}) < \frac{1}{2}$$

and let

$$\mathcal{R}^+ = \left\{ b \in \mathcal{A}_I(B) : \exists (i_1, \ldots, i_k) \in \mathcal{I}_k \text{ such that } b_j = \begin{cases} \hat{b}_j - 1 & \text{if } j \in \{i_1, \ldots, i_t\} \\ \hat{b}_j & \text{if } j \in \{i_{t+1}, \ldots, i_k\} \end{cases} \right\}$$

$$\mathcal{R}^- = \left\{ b \in \mathcal{A}_I(B) : \exists (i_1, \ldots, i_k) \in \mathcal{I}_k \text{ such that } b_j = \begin{cases} \hat{b}_j + 1 & \text{if } j \in \{i_{k+t+1}, \ldots, i_k\} \\ \hat{b}_j & \text{if } j \in \{i_1, \ldots, i_{k+t}\} \end{cases} \right\}.$$

Then

$$\mathcal{A}_{ci}(B) = \begin{cases} \{\hat{b}\} & \text{if } t = 0 \\ \mathcal{R}^+ & \text{if } t > 0 \\ \mathcal{R}^- & \text{if } t < 0. \end{cases}$$

Proof. For any $w \in \mathbb{Z}^k$,

$$\left\| a_{or}(B) - \hat{b} \right\| \leq \| a_{or}(B) - w \| .$$

Suppose $t = 0$. Then $\hat{b} \in \mathcal{A}_I(B) \subset \mathbb{Z}^k$. Thus, $\hat{b}$ is a point in $\mathcal{A}_I(B)$ of minimum distance to $a_{or}(B)$. This means that $\hat{b} \in \mathcal{A}_{ci}(B)$. Since $r(u)$ is a closest integer to u and since r breaks ties by rounding upward, any other integer bit allocation b with minimum distance from $a_{or}(B)$ must satisfy $|b| < |a_{or}(B)|$. Thus, $b \notin \mathcal{A}_I(B)$ and hence $\mathcal{A}_{ci}(B) = \{\hat{b}\}$.

Suppose $t \neq 0$ and let

$$\mathcal{R} = \begin{cases} \mathcal{R}^+ & \text{if} \quad t > 0 \\ \mathcal{R}^- & \text{if} \quad t < 0. \end{cases}$$

It can be seen that every element of $\mathcal{R}$ is a bit allocation $b \in \mathcal{A}_I(B)$ which minimizes the difference between $\|a_{or}(B) - b\|$ and $\|a_{or}(B) - \hat{b}\|$. Since $\|a_{or}(B) - \hat{b}\|$ does not depend on such b, we have

$$\mathcal{R} \subset \{b \in \mathcal{A}_I(B) : \|a_{or}(B) - b\| \leq \|a_{or}(B) - b'\| \ \ \forall b' \in \mathcal{A}_I(B)\} = \mathcal{A}_{ci}(B).$$

To finish the proof, we will show that $\mathcal{A}_{ci}(B) \subset \mathcal{R}$. Let $b \in \mathcal{A}_{ci}(B)$. For any i and j, the following identity holds:

$$\begin{aligned} &[(b_i - 1) - a_{or}(B)_i]^2 + [(b_j + 1) - a_{or}(B)_j]^2 - [b_i - a_{or}(B)_i]^2 - [b_j - a_{or}(B)_j]^2 \\ &= 2[1 + a_{or}(B)_i - b_i + b_j - a_{or}(B)_j]. \end{aligned} \tag{4.4}$$

Suppose there exists an i such that $b_i - a_{or}(B)_i \geq 1$. Then there must exist a j such that $b_j - a_{or}(B)_j < 0$, since $\sum_l b_l = \sum_l a_{or}(B)_l = B$. But then the right-hand

side of (4.4) would be negative which would imply $b \notin \mathcal{A}_{ci}(B)$, since subtracting 1 from b_i and adding 1 to b_j would result in an integer bit allocation closer than b to $a_{or}(B)$. A similar contradiction results in the case where $b_i - a_{or}(B)_i \leq -1$. Thus, for every i, we must have $b_i \in \{\lfloor a_{or}(B)_i \rfloor, \lceil a_{or}(B)_i \rceil\}$. Since $\hat{b}_i \in \{\lfloor a_{or}(B)_i \rfloor, \lceil a_{or}(B)_i \rceil\}$ we conclude that $|\hat{b}_i - b_i| \leq 1$ for all i.

Now, suppose $t > 0$. Then there exists at least one i such that $b_i = \hat{b}_i - 1 = \lfloor a_{or}(B)_i \rfloor < a_{or}(B)_i$. For each j, it cannot be the case that $b_j = \hat{b}_j + 1 = \lceil a_{or}(B)_j \rceil > a_{or}(B)_j$, for otherwise the right-hand side of (4.4) would be positive, which would imply that the Euclidean distance between b and $a_{or}(B)$ can be reduced by adding 1 to $\hat{b}_i$ and subtracting 1 from $\hat{b}_j$, which violates the fact that $b \in \mathcal{A}_{ci}(B)$. Thus, for all i, we have $\hat{b}_i - b_i \in \{0, 1\}$. To minimize the distance between b and $a_{or}(B)$, the t components of b for which $\hat{b}_i - b_i = 1$ must be those components with the smallest values of $\rho(a_{or}(B)_i)$. Thus $b \in \mathcal{R}^+$.

A similar argument shows that if $t < 0$, then for all i, we have $\hat{b}_i - b_i \in \{0, -1\}$; this then implies that the t components of b for which $\hat{b}_i - b_i = -1$ must be those components with the largest values of $\rho(a_{or}(B)_i)$, i.e., $b \in \mathcal{R}^-$. In summary, $b \in \mathcal{R}$. $\square$

Note that in practice $\mathcal{A}_{ci}(B)$ will usually consist of a single bit allocation, although in principle it can contain more than one bit allocation.

We note that Guo and Meng [17] gave a similar algorithm to that implied by Theorem 4.6. Instead of rounding each component of the Huang-Schultheiss solution $a_{or}(B)$ to the nearest integer, they round each component down to the nearest integer from below. Then, they added 1 bit at a time to the rounded components, based on which components were rounded down the most. The technique implied from our Theorem 4.6 uses the same idea, but also adds bits to components which were rounded up too far. The authors of [17] did not claim that their resulting bit allocation gave a closest integer bit allocation. They did, however, assert that their resulting bit allocation was optimal; but,

in fact, their proof was not valid. They attempted to show that adding bits, one at a time, in the manner they described was optimal among all ways to add bits to the rounded bit allocation. However, their proof did not eliminate the possibility of adding more than two bits to multiple components of the rounded bit allocation. Nor did they rule out the possibility of subtracting extra bits from some components in order to add even more bits to other components. We believe their algorithm is indeed correct, despite the lack of proof.

Wintz and Kurtenbach [32, p. 656] also gave a similar algorithm for obtaining integer-valued bit allocations. Their technique was to round the components of the Huang-Schultheiss solution to the nearest integer, and then add or subtract bits to certain components until the bit budget was satisfied. However, their choice of which components to adjust up or down was based on the magnitudes of the components, rather than how much they were initially truncated. The authors of [32] note that their technique is suboptimal.

The algorithm in [17] assumes the Huang-Schultheiss solution has nonnegative components, as does the algorithm implied by our Theorem 4.6. However, in Section 4.3.3, we generalize the result of Theorem 4.6 to give an algorithm for finding optimal nonnegative integer bit allocations without any such assumptions about the Huang-Schultheiss solution.

4.3.2 Equivalence of Closest Integer Bit Allocations and Optimal Integer Bit Allocations

In this subsection, we allow bit allocations to have negative components. In Section 4.3.3 we will add the nonnegativity constraint. The next two technical lemmas are used to prove Lemma 4.9.

Lemma 4.7. *For any k scalar sources and for each bit budget B, let $\beta \in \Delta$ be such that*

$\beta_j \in (-1/2, 1/2]$ *for some* j.

If $\beta_i < -1/2$*, then*

$$\begin{aligned} \beta_i &= -\rho(a_{or}(B)_i) - 1 \\ \beta_j &= -\rho(a_{or}(B)_j) \\ \rho(a_{or}(B)_i) &\leq \rho(a_{or}(B)_j). \end{aligned}$$

If $\beta_i > 1/2$*, then*

$$\begin{aligned} \beta_i &= -\rho(a_{or}(B)_i) + 1 \\ \beta_j &= -\rho(a_{or}(B)_j) \\ \rho(a_{or}(B)_i) &\geq \rho(a_{or}(B)_j). \end{aligned}$$

Lemma 4.8. *For any* k *scalar sources and for each bit budget* B*, let*

$$t = r(a_{or}(B)_1) + \cdots + r(a_{or}(B)_k) - B.$$

Then for any $\beta \in \Delta$ *and for all* i*,*

$$\beta_i \in \begin{cases} (-1/2, 1/2] & \text{if } t = 0 \\ (-1, 1/2] & \text{if } t > 0 \\ (-1/2, 1) & \text{if } t < 0\ . \end{cases}$$

For each i and j, define a k-dimensional vector $\omega(i, j)$ whose components are

$$\omega(i,j)_l = \begin{cases} 1 & \text{if } l = i \\ -1 & \text{if } l = j \\ 0 & \text{otherwise}\ . \end{cases} \tag{4.5}$$

Lemma 4.9. *For any k scalar sources, for each bit budget B, and for any $b \in \mathcal{A}_{ci}(B)$, let $\beta = b - a_{or}(B)$. Then for all i, j,*

$$\beta_j - \beta_i \leq 1.$$

If $\beta_j - \beta_i = 1$, then

$$b + \omega(i, j) \in \mathcal{A}_{ci}(B). \tag{4.6}$$

The following theorem establishes that for each bit budget, the closest integer bit allocations and the optimal integer bit allocations are the same collections.

Theorem 4.10. *For any k scalar sources and for each bit budget B,*

$$\mathcal{A}_{ci}(B) = \mathcal{A}_{oi}(B)$$
$$\mathcal{D}_{ci} = \{d_{oi}\}.$$

Proof. First, we show that $\mathcal{A}_{ci}(B) \subset \mathcal{A}_{oi}(B)$. Let $b \in \mathcal{A}_I(B)$ and $\tilde{b} \in \mathcal{A}_{ci}(B)$, and let d and $\tilde{d}$ denote the resulting MSEs, respectively. It suffices to show that $d \geq \tilde{d}$.

Define

$$\eta^+ = \{l : b_l - \tilde{b}_l > 0\}$$
$$\eta^- = \{l : b_l - \tilde{b}_l < 0\}$$

and consider any sequence of integer bit allocation vectors

$$\tilde{b} = b^{(0)}, \ldots, b^{(n)} = b \tag{4.7}$$

such that for each $m = 0, \ldots, n-1$ there exists an $i \in \eta^+$ and a $j \in \eta^-$ such that

$$b_l^{(m+1)} - b_l^{(m)} = \omega(i, j). \tag{4.8}$$

Such a sequence is guaranteed to exist since $|\tilde{b}| = |b|$. For each m, let $d^{(m)}$ be the MSE achieved by $b^{(m)}$. To establish $d \geq \tilde{d}$, we will show that $d^{(m)}$ is monotonic nondecreasing in m.

The construction of the sequence $b^{(0)}, \ldots, b^{(n)}$ implies that for each $m = 0, \ldots, n-1$

$$(b^{(m)} - b^{(0)})_i \geq 0$$
$$(b^{(m)} - b^{(0)})_j \leq 0$$

where $i \in \eta^+$ and $j \in \eta^-$ are defined by (4.8), and are functions of m. Thus,

$$(b^{(m)} - b^{(0)})_j \leq (b^{(m)} - b^{(0)})_i.$$

Let

$$\beta^{(m)} = b^{(m)} - a_{or}(B).$$

Then,

$$\beta_j^{(0)} - \beta_i^{(0)} \leq 1 \qquad \text{[from Lemma 4.9]}$$

and therefore for each $m = 0, \ldots, n-1$, we get

$$\beta_j^{(0)} - \beta_i^{(0)} - (b^{(m)} - b^{(0)})_i \leq 1 - (b^{(m)} - b^{(0)})_j$$

or equivalently (by the definition of $\beta^{(0)}$)

$$-(b^{(0)} - a_{or}(B))_i - (b^{(m)} - b^{(0)})_i - 1 \leq -(b^{(0)} - a_{or}(B))_j - (b^{(m)} - b^{(0)})_j. \quad (4.9)$$

Canceling terms in (4.9) and raising 4 to the remaining quantity on each side of the inequality gives

$$4^{-\beta_i^{(m)}-1} \leq 4^{-\beta_j^{(m)}} \quad (4.10)$$

or equivalently

$$\begin{aligned} 4^{-\beta_i^{(m)}} + 4^{-\beta_j^{(m)}} &\leq 4^{-(\beta_i^{(m)}+1)} + 4^{-(\beta_j^{(m)}-1)} \\ &= 4^{-\beta_i^{(m+1)}} + 4^{-\beta_j^{(m+1)}} \qquad [\text{from (4.8)}] \end{aligned}$$

which implies

$$d^{(m)} = hg4^{-B/k} \cdot \sum_{l=1}^{k} 4^{-\beta_l^{(m)}} \leq hg4^{-B/k} \cdot \sum_{l=1}^{k} 4^{-\beta_l^{(m+1)}} = d^{(m+1)} \quad [\text{from Lemma 4.2}]. \quad (4.11)$$

Thus $d^{(m)}$ is monotonic and therefore we have shown $\mathcal{A}_{ci}(B) \subset \mathcal{A}_{oi}(B)$. The fact that $\mathcal{D}_{ci} = \{d_{oi}\}$ then immediately follows.

Next, we show that $\mathcal{A}_{oi}(B) \subset \mathcal{A}_{ci}(B)$. Let $b \in \mathcal{A}_{oi}(B)$. Since $\mathcal{A}_{oi}(B) \subset \mathcal{A}_I(B)$, a decomposition as in (4.7) still holds. Our goal is to show $b \in \mathcal{A}_{ci}(B)$, which will be accomplished by showing $b^{(n)} \in \mathcal{A}_{ci}(B)$. By the optimality of b, we must have $d \leq \tilde{d}$, which by the monotonicity of $d^{(m)}$ implies $d^{(n)} = d^{(0)}$. Hence, equality holds in (4.11) and therefore also in (4.10), which implies for each $m = 0, \ldots, n-1$ that

$$\beta_j^{(m)} - \beta_i^{(m)} = 1. \quad (4.12)$$

Now, we use induction to show $b^{(n)} \in \mathcal{A}_{ci}(B)$. The $m = 0$ case holds since $b^{(0)} = \tilde{b} \in \mathcal{A}_{ci}(B)$. Now suppose for all $m \leq l$ (where $l \geq 1$) that $b^{(m)} \in \mathcal{A}_{ci}(B)$. Then we can apply Lemma 4.9 to (4.12) in the case $m = l$, and use (4.8) to obtain $b^{(l+1)} \in \mathcal{A}_{ci}(B)$.

□

Corollary 4.11. *The components of every element in $\mathcal{A}_{oi}(B)$ tend to infinity as the bit budget grows without bound.*

Proof of Lemma 4.11. By Lemma 4.1, the components of $a_{or}(B)$ grow without bound as $B \to \infty$. By Lemma 4.4, Δ is a subset of $V_{k-1}(0)$, which is known to be a bounded convex polytope [8, p. 461–462]. Thus, as $B \to \infty$, for every $b \in \mathcal{A}_{ci}(B)$, the components of b also must grow without bound. The result follows from Theorem 4.10.
□

4.3.3 Equivalence of Closest Nonnegative Integer Bit Allocations and Optimal Nonnegative Integer Bit Allocations

The problem of finding nonnegative bit allocations was addressed by Segall [25], but his solution did not assure integer-valued quantizer resolutions. Fox [11] gave a greedy algorithm for finding nonnegative integer bit allocations by allocating one bit at a time to a set of quantizers. His algorithm is optimal for any convex decreasing distortion function, and in particular, it is optimal for the distortion function we assume in (4.1).

In this section, we prove (in Theorem 4.17) that optimal nonnegative integer bit allocations are equivalent to closest nonnegative integer bit allocations. Our proof leads to an alternate algorithm for finding optimal nonnegative integer bit allocations. The algorithm is faster than Fox's algorithm (as the bit budget grows).

First we introduce some useful notation and then establish five lemmas that will be used to prove Theorem 4.17.

For any bit budget B and any nonempty set $S \subset \{1, 2, \ldots, k\}$, define a vector $a_{or}(B, S) \in \mathbb{R}^k$ whose components are

$$a_{or}(B,S)_i = \begin{cases} \frac{B}{|S|} + \frac{1}{2}\log_2 \frac{\sigma_i^2}{g(S)} & \text{if } i \in S \\ 0 & \text{otherwise} \end{cases}$$

where

$$g(S) = \left(\prod_{i \in S} \sigma_i^2\right)^{1/|S|}.$$

Lemma 4.1 shows that the $|S|$-dimensional vector obtained by extracting the coordinates of $a_{or}(B, S)$, corresponding to the elements of S, is the optimal real-valued bit allocation for the quantizers corresponding to the elements of S. For any given nonempty set $S \subset \{1, \ldots, k\}$, let

$$\theta_1(b) = \|b - a_{or}(B,S)\|$$
$$\theta_2(b) = hg4^{-B/k} \sum_{i=1}^{k} 4^{(a_{or}(B,S)-b)_i}$$

and for any set $T \subset \mathbb{Z}^k$ and any function $f : T \to \mathbb{R}$, let

$$Q(T, f) = \left\{ b \in T : f(b) = \min_{\hat{b} \in T} f(\hat{b}) \right\}.$$

Define the quantities

$$\mathbb{Z}_S^k = \{u \in \mathbb{Z}^k : u_j = 0 \ \ \forall j \notin S\}$$
$$\mathcal{A}_I(B,S) = \{u \in \mathbb{Z}_S^k : |u| = B\}$$
$$\mathcal{A}_I^+(B,S) = \{u \in \mathcal{A}_I(B,S) : u_i \geq 0 \ \forall i\}$$
$$\mathcal{A}_{ci}(B,S) = Q(\mathcal{A}_I(B,S), \theta_1)$$

$$\mathcal{A}_{ci}^{+}(B,S) = Q(\mathcal{A}_{I}^{+}(B,S),\theta_1)$$
$$\mathcal{A}_{oi}^{+}(B,S) = Q(\mathcal{A}_{I}^{+}(B,S),\theta_2).$$

For a given bit budget B, $\mathcal{A}_{ci}(B,S)$ is the set of closest integer bit allocations to $a_{or}(B,S)$, and $\mathcal{A}_{ci}^{+}(B,S)$ is the set of closest nonnegative integer bit allocations to $a_{or}(B,S)$.

Lemma 4.12. *If $Q(W,f) \subset V \subset W$, then $Q(W,f) = Q(V,f)$.*

The following lemma shows that to find a closest integer bit allocation to $a_{or}(B,S)$, one can assume without loss of generality that zeros are located in bit allocation vector components corresponding to integers not in S.

Lemma 4.13. *For each bit budget B and for any nonempty set $S \subset \{1,2,\ldots,k\}$,*

$$\mathcal{A}_{ci}(B,S) = Q(\mathcal{A}_{I}(B),\theta_1).$$

Lemma 4.14. *For any k scalar sources, for each bit budget B, and for any nonempty set $S \subset \{1,2,\ldots,k\}$, if $a_{or}(B,S)$ is nonnegative, then every bit allocation in $\mathcal{A}_{ci}(B,S)$ is nonnegative.*

Lemma 4.15. *Consider k scalar sources with bit budget B and a nonempty set $S \subset \{1,2,\ldots,k\}$. If $\mathcal{A}_{ci}^{+}(B) \subset \mathcal{A}_{I}(B,S)$, then $\mathcal{A}_{ci}^{+}(B) = \mathcal{A}_{ci}^{+}(B,S)$. If $\mathcal{A}_{oi}^{+}(B) \subset \mathcal{A}_{I}(B,S)$, then $\mathcal{A}_{oi}^{+}(B) = \mathcal{A}_{oi}^{+}(B,S)$.*

Lemma 4.16. *Consider k scalar sources with bit budget B. Suppose $\mathcal{A}_{ci}^{+}(B), \mathcal{A}_{oi}^{+}(B) \subset \mathcal{A}_{I}(B,S)$ and there exists an $i \in S \subset \{1,2,\ldots,k\}$ such that $a_{or}(B,S)_i < 0$. Then $b_i = 0$ for all $b \in \mathcal{A}_{ci}^{+}(B) \cup \mathcal{A}_{oi}^{+}(B)$.*

The following theorem shows that optimal nonnegative integer bit allocation is equivalent to closest nonnegative integer bit allocation. In other words, minimizing the

distortion among all nonnegative integer bit allocations is equivalent to finding which nonnegative integer bit allocation vectors are closest in Euclidean distance to the Huang-Schultheiss real-valued bit-allocation vector. This, in turn, can be accomplished with a nearest neighbor search in a lattice. Following Theorem 4.17, we give an efficient algorithm for finding optimal nonnegative integer bit allocation vectors.

Theorem 4.17. *For any k scalar sources and for each bit budget B,*

$$\mathcal{A}_{ci}^{+}(B) = \mathcal{A}_{oi}^{+}(B)$$
$$\mathcal{D}_{ci}^{+} = \{d_{oi}^{+}\}.$$

Proof. Let $S^{(0)} = \{1, \ldots, k\}$ and consider the sequence of bit allocations

$$a_{or}(B, S^{(0)}), \ldots, a_{or}(B, S^{(n)})$$

where

$$S^{(m+1)} = \{i \in S^{(m)} : a_{or}(B, S^{(m)})_i \geq 0\}$$

and n is the smallest nonnegative integer such that $a_{or}(B, S^{(n)})_i \geq 0$ for all i. Such an integer n exists since the following hold:

- $|S^{(m)}| \geq 1$ for all m.
- If $|S^{(m)}| = 1$, then $a_{or}(B, S^{(m)})_i \geq 0$ for all i.
- $|S^{(m)}|$ is monotone decreasing in m.

We will show that both $\mathcal{A}_{ci}^{+}(B)$ and $\mathcal{A}_{oi}^{+}(B)$ are equal to $\mathcal{A}_{ci}(B, S^{(n)})$. The fact that $\mathcal{D}_{ci}^{+} = \{d_{oi}^{+}\}$ then follows from the definition of $\mathcal{D}_{ci}^{+}$.

Note that for any $m \geq 0$, if $\mathcal{A}_{oi}^{+}(B), \mathcal{A}_{ci}^{+}(B) \subset \mathcal{A}_I(B, S^{(m)})$, then (by Lemma 4.16) any optimal or closest nonnegative integer bit allocation b must sat-

isfy $b_i = 0$ for $i \notin S^{(m+1)}$, and therefore $\mathcal{A}_{oi}^{+}(B), \mathcal{A}_{ci}^{+}(B) \subset \mathcal{A}_I(B, S^{(m+1)})$. Thus, since $\mathcal{A}_{oi}^{+}(B), \mathcal{A}_{ci}^{+}(B) \subset \mathcal{A}_I(B) = \mathcal{A}_I(B, S^{(0)})$ we obtain by induction that

$$\mathcal{A}_{oi}^{+}(B), \mathcal{A}_{ci}^{+}(B) \subset \mathcal{A}_I(B, S^{(n)}). \tag{4.13}$$

Now using (4.13) and Lemma 4.15 we have

$$\mathcal{A}_{ci}^{+}(B) = \mathcal{A}_{ci}^{+}(B, S^{(n)}) \tag{4.14}$$

$$\mathcal{A}_{oi}^{+}(B) = \mathcal{A}_{oi}^{+}(B, S^{(n)}). \tag{4.15}$$

Since $a_{or}(B, S^{(n)})$ is nonnegative by definition, Lemma 4.14 implies that each $b \in \mathcal{A}_{ci}(B, S^{(n)})$ is nonnegative, i.e.

$$\mathcal{A}_{ci}(B, S^{(n)}) \subset \mathcal{A}_I^{+}(B, S^{(n)}). \tag{4.16}$$

From (4.16) and the fact that $\mathcal{A}_I^{+}(B, S^{(n)}) \subset \mathcal{A}_I(B, S^{(n)})$, we can apply Lemma 4.12 with $W = \mathcal{A}_I(B, S^{(n)})$, $V = \mathcal{A}_I^{+}(B, S^{(n)})$, and $f = \theta_1$ to obtain $\mathcal{A}_{ci}(B, S^{(n)}) = \mathcal{A}_{ci}^{+}(B, S^{(n)})$. Thus, we have

$$\mathcal{A}_{ci}^{+}(B) = \mathcal{A}_{ci}(B, S^{(n)}) \qquad [\text{from (4.14)}].$$

Now consider a set of sources $\hat{X}_1, \ldots, \hat{X}_k$ with variances

$$\hat{\sigma}_i^2 = \begin{cases} \frac{\sigma_i^2}{g(S^{(n)})} 4^{\frac{B(k-|S^{(n)}|)}{k|S^{(n)}|}} & \text{if } i \in S^{(n)} \\ 4^{-B/k} & \text{if } i \notin S^{(n)}. \end{cases}$$

Lemma 4.1 shows that $a_{or}(B, S^{(n)})$ is the optimal real-valued bit allocation for $\hat{X}_1, \ldots, \hat{X}_k$ (mimicking the argument from the proof of Lemma 4.14). Therefore, by Lemma 4.13, $\mathcal{A}_{ci}(B, S^{(n)})$ is the set of closest integer bit allocations (without requiring

any zero components) for $\hat{X}_1, \ldots, \hat{X}_k$. Hence, by Theorem 4.10, $\mathcal{A}_{ci}(B, S^{(n)})$ is also the set of optimal integer bit allocations for $\hat{X}_1, \ldots, \hat{X}_k$. Thus,

$$\begin{aligned} \mathcal{A}_{ci}(B, S^{(n)}) &= Q(\mathcal{A}_I(B), \theta_2) && \text{[from (4.2)]} \\ &\subset \mathcal{A}_I^+(B, S^{(n)}) && \text{[from (4.16),(4.17)]} \\ &\subset \mathcal{A}_I(B). \end{aligned} \tag{4.17}$$

Now applying Lemma 4.12 with $W = \mathcal{A}_I(B)$, $V = \mathcal{A}_I^+(B, S^{(n)})$, and $f = \theta_2$ gives

$$Q(\mathcal{A}_I(B), \theta_2) = \mathcal{A}_{oi}^+(B, S^{(n)}).$$

Therefore, we have

$$\mathcal{A}_{oi}^+(B) = \mathcal{A}_{ci}(B, S^{(n)}) \qquad \text{[from (4.15),(4.17)]}.$$

□

The proof of Theorem 4.17 yields an alternative procedure to that given by Fox [11] for finding optimal nonnegative integer bit allocations. The main idea is to remove any negative components in the Huang-Schultheiss real-valued solution and then re-compute the Huang-Schultheiss solution for the surviving quantizers, iteratively repeating this procedure until no negative components remain. Then, the set of closest integer-valued vectors (with the same bit budget) to the resulting nonnegative real-valued vector is computed as the output of the algorithm.

Algorithm 4.18. ***(Procedure to find $\mathcal{A}_{oi}^+(B)$ and $\mathcal{A}_{ci}^+(B)$):***
For any k scalar sources and for each bit budget B, the following procedure generates a set of bit allocations which is both the set $\mathcal{A}_{oi}^+(B)$ and the set $\mathcal{A}_{ci}^+(B)$.

- *Step 1: Set $S = \{1, 2, \ldots, k\}$.*

- *Step 2: Compute $a_{or}(B,S)$ and let $J = \{i \in S : a_{or}(B,S)_i \geq 0\}$.*
- *Step 3: If $J = S$ go to Step 4. Otherwise, set $S = J$ and go to Step 2.*
- *Step 4: Set $a_{or}(B)$ equal to $a_{or}(B,S)$ in Theorem 4.6 and then compute $\mathcal{A}_{ci}(B)$. Set $\mathcal{A}^+_{oi}(B) = \mathcal{A}^+_{ci}(B) = \mathcal{A}_{ci}(B)$.*

Remark:

We briefly remark on the computational complexity of the algorithm above as a function of the bit budget B, for a fixed k. When there exists a unique closest nonnegative integer bit allocation, the computational complexity of the algorithm reduces to the complexity of determining $\mathcal{A}_{ci}(B)$. The complexity of this lattice search is known to be constant in B (e.g. see [9, p. 231]). In contrast, Fox's algorithm has complexity linear in B. Thus for large B, Algorithm 4.18 is faster than Fox's algorithm.

4.4 Distortion Penalty for Integer Bit Allocations

In this section and in Section 4.5 all sources will be assumed to have optimal real-valued solutions with nonnegative components. In particular, the following definition is only meaningful when $a_{or}(B)_i \geq 0$ for all i.

For any k scalar sources and for each bit budget B, let

$$p^{oi} = \frac{d_{oi}}{d_{or}}.$$

We call p^{oi} the *distortion penalty* resulting from optimal integer bit allocation. For any $b \in \mathcal{A}_{oi}(B)$, we have

$$p^{oi} = \frac{1}{k}\sum_{i=1}^{k} 4^{(a_{or}(B)-b)_i} \qquad \text{[from (4.2), Lemma 4.1]}. \tag{4.18}$$

Also, clearly $p^{oi} \geq 1$.

Theorem 4.19. *For any k scalar sources with variances $\sigma_1^2, \ldots, \sigma_k^2$ and a bit budget B, the following three statements are equivalent:*

(i) $p^{oi} = 1$.

(ii) The optimal real-valued bit allocation is an integer bit allocation.

(iii) $\frac{1}{2}\log_2 \frac{\sigma_i^2}{g} + \frac{B \bmod k}{k} \in \mathbb{Z} \quad \forall i$.

Proof. If $a_{or}(B)$ is an integer bit allocation, then $a_{or}(B) \in \mathcal{A}_{oi}(B)$, so $p^{oi} = 1$, i.e. $(ii) \Longrightarrow (i)$. Conversely, suppose $p^{oi} = 1$. Then, for any $b \in \mathcal{A}_{oi}(B)$, by (4.18) and the arithmetic-geometric mean inequality, we have

$$1 = \frac{1}{k}\sum_{i=1}^{k} 4^{(a_{or}(B)-b)_i} \geq 4^{\sum_{i=1}^{k}\frac{1}{k}(a_{or}(B)-b)_i} = 4^{\frac{1}{k}(B-B)} = 1 \tag{4.19}$$

so the inequality in (4.19) is, in fact, equality. Thus, $a_{or}(B)_i - b_i$ is a constant for all i, which must equal zero since $|a_{or}(B)| = |b|$. This proves $(i) \Longrightarrow (ii)$. Lemma 4.1 and the equivalence of (i) and (ii) imply: $p^{oi} = 1$ if and only if for all i the quantity $\frac{1}{2}\log_2 \frac{\sigma_i^2}{g} + \frac{B}{k}$ is an integer. Then $(i) \Longleftrightarrow (iii)$ follows from

$$\frac{1}{2}\log_2 \frac{\sigma_i^2}{g} + \frac{B}{k} = \left(\frac{1}{2}\log_2 \frac{\sigma_i^2}{g} + \frac{B \bmod k}{k}\right) + \left(\frac{B - B \bmod k}{k}\right).$$

□

Lemma 4.20. *For any $w \in \mathcal{A}_R(0)$,*

$$4^{-\|w\|\sqrt{(k-1)/k}} + (k-1)4^{\|w\|\sqrt{1/(k(k-1))}} \leq \sum_{i=1}^{k} 4^{-w_i} \leq 4^{\|w\|\sqrt{(k-1)/k}} + (k-1)4^{-\|w\|\sqrt{1/(k(k-1))}}.$$

For any $b \in \mathcal{A}_R(B)$, if $w = b - a_{or}(B)$, then Lemma 4.20 gives bounds on the sum in Lemma 4.2. Moreover, both the upper and lower bounds in Lemma 4.20 are functions only of k and $\|w\|$, both bounds are monotone increasing with $\|w\|$, and as $\|w\| \to 0$ the bounds become tight.

Lemma 4.21. *For any k scalar sources, for each bit budget B, and for any bit allocation $b \in \mathcal{A}_I(B)$, the mean-squared error d resulting from b satisfies*

$$hg4^{-B/k} \cdot \left(4^{-\|b-a_{or}(B)\|\sqrt{(k-1)/k}} + (k-1)4^{\|b-a_{or}(B)\|\sqrt{1/(k(k-1))}}\right)$$
$$\leq d \leq$$
$$hg4^{-B/k} \cdot \left(4^{\|b-a_{or}(B)\|\sqrt{(k-1)/k}} + (k-1)4^{-\|b-a_{or}(B)\|\sqrt{1/(k(k-1))}}\right).$$

For any k scalar sources, define

$$\delta = \min_{B \geq 1} \min_{b \in \mathcal{A}_{oi}(B)} \|b - a_{or}(B)\|.$$

δ is the minimum distance, for k fixed sources, between an optimal integer bit allocation and the optimal real-valued bit allocation vector, over all bit budgets. The quantity δ is well defined by Lemma 4.5 and Theorem 4.10.

The following theorem shows that either $a_{or}(B) \in \mathcal{A}_{oi}(B)$ for all bit budgets B congruent to some constant modulo k, or else $a_{or}(B)$ is never an element of $\mathcal{A}_{oi}(B)$, in which case the distortion penalty resulting from optimal integer bit allocation is bounded away from 1 for all bit budgets.

Theorem 4.22. *Consider k scalar sources.*

(i) If $\delta = 0$, then there exists a nonnegative integer $n \leq k-1$ such that for each bit budget B, the following holds:

$$p^{oi} = 1 \quad \textit{if and only if} \quad B \bmod k = n.$$

(ii) If $\delta > 0$, then for each bit budget B, the following holds:

$$p^{oi} \geq \frac{1}{k}\left(4^{-\delta\sqrt{(k-1)/k}} + (k-1)4^{\delta\sqrt{1/(k(k-1))}}\right) > 1.$$

Proof. Suppose $\delta = 0$. Then there exists a bit budget $\hat{B}$ such that $a_{or}(\hat{B}) \in \mathcal{A}_{oi}(\hat{B})$. Theorem 4.19 implies, for each bit budget B, that $p^{oi} = 1$ if and only if $a_{or}(B) \in \mathcal{A}_I(B)$. Let $n = \hat{B} \bmod k$. We show that $a_{or}(B) \in \mathcal{A}_I(B)$ if and only if $B \bmod k = n$. Suppose $a_{or}(B) \in \mathcal{A}_I(B)$. Then, for each i, the quantities $a_{or}(B)_i$ and $a_{or}(\hat{B})_i$ are both integers, so

$$\begin{aligned} a_{or}(B)_i - a_{or}(\hat{B})_i &= \left(\frac{B}{k} + \frac{1}{2}\log_2\frac{\sigma_i^2}{g}\right) - \left(\frac{\hat{B}}{k} + \frac{1}{2}\log_2\frac{\sigma_i^2}{g}\right) \quad \text{[from Lemma 4.1]} \\ &= \frac{B - \hat{B}}{k} \in \mathbb{Z} \end{aligned}$$

which implies $B \bmod k = n$. Now suppose $B \bmod k = n$. This implies there exists an integer m such that $B = \hat{B} + km$. Hence, for each i,

$$\begin{aligned} a_{or}(B)_i &= \frac{B}{k} + \frac{1}{2}\log_2\frac{\sigma_i^2}{g} && \text{[from Lemma 4.1]} \\ &= m + \frac{\hat{B}}{k} + \frac{1}{2}\log_2\frac{\sigma_i^2}{g} \\ &= m + a_{or}(\hat{B})_i && \text{[from Lemma 4.1]} \\ \therefore a_{or}(B) &\in \mathcal{A}_I(\hat{B} + km) && \text{[from } a_{or}(\hat{B}) \in \mathcal{A}_I(\hat{B})\text{]} \\ &= \mathcal{A}_I(B). \end{aligned}$$

Now suppose $\delta > 0$. Then for each bit budget B and for any $b \in \mathcal{A}_{oi}(B)$,

$$\|b - a_{or}(B)\| \geq \delta > 0. \tag{4.20}$$

Define a function $f : [0, \infty) \to (0, \infty)$ by

$$f(u) = 4^{-u\sqrt{(k-1)/k}} + (k-1)4^{u\sqrt{1/(k(k-1))}}.$$

For each bit budget B and for every $b \in \mathcal{A}_{oi}(B)$,

$$\begin{aligned}
p^{oi} &= \frac{d_{oi}}{d_{or}} \\
&\geq \frac{hg4^{-B/k}}{d_{or}} \cdot f(\|b - a_{or}(B)\|) && \text{[from Lemma 4.21]} \\
&= \frac{1}{k} f(\|b - a_{or}(B)\|) && \text{[from Lemma 4.1]} \\
&\geq \frac{1}{k} f(\delta) && \text{[from (4.20) and the monotonicity of } f\text{]} \\
&> 1 && \text{[from the arithmetic-geometric mean inequality]}.
\end{aligned}$$

□

4.4.1 Lower Bound on Worst Case Distortion Penalty for Integer Bit Allocations

For any particular set of k sources, the distortion obtained by using optimal integer-valued bit allocation may be larger than the distortion predicted by optimal real-valued bit allocation. Theorem 4.24 below illustrates how much worse integer-valued bit allocation can be compared to real-valued bit allocation.

Let

$$\gamma_k = \frac{1}{2k+2}\big(-k, -k+2, \ldots, k-2, k\big).$$

Lemma 4.23. *If the variances $\sigma_1^2, \ldots, \sigma_k^2$ of k scalar sources satisfy*

$$\frac{1}{2}\left(\log_2 \frac{\sigma_1^2}{g}, \ldots, \log_2 \frac{\sigma_k^2}{g}\right) = \gamma_{k-1}$$

then for each bit budget B and for any $b \in \mathcal{A}_{oi}(B)$, the vector $b - a_{or}(B)$ is a permutation of γ_{k-1}.

Theorem 4.24. *For each k, there exist k scalar sources, such that for any bit budget, the distortion penalty resulting from optimal integer bit allocation satisfies*

$$p^{oi} = \frac{3 \cdot 2^{(k-1)/k}}{k(4 - 4^{(k-1)/k})} > 1.$$

The distortion penalty in Theorem 4.24 is monotone increasing with k and is bounded as:

$$1.06 \approx \frac{3\sqrt{2}}{4} \leq p^{oi} \leq \frac{3}{4 \ln 2} \approx 1.08$$

where the lower bound is attained at $k = 2$ and the upper bound is approached as $k \to \infty$. Thus, the theorem guarantees that for some sources, the mean-squared error due to optimal integer-valued bit allocation is at least 6% greater (and as much as 8% greater for large k) than the mean-squared error due to optimal real-valued bit allocation. We do not claim this is the largest possible distortion penalty – rather, it demonstrates that p^{oi} can be bounded away from 1.

Proof. Let $a > 0$ be arbitrary. For each $i \leq k$, consider a scalar source whose variance is given by

$$\sigma_i^2 = a 4^{(\gamma_{k-1})_i}.$$

Then

$$g = \left(\prod_{i=1}^{k} \sigma_i^2 \right)^{1/k} = a$$
$$\frac{1}{2} \left(\log_2 \frac{\sigma_1^2}{g}, \ldots, \log_2 \frac{\sigma_k^2}{g} \right) = \gamma_{k-1}$$

and Lemma 4.23 implies that for each bit budget B and for any $b \in \mathcal{A}_{oi}(B)$, the vector

$b - a_{or}(B)$ is a permutation of γ_{k-1}. Hence, for each B,

$$\begin{aligned} p^{oi} &= \frac{1}{k}\sum_{i=1}^{k} 4^{-(\gamma_{k-1})_i} && \text{[from (4.18)]} \\ &= \frac{1}{k}\sum_{i=0}^{k-1} 4^{-[(-(k-1)+2i)/2k]} \\ &= \frac{2^{(k-1)/k}}{k}\sum_{i=0}^{k-1} 4^{-i/k} \\ &= \frac{2^{(k-1)/k}}{k}\cdot\frac{1-\left(4^{-1/k}\right)^k}{1-4^{-1/k}} \\ &= \frac{3\cdot 2^{(k-1)/k}}{k\left(4-4^{(k-1)/k}\right)}. \end{aligned} \tag{4.21}$$

Applying the arithmetic-geometric mean inequality to (4.21) gives $p^{oi} > 1$. □

We note that for the sources used in the proof of Theorem 4.24, the lower bound in Theorem 4.24 is greater than that given in case (ii) of Theorem 4.22, for all k.

4.5 Upper Bound on Distortion Penalty for Integer Bit Allocations

Lemma 4.8 and Theorem 4.10 imply that each component of any optimal integer bit allocation b differs from the corresponding component of the optimal real-valued bit allocation by less than 1. Hence, using (4.2) and Lemma 4.1, one easily obtains the bound

$$p^{oi} = \frac{hg4^{-B/k}\sum_{i=1}^{k} 4^{(a_{or}(B)-b)_i}}{khg4^{-B/k}} < 4.$$

In the following theorem we give a tighter upper bound on the distortion penalty resulting from optimal integer bit allocation. The bound does not depend on the source distribution or the bit budget.

Theorem 4.25. *For each $k \geq 2$, for any k scalar sources, and for any bit budget, the distortion penalty resulting from optimal integer bit allocation is upper bounded as*

$$p^{oi} \leq 4^{\tau} \left(1 - \frac{3\tau}{4}\right)$$

where $\tau = \frac{1}{k}\left\lceil \frac{4k}{3} - \frac{1}{1-4^{-1/k}} \right\rceil$.

The upper bound on p^{oi} in Theorem 4.25 is tight since, for arbitrary $a > 0$, if

$$\sigma_i^2 = a4^{-c(k-1,k\tau)_i} \qquad\qquad 1 \leq i \leq k$$

and the bit budget B is a multiple of k, then by Theorem 4.6, Theorem 4.10, and (4.18) we have $p^{oi} = 4^{\tau}\left(1 - \frac{3\tau}{4}\right)$. For all $k \geq 2$, the upper bound on p^{oi} in Theorem 4.25 satisfies

$$1.25 \leq 4^{\tau}\left(1 - \frac{3\tau}{4}\right) < \frac{3}{e2^{1/3}\ln 2} \approx 1.26 \tag{4.22}$$

where the lower bound in (4.22) is attained at $k = 2$ and $k = 4$ and the upper bound in (4.22) is approached as $k \to \infty$. Thus, Theorem 4.25 guarantees that for any k scalar sources and for all bit budgets, the mean-squared error due to optimal integer-valued bit allocation is at most 26% greater than the mean-squared error due to optimal real-valued bit allocation.

Figure 4.1 compares the upper bound in Theorem 4.25 with the distortion penalty from Theorem 4.24.

Proof. We show that

$$\sup_B \sup_{\sigma_1^2,\ldots,\sigma_k^2} \frac{d_{oi}}{d_{or}} = 4^\tau \left(1 - \frac{3\tau}{4}\right)$$

where, for a fixed k, the suprema are taken over all possible k-tuples of sources and over all bit budgets.

Define a mapping $f : \mathbb{R}^k \to \mathbb{R}$ by

$$f(u) = \sum_{i=1}^{k} 4^{-u_i}.$$

Then we have

$$\begin{aligned}
&\sup_B \sup_{\sigma_1^2,\ldots,\sigma_k^2} \frac{d_{oi}}{d_{or}} \\
&= \frac{1}{k} \sup_B \sup_{\sigma_1^2,\ldots,\sigma_k^2} \sum_{i=1}^{k} 4^{(a_{or}(B)-b)_i} \qquad \forall b \in \mathcal{A}_{oi}(B) && \text{[from (4.18)]} \\
&= \frac{1}{k} \sup_B \sup_{\sigma_1^2,\ldots,\sigma_k^2} \sum_{i=1}^{k} 4^{(a_{or}(B)-b)_i} \qquad \forall b \in \mathcal{A}_{ci}(B) && \text{[from Theorem 4.10]} \\
&= \frac{1}{k} \sup_B \sup_{\sigma_1^2,\ldots,\sigma_k^2} \sum_{i=1}^{k} 4^{-u_i} \qquad \forall u \in \Delta && \text{[from the definition of } \Delta\text{]} \\
&= \frac{1}{k} \sup_B \sup_{u \in M_B} f(u) && \text{[from the definition of } M_B\text{]} \\
&= \frac{1}{k} \sup_{u \in V_{k-1}(0)} f(u) && \text{[from Lemma 4.4]} \\
&= \frac{1}{k} \max_{0 \le j \le k-1} \sum_{i=1}^{k} 4^{-c(k-1,j)_i} && (4.23) \\
&= \max_{0 \le j \le k-1} 4^{j/k} \left(1 - \frac{3}{4k} j\right) && \text{[from (4.3)]}
\end{aligned}$$

where (4.23) follows from the fact that the convex function f, restricted to the closed and bounded polytope $V_{k-1}(0)$, achieves a global maximum (e.g., see [27, Theorem 6.12 on

p. 154]) on the polytope's set of vertices, which consists of all coordinate permutations of $c(k-1,0),\ldots,c(k-1,k-1)$ [8, p. 461–462].

For $j = 0,\ldots,k-1$, define

$$g(j) = 4^{j/k}\left(1 - \frac{3}{4k}j\right).$$

Since $g(j) > 0$ if and only if $j < 4k/3$, the function g must attain it's maximum when $j < 4k/3$. In the range $0 \le j < 4k/3$, the ratio

$$\frac{g(j+1)}{g(j)} = 4^{1/k}\left(1 - \frac{1}{\frac{4k}{3} - j}\right)$$

is greater than 1 if and only if

$$j < \frac{4k}{3} - \frac{1}{1 - 4^{-1/k}}$$

so g attains is maximum when $j = \lceil \frac{4k}{3} - \frac{1}{1-4^{-1/k}} \rceil$. □

4.6 Acknowledgment

The authors wish to thank Bob Gray, Khalid Sayood, and Vivek Goyal for suggesting at the 2005 Data Compression Conference the nonnegative component version of the bit allocation problem.

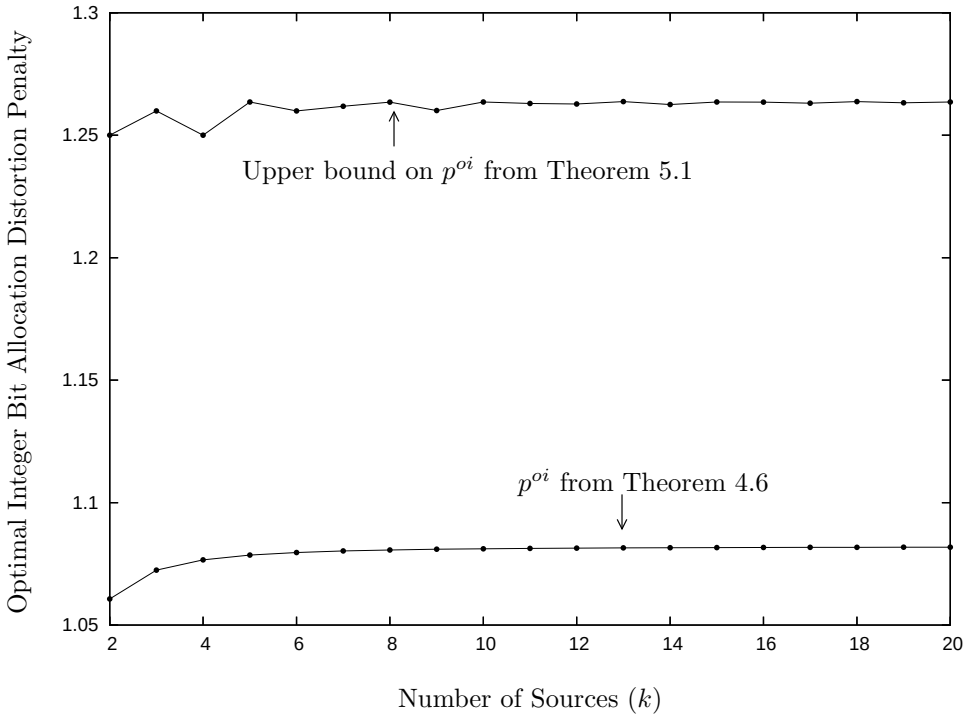

Figure 4.1: Plot of the achievable distortion penalty from Theorem 4.24 and the upper bound on the distortion penalty from Theorem 4.25.

Appendix

Proof of Lemma 4.3.

$$\begin{aligned}
&\left\{u \in \Lambda_k + y : \|w-u\| = \min_{v\in\Lambda_k+y} \|w-v\|\right\} \\
&= y + \left\{u \in \Lambda_k : \|w-(u+y)\| = \min_{v\in\Lambda_k} \|w-(v+y)\|\right\} \\
&= \Phi_k(w-y) + y.
\end{aligned}$$

□

Proof of Lemma 4.4. First, note that for each $k \geq 1$ and for any $u \in \mathcal{H}^k$, the symmetry of Λ_k implies that

$$u \in V_k(0) \text{ if and only if } -u \in V_k(0). \tag{4.24}$$

Also, note that since Λ_{k-1} consists of all vectors with k integer coordinates which sum to 0, and since

$$\frac{B}{k}(\underbrace{1,\ldots,1}_{k}) + c(k-1, B \bmod k) \in \mathcal{A}_I(B)$$

it follows that

$$\mathcal{A}_I(B) = \Lambda_{k-1} + \frac{B}{k}(\underbrace{1,\ldots,1}_{k}) + c(k-1, B \bmod k). \tag{4.25}$$

Now, Lemma 4.3 and (4.25) imply that

$$\begin{aligned}
\mathcal{A}_{ci}(B) = \Phi_{k-1}&\left(a_{or}(B) - \left[\frac{B}{k}(\underbrace{1,\ldots,1}_{k}) + c(k-1, B \bmod k)\right]\right) \\
&+ \frac{B}{k}(\underbrace{1,\ldots,1}_{k}) + c(k-1, B \bmod k).
\end{aligned}$$

Thus, Lemma 4.1 gives

$$\Delta = \Phi_{k-1}(\mu) - \mu.$$

Since $\mu \in V_{k-1}\big(w\big)$ for all $w \in \Phi_{k-1}(\mu)$, we have that for each $w \in \Phi_{k-1}(\mu)$

$$\|\mu - w\| \leq \|\mu - y\| \qquad \forall y \in \Lambda_{k-1} \tag{4.26}$$

$$= \|(\mu - w) - (y - w)\| \qquad \forall y \in \Lambda_{k-1}. \tag{4.27}$$

Since $w \in \Lambda_{k-1}$, we have $y - w \in \Lambda_{k-1}$ for all $y \in \Lambda_{k-1}$. Thus, by (4.24) and the definition of $V_{k-1}(0)$, (4.26)–(4.27) imply $w - \mu \in V_{k-1}(0)$. Hence, $\Delta \subset V_{k-1}(0)$, and therefore $M_B \subset V_{k-1}(0)$.

Now, for any $v \in V_{k-1}(0)$ and for arbitrary $a > 0$, setting

$$\sigma_i^2 = a4^{c(k-1,\, B \bmod k)_i} - v_i$$

for $1 \leq i \leq k$ results in

$$g = \left(\prod_{i=1}^{k} \sigma_i^2\right)^{1/k} = a$$

$$\frac{1}{2}\left(\log_2 \frac{\sigma_1^2}{g}, \ldots, \log_2 \frac{\sigma_k^2}{g}\right) = c(k-1, B \bmod k) - v$$

and therefore

$$\Delta = \Phi_{k-1}(-v) + v.$$

Since $v \in V_{k-1}(0)$, by (4.24), we also have $-v \in V_{k-1}(0)$. Hence, $0 \in \Phi_{k-1}(-v)$, and thus $v \in \Delta$. So, $V_{k-1}(0) \subset M_B$ and therefore $M_B = V_{k-1}(0)$. □

Proof of Lemma 4.5. From (4.3) we have

$$\{c(k-1, B \bmod k) : B \geq 1\} = \{c(k-1, B \bmod k) : 1 \leq B \leq k\}.$$

Lemmas 4.3 and 4.4 imply that for each B, any element of Δ is the difference between the vector $(1/2)\big(\log_2(\sigma_1^2/g), \ldots, \log_2(\sigma_k^2/g)\big)$ and a point (not necessarily unique) closest to it from the set $\Lambda_{k-1} + c(k-1, B \bmod k)$. Hence,

$$\bigcup_{B \geq 1} \Delta \subset \bigcup_{1 \leq B \leq k} \Delta$$

so, in fact, these two unions are equal. The result then follows from the fact that for each B, the set Δ is finite. □

Proof of Lemma 4.7. Let t and $\mathcal{I}_k$ be defined as in Theorem 4.6 and let $b = \beta + a_{or}(B) \in \mathcal{A}_{ci}(B)$. Then for all i,

$$\beta_i = b_i - r(a_{or}(B)_i) - \rho(a_{or}(B)_i) \quad \text{[from the definitions of } \rho \text{ and } r] \tag{4.28}$$

$$-\frac{1}{2} \leq \rho(a_{or}(B)_i) < \frac{1}{2} \quad \text{[from the definitions of } \rho \text{ and } r] \tag{4.29}$$

$$b_i - r(a_{or}(B)_i) \in \begin{cases} \{0, -1\} & \text{if } t > 0 \\ \{0, 1\} & \text{if } t < 0 \end{cases} \quad \text{[from Theorem 4.6]}. \tag{4.30}$$

Since $\beta_j \in (-1/2, 1/2]$, we have

$$-\frac{1}{2} < b_j - r(a_{or}(B)_j) - \rho(a_{or}(B)_j) \leq \frac{1}{2} \quad \text{[from (4.28)]} \tag{4.31}$$

$$b_j - r(a_{or}(B)_j) = 0 \quad \text{[from (4.29),(4.30),(4.31)]} \tag{4.32}$$

$$\beta_j = -\rho(a_{or}(B)_j) \quad \text{[from (4.28),(4.32)]}.$$

Suppose $\beta_i < -1/2$. Then

$$\begin{aligned} b_i - r(a_{or}(B)_i) &= \beta_i + \rho(a_{or}(B)_i) && \text{[from (4.28)]} \\ &< -\frac{1}{2} + \rho(a_{or}(B)_i) \\ &< 0 && \text{[from (4.29)]} \qquad (4.33) \\ b_i - r(a_{or}(B)_i) &= -1 && \text{[from (4.30),(4.33)]} \qquad (4.34) \\ \beta_i &= -\rho(a_{or}(B)_i) - 1 && \text{[from (4.28),(4.34)].} \end{aligned}$$

By (4.32), (4.34), the fact that $b \in \mathcal{A}_{ci}(B)$, and Theorem 4.6, there exists $(i_1, \ldots, i_k) \in \mathcal{I}_k$ such that $i \in \{i_1, \ldots, i_t\}$, $j \in \{i_{t+1}, \ldots, i_k\}$, and $\rho(a_{or}(B)_i) \leq \rho(a_{or}(B)_j)$.

Suppose $\beta_i > 1/2$. Then

$$\begin{aligned} b_i - r(a_{or}(B)_i) &= \beta_i + \rho(a_{or}(B)_i) && \text{[from (4.28)]} \\ &> \frac{1}{2} + \rho(a_{or}(B)_i) \\ &> 0 && \text{[from (4.29)]} \qquad (4.35) \\ b_i - r(a_{or}(B)_i) &= 1 && \text{[from (4.30),(4.35)]} \qquad (4.36) \\ \beta_i &= -\rho(a_{or}(B)_i) + 1 && \text{[from (4.28),(4.36)].} \end{aligned}$$

By (4.32), (4.36), the fact that $b \in \mathcal{A}_{ci}(B)$, and Theorem 4.6, there exists $(i_1, \ldots, i_k) \in \mathcal{I}_k$ such that $i \in \{i_{k+t+1}, \ldots, i_k\}$, $j \in \{i_1, \ldots, i_{k+t}\}$, and $\rho(a_{or}(B)_i) \geq \rho(a_{or}(B)_j)$. □

Proof of Lemma 4.8. Let $\hat{b}$ and $\mathcal{I}_k$ be defined as in Theorem 4.6. If $t = 0$, then the result follows from Theorem 4.6 and the definitions of Δ and $r(\cdot)$.

Suppose $t > 0$ and let $b = \beta + a_{or}(B) \in \mathcal{A}_{ci}(B)$. By Theorem 4.6, there exists

$(i_1, \ldots, i_k) \in \mathcal{I}_k$ such that

$$b_j = \begin{cases} \hat{b}_j - 1 & \text{if} \quad j \in \{i_1, \ldots, i_t\} \\ \hat{b}_j & \text{if} \quad j \in \{i_{t+1}, \ldots, i_k\}. \end{cases} \tag{4.37}$$

Subtracting $a_{or}(B)$ from both sides of (4.37) gives

$$\begin{aligned} \beta_j &= \begin{cases} \hat{b}_j - 1 - a_{or}(B)_j & \text{if} \quad j \in \{i_1, \ldots, i_t\} \\ \hat{b}_j - a_{or}(B)_j & \text{if} \quad j \in \{i_{t+1}, \ldots, i_k\} \end{cases} \\ &= \begin{cases} -\rho(a_{or}(B)_j) - 1 & \text{if} \quad j \in \{i_1, \ldots, i_t\} \\ -\rho(a_{or}(B)_j) & \text{if} \quad j \in \{i_{t+1}, \ldots, i_k\}. \end{cases} \end{aligned}$$

Since $-1/2 \leq \rho(a_{or}(B)_j) < 1/2$, we have $-\rho(a_{or}(B)_j) \in (-1/2, 1/2] \subset (-1, 1/2]$. Thus, it suffices to show that $\rho(a_{or}(B)_j) < 0$ for $j \in \{i_1, \ldots, i_t\}$, since then $-\rho(a_{or}(B)_j) - 1 \in (-1, -1/2]$.

Let n denote the number of components of $a_{or}(B)$ such that $\rho(a_{or}(B)_j) < 0$. Since the subscripts i_j are ordered by increasing value of $\rho(a_{or}(B)_j)$, we have $\rho(a_{or}(B)_j) < 0$ for $j \in \{i_1, \ldots, i_n\}$. Hence, it suffices to show $t \leq n$. We have

$$\begin{aligned} t &= \left(\sum_{i=1}^{k} r(a_{or}(B)_i) \right) - B \\ &= n + \left(\sum_{i=1}^{k} \lfloor a_{or}(B)_i \rfloor \right) - B \\ &= n - \sum_{i=1}^{k} (a_{or}(B)_i - \lfloor a_{or}(B)_i \rfloor) \\ &\leq n. \end{aligned}$$

The result then follows by symmetry for $t < 0$. □

Proof of Lemma 4.9. Since $\beta \in \Delta$, Lemma 4.8 gives $\beta_i, \beta_j \in (-1, 1)$. It is easy to

verify that $\beta_j - \beta_i \leq 1$ in the following three cases:

- $\beta_i, \beta_j \in [0, 1)$
- $\beta_i \in (-1, 1)$, $\beta_j \in (-1, 0]$
- $\beta_i \in [-1/2, 0]$, $\beta_j \in [0, 1/2]$.

The inequality also holds for $\beta_i \in (-1, -1/2)$ and $\beta_j \in [0, 1/2]$ since

$$\beta_j - 1 = -\rho(a_{or}(B)_j) - 1 \leq -\rho(a_{or}(B)_i) - 1 = \beta_i \qquad \text{[from Lemma 4.7]}$$

and it holds for $\beta_i \in (-1/2, 0]$ and $\beta_j \in (1/2, 1)$ since

$$\beta_j - 1 = -\rho(a_{or}(B)_j) \leq -\rho(a_{or}(B)_i) = \beta_i \qquad \text{[from Lemma 4.7]}.$$

Finally, Lemma 4.8 implies that it cannot be the case that $\beta_i \in (-1, -1/2]$ and $\beta_j \in (1/2, 1)$. Thus, $\beta_j - \beta_i \leq 1$ for all i and j.

Let $\tilde{b} = b + \omega(i, j)$ and suppose $\beta_j - \beta_i = 1$. Then

$$\tilde{b}_i = b_i + 1 = \beta_i + 1 + a_{or}(B)_i = \beta_j + a_{or}(B)_i$$
$$\tilde{b}_j = b_j - 1 = \beta_j - 1 + a_{or}(B)_j = \beta_i + a_{or}(B)_j.$$

Hence,

$$\tilde{b}_l - a_{or}(B)_l = \begin{cases} \beta_j & \text{if } l = i \\ \beta_i & \text{if } l = j \\ \beta_l & \text{otherwise .} \end{cases}$$

Therefore, $\|\tilde{b} - a_{or}(B)\| = \|\beta\|$, which by the definition of Δ, implies $\tilde{b} \in \mathcal{A}_{ci}(B)$. $\square$

Proof of Lemma 4.12. Assume $Q(W,f) \subset V \subset W$. If $b \in Q(W,f)$, then

$$\begin{aligned} f(b) &= \min_{\hat{b}\in W} f(\hat{b}) && [\text{from } b \in Q(W,f)] \\ &\leq \min_{\hat{b}\in V} f(\hat{b}) && [\text{from } V \subset W] \\ &\leq f(b) && [\text{from } b \in V] \end{aligned}$$

and therefore $b \in Q(V,f)$. Thus, $Q(W,f) \subset Q(V,f)$.

If $b \in Q(V,f)$, then

$$\begin{aligned} f(b) &= \min_{\hat{b}\in V} f(\hat{b}) && [\text{from } b \in Q(V,f)] \\ &\leq \min_{\hat{b}\in Q(W,f)} f(\hat{b}) && [\text{from } Q(W,f) \subset V] \\ &= \min_{\hat{b}\in W} f(\hat{b}) && [\text{from the definition of } Q(W,f)] \\ &\leq f(b) && [\text{from } b \in V \subset W] \end{aligned}$$

and therefore $b \in Q(W,f)$. Thus, $Q(V,f) \subset Q(W,f)$. □

Proof of Lemma 4.13. Suppose $b \in Q(\mathcal{A}_I(B),\theta_1)$. For any i and j, the following identity holds:

$$\begin{aligned} &[(b_i - 1) - a_{or}(B,S)_i]^2 + [(b_j + 1) - a_{or}(B,S)_j]^2 \\ &- [b_i - a_{or}(B,S)_i]^2 - [b_j - a_{or}(B,S)_j]^2 \\ &= 2[1 + a_{or}(B,S)_i - b_i + b_j - a_{or}(B,S)_j]. \end{aligned} \tag{4.38}$$

Now, suppose there exists an i such that $b_i - a_{or}(B,S)_i \geq 1$. Then there must exist a j such that $b_j - a_{or}(B,S)_j < 0$, since $\sum_l b_l = \sum_l a_{or}(B,S)_l = B$. But then the right-hand side of (4.38) would be negative which would imply $b \notin Q(\mathcal{A}_I(B),\theta_1)$, since

subtracting 1 from b_i and adding 1 to b_j would result in an integer bit allocation closer than b to $a_{or}(B,S)$. A similar contradiction results in the case where $b_i - a_{or}(B,S)_i \leq -1$. Thus, for every i, we must have $b_i \in \{\lfloor a_{or}(B,S)_i \rfloor, \lceil a_{or}(B,S)_i \rceil\}$.

The definition of $a_{or}(B,S)$ then implies $b_i = 0$ for all $i \notin S$. Thus, $b \in \mathcal{A}_I(B,S)$, and therefore $Q(\mathcal{A}_I(B),\theta_1) \subset \mathcal{A}_I(B,S)$. Now applying Lemma 4.12 with $W = \mathcal{A}_I(B)$, $V = \mathcal{A}_I(B,S)$, and $f = \theta_1$ gives $Q(\mathcal{A}_I(B),\theta_1) = \mathcal{A}_{ci}(B,S)$. □

Proof of Lemma 4.14. Consider a set of sources $\hat{X}_1,\ldots,\hat{X}_k$ with variances $\hat{\sigma}_1^2,\ldots,\hat{\sigma}_k^2$ given by

$$\hat{\sigma}_i^2 = \begin{cases} \frac{\sigma_i^2}{g(S)} 4^{\frac{B(k-|S|)}{k|S|}} & \text{if } i \in S \\ 4^{-B/k} & \text{if } i \notin S. \end{cases}$$

The geometric mean of the variances is

$$\begin{aligned}
&\left(\prod_{i=1}^{k} \hat{\sigma}_i^2\right)^{1/k} \\
&= \left(\prod_{i\in S} \frac{\sigma_i^2}{g(S)} \cdot 4^{\frac{B(k-|S|)}{k|S|}}\right)^{1/k} \left(\prod_{i\notin S} 4^{-B/k}\right)^{1/k} \\
&= \left(\frac{\prod_{i\in S} \sigma_i^2}{g(S)^{|S|}} \cdot 4^{\frac{B(k-|S|)}{k}}\right)^{1/k} \left(4^{\frac{-B(k-|S|)}{k}}\right)^{1/k} \\
&= \left(\frac{g(S)^{|S|}}{g(S)^{|S|}} \cdot 4^{\frac{B(k-|S|)}{k}}\right)^{1/k} \left(4^{\frac{-B(k-|S|)}{k}}\right)^{1/k} \qquad [\text{from } g(S) = \left(\textstyle\prod_{i\in S} \sigma_i^2\right)^{1/|S|}] \\
&= 1.
\end{aligned}$$

Therefore, substituting the variances and their geometric mean into Lemma 4.1 gives

$$a_{or}(B)_i = \frac{B}{k} + \frac{1}{2}\log_2 \frac{\hat{\sigma}_i^2}{1}$$

$$= \begin{cases} \frac{B}{|S|} + \frac{1}{2}\log_2 \frac{\sigma_i^2}{g(S)} & \text{if } i \in S \\ 0 & \text{otherwise} \end{cases}$$

$$= a_{or}(B,S)_i.$$

Hence, $a_{or}(B,S)$ is the optimal real-valued bit allocation for $\hat{X}_1, \ldots, \hat{X}_k$. Thus, by Lemma 4.13, $\mathcal{A}_{ci}(B,S)$ is the set of closest integer bit allocations for $\hat{X}_1, \ldots, \hat{X}_k$ (regardless of S). Let

$$t = |(r(a_{or}(B,S)_1), \ldots, r(a_{or}(B,S)_k))| - B$$

and for $b \in \mathcal{A}_{ci}(B,S)$, let

$$\beta = b - a_{or}(B,S) \in \mathcal{A}_{ci}(B,S) - a_{or}(B,S).$$

Then Lemma 4.8 implies that for all i,

$$\beta_i \in \begin{cases} (-1/2, 1/2] & \text{if } t = 0 \\ (-1, 1/2] & \text{if } t > 0 \\ (-1/2, 1) & \text{if } t < 0\,. \end{cases} \tag{4.39}$$

Combining the fact that $a_{or}(B,S)_i \geq 0$ for all i with (4.39) gives $b_i \geq 0$ for all i. □

Proof of Lemma 4.15. Let

$$m = \left(\frac{B(k - |S|)}{k|S|} + \frac{1}{2}\log_2 \frac{g}{g(S)} \right).$$

Then for all $i \in S$,

$$a_{or}(B)_i + m = a_{or}(B,S)_i. \tag{4.40}$$

Suppose $\mathcal{A}_{ci}^{+}(B) \subset \mathcal{A}_I(B,S)$. Since every vector in $\mathcal{A}_{ci}^{+}(B)$ is nonnegative, we have

$$\begin{aligned}\mathcal{A}_{ci}^{+}(B) &\subset \mathcal{A}_I^{+}(B,S) \\ &\subset \mathcal{A}_I^{+}(B). \end{aligned} \tag{4.41}$$

From (4.41), we can apply Lemma 4.12 with $W = \mathcal{A}_I^{+}(B)$, $V = \mathcal{A}_I^{+}(B,S)$ and $f(b) = \|b - a_{or}(B)\|$ to obtain

$$\mathcal{A}_{ci}^{+}(B) = Q\big(\mathcal{A}_I^{+}(B,S), \|b - a_{or}(B)\|\big). \tag{4.42}$$

For any $b \in \mathcal{A}_I^{+}(B,S)$,

$$\begin{aligned}&\|b - a_{or}(B)\|^2 \\ &= \sum_{i \notin S} |a_{or}(B)_i|^2 + \sum_{i \in S} |b_i - a_{or}(B)_i|^2 \qquad [\text{from } b_i = 0 \;\forall\; i \notin S] \qquad (4.43) \\ &= \sum_{i \notin S} |a_{or}(B)_i|^2 + \sum_{i \in S} |m + b_i - (m + a_{or}(B)_i)|^2 \\ &= \sum_{i \notin S} |a_{or}(B)_i|^2 + |S| \cdot m^2 + \sum_{i \in S} 2m\,(b_i - a_{or}(B,S)_i) + (b_i - a_{or}(B,S)_i)^2 \\ &\qquad\qquad [\text{from (4.40)}] \\ &= \sum_{i \notin S} |a_{or}(B)_i|^2 + |S| \cdot m^2 + \sum_{i \in S} (b_i - a_{or}(B,S)_i)^2 \\ &\qquad\qquad [\text{from } \sum_{i \in S} b_i = \sum_{i \in S} a_{or}(B,S)_i = B] \qquad (4.44) \\ &= \sum_{i \notin S} |a_{or}(B)_i|^2 + |S| \cdot m^2 + \sum_{i=1}^{k} |b_i - a_{or}(B,S)_i|^2\end{aligned}$$

$$[\text{from } b_i = a_{or}(B,S)_i = 0 \;\forall\; i \notin S]$$

(4.45)

$$= \sum_{i \notin S} |a_{or}(B)_i|^2 + |S| \cdot m^2 + \|b - a_{or}(B,S)\|^2 . \tag{4.46}$$

Equations (4.43)–(4.46) show that the quantities $\|b - a_{or}(B)\|$ and $\|b - a_{or}(B,S)\|$ differ by a constant which is independent of b. Hence, among all bit allocations in $\mathcal{A}_I^+(B,S)$, we see that b is of minimal distance from $a_{or}(B,S)$ if and only if b is of minimal distance from $a_{or}(B)$, i.e. $\mathcal{A}_{ci}^+(B,S) = Q\big(\mathcal{A}_I^+(B,S), \|b - a_{or}(B)\|\big)$. Thus, by (4.42), we have $\mathcal{A}_{ci}^+(B) = \mathcal{A}_{ci}^+(B,S)$.

Now, to show the second part of the lemma, suppose $\mathcal{A}_{oi}^+(B) \subset \mathcal{A}_I(B,S)$. Since every vector in $\mathcal{A}_{oi}^+(B)$ is nonnegative, we have

$$\begin{aligned} \mathcal{A}_{oi}^+(B) &\subset \mathcal{A}_I^+(B,S) \\ &\subset \mathcal{A}_I^+(B). \end{aligned} \tag{4.47}$$

From (4.47), we can apply Lemma 4.12 with $W = \mathcal{A}_I^+(B)$, $V = \mathcal{A}_I^+(B,S)$, and $f(b) = d$ to obtain

$$\mathcal{A}_{oi}^+(B) = Q\left(\mathcal{A}_I^+(B,S), d\right). \tag{4.48}$$

For any $b \in \mathcal{A}_I^+(B,S)$,

$$\begin{aligned} d &= hg4^{-B/k} \cdot \sum_{i=1}^{k} 4^{(a_{or}(B)-b)_i} && [\text{from Lemma 4.2}] \\ &= hg4^{-B/k} \cdot \sum_{i \notin S} 4^{a_{or}(B)_i} + hg4^{-B/k} \cdot \sum_{i \in S} 4^{(a_{or}(B)-b)_i} && [\text{from } b_i = 0 \;\forall\; i \notin S] \\ &= hg4^{-B/k} \cdot \sum_{i \notin S} 4^{a_{or}(B)_i} + hg4^{-B/k} \cdot \sum_{i \in S} 4^{-m+(m+a_{or}(B)_i)-b_i} \\ &= hg4^{-B/k} \cdot \sum_{i \notin S} 4^{a_{or}(B)_i} + hg4^{(-B/k)-m} \cdot \sum_{i \in S} 4^{(a_{or}(B,S)-b)_i} && [\text{from (4.40)}] \end{aligned}$$

$$
\begin{aligned}
&= hg4^{-B/k} \cdot \sum_{i \notin S} 4^{a_{or}(B)_i} \\
&\quad + hg4^{(-B/k)-m} \cdot \left(\sum_{i \in S} 4^{(a_{or}(B,S)-b)_i} + \sum_{i \notin S} 4^{(a_{or}(B,S)-b)_i} - \sum_{i \notin S} 4^{(a_{or}(B,S)-b)_i} \right) \\
&= hg4^{-B/k} \cdot \sum_{i \notin S} \left(4^{a_{or}(B)_i} - 4^{-m} \right) + 4^{-m} \cdot \theta_2(b) \quad [\text{from } a_{or}(B,S)_i = b_i = 0 \; \forall \; i \notin S]
\end{aligned}
$$

which is an affine function of $\theta_2(b)$, with coefficients which are independent of b. Therefore, among all bit allocations in $\mathcal{A}_I^+(B,S)$, we see that b minimizes $\theta_2(b)$ if and only if b minimizes d, i.e. $\mathcal{A}_{oi}^+(B,S) = Q\big(\mathcal{A}_I^+(B,S), d\big)$. Thus, by (4.48), we have $\mathcal{A}_{oi}^+(B) = \mathcal{A}_{oi}^+(B,S)$. □

Proof of Lemma 4.16. We prove that for all $b \in \mathcal{A}_I(B,S)$, if $b_i > 0$, then $b \notin \mathcal{A}_{ci}^+(B) \cup \mathcal{A}_{oi}^+(B)$.

Let $b \in \mathcal{A}_I(B,S)$ satisfy $b_i > 0$. By Lemma 4.15, we know that $\mathcal{A}_{ci}^+(B) = \mathcal{A}_{ci}^+(B,S)$ and $\mathcal{A}_{oi}^+(B) = \mathcal{A}_{oi}^+(B,S)$. We will show that $b \notin \mathcal{A}_{ci}^+(B,S)$ and $b \notin \mathcal{A}_{oi}^+(B,S)$. In particular, we demonstrate that there exists $j \in S$ such that adding 1 bit to b_j and subtracting 1 bit from b_i reduces both $\theta_1(b)$ and $\theta_2(b)$, i.e., the original b chosen could not have been an optimal nor a closest nonnegative integer bit allocation.

Suppose

$$
a_{or}(B,S)_i - b_i \geq a_{or}(B,S)_l - b_l - 1 \;\; \forall l \in S. \tag{4.49}
$$

Then we get

$$
\begin{aligned}
&(|S|-1)(a_{or}(B,S)_i - b_i) \\
&= \sum_{\substack{l \in S \\ l \neq i}} a_{or}(B,S)_i - b_i \\
&\geq \sum_{\substack{l \in S \\ l \neq i}} (a_{or}(B,S)_l - b_l - 1) \qquad [\text{from (4.49)}]
\end{aligned}
$$

$$= (B - a_{or}(B,S)_i) - (B - b_i) - (|S| - 1) \quad [\text{from } \sum_{l \in S} b_l = \sum_{l \in S} a_{or}(B,S)_l = B]$$

$$= -(a_{or}(B,S)_i - b_i) - (|S| - 1)$$

which implies

$$|S|(1 - b_i + a_{or}(B,S)_i) \geq 1,$$

a contradiction, since $b_i \geq 1$ and $a_{or}(B,S)_i < 0$. Therefore, (4.49) is false, so there exists $j \in S$ such that $j \neq i$ and

$$a_{or}(B,S)_i - b_i < a_{or}(B,S)_j - b_j - 1. \tag{4.50}$$

Multiplying each side of (4.50) by -2 and adding $(a_{or}(B,S)_i - b_i)^2$ $+ (a_{or}(B,S)_j - b_j)^2$ to each side gives

$$(a_{or}(B,S)_i - b_i)^2 - 2(a_{or}(B,S)_i - b_i) + (a_{or}(B,S)_j - b_j)^2$$
$$> (a_{or}(B,S)_i - b_i)^2 - 2(a_{or}(B,S)_j - b_j) + 2 + (a_{or}(B,S)_j - b_j)^2$$

or equivalently

$$\begin{aligned}
&(a_{or}(B,S)_i - b_i)^2 + (a_{or}(B,S)_j - b_j)^2 \\
&> (a_{or}(B,S)_i - b_i)^2 + 2(a_{or}(B,S)_i - b_i) + 1 \\
&\quad + (a_{or}(B,S)_j - b_j)^2 - 2(a_{or}(B,S)_j - b_j) + 1 \\
&= (a_{or}(B,S)_i - (b_i - 1))^2 + (a_{or}(B,S)_j - (b_j + 1))^2.
\end{aligned}$$

Thus, subtracting 1 bit from b_i and adding 1 bit to b_j reduces $\theta_1(b)$. Some algebra shows that the inequality in (4.50) is equivalent to

$$4^{a_{or}(B,S)_i - b_i} + 4^{a_{or}(B,S)_j - b_j} > 4^{a_{or}(B,S)_i - (b_i - 1)} + 4^{a_{or}(B,S)_j - (b_j + 1)}$$

from which it follows that $\theta_2(b)$ can be reduced by adding 1 bit to b_j and subtracting 1 bit from b_i. □

Proof of Lemma 4.20. The proof is trivial for $\|w\| = 0$, so assume $\|w\| > 0$. We determine the extrema of

$$\sum_{i=1}^{k} 4^{-w_i} \tag{4.51}$$

subject to the constraints

$$\sum_{i=1}^{k} w_i = 0 \tag{4.52}$$

$$\sum_{i=1}^{k} w_i^2 = a^2. \tag{4.53}$$

Define a Lagrangian J associated with multipliers λ_1 and λ_2 by:

$$J = \sum_{i=1}^{k} 4^{-w_i} + \lambda_1 \sum_{i=1}^{k} w_i + \lambda_2 \left(\sum_{i=1}^{k} w_i^2 - a^2 \right).$$

The extrema of J must satisfy (for $1 \leq i \leq k$):

$$0 = \frac{\partial J}{\partial w_i} = -(\ln 4) 4^{-w_i} + \lambda_1 + 2\lambda_2 w_i. \tag{4.54}$$

Suppose $\lambda_2 > 0$. Then $\frac{\partial J}{\partial w_i}$ is monotone increasing in w_i and approaches $\pm\infty$ as $w_i \longrightarrow \pm\infty$. Thus, exactly one w_i satisfies (4.54) for each i, and therefore $w_i = w_j$ for all i, j. So, by (4.52) it follows that $w_i = 0$ for all i, contradicting $\|w\| > 0$.

Thus we can assume $\lambda_2 < 0$. Since $\frac{\partial J}{\partial w_i}$ is strictly concave, (4.54) can have at most two solutions. It cannot be the case that (4.54) has only one solution, for otherwise (4.52) would again imply that $w_i = 0$ for all i, contradicting $\|w\| > 0$. So (4.54) has exactly two solutions and by (4.52) these two solutions must be of different signs.

Thus, the extrema of J must lie in the set

$$P = P_1 \cup \cdots \cup P_{k-1}$$

where P_j is the set of all $\binom{k}{j}$ component-wise permutations of the vector

$$a\left(\frac{j}{k(k-j)}\right)^{1/2} \cdot \left(\underbrace{-\left(\frac{k-j}{j}\right), \ldots, -\left(\frac{k-j}{j}\right)}_{j}, \underbrace{1, \ldots, 1}_{k-j} \right). \tag{4.55}$$

The constant factor in (4.55) ensures that the elements of P satisfy (4.52) and (4.53).

Summing both sides of (4.54) over i and solving for λ_1 yields

$$\lambda_1 = \frac{\ln 4}{k} \sum_{i=1}^{k} 4^{-w_i}. \tag{4.56}$$

From (4.54), we obtain

$$(\ln 4)4^{-w_i} - \lambda_1 = 2\lambda_2 w_i$$

which when squared, summed over i, and simplified using (4.53) and (4.56) gives

$$\lambda_2 = -\frac{\ln 2}{a}\left[\sum_{i=1}^{k} 16^{-w_i} - \frac{1}{k}\left(\sum_{i=1}^{k} 4^{-w_i}\right)^2\right]^{1/2}. \tag{4.57}$$

Now, for any component w_i of any $w \in P_j$, using (4.55), (4.56), and (4.57) gives

$$\begin{aligned}
&-(\ln 4)4^{-w_i} + \lambda_1 + 2\lambda_2 w_i \\
&= -(\ln 4)4^{-w_i} + \frac{\ln 4}{k}\sum_{i=1}^{k} 4^{-w_i} - \frac{2w_i \ln 2}{a}\left[\sum_{i=1}^{k} 16^{-w_i} - \frac{1}{k}\left(\sum_{i=1}^{k} 4^{-w_i}\right)^2\right]^{1/2} \\
&= -(\ln 4)4^{-w_i} + \frac{\ln 4}{k}\left(j4^{a\sqrt{(k-j)/(kj)}} + (k-j)4^{-a\sqrt{j/(k(k-j))}}\right)
\end{aligned}$$

$$
\begin{aligned}
&- \frac{w_i \ln 4}{a} \left[j16^{a\sqrt{(k-j)/(kj)}} + (k-j)16^{-a\sqrt{j/(k(k-j))}} \right. \\
&\left. - \frac{1}{k} \left(j4^{a\sqrt{(k-j)/(kj)}} + (k-j)4^{-a\sqrt{j/(k(k-j))}} \right)^2 \right]^{1/2} \\
&= -(\ln 4)4^{-w_i} + \frac{\ln 4}{k} \left(j4^{a\sqrt{(k-j)/(kj)}} + (k-j)4^{-a\sqrt{j/(k(k-j))}} \right) \\
&- \frac{w_i \ln 4}{a} \sqrt{\frac{j(k-j)}{k}} \left[4^{a\sqrt{(k-j)/(kj)}} - 4^{-a\sqrt{j/(k(k-j))}} \right] \\
&= 0 \qquad (4.58)
\end{aligned}
$$

where (4.58) follows by considering the cases $w_i = -a\sqrt{(k-j)/(kj)}$ and $w_i = a\sqrt{j/(k(k-j))}$. Hence every $w \in P$ satisfies (4.54), and therefore P is the set of solutions to (4.54) subject to the constraints in (4.52) and (4.53).

Substituting an arbitrary element $w \in P$ (i.e. an extremum of J) into (4.51) gives

$$
\begin{aligned}
\sum_{i=1}^{k} 4^{-w_i} &= j4^{a\sqrt{(k-j)/(kj)}} + (k-j)4^{-a\sqrt{j/(k(k-j))}} \\
&= j4^{\|w\|\sqrt{(k-j)/(kj)}} + (k-j)4^{-\|w\|\sqrt{j/(k(k-j))}} \qquad [\text{from (4.53)}]. \qquad (4.59)
\end{aligned}
$$

To complete the proof it suffices to show that (4.59) is decreasing in j. This implies (4.51) is upper bounded by (4.59) when $j = 1$ and lower bounded by (4.59) when $j = k - 1$.

Note that if the right-hand side of (4.59) is viewed as a continuous function of j, then its derivative with respect to j is

$$
\begin{aligned}
&4^{\|w\|\sqrt{(k-j)/(kj)}} \left[1 - \|w\| \ln 2 \left(\frac{k}{j(k-j)} \right)^{1/2} \right] \\
&- 4^{-\|w\|\sqrt{j/(k(k-j))}} \left[1 + \|w\| \ln 2 \left(\frac{k}{j(k-j)} \right)^{1/2} \right]
\end{aligned}
$$

which is negative if and only if $f\left(\|w\| \ln 2\sqrt{k/(j(k-j))}\right) > 0$, where

$$f(u) = 1 + u - (1-u)e^{2u}.$$

Since $f(0) = f'(0) = 0$ and $f''(u) = 4ue^{2u} > 0$ for all $u > 0$, we have $f(u) > 0$ for all $u > 0$. □

Proof of Lemma 4.21. The result follows from Lemma 4.2 and Lemma 4.20 with $w = b - a_{or}(B)$. □

Proof of Lemma 4.23. For any vector u and any permutation π of the positive integers less than or equal to the dimension of u, let $\pi(u)$ denote the component-wise permutation of u according to π. First observe that $\Phi_k\big(\pi(\gamma_k)\big) = \{0\}$ for any k and any permutation π of $\{1, \ldots, k+1\}$. To see this, note that for any $w \in \Lambda_k \setminus \{0\}$, since $|(\gamma_k)_i| < 1/2$ for all i, we have

$$|(\pi(\gamma_k))_i - w_i| > |(\pi(\gamma_k))_i| \text{ if } w_i \neq 0$$
$$|(\pi(\gamma_k))_i - w_i| = |(\pi(\gamma_k))_i| \text{ if } w_i = 0$$

which implies

$$\|\pi(\gamma_k) - w\| > \|\pi(\gamma_k)\|$$

and therefore

$$\Phi_k\big(\pi(\gamma_k)\big) = \left\{u \in \Lambda_k : \|\pi(\gamma_k) - u\| = \min_{v \in \Lambda_k} \|\pi(\gamma_k) - v\|\right\} = \{0\}.$$

Now observe that $\gamma_{k-1} - c(k-1, j)$ is the left-cyclic shift of γ_{k-1} by j positions, for any j, since

$$\gamma_{k-1} - c(k-1, j)$$

$$= \frac{1}{2k}\big(-(k-1)+2j, -(k-1)+2j+2, \ldots, k-1,$$
$$-(k-1), \ldots, -(k-1)+2j-4, -(k-1)+2j-2\big).$$

In particular, for each bit budget B, Theorem 4.10 and Lemma 4.4 imply that for every $b \in \mathcal{A}_{oi}(B)$,

$$\begin{aligned} b - a_{or}(B) &\in \Phi_{k-1}\big(\gamma_{k-1} - c(k-1, B \bmod k)\big) - \big(\gamma_{k-1} - c(k-1, B \bmod k)\big) \\ &= \Phi_{k-1}\big(\hat{\gamma}_{k-1}\big) - \hat{\gamma}_{k-1} \\ &= \{-\hat{\gamma}_{k-1}\} \end{aligned}$$

where $\hat{\gamma}_{k-1}$ denotes γ_{k-1} left-cyclic shifted by $B \bmod k$ positions. Since the components of γ_{k-1} are the same as those of $-\gamma_{k-1}$, so are the components of $-\hat{\gamma}_{k-1}$. Thus, $-\hat{\gamma}_{k-1}$ is a permutation of γ_{k-1}. □

This chapter, in full, has been submitted for publication as: Benjamin Farber and Kenneth Zeger, "Quantization of Multiple Sources Using Nonnegative Integer Bit Allocation," *IEEE Transactions on Information Theory*, May 2005. The dissertation author was the primary investigator of this paper.

References

[1] P. Batra and A. Eleftheriadis, "Alternative formulations for bit allocation with dependent quantization," in *Proceedings of the 2002 IEEE International Conference on Acoustics, Speech, and Signal Processing*, vol. 4, pp. 3505–3508, Orlando, Florida, May 2002.

[2] L. Breiman, J.H. Friedman, R.A. Olshen, and C.J. Stone, *Classification and Regression Trees. The Wadsworth Statistics / Probability Series*, Belmont, California: Wadsworth, 1984.

[3] A.M. Bruckstein, "On 'soft' bit allocation," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 35, no. 5, pp. 614–617, May 1987.

[4] J. Chen and D.W. Lin, "Optimal bit allocation for coding of video signals over ATM networks," *IEEE Journal on Selected Areas in Communications*, vol. 15, no. 6, pp. 1002–1015, August 1997.

[5] G. Cheung and A. Zakhor, "Bit allocation for joint source/channel coding of scalable video," *IEEE Transactions on Image Processing*, vol. 9, no. 3, pp. 340–356, March 2000.

[6] P.A. Chou, "Applications of Information Theory to Pattern Recognition and the Design of Decision Trees and Trellises," *Ph.D. Thesis*, Stanford University, 1988.

[7] P.A. Chou, T. Lookabaugh, and R. Gray, "Optimal pruning with applications to tree-structured source coding and modeling," *IEEE Transactions on Information Theory*, vol. 35, no. 2, pp. 299–315, March 1989.

[8] J.H. Conway and N.J.A. Sloane, *Sphere Packings, Lattices and Groups*, 3rd edition, Springer-Verlag, 1999.

[9] J.H. Conway and N.J.A. Sloane, "Fast quantizing and decoding algorithms for lattice quantizers and codes ," *IEEE Transactions on Information Theory*, vol. 28, no. 2, pp. 227–232, March 1982.

[10] J.J. Dubnowski and R.E. Crochiere, "Variable rate coding," in *Proceedings of the 1979 IEEE International Conference on Acoustics, Speech, and Signal Processing*, vol. 4, pp. 445–448, Washington, DC, April 1979.

[11] B. Fox, "Discrete optimization via marginal analysis," *Management Science*, vol. 13, no. 3, pp. 210–216, November, 1966.

[12] H. Gazzah and A.K. Khandani, "Optimum non-integer rate allocation using integer programming," *Electronics Letters*, vol. 33 , no. 24, pp. 2034, Nov. 20, 1997.

[13] A. Gersho and R. Gray, *Vector Quantization and Signal Compression*, Kluwer Academic Publishers, 1991.

[14] J.D. Gibson, "Bounds on performance and dynamic bit allocation for sub-band coders," in *Proceedings of the 1981 IEEE International Conference on Acoustics, Speech, and Signal Processing*, vol. 2, pp. 836–839, Atlanta, Georgia, March 1981.

[15] J.D. Gibson, "Notes on bit allocation in the time and frequency domains," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 33, no. 6, pp. 1609–1610, December 1985.

[16] L.M. Goodman, "Optimum rate allocation for encoding sets of analog messages," *Proceedings of the IEEE*, vol. 53, pp. 1776-1777, November 1965.

[17] L. Guo and Y. Meng, "Round-up of integer bit allocation," *Electronics Letters*, vol. 38 , no. 10, pp. 466-467, May 9, 2002.

[18] M. Honda and F. Itakura, "Bit allocation in time and frequency domains for predictive coding of speech," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 32, no. 3, pp. 465–473, June 1984.

[19] J.J. Huang and P.M. Schultheiss, "Block quantization of correlated Gaussian random variables," *IEEE Transactions on Communications Systems*, vol. 11, pp. 289–296, September 1963.

[20] A.E. Mohr, "Bit allocation in sub-linear time and the multiple-choice knapsack problem," in *Proceedings of the 2002 Data Compression Conference*, pp. 352–361, Snowbird, Utah, March 2002.

[21] A. Ortega, K. Ramchandran, and M. Vetterli, "Optimal trellis-based buffered compression and fast approximations," *IEEE Transactions on Image Processing*, vol. 3, no. 1, pp. 26–40, January 1994.

[22] K. Ramchandran, A. Ortega, and M. Vetterli, "Bit allocation for dependent quantization with applications to multiresolution and MPEG video coders," *IEEE Transactions on Image Processing*, vol. 3, no. 5, pp. 533–545, September 1994.

[23] T.A. Ramstad, "Considerations on quantization and dynamic bit-allocation in subband coders," in *Proceedings of the 1986 IEEE International Conference on Acoustics, Speech, and Signal Processing*, vol. 2, pp. 841–844, Tokyo, Japan, April 1986.

[24] E.A. Riskin, "Optimal bit allocation via the generalized BFOS algorithm," *IEEE Transactions on Information Theory*, vol. 37, no. 2, pp. 400–402, March 1991.

[25] A. Segall, "Bit allocation and encoding for vector sources," *IEEE Transactions on Information Theory*, vol. 22, no. 2, pp. 162–169, March 1976.

[26] Y. Sermadevi and S.S. Hemmami, “Efficient bit allocation for dependent video coding,” in *Proceedings of the 2004 Data Compression Conference*, pp. 232–241, Snowbird, Utah, March 2004.

[27] B. Schölkopf and A. Smola, *Learning with Kernels*, MIT Press, Cambridge, MA, 2002.

[28] Y. Shoham and A. Gersho, “Efficient bit allocation for an arbitrary set of quantizers,” *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 36, no. 9, pp. 1445–1453, September 1988.

[29] A.V. Trushkin, “Bit number distribution upon quantization of a multivariate random variable,” *Problems of Information Transmission*, vol. 16, no. 1, pp. 76–79, 1980 (translated from Russian).

[30] A.V. Trushkin, “Optimal bit allocation algorithm for quantizing a random vector,” *Problems of Information Transmission*, vol. 17, no. 3, pp. 156–161, 1981 (translated from Russian).

[31] P. H. Westerink, J. Biemond, and D. E. Boekee, “An optimal bit allocation algorithm for sub-band coding”, *International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, pp. 757–760, 1988.

[32] P.A. Wintz and A.J. Kurtenbach, “Waveform error control in PCM telemetry”, *IEEE Transactions on Information Theory*, vol. 14, no. 5, pp. 650–661, September 1968.

[33] J.W. Woods and S.D. O’Neil, “Subband coding of images,” *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 34, no. 5, pp. 1278–1288, October 1986.

[34] X. Wu, “Globally optimal bit allocation,” *Proceedings of the 1993 Data Compression Conference*, pp. 22–31, Snowbird, Utah, March 1993.

Chapter 5

Conclusion

In this dissertation we have presented work on two important problems having to do with scalar quantization. This study raises some addition interesting problems and ideas. We, therefore, conclude by discussing pertinent extensions of our work.

Chapter 2 deals exclusively with scalar quantization. However, one can also apply the weighted centroid condition to vector quantizer decoders. A challenging aspect of generalizing Chapter 2 to $k \geq 2$ dimensions is how to define a uniform quantizer encoder with a reasonable k-dimensional analog of the assumption that the encoding cells are ordered from left to right. With a well defined k-dimensional analog of a decoder optimized uniform quantizer, one can tackle the same questions that Chapter 2 answers:

1. For a given source, is there an optimal family of index assignments for all transmission rates and all bit error probabilities?

2. Are most index assignments asymptotically bad?

3. What do the point density functions look like for different families of index assignments?

Similarly, with a reasonable definition of a k-dimensional uniform quantizer decoder, one can generalize the notion of an encoder optimized uniform quantizer to $k \geq 2$

dimensions. Extending the work in Chapter 3 to higher dimensions could then involve answering questions such as:

1. Is the Natural Binary Code index assignment still sub-optimal for a large range of transmission rates and bit error probabilities?

2. How many empty cells do particular families of index assignments induce in the quantizer encoder?

3. What do the cell density functions look like for particular families of index assignments?

Another extension of the work in Chapter 3 would be to characterize the occurrence of empty cells and compute the expected mean squared error of an encoder optimized uniform quantizer with an index assignment chosen uniformly at random.

Both Chapters 2 and 3 restrict index assignments to be a permutation operation on the input to the channel and the inverse permutation operation on the output of the channel. A relaxed notion of index assignments would not require the permutation on the output of the channel to be the inverse of the permutation on the input to the channel. Both Chapters 2 and 3 also consider only binary symmetric channels. An interesting and challenging problem would be to consider any of the main results in Chapters 2 and 3 in the context of a different channel and/or a relaxed notion of index assignments.

One other interesting problem stemming from Chapters 2 and 3 is how well scalar quantizers can perform, and how they are structured when both the quantizer encoder and decoder are optimized to the channel statistics (i.e. when both the weighted centroid condition and the weighted nearest neighbor condition are satisfied). We found this to be a very difficult problem analytically, even for relatively small rate quantizers. One approach is to simplify the source so there are a finite number of possible quantizer encoders and decoders. This can be done by assuming the source consists of a discrete

random variable. Perhaps by making progress in this simplified case, one could gain understanding for the case of a continuous source random variable.

In Chapter 4 we assumed an individual quantizer with rate b achieves a mean squared error (MSE) proportional to 4^{-b}. However, we suspect our results might generalize to the case when the MSE is instead proportional to an arbitrary decreasing convex function of b. Two main issues must be resolved in order to make such a generalization. First of all, the optimal real-valued bit allocation derived by Huang and Schultheiss [1] must be generalized. Since their solution guarantees a unique optimal real-valued bit allocation, any generalization of their work would hopefully do the same. Secondly, one would need to find a way of relating the MSE achieved by an integer bit allocation to its component-wise difference from the generalized notion of an optimal real-valued bit allocation. Without such a relation, one cannot show that closest integer bit allocations are the same as optimal integer bit allocations. We relied on this fact to analyze the MSE of optimal integer bit allocations.

References

[1] J.J. Huang and P.M. Schultheiss, “Block quantization of correlated Gaussian random variables,” *IEEE Transactions on Communications Systems*, vol. 11, pp. 289–296, September 1963.