Study and Design of Globally Optimal Distributed Scalar Quantizer for Binary Linear Classification

Study and Design of Globally Optimal Distributed Scalar Quantizer for Binary Linear Classification

By Sara Zendehboodi, M.Sc.,

A Thesis Submitted to the School of Graduate Studies in the Partial Fulfillment of the Requirements for the Degree Doctor of Philosophy

McMaster University

Doctor of Philosophy (2024)

Hamilton, Ontario (Department of Electrical and Computer Engineering)

TITLE: Study and Design of Globally Optimal Distributed Scalar Quantizer for Binary Linear Classification

AUTHOR: Sara ZENDEHBOODI , M.Sc., (McMaster University)

SUPERVISOR: Dr. Sorina DUMITRESCU

NUMBER OF PAGES: xiv, 152

To Amirreza, my one and only.

Abstract

This thesis addresses the design of distributed scalar quantizers (DSQs) for two sensors, tailored to maximize the classification accuracy for a pre-trained binary linear classifier at the central node, diverging from traditional designs that prioritize data reconstruction quality.

The first contribution of this thesis is the development of efficient globally optimal DSQ design algorithms for two correlated discrete sources when the quantizer cells are assumed to be convex. First, it is shown that the problem is equivalent to a minimum weight path problem (with certain constraints) in a weighted directed acyclic graph. The latter problem can be solved using dynamic programming with $O(K_1K_2M^4)$ computational complexity, where K_i, is the number of cells for the quantizer of source i, $i = 1, 2$, and M is the size of the union of the sources' alphabets. Additionally, it is proved that the dynamic programming algorithm can be expedited by a factor of M by exploiting the so called Monge property, for scenarios where the pre-trained classifier is the optimal classifier for the unquantized sources.

Next, the design of so-called staggered DSQs (SDSQs) is addressed, i.e., DSQ's with $K_1 = K_2 = K$ and with the thresholds of the two quantizers being interleaved. First, a faster dynamic programming algorithm with only $O(KM^2)$ time complexity is devised for the design of the SDSQ that minimizes an upperbound on the classification error. This sped up is obtained by simplifying the graph model for the problem. Moreover, it is shown that this algorithm can also be further accelerated by a factor of M when the pre-trained linear classifier is the optimal classifier. Furthermore, some theoretical results are derived that provide support to imposing the above constraints to the DSQ

design problem in the case when the pre-trained classifier is optimal. First, it is shown that when the sources (discrete or continuous) satisfy a certain symmetry property, the SDSQ that minimizes the modified cost also minimizes the original cost within the class of DSQs without the staggerness constraint. For continuous sources, it is also shown that the SDSQ that minimizes the modified cost also minimizes the original cost and all quantizer thresholds are distinct, even if the sources do not satisfy the aforementioned symmetry condition. The latter result implies that DSQs with identical encoders are not optimal even when the sources has the same marginal distribution, a fact which is proved here for the first time, up to our knowledge.

The last (but not least) contribution of this thesis resides in leveraging the aforementioned results to obtain efficient globally optimal solution algorithms for the problem of decentralized detection under the probability of error criterion of two discrete vector sources that are conditionally independent given any class label. The previously known globally optimal solution has $O(N^{K_1+K_2+1})$ time complexity, where N is the size of the union of the alphabets of the two sources. We show that by applying an appropriate transformation to each vector source, the problem reduces to the problem of designing the optimal DSQ with convex cells in the transformed scalar domain for a scenario where the pre-trained linear classifier is the optimal classifier. We conclude that the problem can be solved by a much faster algorithm with only $O(K_1K_2N^3)$ time complexity. Similarly, for the case of equal quantizer rates, the problem can be solved in $O(KN)$ operations if the sources satisfy an additional symmetry condition. Furthermore, our results prove the conjecture that for continuous sources, imposing the constraint that the encoders be identical precludes optimality, even when the marginal distributions of the sources are the same.

Acknowledgements

I would like to extend my profound gratitude and utmost respect to Professor Sorina Dumitrescu. Her unwavering support, vast knowledge, and boundless patience have been the cornerstones of this journey, guiding me with grace and wisdom. Her invaluable guidance and dedication have brought this thesis to life.

I extend my heartfelt appreciation to my committee members, Dr. Jun Chen and Dr. Shahram Shirani, whose insightful guidance and thoughtful feedback have significantly enriched my research.

Moreover, I am deeply grateful to Ms. Cheryl Gies, the graduate administrative assistant of the Department of Electrical and Computer Engineering, for her indispensable help and support.

I must express my boundless gratitude to my parents for their unconditional love and support throughout these years. Last but not least, I thank my husband, Amirreza Mousavi, for his unwavering support and enduring patience throughout my PhD journey.

Contents

List of Figures

List of Tables

Abbreviations

DSQ	Distributed Scalar Quantizer
MS	Matrix Search
OC	Optimal Classifier Condition
OIE	Optimized Identical Encoders
ONIE	Optimized Non-Identical Encoders
ONIE10k	Optimized Non-Identical Encoders with 10k data
ONIE4k	Optimized Non-Identical Encoders with 4k data
pdf	probability density function
pmf	probability mass function
S	Symmetry Condition
SCS	Sufficient Condition for Symmetry
SDSQ	Staggered Distributed Scalar Quantizer
UL	Uniform Length
WDAG	Weighted Directed Acyclic Graph

Chapter 1

Introduction

Quantization plays a pivotal role in nearly all lossy data compression algorithms, serving to reduce the number of bits necessary for storage and communication. These techniques aim to optimize a rate-distortion trade-off, striving to represent data as accurately as possible with a limited number of bits. In contrast, this work introduces distributed quantization schemes specifically designed for data intended for classification purposes. We focus on creating distributed quantizers that balance the rate-classification error trade-off, optimizing not for reconstruction accuracy, but for correct classification. Intuitively, for data reconstruction, the objective is to represent regions with high signal concentration more precisely. For classification, however, the focus shifts to more finely representing areas near the decision boundary, where errors are more likely to occur.

In this thesis, our goal is to study and design distributed scalar quantizers for the following system. Two sensor nodes collect data, independently from one another. These sensors do not communicate with each other. The collected data is sent to a central node in which a known pre-trained linear classifier exists, for classification. The communication between the sensors and the central node is rate limited. Therefore, the collected

data cannot be sent to the server with full precision. Instead, each sensor node applies a scalar quantization step, independent from the other node, to encode its measurements into bit representations as efficiently as possible. As a result of this data compression, some information will be lost. The goal is to design quantizers at each node such that this information loss has the least negative effect on the classification results. The following section reviews some prior works and Section 1.2 describes our contribution and thesis organization.

1.1 Prior Work

The problem of distributed quantization was framed in a variety of scenarios, including traditional lossy multiterminal source coding [3, 53, 4, 37, 45, 58, 59, 57, 15], distributed source coding for functional computation [29, 34, 50, 8, 54, 56, 49], and statistical inference under multiterminal data compression [23, 22, 6, 55, 28, 7, 52]. In traditional multiterminal source coding, two or more correlated sources are encoded separately and sent to a joint decoder that aims at reconstructing all sources. The practical design of distributed quantizers for this task was addressed in [40, 18, 42, 60, 43, 65, 67, 39]. In distributed source coding for functional computation, the goal of the common decoder is to reconstruct a specified function of the sources.

The problem of task-based scalar quantizer design using neural networks is considered in [47, 48, 46]. In all of these works, the encoders and decoders are jointly trained using fully connected neural networks. The scalar quantizers are modelled as the activation functions of the last layer of the encoder neural network. The goal is to minimize the distortion between the parameter to be estimated and the true parameter value for all

examples of a training sequence. In all of these works, all quantizers are constrained to be identical.

Standard tasks in multiterminal inference problems are estimation of an unknown parameter that is dependent on the encoded sources and detection (or hypothesis testing). Algorithms for optimal design of distributed quantizers tailored for the estimation problem were proposed in [51, 19, 20, 30, 33, 38, 17, 24]. Decentralized detection was first considered in [51] for the case of two sensors and two hypotheses with equal encoder rates of 2. In all these works, except for [33], the joint probability distribution of the data is assumed to be known. Among the optimization objectives considered are minimizing the average distortion for a general distortion function [19] and minimizing the mean squared error [20, 30, 33, 24]. Other objectives are maximization of Fischer's information [30], maximization of the minimum asymptotic relative efficiency between the quantized and unquantized estimators [38] and the minimization of the trace of the Cramer-Rao bound matrix [17]. The authors of [30] also considered the minimization of the probability of estimation error (the same criterion as in our work), but their solution algorithm cannot ensure global optimality. In all of these works, an iterative algorithm that successively optimizes each encoder while keeping the other encoders fixed is used. This strategy is known as the "person-by-person" optimization strategy, which does not guarantee the globally optimal solution.

The problem of distributed quantizers design for binary hypothesis testing was addressed in [32, 2, 36, 52]. The authors of [32] propose a locally optimal algorithm to maximize the Bhattacharyya distance between the distributions of the space of interest under the two hypotheses. In [2] the cost to be minimized is an approximation of the probability of error, known as the saddlepoint approximation. While in [32, 2] the data

distribution is assumed to be known, in [36], only a training set is available. The authors of [36] assume that probabilistic decision rules are used at the encoders and consider the average loss combined with a regularization term as the cost function, where the loss function is a convex upper bound of the $0-1$ loss[1]. In [52], a general scenario, i.e., general number of sensors and of hypotheses, and general error functions was addressed. For the case of conditionally independent sources, Tsitsiklis derived necessary optimality conditions [52]. Based on these, he proposed a person-by-person algorithm. Tsitsiklis emphasized that the person-by-person algorithm does not guarantee convergence to a globally optimal solution. For the scenario of two binary hypotheses, conditionally independent sources and the probability of error criterion (i.e., our scenario), Tsitsiklis noted that when the sources are discrete, an exhaustive search over all possible threshold rules solves the problem globally optimally, but this algorithm is computationally extensive.

The problem of Distributed Scalar Quantizer (DSQ) design for a classification task is also considered in [16, 25]. The authors of [16] address the problem of designing a DSQ that minimizes the mismatch between the classifier applied to the quantized training data versus unquantized data, but impose a constraint on the structure of the DSQ. In particular, in the case when the rates of the two encoders are equal, the constraint corresponds to considering identical encoders. The proposed solution for this scenario is a greedy design algorithm. The problem studied in [25] is the problem of practical distributed quantizer design for a general classification task with the goal of minimizing the training misclassification rate. A general number of features and of sensors that collect these features is considered. The authors of [25] prove that the problem is NP-hard in the general case and propose a greedy algorithm of polynomial time complexity.

[1]Note that the expected $0-1$ loss equals the probability of error.

Furthermore, for the case of two features, two sensors and linearly separable data, they provide an efficient algorithm for the optimal DSQ design within the class of DSQs with identical encoders. Note that the latter algorithm also provides an optimal solution to the constrained problem considered in [16].

1.2 Contribution and Thesis Organization

Previous research on quantizer design primarily focused on achieving high-quality reconstruction of original samples. However, in our context, precise reconstruction of all samples is not crucial, as the output of the DSQ is used solely for a classification task. The design process could be more efficient if the optimization criterion were tailored specifically to the classification task. Thus, in this thesis we aim to minimize the error of the classifier applied to the quantized output.

The scenario we consider involves two sensor nodes collecting data independently. Each sensor node quantizes its data separately before sending it to a server node, which hosts a known linear binary classifier. Our objective is to design the quantizers in a way that minimizes the classifier's error on the quantized output. To reduce quantization delay, we focus on the use of scalar quantizers. In addition, we assume that each quantizer has convex cells (or bins, or regions), i.e., each quantization region is an interval of the real line (or the intersection of the source alphabet with an interval of the real line).

The assumption of a pre-trained classifier is practical in situations where the communication channel constraints are unknown during classifier design, allowing the same classifier to be used across systems with varying communication channels.

The selection of a linear classifier is also optimal in many scenarios. For linearly separable data or for cases where the joint conditional probability of the sources given each class label is Gaussian with equal variances, the optimal classifier is linear. Additionally, some scenarios allow for separate data transformation on each source such that the optimal decision boundary becomes linear in the transformed domain, even if it is not linear in the original domain. An example is the detection problem described in [52], when the sources are conditionally independent given the class label. In this case, a transformation related to the likelihood ratio results in an optimal linear decision boundary in the transformed domain, without requiring any specific conditions or knowledge about the optimal decision boundary in the original domain.

Up to our knowledge, there is no efficient solution algorithm available that guarantees optimality, even for the case of two sensors and linear binary classifier, without constraining the classes to be linearly separable and imposing identical encoders [16, 25, 14]. In Tsitsiklis' work [52], the proposed optimal solution for conditionally independent sources given the class label is computationally expensive and a person-by-person approach is taken for efficient quantizer design, which does not guarantee global optimality. In this thesis, we first address the problem in the general case, then we study some special scenarios, including the case when the given linear classifier is optimal, the case of equal quantizer rates, symmetric data distribution, identical encoders, and conditionally independent sources given the class label.

In particular, in Chapter 2, we study the problem of optimal design of DSQs with two correlated sources and for a known linear binary classifier present at the decoder. In contrast to prior works, we do not impose any condition on the data distribution, linear separability, optimality of the linear classifier, or structure of the encoders. The only

assumption is that the quantizer cells are convex. For the case of discrete sources, we establish that optimizing the DSQ equates to solving a minimum weight path problem, subject to constraints on the number of vertices of certain types, in a weighted directed acyclic graph (WDAG). We devise a dynamic programming solution algorithm that operates with a time complexity of $O(K_1K_2M^4)$ where K_i is the number of quantizer cells at encoder $i, i = 1, 2$, and M is the size of the union of the sources' alphabets. Next, we showcase a significant acceleration of the proposed algorithm, achieving a speedup by a factor of M through leveraging a fast matrix search technique in matrices with the Monge property, for the case where the given linear classifier is the optimal classifier. Furthermore, we emphasize the adaptability of this algorithm to situations where the source distribution is continuous or only a training sequence is available. Some of these results were presented in [14] and [68]. It is important to note that in [14], the goal is to minimize the mismatch between the classifier applied to the unquantized data and the classifier applied to the quantized data. Thus, the cost is different from the cost considered in this thesis unless the linear classifier separates the two classes completely. Therefore, the algorithm proposed in [14] does not guarantee the minimization of the classification error unless the data is separable by the linear classifier. Nevertheless, a similar algorithm can be used to minimize the classification error, as will be shown, since the graph model used there can be translated to the scenario of this thesis.

Chapter 3 explores the characteristics of the optimal DSQ for the equal-rate scenario, i.e., where $K_1 = K_2 = K$, with the aim of streamlining the design process. In several prior works addressing this case, the encoders are constrained to be identical [16, 25, 47]. In this chapter, we consider continuous sources for which the optimal classifier is

linear. We show that for such sources, the identical-encoders constraint is highly restrictive. Our approach involves investigating staggered DSQ (SDSQ) designs, i.e., where the threshold values of the two encoders are interleaved. We prove that the optimal SDSQ must be strict, i.e., the two encoders cannot have any identical thresholds. In addition, we demonstrate that the optimal SDSQ also minimizes an upper bound on the classification error that is easier to compute. Furthermore, we establish that for distributions that additionally satisfy a certain symmetry condition, the globally optimal DSQ must be strictly staggered.

Given the results of Chapter 3 for the case of equal quantizer rates for continuous sources, in Chapter 4, we establish similar properties that hold when the sources are discrete. Specifically, we show that in the case where the linear classifier is the optimal classifier for the unquantized data and the data distribution satisfies a certain symmetry condition, the optimal SDSQ is also optimal among all DSQs. Further, we address the problem of minimizing the modified cost that is the upper bound on the probability of error considered in the previous chapter. This strategic choice significantly simplifies the graph model, reducing the solution algorithm's time complexity to $O(KM^2)$. Furthermore, we prove that the Monge property holds for sources for which the optimal classifier is linear, facilitating a reduction in time complexity by an order of M. Moreover, we propose a globally optimal solution for the case where encoders have identical thresholds. Experimental results with training data illustrate the superiority of the designed SDSQs with this modified cost over prior work that assume identical encoders, despite our focus on minimizing an upper bound of the error.

Chapter 5 extends the discussions from Chapters 2 and 4 by exploring a practical application for the proposed designs. Specifically, we consider the detection problem of

[52] in which the sources are vectors, are discrete and conditionally independent given the class label. Note that for this case, it was shown in [52] that the optimal encoders are scalar quantizers with convex cells in the likelihood ratio domain. Therefore, an exhaustive search over all possible threshold rules solves the problem globally optimally as observed by Tsitsiklis [52]. Up to our knowledge, the aforementioned exhaustive search is the fastest globally optimal algorithm known so far, but it is very inefficient since it requires $N^{K_1+K_2}(N + K_1K_2)$ operations, where N is the size of the union of the alphabets of the input vectors. We propose a considerably faster globally optimal solution with time complexity $O(K_1K_2N^3)$. To achieve this, we show that the problem can be mapped to the DSQ design problem of Chapter 2 in a transformed space, and therefore can be solved with a similar solution algorithm. This result was presented in [68]. Moreover, we show that the properties discussed in Chapter 4 hold in the transformed space for the case of encoders with equal rates. We further derive a sufficient condition on the vectors sources that guarantees that the symmetry property introduced in Chapter 3 holds in the transformed domain and provide some examples.

Finally, Chapter 6 concludes the thesis and discusses some directions for future work.

Chapter 2

Globally Optimal Design of DSQ for Binary Linear Classification. General Problem

In this chapter, we study the problem of designing an optimal (K_1, K_2)-level distributed scalar quantizer, where K_k denotes the number of quantizer regions of the encoder at node k, for $k = 1, 2$, such that the error of the given classifier applied to the quantized output is minimized. We consider the case of two scalar and correlated sources, two classes, and a given linear classifier at the decoder. We assume that the quantizer cells are convex. We consider the scenario in which the joint distribution of the data and labels is known and discrete. We prove that the problem of optimal DSQ design is equivalent to a minimum weight path problem with some constraints on the number of vertices of certain types in a certain weighted directed acyclic graph (WDAG). The proposed solution algorithm has time complexity $O(K_1 K_2 M^4)$, where M is the size of the union of the alphabets of the two sources. Next, we demonstrate that the proposed

dynamic programming algorithm can be sped up by a factor of M by exploiting the so-called Monge property, if the given linear classifier is the optimal classifier. Moreover, we show how the proposed algorithm can be adopted in the scenario where the joint distribution of the data in each class is known and continuous as well as the scenario where the source distribution is not known but a training sequence is available.

This chapter is organized into seven sections. Section 2.1 describes the general framework and introduces the notations. Section 2.2 formulates the optimization problem. Section 2.3 shows that the problem of optimizing the DSQ is equivalent to a constrained shortest path problem in a specific WDAG and describes the graph model. In Section 2.4, the solution algorithm for the optimal DSQ design is proposed. Section 2.5 demonstrates that the Monge property is satisfied and can be used to speed up the solution algorithm when the linear classifier is the optimal classifier. Section 2.6 discusses the adaptation of the algorithm to the cases when the sources are continuous or when the distribution is unknown but a training sequence is available instead. Finally, Section 2.7 concludes the chapter.

2.1 Notations and Problem Description

Consider two distributed sensor nodes as in Fig. 2.1. The data collected by each node is quantized using a scalar quantizer with convex cells. Let Q_k denote the quantizer at node k, let α_k denote the encoder of Q_k and let R_k denote the rate of Q_k, for $k = 1, 2$. Then $\alpha_k : \mathbb{R} \to \{1, 2, \ldots, 2^{R_k}\}$, for $k = 1, 2$. The messages $m_k = \alpha_k(x_k)$ transmitted by the sensor nodes are received at a server node, where they are decoded using the decoding mapping $\beta : \{1, 2, \ldots, 2^{R_1}\} \times \{1, 2, \ldots, 2^{R_2}\} \to \mathbb{R}^2$. Let x_k denote the input to the encoder at node k, for $k = 1, 2$, and $\mathbf{x} = (x_1, x_2)$. The output $\hat{\mathbf{x}}$ of the decoder

is further used as input to a linear binary classifier $\gamma : \mathbb{R}^2 \to \{1, -1\}$. We assume that the joint probability distribution of the sources and labels is known and that the linear classifier γ is given. Note that the system described by the triple $(\alpha_1, \alpha_2, \beta)$ forms a distributed scalar quantizer. We will also use the notation $\mathbf{Q} = (Q_1, Q_2)$ for this DSQ and will say that $\mathbf{Q}$ is a (K_1, K_2)-level DSQ, where $K_1 = 2^{R_1}$ and $K_2 = 2^{R_2}$.

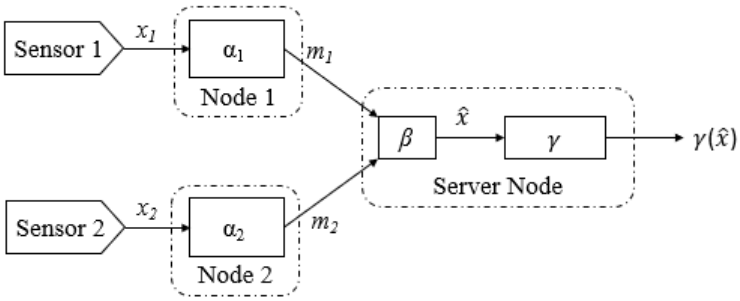


FIGURE 2.1: System block diagram.

Without restricting the generality, we assume that the decision boundary of the linear classifier obeys the equation $x_1 - x_2 = 0$. Note that any affine equation $ax_1 + bx_2 + c = 0$ with $a \neq 0$ and $b \neq 0$ can be reduced to the above after applying an affine transformation to each variable[1].

Recall that, for $k = 1, 2$, $K_k = 2^{R_k}$. It follows that the encoder of Q_k partitions the real line into K_k intervals called cells or bins. All elements in the same bin are assigned the same index by the encoder. Let us denote $U_i = \alpha_1^{-1}(i)$, for $1 \leq i \leq K_1$ and $V_j = \alpha_2^{-1}(j)$, for $1 \leq j \leq K_2$. Since each cell of Q_1 is an interval, the encoder partition of Q_1 is completely specified by the separators between cells. Consider the $(K_1 - 1)$-tuple obtained by ordering these separators in increasing order. We denote it by $\tilde{\mathbf{u}} = (\tilde{u}_1, \cdots, \tilde{u}_{K_1 - 1})$, where $-\infty = \tilde{u}_0 < \tilde{u}_1 < \cdots < \tilde{u}_{K_1 - 1} < \tilde{u}_{K_1} = \infty$,

[1] If either $a = 0$ or $b = 0$, then the output of the classifier depends only on one the inputs, therefore only the data collected at one of the sending nodes is needed at the server.

and $U_j = [\tilde{u}_{j-1}, \tilde{u}_j)$, for $1 \leq j \leq K_1$. We will refer to the components of $\tilde{\mathbf{u}}$ as the thresholds of quantizer Q_1. Likewise, the encoder partition of Q_2 is determined by the ordered $(K_2 - 1)$-tuple of integers $\tilde{\mathbf{v}} = (\tilde{v}_1, \cdots, \tilde{v}_{K_2-1})$, where $-\infty = \tilde{v}_0 < \tilde{v}_1 < \tilde{v}_2 < \cdots < \tilde{v}_{K_2-1} < v_{K_2} = \infty$, and $V_k = [\tilde{v}_{k-1}, \tilde{v}_k)$, for $1 \leq k \leq K_2$. The components of $\tilde{\mathbf{v}}$ are called the thresholds of quantizer Q_2. Recall that the DSQ $\mathbf{Q}$ is said to be a (K_1, K_2)-level DSQ.

The DSQ $\mathbf{Q} = (Q_1, Q_2)$ induces a partition of the set of the two-dimensional vectors $\mathbf{x} = (x_1, x_2)$ into the rectangular regions $U_i \times V_j$ for all $1 \leq i \leq K_1$, $1 \leq j \leq K_2$. They can be regarded as the bins of the product quantizer $Q_1 \times Q_2$. Note that for each region $U_i \times V_j$, all vectors $\mathbf{x}$ belonging to it are assigned to the same class, namely $\gamma(\beta(i, j))$, where $\beta(i, j) \in U_i \times V_j$.

We will use the term *classifier line* for the decision boundary of the linear classifier γ and we will denote it by $\mathcal{L}$. We also denote by $\mathcal{H}_-$ the closed half-plane below $\mathcal{L}$ and by $\mathcal{H}_+$ the closed half-plane above $\mathcal{L}$. In other words, $\mathcal{L} = \{(x_1, x_2) : x_1 = x_2\}$, $\mathcal{H}_- = \{(x_1, x_2) : x_1 \geq x_2\}$ and $\mathcal{H}_+ = \{(x_1, x_2) : x_1 \leq x_2\}$. We further assume that

$$\gamma(\mathbf{x}) = \begin{cases} 1 & \text{if } \mathbf{x} \in \mathcal{H}_+ \\ -1 & \text{if } \mathbf{x} \in \breve{\mathcal{H}}_- \end{cases},$$

where $\breve{\mathcal{H}}_- = \mathcal{H}_- \setminus \mathcal{L}$.

Then the problem we seek to solve is to find the triple $(\alpha_1, \alpha_2, \beta)$ that minimizes the probability of classification error, i.e.,

$$\min_{(\alpha_1,\alpha_2,\beta)} Pr(\gamma(\hat{\mathbf{x}}) \neq \ell) \tag{2.1}$$

where $Pr(E)$ denotes the probability of the event E, $\hat{\mathbf{x}} = \beta(\alpha_1(x_1), \alpha_2(x_2))$ and $\beta(i,j) \in \alpha_1^{-1}(i) \times \alpha_2^{-1}(j)$ for all (i,j). Throughout this thesis, we will refer to this problem as the optimal DSQ design problem.

2.2 Equivalent Problem Formulation

In this section, we propose an equivalent formulation of the optimization problem, which facilitates the solution algorithm. We will denote by $\mathcal{X}$ the set where $\mathbf{x}$ takes values in, where $\mathcal{X} \subset \mathbb{R}^2$. For $\mathbf{x} \in \mathcal{X}$ and $\ell \in \{1, -1\}$, we denote by $P(\mathbf{x}, \ell)$ the joint probability mass function (pmf) of $\mathbf{x}$ and label ℓ when the sources are discrete, and by $f(\mathbf{x}, \ell)$ we denote the joint probability density function (pdf) when the sources are continuous, where $f(\mathbf{x}, \ell)$ is a continuous function in $\mathbf{x}$.

Note the definitions introduced in this sections apply to both cases when the sources are discrete or continuous. At times, we will spell out the definitions for the discrete case. The corresponding relations for the continuous case can be obtained by replacing the pmf by the pdf, and summations over $\mathbf{x}$ by integrals.

The expected error of the DSQ is defined as the expected classification error as follows.

$$\rho(\mathbf{Q}) = Pr\{\gamma(\hat{\mathbf{x}}) \neq \ell\} = \sum_{\ell \in \{-1,1\}} \sum_{\mathbf{x} \in \mathcal{X}} P(\mathbf{x}, \ell) I(\gamma(\hat{\mathbf{x}}) \neq \ell), \tag{2.2}$$

where $I(\cdot)$ denotes the indicator function.

Additionally, we denote by ρ_u the probability of error of the classifier applied to the unquantized input. In other words,

$$\rho_u = Pr\{\gamma(\mathbf{x}) \neq \ell\} = \sum_{\ell \in \{-1,1\}} \sum_{\mathbf{x} \in \mathcal{X}} P(\mathbf{x}, \ell) I(\gamma(\mathbf{x}) \neq \ell). \tag{2.3}$$

The first crucial observation that enables our approach is that since γ is given, it follows that ρ_u is fixed, therefore, in order to minimize $\rho(\mathbf{Q})$ defined in (2.2), it is sufficient to minimize $\rho(\mathbf{Q}) - \rho_u$. Note that

$$\rho(\mathbf{Q}) = Pr\{\gamma(\hat{\mathbf{x}}) \neq \ell \,\&\, \gamma(\mathbf{x}) \neq \ell\} + Pr\{\gamma(\hat{\mathbf{x}}) \neq \ell \,\&\, \gamma(\mathbf{x}) = \ell\}, \tag{2.4}$$

and

$$\rho_u = Pr\{\gamma(\mathbf{x}) \neq \ell \,\&\, \gamma(\hat{\mathbf{x}}) \neq \ell\} + Pr\{\gamma(\mathbf{x}) \neq \ell \,\&\, \gamma(\hat{\mathbf{x}}) = \ell\}. \tag{2.5}$$

Relations (2.4) and (2.5) imply that

$$\rho(\mathbf{Q}) - \rho_u = Pr\{\gamma(\hat{\mathbf{x}}) \neq \ell \,\&\, \gamma(\mathbf{x}) = \ell\} - Pr\{\gamma(\mathbf{x}) \neq \ell \,\&\, \gamma(\hat{\mathbf{x}}) = \ell\}. \tag{2.6}$$

Further, we associate to each measurable set $A \in \mathbb{R}^2$ the following value

$$\begin{aligned} c(A) = Pr\{\mathbf{x} \in A \,\&\, \gamma(\hat{\mathbf{x}}) \neq \ell \,\&\, \gamma(\mathbf{x}) = \ell\} \\ - Pr\{\mathbf{x} \in A \,\&\, \gamma(\mathbf{x}) \neq \ell \,\&\, \gamma(\hat{\mathbf{x}}) = \ell\}. \end{aligned} \tag{2.7}$$

Using the notation

$$c(\mathbf{Q}) = \sum_{i=1}^{K_1} \sum_{j=1}^{K_2} c\left(U_i \times V_j\right),$$

we obtain, according to (2.6) and (2.7), that $\rho(\mathbf{Q}) = c(\mathbf{Q}) + \rho_u$. We conclude that minimizing $\rho(\mathbf{Q})$ is equivalent to minimizing $c(\mathbf{Q})$.

Recall our assumption that $\beta(i,j) \in U_i \times V_j$. This implies that if $U_i \times V_j \subset \mathcal{H}_+$ or $U_i \times V_j \subset \breve{\mathcal{H}}_-$, then $\gamma(\mathbf{x}) = \gamma(\beta(i,j)) = \gamma(\hat{\mathbf{x}})$ for all $\mathbf{x} \in U_i \times V_j$, thus both terms in (2.7) are 0, which leads to $c\left(U_i \times V_j\right) = 0$.

For the quantization regions $U_i \times V_j$ that are not entirely included in $\mathcal{H}_+$ or in $\breve{\mathcal{H}}_-$, the value of $c(U_i \times V_j)$ depends on whether $\beta(i,j)$ is in $\mathcal{H}_+$ or in $\breve{\mathcal{H}}_-$. Before analyzing the two options, we need to introduce some more notations. For any measurable set A satisfying $A \subseteq \mathcal{H}_+$ or $A \subseteq \mathcal{H}_-$, define $\mu(A)$ as follows:

a) in the discrete case

$$\mu(A) = \begin{cases} \sum_{\mathbf{x} \in A} (P(\mathbf{x}, 1) - P(\mathbf{x}, -1)) & \text{if } A \subseteq \mathcal{H}_+ \\ \sum_{\mathbf{x} \in A} (P(\mathbf{x}, -1) - P(\mathbf{x}, 1)) & \text{if } A \subseteq \mathcal{H}_- \end{cases},$$

b) in the continuous case

$$\mu(A) = \begin{cases} \int_A (f(\mathbf{x},1) - f(\mathbf{x},-1))\, d\mathbf{x} & \text{if } A \subseteq \mathcal{H}_+ \\ \int_A (f(\mathbf{x},-1) - f(\mathbf{x},1))\, d\mathbf{x} & \text{if } A \subseteq \mathcal{H}_- \end{cases}.$$

Clearly, $\mu(\emptyset) = 0$. Note that in the discrete case $\mu(A) = \mu(\bar{A})$, where $\bar{A}$ denotes the closure of A as $[a_1, a_2) \times [a_3, a_4)$. In addition, the relation $\mu(A) = 0$ holds in the continuous case whenever $area(A) = 0$, where $area(A) = \int_A 1\, d\mathbf{x}$.

Next, by considering the values 1 and -1 for ℓ, we rewrite (2.7) as follows

$$\begin{aligned} c(A) = & Pr\{\mathbf{x} \in A \,\&\, \gamma(\hat{\mathbf{x}}) \neq \ell \,\&\, \gamma(\mathbf{x}) = \ell \,\&\, \ell = 1\} \\ & + Pr\{\mathbf{x} \in A \,\&\, \gamma(\hat{\mathbf{x}}) \neq \ell \,\&\, \gamma(\mathbf{x}) = \ell \,\&\, \ell = -1\} \\ & - Pr\{\mathbf{x} \in A \,\&\, \gamma(\mathbf{x}) \neq \ell \,\&\, \gamma(\hat{\mathbf{x}}) = \ell \,\&\, \ell = 1\} \\ & - Pr\{\mathbf{x} \in A \,\&\, \gamma(\mathbf{x}) \neq \ell \,\&\, \gamma(\hat{\mathbf{x}}) = \ell \,\&\, \ell = -1\}. \end{aligned}$$

Further, if $\beta(i,j) \in \mathcal{H}_+$, then $\gamma(\hat{\mathbf{x}}) = 1$ for $\mathbf{x} \in U_i \times V_j$ leading to

$$\begin{aligned} c(U_i \times V_j) &= Pr\{\mathbf{x} \in U_i \times V_j \cap \breve{\mathcal{H}}_- \,\&\, \ell = -1\} - Pr\{\mathbf{x} \in U_i \times V_j \cap \breve{\mathcal{H}}_- \,\&\, \ell = 1\} \\ &= \sum_{\mathbf{x} \in U_i \times V_j \cap \breve{\mathcal{H}}_-} (P(\mathbf{x},-1) - P(\mathbf{x},1)) = \mu(U_i \times V_j \cap \breve{\mathcal{H}}_-). \end{aligned}$$

On the other hand, if $\beta(i,j) \in \breve{\mathcal{H}}_-$, then $\gamma(\hat{\mathbf{x}}) = -1$ for all $\mathbf{x} \in U_i \times V_j$ implying that

$$\begin{aligned} c(U_i \times V_j) &= Pr\{\mathbf{x} \in U_i \times V_j \cap \mathcal{H}_+ \ \& \ \ell = 1\} - Pr\{\mathbf{x} \in U_i \times V_j \cap \mathcal{H}_+ \ \& \ \ell = -1\} \\ &= \sum_{\mathbf{x} \in U_i \times V_j \cap \mathcal{H}_+} (P(\mathbf{x}, 1) - P(\mathbf{x}, -1)) = \mu(U_i \times V_j \cap \mathcal{H}_+). \end{aligned}$$

Note that for the case of continuous sources, similar relations hold by replacing pmf by pdf and summations on $\mathbf{x}$ by integrals.

When $U_i \times V_j \cap \breve{\mathcal{H}}_- \neq \emptyset$ and $U_i \times V_j \cap \mathcal{H}_+ \neq \emptyset$, we choose $\beta(i,j)$ such that $c(U_i \times V_j)$ is minimized. Therefore, for any $S = U_i \times V_j$, $c(S)$ satisfies the following relation

$$c(S) = \begin{cases} 0 & \text{if } S \subset \breve{\mathcal{H}}_- \text{ or } S \subset \mathcal{H}_+ \\ \min\{\mu(S \cap \mathcal{H}_+), \mu(S \cap \breve{\mathcal{H}}_-)\} & \text{if } S \cap \breve{\mathcal{H}}_- \neq \emptyset \text{ and } S \cap \mathcal{H}_+ \neq \emptyset \end{cases}.$$

Note that since both U_i and V_j are intervals and V_j does not contain its right boundary, it follows that whenever $U_i \times V_j \cap \mathcal{H}_+ \neq \emptyset$, we have $U_i \times V_j \cap \breve{\mathcal{H}}_+ \neq \emptyset$, where $\breve{\mathcal{H}}_+ = \mathcal{H}_+ \setminus \mathcal{L}$. Consequently, only the regions $U_i \times V_j$ that satisfy $U_i \times V_j \cap \breve{\mathcal{H}}_- \neq \emptyset$ and $U_i \times V_j \cap \breve{\mathcal{H}}_+ \neq \emptyset$ can contribute to $c(\mathbf{Q})$. We will call these cells of the product quantizer the *relevant cells* of the DSQ.

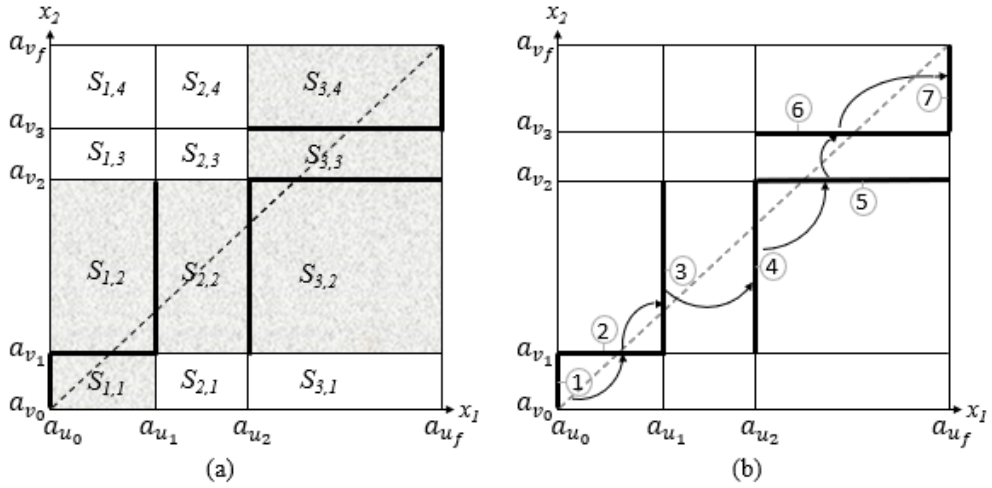


FIGURE 2.2: Example of a DSQ: a) partition of $Q_1 \times Q_2$ (each cell $U_i \times V_j$ is denoted by $S_{i,j}$); b) corresponding path in the associated WDAG; the thick segment lines represent the nodes on the path, while the arcs represent the edges.

2.3 Optimal DSQ Design Problem for Discrete Sources and Its Graph Model

In this section we consider the optimal DSQ design problem (2.1) for the case when the sources take values in a finite alphabet and will show that it can be modelled as a constrained shortest path problem in a certain WDAG. We will assume that the set $\mathcal{X}$ consist only of the pairs $\mathbf{x}$ for which either $P(\mathbf{x}, 1)$ or $P(\mathbf{x}, -1)$ is nonzero.

For $k = 1, 2$, let $\mathcal{A}_k$ denote the projection of $\mathcal{X}$ on the k-th axis. In other words, $\mathcal{A}_1 = \{x_1 : (x_1, x_2) \in \mathcal{X} \text{ for some } x_2\}$ and $\mathcal{A}_2 = \{x_2 : (x_1, x_2) \in \mathcal{X} \text{ for some } x_1\}$. It can be easily seen that it is sufficient to consider only the scalar quantizers Q_k that have as separators between cells only values from the set $\mathcal{A}_k$, for $k = 1, 2$.

Let M_k denote the size of $\mathcal{A}_k$, $k = 1, 2$. Further, denote $\mathcal{A} = \mathcal{A}_1 \cup \mathcal{A}_2$, $M = |\mathcal{A}|$ and let us label the elements of $\mathcal{A}$ in increasing order, i.e., $\mathcal{A} = \{a_1, \cdots, a_M\}$ where

$a_m < a_{m+1}$ for $1 \leq m \leq M-1$.

Also, denote $a_0 = -\infty$ and $a_{M+1} = \infty$. By convention $[-\infty, x) = (-\infty, x)$. Additionally, for $k = 1, 2$, let $\mathcal{M}_k = \{m : a_m \in \mathcal{A}_k\}$. Let $\mathcal{Q}(K_1, K_2, \mathcal{A}_1, \mathcal{A}_2)$ denote the set of (K_1, K_2)-level DSQs determined by all possible pairs $\tilde{\mathbf{u}}$, $\tilde{\mathbf{v}}$, where $\tilde{\mathbf{u}} = (\tilde{u}_1, \cdots, \tilde{u}_{K_1-1})$, $\tilde{\mathbf{v}} = (\tilde{v}_1, \cdots, \tilde{v}_{K_2-1})$, $\tilde{u}_i \in \mathcal{A}_1$ and $\tilde{v}_j \in \mathcal{A}_2$, for all i and j.

According to the discussion in the previous sections, minimizing $\rho(\mathbf{Q})$ is equivalent to minimizing $c(\mathbf{Q})$. Then the problem of optimal (K_1, K_2)-level DSQ design is equivalent to

$$\min_{\mathbf{Q} \in \mathcal{Q}(K_1, K_2, \mathcal{A}_1, \mathcal{A}_2)} c(\mathbf{Q}). \tag{2.8}$$

Since each threshold of Q_k is an element in the alphabet $\mathcal{A}_k$, $k = 1, 2$, we can identify it with the integer index corresponding to that element. For this, let u_i denote the integer such that $\tilde{u}_i = a_{u_i}$ and let v_j denote the integer such that $\tilde{v}_j = a_{v_j}$. The encoder partition of Q_1 is determined by the ordered (K_1+1)-tuple of integers $\mathbf{u} = (u_0, \cdots, u_{K_1})$ satisfying $0 = u_0 < u_1 < \cdots < u_{K_1-1} < u_{K_1} = M+1$, $u_j \in \mathcal{M}_1$, for $1 \leq j \leq K_1 - 1$, and $U_j = [a_{u_{j-1}}, a_{u_j})$, for $1 \leq j \leq K_1$. We will refer to the components of $\mathbf{u}$ as the integral thresholds of quantizer Q_1. Likewise, the encoder partition of Q_2 is determined by the ordered (K_2+1)-tuple of integers $\mathbf{v} = (v_0, \cdots, v_{K_2})$ satisfying $0 = v_0 < v_1 < \cdots < v_{K_2-1} < v_{K_2} = M+1$, $v_k \in \mathcal{M}_2$, for $1 \leq k \leq K_2 - 1$, and $V_k = [a_{v_{k-1}}, a_{v_k})$, for $1 \leq k \leq K_2$. The components of $\mathbf{v}$ are called the integral thresholds of quantizer Q_2.

Next, we will show that this problem is equivalent to the shortest path problem with

some constraints in a certain WDAG. We will first explain the intuition behind the graph construction. Our goal is to map the DSQ to a path in the WDAG such that the weight of the path (i.e., the sum of the weights of its edges) is equal to the cost of the DSQ (Fig. 2.2). On the other hand, the cost of the DSQ is equal to the sum of the costs of the relevant cells of $Q_1 \times Q_2$. Then the relevant cells can represent the edges in the path. This leads to the idea of associating each rectangular region that intersects $\mathcal{L}$ to an edge in the graph. The vertices (also called nodes) forming the edge are the two sides of the rectangle that intersect the line $\mathcal{L}$. In other words, the vertices of the graph correspond to horizontal and vertical segment lines that intersect $\mathcal{L}$. Then the vertices on the path corresponding to the DSQ are the vertices corresponding to the boundaries between the relevant cells of $Q_1 \times Q_2$. The path starts in the source node ν_0, visits the aforementioned vertices in increasing order of their intersection with $\mathcal{L}$ and ends in the final node ν_f. An illustration of a DSQ and its corresponding path is provided in Fig. 2.2(b).

Let us introduce now the WDAG formally: $G = (V, E, w)$ has $V = V_v \cup V_h \cup \{\nu_0, \nu_f\}$, where

$$V_h = \{(j,k,i)_h : 0 \leq j < k \leq i \leq M+1, j, i \in \mathcal{M}_1, k \in \mathcal{M}_2\},$$
$$V_v = \{(k,i,\ell)_v : 0 \leq k \leq i < \ell \leq M+1, i \in \mathcal{M}_1, k, \ell \in \mathcal{M}_2\}.$$

The node $(j,k,i)_h$ represents the set $[a_j, a_i) \times \{a_k\}$, i.e., the horizontal segment line connecting the points of coordinates (a_j, a_k) and (a_i, a_k). The node $(k,i,\ell)_v$ represents the set $\{a_i\} \times [a_k, a_\ell)$, i.e., the vertical segment line connecting the points of coordinates (a_i, a_k) and (a_i, a_ℓ). The nodes in V_v are called vertical nodes (e.g., the nodes labeled $1, 3, 4, 7$ in Fig. 2.2(b)) and the nodes in V_h are called horizontal nodes (e.g., the nodes

labeled $2, 5, 6$ in Fig. 2.2(b)). ν_0 is the source node and ν_f is the final node.

The set of edges is $E = E_{0h} \cup E_{0v} \cup E_{Mh} \cup E_{Mv} \cup E_{hh} \cup E_{hv} \cup E_{vh} \cup E_{vv}$, where

$$
\begin{aligned}
E_{0h} &= \{\nu_0 \to (0,k,i)_h : 1 \le k \le i \le M+1\},\\
E_{0v} &= \{\nu_0 \to (0,i,\ell)_v : 1 \le i < \ell \le M+1\},\\
E_{hf} &= \{(j,k,M+1)_h \to \nu_f : 0 \le j < k < M+1\},\\
E_{vf} &= \{(k,i,M+1)_v \to \nu_f : 0 \le k \le i < M+1\},\\
E_{hh} &= \{(j,k,i)_h \to (j,k',i)_h : 0 \le j < k < k' \le i \le M+1\},\\
E_{hv} &= \{(j,k,i)_h \to (k,i,\ell)_v : 0 \le j < k \le i < \ell \le M+1\},\\
E_{vh} &= \{(\ell,j,k)_v \to (j,k,i)_h : 0 \le \ell \le j < k \le i \le M+1\},\\
E_{vv} &= \{(k,i,\ell)_v \to (k,i',\ell)_v : 0 \le k \le i < i' < \ell \le M+1\}.
\end{aligned}
$$

Each edge $e \in E$ represents a possible relevant cell of $Q_1 \times Q_2$, namely, the rectangular region determined by the segment lines corresponding to the nodes connected by the edge. Let us denote by $\mathcal{R}(e)$ the set represented by edge e. Then

- for $e = (j,k,i)_h \to (j,k',i)_h$, $\mathcal{R}(e) = [a_j, a_i) \times [a_k, a_{k'})$ (e.g., cell $S_{3,3}$ in Fig. 2.2(a));
- for $e = (j,k,i)_h \to (k,i,\ell)_v$, $\mathcal{R}(e) = [a_j, a_i) \times [a_k, a_\ell)$ (e.g., cell $S_{1,2}$ in Fig. 2.2(a));
- for $e = (\ell,j,k)_v \to (j,k,i)_h$, $\mathcal{R}(e) = [a_j, a_i) \times [a_\ell, a_k)$ (e.g., cell $S_{3,2}$ in Fig. 2.2(a));

- for $e = (k, i, \ell)_v \rightarrow (k, i', \ell)_v$, $\mathcal{R}(e) = [a_i, a_{i'}) \times [a_k, a_\ell)$ (e.g., cell $S_{2,2}$ in Fig. 2.2(a));
- for $e = \nu_0 \rightarrow (0, k, i)_h$, $\mathcal{R}(e) = (-\infty, a_i) \times (-\infty, a_k)$ (e.g., cell $S_{1,1}$ in Fig. 2.2(a));
- for $e = \nu_0 \rightarrow (0, i, \ell)_v$, $\mathcal{R}(e) = (-\infty, a_i) \times (-\infty, a_\ell)$;
- for $e = (j, k, M+1)_h \rightarrow \nu_f$, $\mathcal{R}(e) = [a_j, \infty) \times [a_k, \infty)$ (e.g., cell $S_{3,4}$ in Fig. 2.2(a));
- for $e = (k, i, M+1)_v \rightarrow \nu_f$, $\mathcal{R}(e) = [a_i, \infty) \times [a_k, \infty)$.

Finally, for each edge $e \in E$, its weight is $w(e) = c(\mathcal{R}(e))$.

The key result of this section is the following proposition, whose proof is deferred to Appendix A.

Proposition 1. *There is a one-to-one mapping $\mathcal{P}$ from the set of all possible DSQs $\mathbf{Q}$ in $\mathcal{Q}(K_1, K_2, \mathcal{A}_1, \mathcal{A}_2)$ to the set of $(K_1 + K_2 - 1)$-edge paths from ν_0 to ν_f in G with $K_1 - 1$ vertical nodes and $K_2 - 1$ horizontal nodes such that $c(\mathbf{Q}) = w(\mathcal{P}(\mathbf{Q}))$.*

Corollary 1. *In virtue of Proposition 1, the problem (2.8) is equivalent to finding the path of minimum-weight among all $(K_1 + K_2 - 1)$-edge paths from ν_0 to ν_f in G that have $K_1 - 1$ vertical nodes and $K_2 - 1$ horizontal nodes.*

2.4 Solution Algorithm

This section presents the solution algorithm to problem (2.8). Proposition 1 indicates that the problem (2.8) can be solved by finding the path of minimum weight among all $(K_1 + K_2 - 1)$-edge paths in G from ν_0 to ν_f, with $K_1 - 1$ vertical nodes and $K_2 - 1$ horizontal nodes. In order to describe the algorithm, let us introduce a few notations. For each $0 \leq k_1 \leq K_1 - 1$, $0 \leq k_2 \leq K_2 - 1$ and $v \in V$, let $\hat{W}_{k_1,k_2}(v)$ denote the weight of the minimum-weight path from ν_0 to v with k_1 vertical nodes and k_2 horizontal nodes without counting v. Then the following recursive formula holds

$$\hat{W}_{k_1,k_2}(v) = \min(\hat{W}'_{k_1,k_2}(v), \hat{W}''_{k_1,k_2}(v)), \tag{2.9}$$

$$\hat{W}'_{k_1,k_2}(v) = \min_{u \in V_v, (u,v) \in E} (\hat{W}_{k_1-1,k_2}(u) + w(u,v)), \tag{2.10}$$

$$\hat{W}''_{k_1,k_2}(v) = \min_{u \in V_h, (u,v) \in E} (\hat{W}_{k_1,k_2-1}(u) + w(u,v)), \tag{2.11}$$

where $1 \leq k_1 \leq K_1 - 1$, $1 \leq k_2 \leq K_2 - 1$ and $v \in V$.

For many quantizer systems, for finite-alphabet sources, globally optimal design is possible using dynamic programming. This includes single description scalar quantizers [35, 44, 64, 66], Wyner-Ziv scalar quantizers [35, 69], sequential scalar quantizers [61, 27], successively refined and multiple description scalar quantizers [35, 11, 12, 13, 41, 21], successively refinable polar quantizers [62, 63], joint source-channel scalar quantizers with random index assignment [10], and in zero-delay coding of Markov sources [31]. The current problem can also be solved using a dynamic programming algorithm that computes $\hat{W}_{k_1,k_2}(v)$ for all $1 \leq k_1 \leq K_1 - 1$, $1 \leq k_2 \leq K_2 - 1$ and

$v \in V$ by processing the pairs k_1, k_2 in lexicographical order. The pseudo code for the solution algorithm is provided in Algorithm 1.

Note that if the weight of any edge can be calculated in constant time, then the time complexity of the solution algorithm is $O(K_1 K_2 M^4)$. For this requirement to be satisfied, we perform a preprocessing stage that will be described shortly.

For each pair $(\xi_1, \xi_2) \in \mathbb{R}^2$ with $\xi_1 < \xi_2$, let us denote by $T(\xi_1, \xi_2)$ the triangular region in $\mathcal{H}_+$ with vertices (ξ_1, ξ_1), (ξ_2, ξ_2) and (ξ_1, ξ_2), more specifically, $T(\xi_1, \xi_2) = \{(x_1, x_2) \in \mathbb{R}^2 : x_1 \leq x_2 < \xi_2\}$. Additionally, denote by $T'(\xi_1, \xi_2)$ the reflection of $T(\xi_1, \xi_2)$ across $\mathcal{L}$ from which the points in $\mathcal{L}$ have been removed, i.e., $T'(\xi_1, \xi_2) = \{(x_1, x_2) \in \mathbb{R}^2 : \xi_1 \leq x_2 < x_1 < \xi_2\}$.

In the preprocessing stage we calculate and store the values $\mu(T(a_m, a_n))$ and $\mu(T'(a_m, a_n))$ for all m, n, $0 \leq m < n \leq M + 1$. In order to compute the quantities $\mu(T(a_m, a_n))$, the elements $\mathbf{x}$ of $\mathcal{X} \cap \mathcal{H}_+$ are first stored in an array $\mathcal{B}$ in reverse lexicographical order along with their probabilities $P(\mathbf{x}, 1)$ and $P(\mathbf{x}, -1)$. Note that by the reverse lexicographic order of pairs (x_1, x_2), we understand the lexicographic order of the reversed pairs, i.e. of (x_2, x_1).

The calculation of the values $\mu(T(a_m, a_n))$ is performed in increasing order of m, $1 \leq m \leq M$, and for each m, in increasing order of $n, m < n \leq M$. Note that for each pair (m, n), the triangular region $T(a_m, a_n)$ is included in $T(a_m, a_{n+1})$. Therefore, $\mu(T(a_m, a_{n+1}))$ is computed by adding to $\mu(T(a_m, a_n))$ the value $\mu(A(m, n))$, where $A(m, n) = T(a_m, a_{n+1}) \setminus T(a_m, a_n)$. In other words, $A(m, n)$ consists of the points $\mathbf{x}$ with $x_2 = a_n$ and $a_m \leq x_1 < a_n$. These points are stored in consecutive positions in the array $\mathcal{B}$. In order to compute the values $\mu(A(m, n))$ for fixed m and all $n > m$, the

Algorithm 1. Solution to the minimum-weight $(K_1 + K_2 - 1)$-edge path problem with $K_1 - 1$ vertical and $K_2 - 1$ horizontal nodes.

begin
- $W_{0,0}(s) \leftarrow 0$
- **for** $k_1 = 1$ **to** $K_1 - 1$ **do**
  - $k_2 \leftarrow 0$
  - **for** $v \in V$ **do**
    - $W_{k_1,k_2}(v) \leftarrow \min_{u:(u,v)\in(E_{vv}\cup E_{vh})}(W_{k_1-1,k_2}(u) + w(u,v))$
    - $prev(k_1,k_2,v) \leftarrow \arg\min_{u:(u,v)\in(E_{vv}\cup E_{vh})}(W_{k_1-1,k_2}(u) + w(u,v))$
- **for** $k_2 = 1$ **to** $K_2 - 1$ **do**
  - $k_1 \leftarrow 0$
  - **for** $v \in V$ **do**
    - $W_{k_1,k_2}(v) \leftarrow \min_{u:(u,v)\in(E_{hv}\cup E_{hh})}(W_{k_1,k_2-1}(u) + w(u,v))$
    - $prev(k_1,k_2,v) \leftarrow \arg\min_{u:(u,v)\in(E_{hv}\cup E_{hh})}(W_{k_1,k_2-1}(u) + w(u,v))$
- **for** $k_1 = 1$ **to** $K_1 - 1$ **do**
  - **for** $k_2 = 1$ **to** $K_2 - 1$ **do**
    - **for** $v \in V$ **do**
      - $W_1(v) \leftarrow \min_{u:(u,v)\in(E_{vv}\cup E_{vh})}(W_{k_1-1,k_2}(u) + w(u,v))$
      - $W_2(v) \leftarrow \min_{u:(u,v)\in(E_{hv}\cup E_{hh})}(W_{k_1,k_2-1}(u) + w(u,v))$
      - $W_{k_1,k_2}(v) \leftarrow \min(W_1(v), W_2(v))$
      - $prev(k_1,k_2,v) \leftarrow \arg\min_{u:(u,v)\in E}(W_{k_1,k_2}(v))$
- Recover the best path from s to t using array $prev$

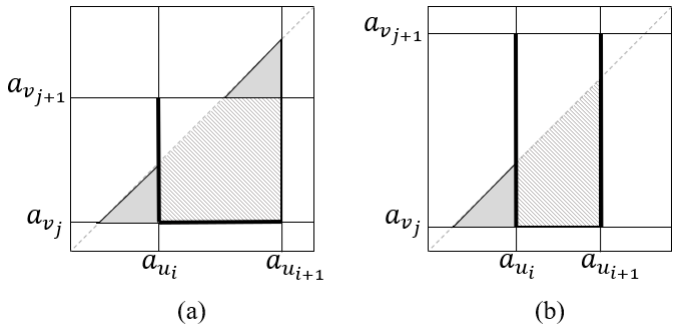


FIGURE 2.3: Illustration for the computation of $\mu(\mathcal{R}(e) \cap \breve{\mathcal{H}}_-)$; the striped region represents the set $\mathcal{R}(e) \cap \breve{\mathcal{H}}_-$. By appending to $\mathcal{R}(e) \cap \breve{\mathcal{H}}_-$ two triangular regions (a) or one triangular region (b), a bigger triangular region is obtained.

portion of the array $\mathcal{B}$ starting at the first occurrence of the element (a_m, a_m) is scanned only once. After finishing the computation of $T(a_m, a_n)$, this pointer is situated at the first element in the array with $x_2 = a_n$ (recall that pairs (x_1, x_2) with $x_2 = a_n$ are not included in $T(a_m, a_n)$). Then the pointer advances until reaching the first pair with $x_1 = a_m$. Here is where the block storing the alphabet points corresponding to $A(m, n)$ starts. While the block is scanned, $\mu(A(m, n))$ is updated correspondingly. The first element after the block ends is the first pair with $x_2 = a_{n+1}$. When the pointer reaches this position, the computation of $\mu(A(m, n))$ is finished. We conclude that the number of operations to compute all weights $\mu(T(a_m, a_n))$ for fixed m and all $n > m$ is $O(M)$, further amounting to $O(M^2)$ over all m. The computation of the values $\mu(T'(a_m, a_n))$ proceeds similarly, therefore the preprocessing takes $O(M^2)$ operations.

When the weight of an edge e is needed, each of the values $\mu(\mathcal{R}(e) \cap \mathcal{H}_+)$ and $\mu(\mathcal{R}(e) \cap \breve{\mathcal{H}}_-)$ is computed by subtracting the μ-value of a smaller triangle from that of a bigger triangle (as in Fig. 2.3(b)) or the μ-values of two smaller triangles from the μ-value of a bigger triangle (as in Fig. 2.3(a)).

2.5 Faster Solution Algorithm using the Monge Property

In this section, we show how the dynamic programming algorithm presented in Section 2.4 can be sped up by a factor of M by exploiting the so-called Monge property [5], in the case where the given linear classifier at the decoder is the optimal classifier for the unquantized data. This is to say that for discrete sources, the joint probability distribution P satisfies the following condition.

Condition OC$_d$: The following relations hold

- $P(\mathbf{x}, 1) = P(\mathbf{x}, -1)$, for all $\mathbf{x} \in \mathcal{L} \cap \mathcal{X}$;
- $P(\mathbf{x}, 1) > P(\mathbf{x}, -1)$, for all $\mathbf{x} \in (\mathcal{H}_+ \cap \mathcal{X}) \setminus \mathcal{L}$;
- $P(\mathbf{x}, -1) > P(\mathbf{x}, 1)$, for all $\mathbf{x} \in (\mathcal{H}_- \cap \mathcal{X}) \setminus \mathcal{L}$.

Remark 1. *If condition* **OC**$_d$ *holds, then for any measurable set satisfying* $A \subseteq \mathcal{H}_+$ *or* $A \subseteq \mathcal{H}_-$*, we have* $\mu(A) \geq 0$*.*

Remark 2. *When condition* **OC**$_d$ *holds, for any measurable set* $A \subset \mathbb{R}^2$*, we have* $c(A) \geq 0$ *and*

$$c(A) = \min(\mu(A \cap \mathcal{H}_+), \mu(A \cap \breve{\mathcal{H}}_-)). \tag{2.12}$$

Remark 3. *If the probability distribution given by* $P(\mathbf{x}, \ell)$ *satisfies* $P(\mathbf{x}, 1) = \beta(||\mathbf{x} - \boldsymbol{\mu}_1||)$*, and* $P(\mathbf{x}, -1) = \beta(||\mathbf{x} - \boldsymbol{\mu}_2||)$*, where* $\beta(\cdot)$ *is a strictly decreasing function,*

$\boldsymbol{\mu}_1 \in \mathcal{H}_+$, $\boldsymbol{\mu}_2$ is the reflection of $\boldsymbol{\mu}_1$ across $\mathcal{L}$ and $||\mathbf{x}-\mathbf{y}||$ denotes the Euclidian distance between $\mathbf{x}$ and $\mathbf{y}$, then condition $\mathbf{OC}_d$ holds.

Proof. The optimal decision boundary is defined by the equation $P(\mathbf{x},1) = P(\mathbf{x},-1)$. Based on the hypothesis of this remark, this happens if and only if (iff) $||\mathbf{x}-\boldsymbol{\mu}_1|| = ||\mathbf{x}-\boldsymbol{\mu}_2||$, i.e., iff $\mathbf{x}$ is on the mediator line of the line segment connecting $\boldsymbol{\mu}_1$ and $\boldsymbol{\mu}_2$, which is $\mathcal{L}$. On the other hand, since $\beta(\cdot)$ is a strictly decreasing function, if $\mathbf{x}$ is closer to $\boldsymbol{\mu}_1$ compared to $\boldsymbol{\mu}_2$, then $P(\mathbf{x},1) > P(\mathbf{x},-1)$ and otherwise, $P(\mathbf{x},1) < P(\mathbf{x},-1)$. Therefore, condition $\mathbf{OC}_d$ holds. □

Example 1. Another example when the condition $\mathbf{OC}_d$ holds is when the class conditional distributions are Gaussian with flipped means and covariance matrices, $\boldsymbol{\mu}_1 = \boldsymbol{\mu}_{-1}^{\tau}$ and $\boldsymbol{\Sigma}_1 = \boldsymbol{\Sigma}_{-1}^{\tau}$, where $(a_1,a_2)^{\tau} = (a_2,a_1)$ and

$$\begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix}^{\tau} = \begin{bmatrix} a_{22} & a_{12} \\ a_{21} & a_{11} \end{bmatrix}.$$

In other words, $P(x_1,x_2,1) = \frac{c}{2\pi\sqrt{|\Sigma_1|}}\exp(-\frac{1}{2}\mathbf{z_1}^T\Sigma_1^{-1}\mathbf{z_1})$ and $P(x_1,x_2,-1) = \frac{c}{2\pi\sqrt{|\Sigma_{-1}|}}\exp(-\frac{1}{2}\mathbf{z_{-1}}^T\Sigma_{-1}^{-1}\mathbf{z_{-1}})$, for $(x_1,x_2) \in \mathcal{X}$, where c is an appropriate constant and

$$\mathbf{z_1} = \begin{pmatrix} x_1-\mu_1 \\ x_2-\mu_2 \end{pmatrix}, \mathbf{z_{-1}} = \begin{pmatrix} x_1-\mu_2 \\ x_2-\mu_1 \end{pmatrix},$$
$$\Sigma_1 = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix}, \Sigma_{-1} = \begin{bmatrix} a_{22} & a_{12} \\ a_{21} & a_{11} \end{bmatrix}.$$

An illustration of this example is shown in Fig. 2.4.

Note that when the optimal decision boundary has the equation $ax_1 + bx_2 + c = 0$, with $a \neq 0$ and $b \neq 0$. In this case, with the simple transformation $x'_1 = ax_1$ and $x'_2 = -bx_2 + c$, the decision boundary becomes $x'_1 - x'_2 = 0$.

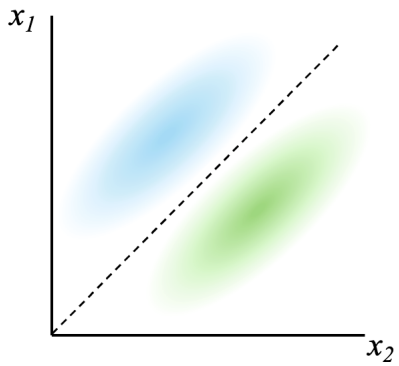


FIGURE 2.4: Example of distribution with $x_1 - x_2 = 0$ as the optimal classifier.

To show that Monge property is satisfied, for each pair (k_1, k_2), we will organize the problems (2.10) and (2.11) for all nodes v as a series of matrix search problems where each matrix satisfies the Monge property. The Matrix Search (MS) problem is the problem of finding the minimum element on each column. For a general $k \times n$ matrix, this can be done in $O(kn)$ operations. If the matrix satisfies the Monge property (to be defined shortly), the solution can be obtained by using a fast matrix search algorithm nicknamed SMAWK [1], which requires $O(k + n)$ operations. A $k \times n$ matrix G is said to satisfy the Monge property if $G(i, j) + G(i', j') \leq G(i, j') + G(i', j)$ for any $1 \leq i < i' \leq k, 1 \leq j < j' \leq n$.

Proposition 2. *For each (k_1, k_2) with $1 \leq k_1 \leq K_1 - 1$, $1 \leq k_2 \leq K_2 - 1$, the problems (2.10) and (2.11) can be solved for all nodes $v \in V$ in $O(M^3)$ time.*

Corollary 2. *The problem (2.8) can be solved in $O(K_1 K_2 M^3)$ operations.*

Proof of Proposition 2. For each edge $e = (m, n) \in E$, define $w_1(e) = \mu(\mathcal{R}(e) \cap \mathcal{H}_+)$, $w_2(e) = \mu(\mathcal{R}(e) \cap \mathcal{H}_-)$. Clearly, $w(e) = \min(w_1(e), w_2(e))$. Next define

$$\Gamma'_{k_1,k_2,1}(v) = \min_{u \in V_v, (u,v) \in E} (\hat{W}_{k_1-1,k_2}(u) + w_1(u, v)) \tag{2.13}$$

$$\Gamma'_{k_1,k_2,2}(v) = \min_{u \in V_v, (u,v) \in E} (\hat{W}_{k_1-1,k_2}(u) + w_2(u, v)) \tag{2.14}$$

Then relation (2.10) becomes equivalent to

$$\hat{W}'_{k_1,k_2}(v) = \min(\Gamma'_{k_1,k_2,1}(v), \Gamma'_{k_1,k_2,2}(v)) \tag{2.15}$$

For simplicity, we will assume that $\mathcal{M}_1 = \mathcal{M}_2 = \{1, \cdots, M-1\}$ since the argument can be easily extended to the case $\mathcal{M}_1 \neq \mathcal{M}_2$. Let v be a vertical node, then $v = (k, i', \ell)_v$. The search in (2.13) is over the edges $(k, i, \ell)_v \rightarrow (k, i', \ell)_v$, i.e., for all i such that $k \leq i < i'$. For each pair (k, ℓ), with $k + 1 < \ell$, we form two $(\ell - k) \times (\ell - k)$ matrices $G'_{k,\ell,1}$ and $G'_{k,\ell,2}$ as follows. For $s = 1, 2$, the elements of $G'_{k,\ell,s}$ are $G'_{k,\ell,s}(i, i')$ for $k \leq i, i' \leq \ell - 1$, where

$$G'_{k,\ell,s}(i, i') = \hat{W}_{k_1-1,k_2}(k, i, \ell)_v + w_s((k, i, \ell)_v, (k, i', \ell)_v)$$

if $i < i'$ and $G'_{k,\ell,s}(i, i') = \infty$ if $i \geq i'$. Then

$$\Gamma'_{k_1,k_2,s}((k, i', \ell)_v) = \min_{i:k \leq i < i'} G'_{k,\ell,s}(i, i')$$

This implies that finding $\Gamma'_{k_1,k_2,s}(v)$ for all vertical nodes v is equivalent to solving the MS problem in the matrix $G'_{k,\ell,s}$ for all pairs (k, ℓ) such that $0 \leq k < \ell - 1$ and $s = 1, 2$.

For the case of horizontal node v, we have $v = (j, k, i)_h$, and the search in (2.13) is over the edges $(\ell, j, k)_v \to (j, k, i)_h$, i.e., for all ℓ such that $0 \leq \ell \leq j$. For each pair (j, k) with $j < k$, we form two $(j + 1) \times (M + 2 - k)$ matrices $G''_{j,k,1}$ and $G''_{j,k,2}$ with elements $G''_{j,k,s}(\ell, i)$ for $0 \leq \ell \leq j$, $k \leq i \leq M + 1$, $s = 1, 2$, where

$$G''_{j,k,s}(\ell, i) = \hat{W}_{k_1-1,k_2}(\ell, j, k)_v + w_s((\ell, j, k)_v, (j, k, i)_h).$$

This implies that finding $\Gamma'_{k_1,k_2,s}(v)$ for all horizontal nodes v is equivalent to solving the MS problem in $G''_{j,k,s}$ for all pairs (j, k) such that $0 \leq j < k \leq M + 1$ and $s = 1, 2$. According to Lemma 1, all matrices $G'_{k,\ell,s}$ and $G''_{j,k,s}$ satisfy the Monge property, therefore the MS problem can be solved in $O(M)$ time for each of them. Since their total number is $O(M^2)$, solving all the aforementioned MS problems requires $O(M^3)$ operations. In view of the above discussion and using (2.15), we conclude that computing $\hat{W}'_{k_1,k_2}(v)$ for all nodes v can be done with $O(M^3)$ operations. Similarly, it can be shown that computing $\hat{W}''_{k_1,k_2}(v)$ for all nodes v can be performed in $O(M^3)$ time. With these observations, the conclusion follows. □

Lemma 1. *The matrices $G'_{k,\ell,s}$ and $G''_{j,k,s}$ satisfy the Monge property.*

Proof. For the Monge property to hold for the matrix $G'_{k,\ell,s}$, for any i_1, i_2, i'_1, i'_2 satisfying $k \leq i_1 < i_2 < \ell$, $k \leq i'_1 < i'_2 < \ell$, the following should hold

$$G'_{k,\ell,s}(i_1, i'_1) + G'_{k,\ell,s}(i_2, i'_2) \leq G'_{k,\ell,s}(i_1, i'_2) + G'_{k,\ell,s}(i_2, i'_1). \tag{2.16}$$

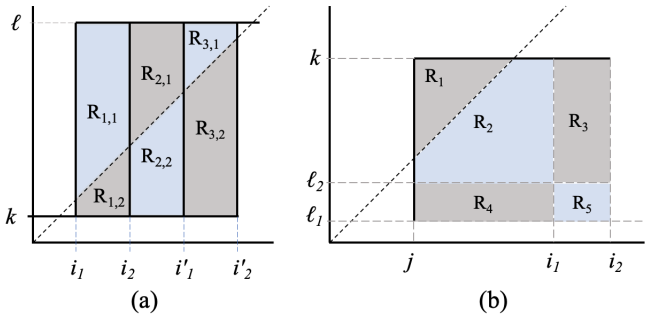


FIGURE 2.5: (a) example for $G_{k,\ell,1}(i,i')$. (b) example for $G_{j,k,2}(\ell,i)$.

If $i'_2 \geq i_1$, $G'_{k,\ell,s}(i_2,i'_1) = \infty$ and (2.16) is satisfied. It remains to consider the case when $i'_2 < i_1$. Then (2.16) reduces to

$$\begin{aligned} &w_s((k,i_1,\ell)_v,(k,i'_1,\ell)_v) + w_s((k,i_2,\ell)_v,(k,i'_2,\ell)_v) \\ &\leq w_s((k,i_1,\ell)_v,(k,i'_2,\ell)_v) + w_s((k,i_2,\ell)_v,(k,i'_1,\ell)_v). \end{aligned} \tag{2.17}$$

This situation is illustrated in Fig. 2.5(a). The above weights are defined in terms of the regions $R_{t,s}$, where $t = 1,2,3$ and $s = 1,2$, shown in the figure. More specifically,

$$\begin{aligned} w_s((k,i_1,\ell)_v,(k,i'_1,\ell)_v) &= \mu(R_{1,s} \cup R_{2,s}) \\ w_s((k,i_2,\ell)_v,(k,i'_2,\ell)_v) &= \mu(R_{2,s} \cup R_{3,s}) \\ w_s((k,i_1,\ell)_v,(k,i'_2,\ell)_v) &= \mu(R_{1,s} \cup R_{2,s} \cup R_{3,s}) \\ w_s((k,i_2,\ell)_v,(k,i'_1,\ell)_v) &= \mu(R_{2,s}). \end{aligned}$$

By further applying the fact that $\mu(A \cup B) = \mu(A) + \mu(B)$, for any disjoint sets A and B, we obtain that (2.17) holds with equality.

For the Monge property to hold for the matrix $G''_{j,k,s}$, for any ℓ_1, ℓ_2, i_1, i_2, satisfying $0 \leq \ell_1 < \ell_2 \leq j$, $k \leq i_1 < i_2 \leq M+1$, the following must hold

$$G''_{j,k,s}(\ell_1, i_1) + G''_{j,k,s}(\ell_2, i_2) \leq G''_{j,k,s}(\ell_1, i_2) + G''_{j,k,s}(\ell_2, i_1).$$

The above relation can be simplified to

$$\begin{aligned} &w_s((\ell_1, j, k)_v, (j, k, i_1)_h) + w_s((\ell_2, j, k)_v, (j, k, i_2)_h) \\ &\leq w_s((\ell_1, j, k)_v, (j, k, i_2)_h) + w_s((\ell_2, j, k)_v, (j, k, i_1)_h) \end{aligned} \tag{2.18}$$

This case is depicted in Fig. 2.5(b). The above weights are defined in terms of the regions $R_1, \cdots, R_6$ shown in the figure. More specifically, for $s = 1$, all terms in (2.18) are equal to $\mu(R_1)$, therefore the relation holds with equality. For $s = 2$, we have

$$\begin{aligned} w_2((\ell_1, j, k)_v, (j, k, i_1)_h) &= \mu(R_2 \cup R_4) \\ w_2((\ell_2, j, k)_v, (j, k, i_2)_h) &= \mu(R_2 \cup R_3) \\ w_2((\ell_1, j, k)_v, (j, k, i_2)_h) &= \mu(R_2 \cup R_3 \cup R_4 \cup R_5) \\ w_2((\ell_2, j, k)_v, (j, k, i_1)_h) &= \mu(R_2) \end{aligned}$$

After replacing the above in (2.18) and applying the additivity of μ over unions of disjoint sets, (2.18) becomes

$$\begin{aligned} &2\mu(R_2) + \mu(R_3) + \mu(R_4) \leq \\ &2\mu(R_2) + \mu(R_3) + \mu(R_4) + \mu(R_5), \end{aligned}$$

which holds since $\mu(A) \geq 0$ for any set A.

□

2.6 Discussion

In this section, we discuss how the algorithm can be adapted to the case when the sources are continuous and their pdf is known, or when the probability distribution is not known.

In the context of continuous sources, it is crucial to acknowledge the existence of an infinite number of potential thresholds for each quantizer. In this case, an intuitive approach for obtaining approximate solutions to the DSQ design problem resides in applying the proposed algorithm to a discretization of the original sources. The DSQ derived through this method should progressively converge to the performance of the optimal DSQ for the original sources as the accuracy of the discretization process improves.

The discretization of the sources can be done by randomly sampling N examples (x_1, x_2) from the joint density functions $f(\mathbf{x}, 1)$ and $f(\mathbf{x}, -1)$, leading to the creation of a sample set denoted as $\mathcal{X}'$. Further, the set $\mathcal{A}$ of possible boundaries is constructed as in the discrete case. It is noteworthy that for sufficiently large values of N, by this spatial discretization, the resultant set of boundaries aligns effectively with the pdf.

In practice, the probability distribution of the data and the labels is generally not known and only a training sequence is available. In this case, we can find the empirical probability distribution of the training sequence, which is discrete, and can apply the algorithm of Section 2.4 to find the DSQ that minimizes the classification error on the given training sequence. Assume that only a training sequence $\mathcal{T}$ is given, where $\mathcal{T} = (\mathbf{x}_t, \ell_t)_{1 \leq t \leq N}$, $\mathbf{x}_t = (x_{1,t}, x_{2,t}) \in \mathbb{R}^2$ and ℓ_t denotes the class label of $\mathbf{x}_t$, for $1 \leq t \leq N$.

In this case, we define the empirical probability distribution $P_{emp}(\mathbf{x}, \ell) = \frac{1}{N}|\{t : \mathbf{x}_t = \mathbf{x} \ \& \ \gamma(\mathbf{x}_t) = \ell\}|$. Hence, the values ρ, ρ_u, μ, and c can be calculated as in Section 2.2 and the rest of the conclusions follow.

2.7 Conclusion

In this chapter, the problem of designing a fixed-rate DSQ with convex cells for two discrete correlated sources is studied. We assume that the joint probability distribution of the sources and labels is known and that a fixed binary linear classifier is present at the decoder. The goal is to design the DSQ that minimizes the error of the classifier applied to the quantized outputs. We prove that the problem is equivalent to the problem of finding the shortest path with specific numbers and types of vertices in a certain weighted directed acyclic graph, and propose an efficient dynamic programming solution with a polynomial runtime. Further, we demonstrate that if the given linear classifier is the optimal classifier for the unquantized sources, then the solution algorithm can be expedited by a factor of M by leveraging the Monge property. Moreover, we illustrate how this algorithm can be adapted for scenarios where the source distribution is continuous and known, or when only a training sequence is available.

Chapter 3

Properties of the Optimal DSQ for Continuous Sources in the Equal-rate Case

This chapter is motivated by our goal to streamline the design process in the case of equal rates, i.e., where $K_1 = K_2 = K$. To this aim, we confine our search to the set of DSQs for which the threshold values of the two encoders are interleaved, which we call Staggered DSQs (SDSQs). To give some support to this choice, we study the properties of the SDSQs in the case of continuous sources when the linear classifier given at the decoder is the optimal classifier for the unquantized data. We demonstrate that for such sources, the optimal SDSQ must be strictly staggered meaning that no two thresholds are identical between the two encoders. This result implies that DSQs with identical encoders, used in many prior works, are not optimal for the aforementioned sources. Up to our knowledge, this result was not known before. In addition, we prove that the optimal SDSQ also minimizes a modified cost that is an upperbound of the original

cost. The choice of staggered DSQ and modified cost will enable us to simplify the graph model significantly, consequently reducing the time complexity of the solution algorithm to $O(KM^2)$ as it will be shown in Chapter 4. Moreover, we show that if, in addition, the sources fulfill a certain symmetry condition, the globally optimal DSQ must be strictly staggered.

This chapter is organized as follows. Section 3.1 introduces the SDSQs and the modified cost. Section 3.2 establishes the properties of the optimal SDSQ when the optimal classifier is linear, and also when in addition, the data distribution satisfies a symmetry condition. Moreover, more insight on why the staggered structure performs better compared to identical encoders is provided in Section 3.3. Finally, Section 3.4 concludes the chapter.

3.1 Problem Description and Notations

First, let us introduce the formal definition of Staggered DSQs. We say that the DSQ is *staggered* if the sequences of thresholds $\tilde{\mathbf{u}}$ and $\tilde{\mathbf{v}}$ satisfy

$$\tilde{u}_1 \leq \tilde{v}_1 \leq \tilde{u}_2 \leq \tilde{v}_2 \leq \cdots \leq \tilde{u}_i \leq \tilde{v}_i \leq \tilde{u}_{i+1} \leq \cdots \leq \tilde{u}_{K-1} \leq \tilde{v}_{K-1}, \tag{3.1}$$

or

$$\tilde{v}_1 \leq \tilde{u}_1 \leq \tilde{v}_2 \leq \tilde{u}_2 \leq \cdots \leq \tilde{v}_i \leq \tilde{u}_i \leq \tilde{v}_{i+1} \leq \cdots \leq \tilde{v}_{K-1} \leq \tilde{u}_{K-1}. \tag{3.2}$$

If (3.1) holds, we will say that the thresholds satisfy the u-first order and that the SDSQ is a u-first SDSQ and that it has a *u-first orientation*. If (3.2) is satisfied, we will say

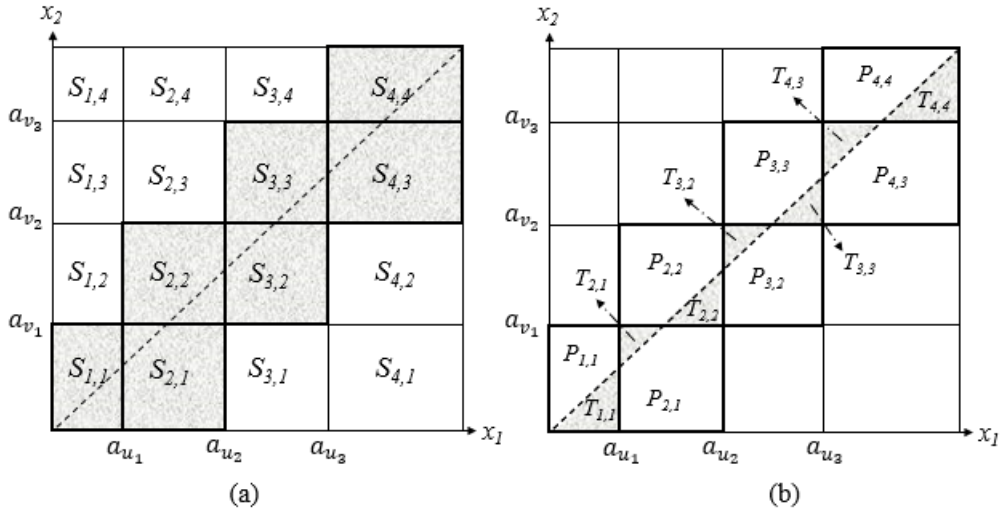


FIGURE 3.1: In (a) the relevant cells are highlighted. In (b) the triangular regions used in the computation of the modified cost are highlighted.

that the thresholds satisfy the v-first order and that the SDSQ is a v-first SDSQ and that it has a *v-first orientation*. Additionally, A DSQ is *strict* if the two encoders do not have any threshold in common. Thus, an SDSQ is strict if (3.1) or (3.2) hold with strict inequalities. The product quantizer partition of a u-first strict SDSQ is depicted in Fig. 3.1.

Recall that a cell $S_{i,j} = U_i \times V_j$ is called relevant if and only if $S_{i,j} \cap \breve{\mathcal{H}}_- \neq \emptyset$ and $S_{i,j} \cap \breve{\mathcal{H}}_+ \neq \emptyset$. Then for a u-first strict SDSQ, the relevant cells are $S_{i,i}$, for $1 \leq i \leq K$, and $S_{i+1,i}$, for $1 \leq i \leq K-1$ (see Fig. 3.1(a)). For a v-first strict SDSQ, the relevant cells are $S_{i,i}$, for $1 \leq i \leq K$, and $S_{i,i+1}$, for $1 \leq i \leq K-1$.

In any SDSQ, each relevant cell $S_{i,j}$ with $\tilde{u}_i \neq \tilde{v}_j$, $\tilde{u}_{i-1} \neq \tilde{v}_{j-1}$, is divided by $\mathcal{L}$ into a triangular region denoted $T_{i,j}$, and a pentagonal region denoted $P_{i,j}$. The definition is extended to the cases when $\tilde{u}_i = \tilde{v}_j$ or $\tilde{u}_{i-1} = \tilde{v}_{j-1}$ as follows. When only one of the above equalities holds, the region $P_{i,j}$ has actually only four sides, while when both

equalities hold, i.e., $\tilde{u}_i = \tilde{v}_j$ and $\tilde{u}_{i-1} = \tilde{v}_{j-1}$, both $T_{i,j}$ and $P_{i,j}$ are triangular. To make more clear the distinction we provide next the formal definition.

Definition 1. *For each relevant cell $S_{i,j}$ of an SDSQ, we define $T_{i,j}$ and $P_{i,j}$ as follows.*

If the SDSQ has a u-first orientation, then

$$T_{i,i} = S_{i,i} \cap \breve{\mathcal{H}}_- = T'(\tilde{v}_{i-1}, \tilde{u}_i),\ P_{i,i} = S_{i,i} \cap \mathcal{H}_+$$
$$T_{i+1,i} = S_{i+1,i} \cap \mathcal{H}_+ = T(\tilde{u}_i, \tilde{v}_i),\ P_{i+1,i} = S_{i+1,i} \cap \breve{\mathcal{H}}_-.$$

If the SDSQ has a v-first orientation, then

$$T_{i,i} = S_{i,i} \cap \mathcal{H}_+ = T(\tilde{u}_{i-1}, \tilde{v}_i),\ P_{i,i} = S_{i,i} \cap \breve{\mathcal{H}}_-$$
$$T_{i,i+1} = S_{i,i+1} \cap \breve{\mathcal{H}}_- = T'(\tilde{v}_i, \tilde{u}_i),\ P_{i,i+1} = S_{i,i+1} \cap \mathcal{H}_+.$$

For each cell $S_{i,j}$ we define a modified cost $c'(S_{i,j})$ as follows. If $S_{i,j}$ is a relevant cell, then

$$c'(S_{i,j}) = \mu(T_{i,j}),$$

otherwise $c'(S_{i,j}) = 0$. The choice of $T_{i,j}$ instead of $P_{i,j}$ is by considering a smaller region for the cost of each cell, aiming for minimum cost possible. Note that these notations are valid for both continuous and discrete probability distributions. It follows that the modified cost is an upperbound of the original cost, i.e.,

$$c(S_{i,j}) \leq c'(S_{i,j}). \tag{3.3}$$

The modified cost of an SDSQ $\mathbf{Q}$, denoted by $c'(\mathbf{Q})$ is defined as the sum of the modified costs of its cells, i.e.,

$$c'(\mathbf{Q}) = \sum_{i=1}^{K}\sum_{j=1}^{K} c'(S_{i,j}).$$

According to (3.3), we have

$$c(\mathbf{Q}) \leq c'(\mathbf{Q}), \tag{3.4}$$

Let $\mathcal{Q}_{st}(K)$ denote the set of (K, K)-level strict SDSQs determined by all possible pairs $\tilde{\mathbf{u}} \in \mathbb{R}^{K-1}$, $\tilde{\mathbf{v}} \in \mathbb{R}^{K-1}$, satisfying relations (3.1) or (3.2). Then we are analyzing properties of DSQs that would minimize this modified cost,

$$\inf_{\mathbf{Q}\in\mathcal{Q}_{st}(K)} c'(\mathbf{Q}). \tag{3.5}$$

Note that if the distribution is discrete, the infimum is actually the minimum. In the next section, we study the properties that support the use of this modified cost instead of the original cost.

3.2 Properties of Optimal Staggered DSQ for Continuous Sources when the Optimal Classifier is Linear

In this section, we assume that the two sources are continuous and that the linear classifier γ is the optimal classifier for the unquantized data. In this scenario, we prove that the (K, K)-level staggered DSQ that minimizes the expected classification error is strict

and satisfies relation (3.4) with equality, i.e., the DSQ that minimizes the modified cost, in fact, minimizes the true cost. Further, we show that if the distribution additionally satisfies a certain symmetry property, the DSQ that minimizes the expected classification error has to be staggered and strict. These results support the use of the algorithm developed in the next chapter.

More specifically, we consider the situation when the classifier γ minimizes the classification error ρ_u for the unquantized data, and any other classifier that achieves the smallest classification error has the line $\mathcal{L}$ as the decision boundary. In other words, this is to say that the joint distribution $f(\mathbf{x}, \ell)$ satisfies the condition $\mathbf{OC}_c$ presented next.

Condition $\mathbf{OC}_c$: The following relations hold

- $f(\mathbf{x}, 1) = f(\mathbf{x}, -1)$, for all $\mathbf{x} \in \mathcal{L}$;
- $f(\mathbf{x}, 1) > f(\mathbf{x}, -1)$, for all $\mathbf{x} \in (\mathcal{H}_+) \setminus \mathcal{L}$;
- $f(\mathbf{x}, -1) > f(\mathbf{x}, 1)$, for all $\mathbf{x} \in (\mathcal{H}_-) \setminus \mathcal{L}$.

As in Chapter 2, if condition $\mathbf{OC}_c$ holds, then for any measurable set satisfying $A \subseteq \mathcal{H}_+$ or $A \subseteq \mathcal{H}_-$, we have $\mu(A) \geq 0$.

Recall that for a measurable set $S \subset \mathbb{R}^2$, $area(S) = \int_S d\mathbf{x}$. The following lemma is straightforward.

Lemma 2. *Assume that the sources are continuous with joint pdf $f(\mathbf{x}, \ell)$ of sources and labels and that condition $\mathbf{OC}_c$ holds. Then for any measurable sets A and B satisfying $A \subset B \subseteq \mathbb{R}^2$ the following are valid.*

a) $\mu(A \cap \mathcal{H}_+) \leq \mu(B \cap \mathcal{H}_+)$; *if* $area((B \setminus A) \cap \mathcal{H}_+)$ *is larger than* 0, *then the inequality is strict;*

b) $\mu(A \cap \breve{\mathcal{H}}_-) \leq \mu(B \cap \breve{\mathcal{H}}_-)$; *if* $area((B \setminus A) \cap \breve{\mathcal{H}}_-)$ *is larger than* 0, *then the inequality is strict;*

c) $c(A) \leq c(B)$.

Lemma 3. *Assume that the sources are continuous with joint pdf* $f(\mathbf{x}, \ell)$ *of sources and labels and that condition* $\mathbf{OC}_c$ *holds. Then for any DSQ* $\mathbf{Q}$, *by splitting a cell of any encoder into two cells, the cost* $c(\mathbf{Q})$ *does not increase.*

Proof. Splitting a cell in one encoder leads to splitting some cells in the product quantizer $Q_1 \times Q_2$. The split of a non-relevant cell of $Q_1 \times Q_2$ does not have any impact. Let A be a relevant cell of $Q_1 \times Q_2$ that is split into A_1 and A_2. Then $\mu(A \cap \mathcal{H}_+) = \mu(A_1 \cap \mathcal{H}_+) + \mu(A_2 \cap \mathcal{H}_+)$ and $\mu(A \cap \breve{\mathcal{H}}_-) = \mu(A_1 \cap \breve{\mathcal{H}}_-) + \mu(A_2 \cap \breve{\mathcal{H}}_-)$. Combining these with Remark 1, we obtain

$$\begin{aligned} &\min(\mu(A_1 \cap \mathcal{H}_+), \mu(A_1 \cap \breve{\mathcal{H}}_-)) + \min(\mu(A_2 \cap \mathcal{H}_+), \mu(A_2 \cap \breve{\mathcal{H}}_-)) \\ &\leq \min(\mu(A \cap \mathcal{H}_+), \mu(A \cap \breve{\mathcal{H}}_-)), \end{aligned}$$

leading further to $c(A_1) + c(A_2) \leq c(A)$ according to Remark 2. □

Let $\mathcal{O}_K$ denote the set of pairs of threshold vectors $(\tilde{\mathbf{u}}, \tilde{\mathbf{v}}) \in \mathbb{R}^{2K-2}$ satisfying relations (3.1) or (3.2) and $\tilde{u}_i < \tilde{u}_{i+1}$, $\tilde{v}_i < \tilde{v}_{i+1}$, for $1 \leq i \leq K-1$. Thus, $(\tilde{\mathbf{u}}, \tilde{\mathbf{v}})$ are the threshold sequences of a (K, K)-level SDSQ if and only if $(\tilde{\mathbf{u}}, \tilde{\mathbf{v}}) \in \mathcal{O}_K$. Further, let $\bar{\mathbb{R}}$ denote the extended real line, i.e., $\bar{\mathbb{R}} = \mathbb{R} \cup \{-\infty, +\infty\}$, and let $\bar{\mathcal{O}}_K$ denote the pairs

of vectors $(\tilde{\mathbf{u}}, \tilde{\mathbf{v}}) \in \bar{\mathbb{R}}^{2K-2}$ satisfying relations (3.1) or (3.2). Note that $\bar{\mathcal{O}}_K$ equals the closure of the set $\mathcal{O}_K$ in $\bar{\mathbb{R}}^{2K-2}$. Furthermore, any pair $(\tilde{\mathbf{u}}', \tilde{\mathbf{v}}') \in \bar{\mathcal{O}}_K \setminus \mathcal{O}_K$ corresponds to a DSQ where at least one encoder has less than K nonempty cells. Then the cost function c can be defined for $(\tilde{\mathbf{u}}', \tilde{\mathbf{v}}')$ as the summation of the costs of the relevant cells. This way, the function c can be extended to $\bar{\mathbb{R}}^{2K-2}$.

Since $\mathcal{O}_K$ is not a compact set, it is not guaranteed that a continuous function can achieve its minimum on this set. However, we will show that this is true for the cost function c. In other words, there exists an optimum (K, K)-level SDSQ, i.e., a (K, K)-level SDSQ minimizing c. This is stated in the following theorem.

Theorem 1. *Assume that the sources are continuous with joint pdf $f(\mathbf{x}, \ell)$ of sources and labels and that condition* $\mathbf{OC}_c$ *holds. Then there exists $(\tilde{\mathbf{u}}_{opt}, \tilde{\mathbf{v}}_{opt}) \in \mathcal{O}_K$ such that*

$$c(\tilde{\mathbf{u}}_{opt}, \tilde{\mathbf{v}}_{opt}) = \min_{(\tilde{\mathbf{u}}, \tilde{\mathbf{v}}) \in \mathcal{O}_K} c(\tilde{\mathbf{u}}, \tilde{\mathbf{v}}).$$

Proof. It can be easily verified that the function c is continuous on $\bar{\mathcal{O}}_K$. Since $\bar{\mathcal{O}}_K$ contains all its limiting points and the function c is continuous and bounded, it follows that the function c has a finite minimum on $\bar{\mathcal{O}}_K$. Let $(\tilde{\mathbf{u}}_{opt}, \tilde{\mathbf{v}}_{opt}) \in \bar{\mathcal{O}}_K$ denote a point of minimum. If $(\tilde{\mathbf{u}}_{opt}, \tilde{\mathbf{v}}_{opt}) \in \bar{\mathcal{O}}_K \setminus \mathcal{O}_K$, then $(\tilde{\mathbf{u}}_{opt}, \tilde{\mathbf{v}}_{opt})$ represents a SDSQ $\mathbf{Q}$ where at least one encoder has less than K nonempty cells. Then we can construct a new $(\tilde{\mathbf{u}}', \tilde{\mathbf{v}}') \in \mathcal{O}_K$ by appropriately splitting some nonempty cells of encoder 1 and/or encoder 2 until both encoders have K nonempty cells[1]. By Lemma 3, we have $c(\tilde{\mathbf{u}}', \tilde{\mathbf{v}}') \leq c(\tilde{\mathbf{u}}_{opt}, \tilde{\mathbf{v}}_{opt})$, and according to the optimality of $(\tilde{\mathbf{u}}_{opt}, \tilde{\mathbf{v}}_{opt})$, we have $c(\tilde{\mathbf{u}}', \tilde{\mathbf{v}}') \geq c(\tilde{\mathbf{u}}_{opt}, \tilde{\mathbf{v}}_{opt})$. It follows that $c(\tilde{\mathbf{u}}', \tilde{\mathbf{v}}') = c(\tilde{\mathbf{u}}_{opt}, \tilde{\mathbf{v}}_{opt})$ and the proof is completed. □

[1]The splitting has to be done such that relations (3.1) or (3.2) hold, which is possible.

A relevant cell $S_{i,j}$ is called a *Tcell* if $\mu(T_{i,j}) \leq \mu(P_{i,j})$, and it is called a *Pcell* otherwise. By the *status* of a relevant cell, we understand the status of it being a Tcell or a Pcell.

Remark 4. *Obviously, if $S_{i,j}$ is a Tcell, then $c(S_{i,j}) = c'(S_{i,j})$. Thus, if an SDSQ $\mathbf{Q}$ has only Tcells, then $c(\mathbf{Q}) = c'(\mathbf{Q})$.*

The proof of the next result is deferred to Appendix C.

Theorem 2. *Assume that the sources are continuous with joint pdf $f(\mathbf{x}, \ell)$ of sources and labels and that condition $\mathbf{OC}_c$ holds. Then any optimum (K, K)-level SDSQ must be strict and have only Tcells.*

The following corollary is immediate.

Corollary 3. *When condition $\mathbf{OC}_c$ is satisfied, the following holds*

$$\min_{(\tilde{\mathbf{u}},\tilde{\mathbf{v}})\in\mathcal{O}_K} c(\tilde{\mathbf{u}}, \tilde{\mathbf{v}}) = \min_{(\tilde{\mathbf{u}},\tilde{\mathbf{v}})\in\mathcal{O}_K} c'(\tilde{\mathbf{u}}, \tilde{\mathbf{v}}).$$

Remark 5. *According to Corollary 3, although c' is in general only an upperbound of the true cost c, the (K, K)-level SDSQ that minimizes c' also minimizes the true cost c when condition $\mathbf{OC}_c$ holds.*

In order to present the next results of this section, we first introduce the following symmetry condition.

Condition $\mathbf{S}_c$: The following relation is valid for all $(x_1, x_2) \in \mathbb{R}^2$:

$$f(x_1, x_2, 1) - f(x_1, x_2, -1) = f(x_2, x_1, -1) - f(x_2, x_1, 1).$$

Remark 6. *A special case when condition $\mathbf{S}_c$ is satisfied is when the following equality is true for all $(x_1, x_2) \in \mathbb{R}^2$: $f(x_1, x_2, 1) = f(x_2, x_1, -1)$. Note that if $f(\mathbf{x}, \ell)$ fulfills the conditions in Remark 3, then condition $\mathbf{S}_c$ holds as well.*

Example 2. An example of such a case is when the conditional joint distribution given each class is Gaussian with flipped covariance matrix and mean vectors, i.e., $\boldsymbol{\mu}_1 = \boldsymbol{\mu}_{-1}^{\tau}$ and $\boldsymbol{\Sigma}_1 = \boldsymbol{\Sigma}_{-1}^{\tau}$, where $(a_1, a_2)^{\tau} = (a_2, a_1)$ and $\begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix}^{\tau} = \begin{bmatrix} a_{22} & a_{12} \\ a_{21} & a_{11} \end{bmatrix}$. In other words, $f(x_1, x_2, 1) = \frac{1}{2\pi\sqrt{|\Sigma_1|}} \exp(-\frac{1}{2}\mathbf{z_1}^T\Sigma_1^{-1}\mathbf{z_1})$ and $f(x_1, x_2, -1) = \frac{1}{2\pi\sqrt{|\Sigma_{-1}|}} \exp(-\frac{1}{2}\mathbf{z_{-1}}^T\Sigma_{-1}^{-1}\mathbf{z_{-1}})$, where $\mathbf{z_1} = \begin{pmatrix} x_1 - \mu_1 \\ x_2 - \mu_2 \end{pmatrix}$, $\mathbf{z_{-1}} = \begin{pmatrix} x_1 - \mu_2 \\ x_2 - \mu_1 \end{pmatrix}$, $\Sigma_1 = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix}$ and $\Sigma_{-1} = \begin{bmatrix} a_{22} & a_{12} \\ a_{21} & a_{11} \end{bmatrix}$. Note that in this case, $f(x_2, x_1, -1)$ would have $\mathbf{z_{-1}} = \begin{pmatrix} x_2 - \mu_2 \\ x_1 - \mu_1 \end{pmatrix}$, $|\Sigma_1| = |\Sigma_{-1}|$, and $\mathbf{z_1}^T\Sigma_1^{-1}\mathbf{z_1} = \mathbf{z_{-1}}^T\Sigma_{-1}^{-1}\mathbf{z_{-1}}$. Therefore, $f(x_1, x_2, 1) = f(x_2, x_1, -1)$, which is the special case discussed in Remark 6.

Consider now the following notation. For any set $S \subseteq \mathbb{R}^2$, let $\sigma(S) = \{(y, x) : (x, y) \in S\}$. In other words, $\sigma(S)$ is the reflection of S across the line $\mathcal{L}$. We will simply refer to it as the reflection of S. The following lemma is obvious.

Lemma 4. *Assume that the sources are continuous with joint pdf $f(\mathbf{x}, \ell)$ of sources and labels. If condition $\mathbf{S}_c$ holds then for any measurable set $S \subseteq \mathbb{R}^2$, we have $\mu(S \cap \mathcal{H}_+) =$*

$\mu(\sigma(S) \cap \breve{\mathcal{H}}_-)$, $\mu(S \cap \breve{\mathcal{H}}_-) = \mu(\sigma(S) \cap \mathcal{H}_+)$, *and* $c(S) = c(\sigma(S))$.

Lemma 5. *Assume that the sources are continuous with joint pdf* $f(\mathbf{x}, \ell)$ *of sources and labels. If both conditions* $\mathbf{OC}_c$ *and* $\mathbf{S}_c$ *are satisfied, then any SDSQ has only Tcells.*

Proof. Note that if $S_{i,j}$ is a cell in an SDSQ then $\sigma(T_{i,j}) \subseteq P_{i,j}$. By applying further Lemmas 4 and Lemma 2 the claim follows. □

The proof of the following theorem is deferred to Appendix D.

Theorem 3. *Assume that the sources are continuous with joint pdf* $f(\mathbf{x}, \ell)$ *of sources and labels, and that conditions* $\mathbf{OC}_c$ *and* $\mathbf{S}_c$ *hold. Then the* (K, K)*-level DSQ that minimizes the expected misclassification error* ρ *must be strictly staggered.*

This result shows that for the case of equal rates, when the joint probability distribution is symmetric with respect to the classification line and the given linear classifier is the optimal classifier, the globally optimal DSQ is strictly staggered. Further, due to symmetry property, the modified cost is equivalent to the original cost. Hence, the DSQ that minimizes the modified cost is the one that minimizes the expected classification error.

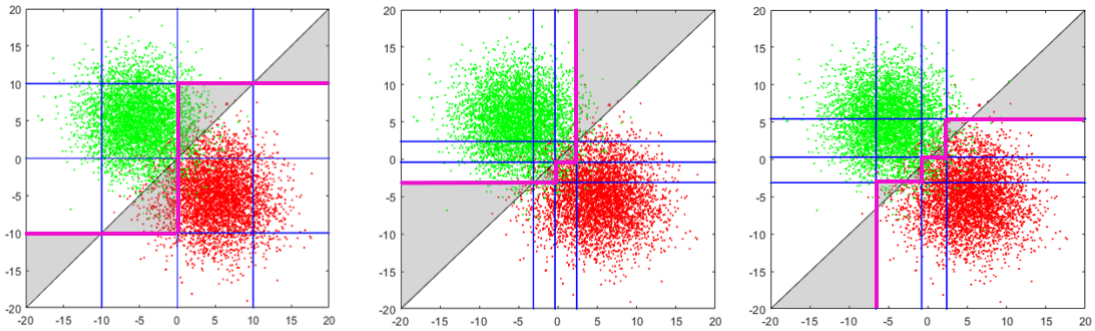
3.3 Suboptimality of DSQs with Identical Encoders

The following result is a corollary of Theorem 2.

Corollary 4. *When the sources are continuous with joint pdf* $f(\mathbf{x}, \ell)$ *of sources and labels, and condition* $\mathbf{OC}_c$ *is satisfied, any optimum SDSQ has strictly better performance than any DSQ with identical encoders.*

Proof. Note that any DSQ with identical encoders is a special case of an SDSQ, but it is not strict. Based on Theorem 2, an SDSQ that is not strictly staggered cannot be optimal. □

In order to gain a better insight into the reason why the constraint of identical encoders is highly restrictive, we illustrate in Fig. 3.2 a training set and the partition of the product quantizer $Q_1 \times Q_2$ obtained with Uniform Length (UL) encoders [9], Optimal Identical Encoders (OIE) and Optimal Non-Identical Encoders (ONIE) when $R = 2$ and there are 10000 data samples, for a symmetric bivariate normal distribution. Fig. 3.3 shows the magnified version of Fig. 3.2 (b) and (c) in the region of high concentration of training points. It can be seen that the decision boundary of the nonlinear classifier obtained by cascading the DSQ and the linear classifier is more complex in the case of ONIE and its number of relevant cells is almost doubled in comparison with OIE. This confers ONIE a higher flexibility and allows it to choose cost regions that are less populated. Note that the plotted quantizer thresholds are obtained using optimization on training data as will be shown in Chapter 4.



(a) UL (b) OIE (c) ONIE

FIGURE 3.2: Training set and partition of the product quantizer for UL, OIE and ONIE when $R = 2$. The cost regions of the relevant cells are depicted in grey. The decision boundary of the nonlinear classifier obtained after incorporating the quantization is shown in pink.

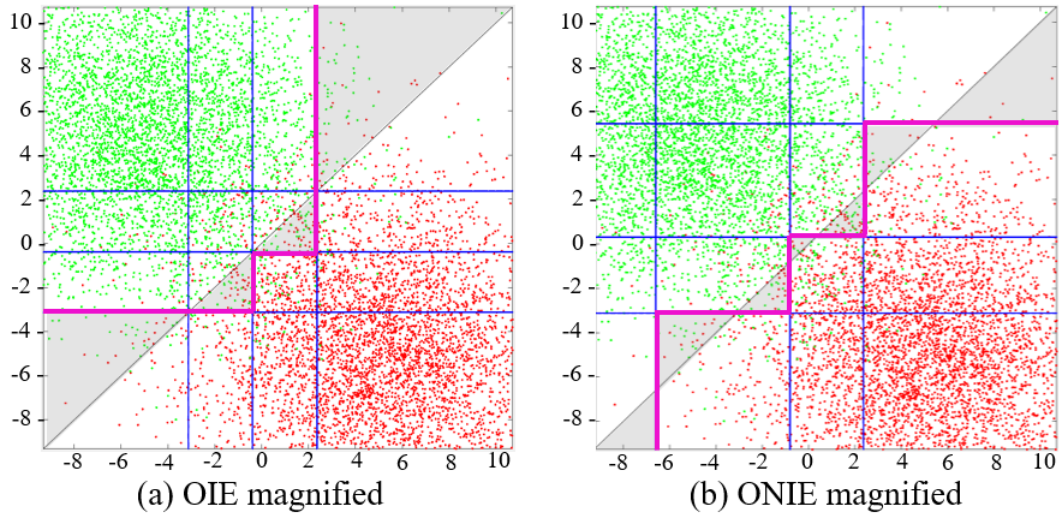


FIGURE 3.3: Quantizer partition for OIE and ONIE, magnified. The cost regions of the relevant cells are depicted in grey. The decision boundary of the nonlinear classifier obtained after incorporating the quantization is shown in pink.

3.4 Conclusion

In this chapter, we study the properties of the DSQs when the quantizer rates are equal. We introduce the structural constraint of staggered thresholds and a modified cost, which is generally an upperbound on the actual classification error. For the scenario of continuous sources and optimal classifier, we show that the optimal staggered DSQ among all staggered DSQs is strict, i.,e., no two thresholds of the quantizers are identical, and minimizing the modified cost is equivalent to minimizing the actual classification error. Further, we show that if, in addition, the joint distribution of sources and labels is symmetric with respect to the decision boundary, then the optimal DSQ that minimizes the misclassification error, must be strictly staggered. These results indicate that any optimal staggered DSQ has better performance than any DSQ with identical encoders.

Chapter 4

Faster DSQ Design for Discrete Sources in the Equal-rate Case

In this chapter, we propose a faster DSQ design algorithm for discrete sources in the equal-rate case, inspired by the results of Chapter 3. To expedite the design algorithm we restrict our search to the set of Staggered DSQ and use the modified cost that was introduced in the previous chapter (which is an upper bound of the classification error) as the objective function to be minimized. We also provide an algorithm for the design of optimal DSQs with identical encoders. Additionally, we establish that in instances when the given linear classifier is the optimal classifier for the unquantized data, the Monge property holds, facilitating a reduction in time complexity of the algorithm by an order of M. Moreover, we show that if the sources additionally satisfy the symmetry property introduced in the previous chapter, the proposed algorithm provides the globally optimal DSQ with respect to the original cost.

The experimental results demonstrate the superiority of the designed SDSQs with this modified cost, compared to the prior work that uses identical encoders, even though

we try to minimize only an upper bound of the error.

This chapter is organized as follows. In Section 4.1, we clarify which results established in the previous chapter for continuous distributions hold for discrete distributions as well. In Section 4.2, the graph model for the problem with the modified cost along with a dynamic programming solution algorithm are proposed. Section 4.3 describes how the algorithm of the previous section can be sped up by a factor of M by leveraging the Monge property when the given classifier is optimal for the unquantized sources. Section 4.4 proposes an optimal solution algorithm for the case when the quantizer thresholds are restricted to be identical for the two sources. Experimental results illustrated in Section 4.5 confirm the superiority of the SDSQs designed by minimizing the modified cost, compared to prior work that imposed identical encoders. Finally, Section 4.6 concludes the chapter.

4.1 Notes on the Optimal SDSQ for Discrete Distribution

The results in Section 3.2 are derived for continuous sources. However, the continuity is not a necessary condition for some of them. In this section, we specify which of those results hold for the discrete case too. Throughout this section, we consider the condition $\mathbf{OC}_d$ to be satisfied.

It can be easily verified that the results of Lemmas 2 and 3 hold in the discrete case as well. For clarity, we will restate them next for the discrete scenario.

Lemma 6. *Assume that the sources are discrete with joint probability $P(\mathbf{x}, \ell)$ of sources*

and labels and that condition $\mathbf{OC}_d$ *holds. Then for any measurable sets* A *and* B *satisfying* $A \subset B \subseteq \mathbb{R}^2$ *the following are valid.*

a) $\mu(A \cap \mathcal{H}_+) \leq \mu(B \cap \mathcal{H}_+)$*; if* $area((B \setminus A) \cap \mathcal{H}_+)$ *is larger than* 0*, then the inequality is strict;*

b) $\mu(A \cap \breve{\mathcal{H}}_-) \leq \mu(B \cap \breve{\mathcal{H}}_-)$*; if* $area((B \setminus A) \cap \breve{\mathcal{H}}_-)$ *is larger than* 0*, then the inequality is strict;*

c) $c(A) \leq c(B)$.

Lemma 7. *Assume that the sources are discrete with joint probability* $P(\mathbf{x}, \ell)$ *of sources and labels and that condition* $\mathbf{OC}_d$ *holds. Then for any DSQ* $\mathbf{Q}$*, by splitting a cell of any encoder into two cells, the cost* $c(\mathbf{Q})$ *does not increase.*

Note that in the discrete case, the set of all possible (K, K)-level SDSQs can be restricted to a finite set. Therefore, result of Theorem 1 is obvious.

The condition $\mathbf{S}_c$ can be easily translated to discrete distributions for which $\mathcal{X} = \mathcal{A}_1 \times \mathcal{A}_2$. Recall that $\mathcal{A}_k$ was defined as the projection of $\mathcal{X}$ on the k-th axis, $k = 1, 2$.

Condition $\mathbf{S}_d$: The following relation is valid for all $(x_1, x_2) \in \mathcal{A}_1 \times \mathcal{A}_2$:

$$P(x_1, x_2, 1) - P(x_1, x_2, -1) = P(x_2, x_1, -1) - P(x_2, x_1, 1).$$

It can be easily verified that Remark 4, and Lemmas 4 and 5 hold in the discrete case too.

Remark 7. *Obviously, if $S_{i,j}$ is a Tcell, then $c(S_{i,j}) = c'(S_{i,j})$. Thus, if an SDSQ $\mathbf{Q}$ has only Tcells, then $c(\mathbf{Q}) = c'(\mathbf{Q})$.*

Lemma 8. *Assume that the sources are discrete with joint probability $P(\mathbf{x}, \ell)$ of sources and labels. If condition $\mathbf{S}_d$ holds then for any measurable set $S \subseteq \mathbb{R}^2$, we have $\mu(S \cap \mathcal{H}_+) = \mu(\sigma(S) \cap \breve{\mathcal{H}}_-)$, $\mu(S \cap \breve{\mathcal{H}}_-) = \mu(\sigma(S) \cap \mathcal{H}_+)$, and $c(S) = c(\sigma(S))$.*

Lemma 9. *Assume that the sources are discrete with joint probability $P(\mathbf{x}, \ell)$ of sources and labels. If both conditions $\mathbf{OC}_d$ and $\mathbf{S}_d$ are satisfied, then any SDSQ has only Tcells.*

Furthermore, a close inspection of the proof of Theorem 3 shows that the continuity of the distribution is only needed to prove that the strictness of the optimal staggered DSQ but not to prove the optimality of staggered DSQs within the general class of DSQs. Therefore, we have the following result.

Then the following theorem is derived in similar steps as Theorem 3.

Theorem 4. *Assume that the sources are discrete with joint probability $P(\mathbf{x}, \ell)$ of sources and labels and that both conditions $\mathbf{OC}_d$ and $\mathbf{S}_d$ hold. Then the optimal (K, K)-level staggered DSQ that minimizes the expected classification error is also optimal within the general class of (K, K)-level DSQs without the staggerness constraint.*

Proof. Lemmas 17 and 18 which are stated in Appendix D do not require the continuity of sources. Based on that it can be concluded that there is a staggered DSQ that is optimal within the class of DSQs. □

For Theorem 2, the continuity of the distribution is essential since it relies on Lemma 16 stated in Appendix B. Therefore, the proof cannot be translated to the discrete case.

Nevertheless, in practice, often the discrete distribution at hand is obtained by discretizing a continuous distribution and the true goal is to optimize the DSQ for the true continuous distribution, which is not known. Therefore, the results on the structure of the optimal DSQ/SDSQ for continuous sources give some support for imposing the staggered structure and using the modified cost in the design of the optimal DSQ for the discrete case.

4.2 Graph Model for the Modified Cost Problem and its Solution

In this section we propose a graph model and a solution to the problem of minimizing modified the cost of strict SDSQs, for the case of equal encoder rates and discrete sources. In other words, we consider the following problem.

$$\min_{\mathbf{Q}\in\mathcal{Q}'_{st}(K,\mathcal{A})} c'(\mathbf{Q}), \tag{4.1}$$

where c' is the modified cost defined in Chapter 3 and $\mathcal{Q}'_{st}(K,\mathcal{A})$ denotes the set of (K,K)-level strict SDSQs determined by all possible pairs $\tilde{\mathbf{u}} \in \mathcal{A}^{K-1}$, $\tilde{\mathbf{v}} \in \mathcal{A}^{K-1}$, satisfying relations (3.1) or (3.2).

Problem (4.1) can be modelled as a minimum weight path problem with certain constraints on the edges in the WDAG $G' = (V', E', w')$ described next. Its set of vertices is $V' = \{m : 0 \leq m \leq M+1\}$ and its set of edges is $E' = \{(m,n)_i : 0 \leq m < n \leq M+1, i \in \{1,2\}\}$. Note that between each two vertices we have two types of

edges, namely $(m,n)_1$ (an edge of type 1), which corresponds to $T'(m,n)$, and $(m,n)_2$ (an edge of type 2), which corresponds to $T(m,n)$. The weights of the edges in G' are $w'((m,n)_1) = \mu(T'(m,n))$ and $w'((m,n)_2) = \mu(T(m,n))$. Then, the following result holds.

Proposition 3. *The problem (4.1) is equivalent to the problem of finding the minimum path in G' among all the paths from 0 to $M+1$ that have $2K-1$ edges of alternating types.*

Proof. We will show that there is a one-to-one correspondence between the paths in G' specified in the statement of Proposition 3 and the strict SDSQs in $\mathcal{Q}_{st}(K,\mathcal{A})$. Let $(z_0, z_1, \cdots, z_{2K-1})$ be the sequence of nodes on such a path. Then $0 = z_0 < z_1 < z_2 < \cdots < z_{2K-2} < z_{2K-1} = M+1$. If the first edge is of type 1, we set $\tilde{u}_i = a_{z_{2i-1}}$ and $\tilde{v}_j = a_{z_{2j}}$ for $1 \le i \le K-1$ and $1 \le j \le K-1$. Then relations (3.1) are satisfied with strict inequality. If the first edge is of type 2, we set $\tilde{u}_i = a_{z_{2i}}$ and $\tilde{v}_j = a_{z_{2j-1}}$ for $1 \le i \le K-1$ and $1 \le j \le K-1$. Then relations (3.2) are satisfied with strict inequality. In both cases the pair $\tilde{\mathbf{u}}$, $\tilde{\mathbf{v}}$ corresponds to a strict SDSQ $\mathbf{Q}$. Furthermore, $c'(\mathbf{Q})$ equals the weight of the path. It can be easily seen that this correspondence is one-to-one. □

The problem (4.1) can be solved using dynamic programming. In order to describe the algorithm, for each triple (n,k,i) with $1 \le n \le M+1$, $1 \le k \le 2K-1$, and $i = 1, 2$, let $\hat{W}_{k,i}(n)$ denote the minimum weight over all k-edge paths starting in 0 and ending in n that consist of edges of alternating types starting with type i. In addition,

let $\hat{W}_{0,i}(0) = 0$ for $i = 1, 2$. Then the following recurrence relation holds

$$\hat{W}_{k,i}(n) = \min_{k-1 \leq m < n} \left(\hat{W}_{k-1}(m) + \begin{cases} w'(m,n)_{(k+1)\%2+1} & \text{if } i = 1 \\ w'(m,n)_{k\%2+1} & \text{if } i = 2 \end{cases} \right) \tag{4.2}$$

for all $1 \leq k \leq 2K-1$, $k \leq n \leq M+1$, and $i = 1, 2$. For each $i = 1, 2$, the values $\hat{W}_{k,i}(n)$ are computed in lexicographical order of the pairs (k, n). At the end, the path achieving $\min_{i=1,2}(\hat{W}_{2K-1,1}(M+1), \hat{W}_{2K-1,2}(M+1))$ is selected.

The time complexity of the solution algorithm is $O(KM^2)$ if the weight of any edge can be calculated in constant time. The latter requirement can be satisfied if the preprocessing stage described in Section 2.4 is included.

Note that with similar reasoning as in Section 2.6, this algorithm can also be used when only a training sequence is available.

4.3 Monge Property and Time Complexity Reduction

In this section we show that the dynamic programming algorithm introduced in the previous section can be sped up by a factor of M, in the case where the linear classifier is optimal, i.e., condition $\mathbf{OC}_d$ holds. Similar to Section 2.5, we will organize the problem (4.2) for all nodes v as a series of matrix search problems where each matrix satisfies the Monge property, hence, using SMAWK [1], the search will be sped up.

Proposition 4. *For each k with $1 \leq k \leq K-1$, the problem (4.2) can be solved for all nodes $v \in V'$ in $O(M)$ time.*

Corollary 5. *The problem (4.1) can be solved in $O(KM)$ operations.*

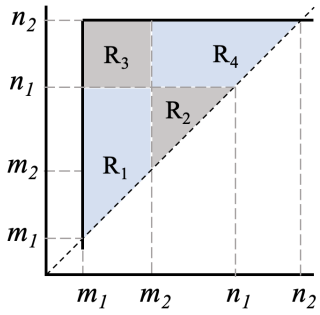


FIGURE 4.1: Illustration of regions created by two pairs of vertices from matrix $G'_{k,s}$.

Proof of Proposition 4. For each edge $e = (m, n) \in E'$, define

$$w'_p(m,n)_1 = \mu(T'(m,n)), w'_p(m,n)_2 = \mu(T(m,n)).$$

For each $i = 1, 2$, the values $\hat{W}_{k,i}(n)$ need to be computed. For the case of $i = 1$, $\hat{W}_{k,1}(n) = \min_{k-1 \le m < n} \hat{W}_{k-1}(m) + w'_p(m,n)_{(k+1)\%2+1}$, and at each $1 \le k \le 2K-1$, we know exactly which type of edge $w'_p(m,n)_{(k+1)\%2+1}$ corresponds to. With this, two possibilities exist for $\hat{W}_{k,1}(n)$.

$$\hat{W}_{k,1,1}(n) = \min_{k-1 \le m < n} \hat{W}_{k-1}(m) + w'_p(m,n)_1 \qquad \text{if } k\%2 = 1, \text{ or,} \tag{4.3}$$

$$\hat{W}_{k,1,2}(n) = \min_{k-1 \le m < n} \hat{W}_{k-1}(m) + w'_p(m,n)_2 \qquad \text{if } k\%2 = 0 \tag{4.4}$$

For each k, a matrix $G'_{k,s}$ is formed based on odd ($s = 1$) or even ($s = 2$) value of k, where $G'_{k,s}(m,n) = \hat{W}_{k-1}(m) + w'_p(m,n)_s$, for $k-1 \le m < n \le M+1$, and $G'_{k,s}(m,n) = \infty$, otherwise.

Then,

$$\hat{W}_{k,1,s}(n) = \min_{m:k-1\leq m<n} G'_{k,s}(m,n).$$

For the case of $i = 2$, similar results will be achieved. Next, it will be shown in Lemma 10 that matrix $G'_{k,s}$ satisfies the Monge property and hence, the matrix search problem can be solved in $O(M)$ for each value of k. Note that for each value of k, there is only one matrix to be searched as the structure is exactly known. □

Lemma 10. *The matrix $G'_{k,s}$ satisfies the Monge property.*

Proof. For the Monge property to hold for this matrix, for any arbitrary m_1, m_2, n_1, n_2, where $k-1 \leq m_1 < m_2 < n_1 < n_2 \leq M+1$ the following should hold

$$G'_{k,s}(m_1,n_1) + G'_{k,s}(m_2,n_2) \leq G'_{k,s}(m_1,n_2) + G'_{k,s}(m_2,n_1)$$

Regions created by these nodes are illustrated in Fig. 4.1. The above inequality reduces to

$$w'_p(m_1,n_1)_s + w'_p(m_2,n_2)_s \leq w'_p(m_1,n_2)_s + w'_p(m_2,n_1)_s. \tag{4.5}$$

Considering Fig. 4.1 and the fact that $\mu(A \cup B) = \mu(A) + \mu(B)$, this can be rewritten as follows

$$\begin{aligned} &\mu(R_1) + \mu(R_2) + \mu(R_2) + \mu(R_4) \\ &\leq \mu(R_1) + \mu(R_2) + \mu(R_3) + \mu(R_4) + \mu(R_2), \end{aligned}$$

which simplifies to $0 \leq \mu(R_3)$. This is true since $\mu(A) \geq 0$ for any measurable set A and the proof is complete.

□

4.4 Design of Optimal DSQ with Identical Encoders

In this section we address the design of an optimal DSQ under the constraint that the encoders be identical. This constraint was considered in some prior work [9, 16, 25]. The authors of [25] propose a design algorithm for this situation, which is globally optimal only when the training data is separable by the classifier line $\mathcal{L}$. Our contribution is to present a solution algorithm that is globally optimal even when the data is not linearly separable.

Let $\mathcal{Q}_{id}(K, \mathcal{A})$ denote the set of (K, K)-level DSQs determined by all possible pairs $\tilde{\mathbf{u}} = \tilde{\mathbf{v}}$, where $\tilde{\mathbf{u}} = (\tilde{u}_1, \cdots, \tilde{u}_{K-1})$, $\tilde{u}_i \in \mathcal{A}$, for all i. Then the problem that we seek to solve in this section is

$$\min_{\mathbf{Q} \in \mathcal{Q}_{id}(K, \mathcal{A})} c(\mathbf{Q}). \tag{4.6}$$

The above problem can be modelled as a minimum weight K-edge path problem in the WDAG $G'' = (V'', E'', w'')$ described next. Its set of vertices is $V'' = \{m : 0 \leq m \leq M+1\}$ and its set of edges is $E'' = \{(m,n) : 0 \leq m < n \leq M+1\}$. For each edge (m,n) its weight is $w''((m,n) = \min(\mu(T'(m,n)), \mu(T(m,n)))$. Then, the following result holds.

Proposition 5. *The problem (4.6) is equivalent to the problem of finding the minimum path in G' among all the paths from 0 to $M+1$ that have K edges.*

Proof. We will show that there is a one-to-one correspondence between the K-edge paths in G'' from 0 to $M+1$ and the DSQs in $\mathcal{Q}_{id}(K, \mathcal{A})$. Let $(z_0, z_1, \cdots, z_K)$ be the sequence of nodes on such a path. Then $0 = z_0 < z_1 < z_2 < \cdots < z_{K-1} < z_K = M+1$. Let $\tilde{u}_i = \tilde{v}_i = a_{z_i}$, for $1 \leq i \leq K-1$ and $1 \leq j \leq K-1$. Then the pair $(\tilde{\mathbf{u}}, \tilde{\mathbf{v}})$ determines a (K,K)-level DSQ $\mathbf{Q}$ in $\mathcal{Q}_{id}(K, \mathcal{A})$. The relevant cells of the DSQ are only the cells $S_{i,i}$, for $1 \leq i \leq K$. In addition, $c(S_{i,i}) = w''(z_{i-1}, z_i)$. It follows that the weight of the path is equal to $c(\mathbf{Q})$. The fact that the correspondence is one-to-one can be easily verified. □

The problem (4.6) can be solved using dynamic programming. Before describing the algorithm, let us introduce a few notations. For each pair (n,k) with $1 \leq n \leq M+1$, $1 \leq k \leq K$, let $\tilde{W}_k(n)$ denote the minimum weight over all k-edge paths starting in 0 and ending in n. In addition, let $\tilde{W}_0(0) = 0$. Then the following recurrence relation holds

$$\tilde{W}_k(n) = \min_{k-1 \leq m < n} \left(\tilde{W}_{k-1}(m) + w''(m,n) \right),$$

for all $1 \leq k \leq K$, $k \leq n \leq M+1$. The quantities $\tilde{W}_k(n)$ are computed in lexicographical order of the pairs (k, n). At the end, the path achieving $\tilde{W}_K(M+1)$ is selected.

The time complexity of the solution algorithm is $O(KM^2)$ if the weight of any edge can be calculated in constant time. The latter requirement can be satisfied if the preprocessing stage described in Section 2.4 is included.

Note that with similar reasoning to the Section 2.6, this algorithm can also be used when only a training sequence is available.

It is worth mentioning that in this case, if condition $\mathbf{OC}_d$ holds, the above search can be expedited by leveraging the Monge property. Since $w''((m,n) = \min(\mu(T'(m,n)), \mu(T(m,n)))$, there are two possibilities for $\tilde{W}_k(n)$.

$$\tilde{W}_{k,1}(n) = \min_{k-1 \leq m < n} \left(\tilde{W}_{k-1}(m) + \mu(T'(m,n)) \right), \text{ or,}$$
$$\tilde{W}_{k,2}(n) = \min_{k-1 \leq m < n} \left(\tilde{W}_{k-1}(m) + \mu(T(m,n)) \right),$$

where $\tilde{W}_k(n) = min(\tilde{W}_{k,1}(n), \tilde{W}_{k,2}(n))$. To find the best m for each n, and eventually finding the best path to node $n = M+1$, for each k, $1 \leq k \leq K$, we build two $(M-k+3) \times (M-k+2)$ search matrices G'_k and G_k, where $G'_k(m,n) = \tilde{W}_{k-1}(m) + \mu(T'(m,n))$ for $m \leq< n$, and $G'_k(m,n) = \infty$ otherwise, and where $G_k(m,n) = \tilde{W}_{k-1}(m) + \mu(T(m,n))$ for $m \leq< n$, and $G_k(m,n) = \infty$ otherwise. Hence, $\tilde{W}_k(n) = \min(\min_{k-1 \leq m < n} G'_k(m,n), \min_{k-1 \leq m < n} G_k(m,n))$. Next, we show that Monge property holds in matrices G'_k and G_k, hence the search in each of these matrices can be sped up to $O(M)$ operations.

For Monge property to hold for G'_k, the following should hold.

$$G'_k(m_1, n_1) + G'_k(m_2, n_2) \leq G'_k(m_1, n_2) + G'_k(m_2, n_1),$$

for all $k-1 \leq m_1 < m_2 < n_1 < n_2 <\leq M+1$. After replacing the values, this inequality simplifies to $\mu(T'(m_1, n_1)) + \mu(T'(m_2, n_2)) \leq \mu(T'(m_1, n_2)) + \mu(T'(m_2, n_1))$. It can be easily concluded that this inequality holds as this is exactly the inequality (4.5) in Section 4.3, for $s = 1$. With similar steps, for G_k, the Monge inequality will reduce to the inequality (4.5), for $s = 2$, and therefore, Monge property holds for this matrix. Hence, the search for the shortest path can be completed in $O(KM)$.

4.5 Experimental Results

In this chapter, we compare empirically the performance of the faster algorithm developed in Section 3.1 for the equal-rate case against two other approaches that assume identical encoders. For the proposed algorithm in this chapter, we use the acronym ONIE (Optimized Non-Identical Encoders). The other two approaches are 1) Uniform Length Quantization (UL) [9], and 2) Optimized Identical Encoders (OIE). For OIE we use the algorithm proposed in Section 4.4. For UL, the two identical encoders are obtained by partitioning the total (finite) range into equal-size intervals. In our experiments, the total range is $[-25, 25]$.

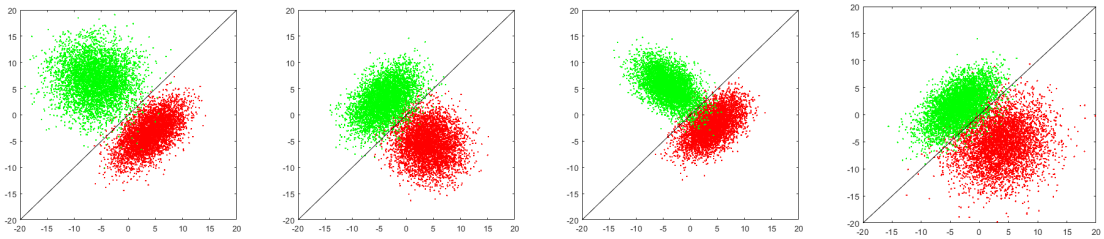
We considered two different scenarios of symmetric and asymmetric distributions. The class-conditional distributions used in our experiments are Gaussian, i.e., $P(\mathbf{x}|\ell = -1) = \mathcal{N}(\boldsymbol{\mu}_1, \boldsymbol{\Sigma}_1)$, $P(\mathbf{x}|\ell = 1) = \mathcal{N}(\boldsymbol{\mu}_2, \boldsymbol{\Sigma}_2)$, and the probabilities of the

two classes are equal. For the symmetric case we consider $\boldsymbol{\mu}_1 = -\boldsymbol{\mu}_2 = (\xi_1, -\xi_1)$ and $\boldsymbol{\Sigma}_2 = \boldsymbol{\Sigma}_1^\tau$, where $\begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix}^\tau = \begin{bmatrix} a_{22} & a_{12} \\ a_{21} & a_{11} \end{bmatrix}$.

For each scenario we employ four different distributions. Their corresponding parameters are presented in Table 4.1 for the asymmetric case and in Table 4.2 for the symmetric case. Note that for all distributions in our study, $\mathcal{L}$ is the decision boundary of the optimal linear classifier. For the symmetric case, both conditions **OC** and **S** hold implying that the optimal equal-rate SDSQ must be staggered in view of Theorem 4. Therefore, we expect ONIE to have a performance close to the optimum.

Asymmetric				
#	$\boldsymbol{\mu}_1^T$	$\boldsymbol{\mu}_2^T$	$\boldsymbol{\Sigma}_1$	$\boldsymbol{\Sigma}_2$
1	$\begin{matrix} 4 \\ -3.6 \end{matrix}$	$\begin{matrix} -6 \\ 6.75 \end{matrix}$	$\begin{matrix} 9 & 5.15 \\ 5.15 & 9.57 \end{matrix}$	$\begin{matrix} 14 & -1.03 \\ -1.03 & 14.88 \end{matrix}$
2	$\begin{matrix} 4 \\ -5.44 \end{matrix}$	$\begin{matrix} -4 \\ 2.8 \end{matrix}$	$\begin{matrix} 10 & -1.03 \\ -1.03 & 10.63 \end{matrix}$	$\begin{matrix} 9 & 4.12 \\ 4.12 & 9.57 \end{matrix}$
3	$\begin{matrix} 4 \\ -2.04 \end{matrix}$	$\begin{matrix} -4 \\ 5.51 \end{matrix}$	$\begin{matrix} 8 & 3.77 \\ 3.77 & 7.12 \end{matrix}$	$\begin{matrix} 8 & -3.77 \\ -3.77 & 7.12 \end{matrix}$
4	$\begin{matrix} 3.5 \\ -5.3 \end{matrix}$	$\begin{matrix} -3.5 \\ 1.7 \end{matrix}$	$\begin{matrix} 17 & 2 \\ 2 & 17 \end{matrix}$	$\begin{matrix} 9 & 5 \\ 5 & 9 \end{matrix}$

TABLE 4.1: Parameters of the data distributions considered in the asymmetric scenario.

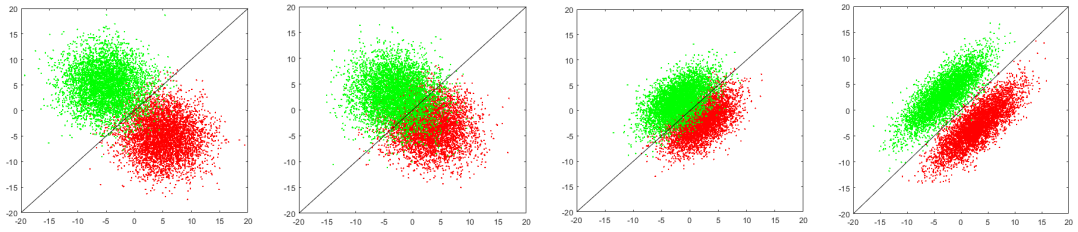


(A) 1 (B) 2 (C) 3 (D) 4

FIGURE 4.2: Data distributions used in the asymmetric scenario.

Symmetric		
#	$\boldsymbol{\mu}_1^T$	$\boldsymbol{\Sigma}_1$
1	$\begin{bmatrix} 5 \\ -5 \end{bmatrix}$	$\begin{bmatrix} 14 & -1 \\ -1 & 14 \end{bmatrix}$
2	$\begin{bmatrix} 3 \\ -3 \end{bmatrix}$	$\begin{bmatrix} 14 & -1 \\ -1 & 14 \end{bmatrix}$
3	$\begin{bmatrix} 2 \\ -2 \end{bmatrix}$	$\begin{bmatrix} 9 & 5 \\ 5 & 9 \end{bmatrix}$
4	$\begin{bmatrix} 3 \\ -3 \end{bmatrix}$	$\begin{bmatrix} 14 & 11 \\ 11 & 14 \end{bmatrix}$

TABLE 4.2: Parameters of the data distributions considered in the symmetric scenario.



(A) 1 (B) 2 (C) 3 (D) 4

FIGURE 4.3: Data distributions used in the symmetric scenario.

For OIE we use training sets of size $N = 10000$, while for ONIE we consider two sizes for the training sequence, namely $N = 10000$ (ONIE 10k) and $N = 4000$ (ONIE 4k). All four approaches (UL, OIE, ONIE10k, ONIE4k) are tested using test sequences with $N = 10000$ examples. The graphical illustration of the 10000-size training sequence of each distribution in the asymmetric and symmetric scenarios is shown in Fig. 4.2 and Fig. 4.3, respectively. Different colors represents different classes.

Figs. 4.4 and 4.5 show the plot of $\rho - \rho_u$ measured on the test sequence versus R for the four compared approaches in the asymmetric and symmetric cases, respectively. We notice that for each of UL, OIE and ONIE10k there is a threshold rate R_0, which

also depends on the distribution, after which the value of $\rho - \rho_u$ becomes very close to 0. This is because as R increases, the number of relevant cells increases and the number of training points covered by the cost regions (i.e., the triangular regions used in the computation of the modified cost) decreases. On the other hand, for smaller rates we observe a significant difference in performance between UL, OIE and ONIE10k, the latter being the best. The superiority of ONIE10k was expected in the symmetric case, but it is pleasing to see that it also holds for the asymmetric distributions in our study. This observation indicates that the conclusion of Theorem 2 might hold under weaker assumptions about the data distribution. Another interesting observation is that the perfomance of ONIE4k is very similar to that of ONIE10k in the symmetric scenario. However, for two of the asymmetric distributions this conclusion holds only for small rates, while for the larger rates the value of $\rho - \rho_u$ for ONIE4k plateaus above 0 suggesting that overfitting occurs. Finally, we see that UL performs very poorly if the rate is not sufficiently high.

4.6 Conclusion

In this chapter, we propose a faster algorithm of the DSQ design for the case of equal quantizer rates. With the support of properties discussed in Chapter 3 for continuous sources, we derive similar properties for the discrete sources case. Mainly, we show that when a certain symmetry condition holds for the joint distribution of sources and labels, the optimal staggered DSQ that minimizes the expected classification error is also optimal among the set of all DSQs. Further, by considering the modified cost and imposing the staggering thresholds condition, we simplify the graph model of Chapter 2 and propose a faster dynamic programming algorithm that finds the best staggered

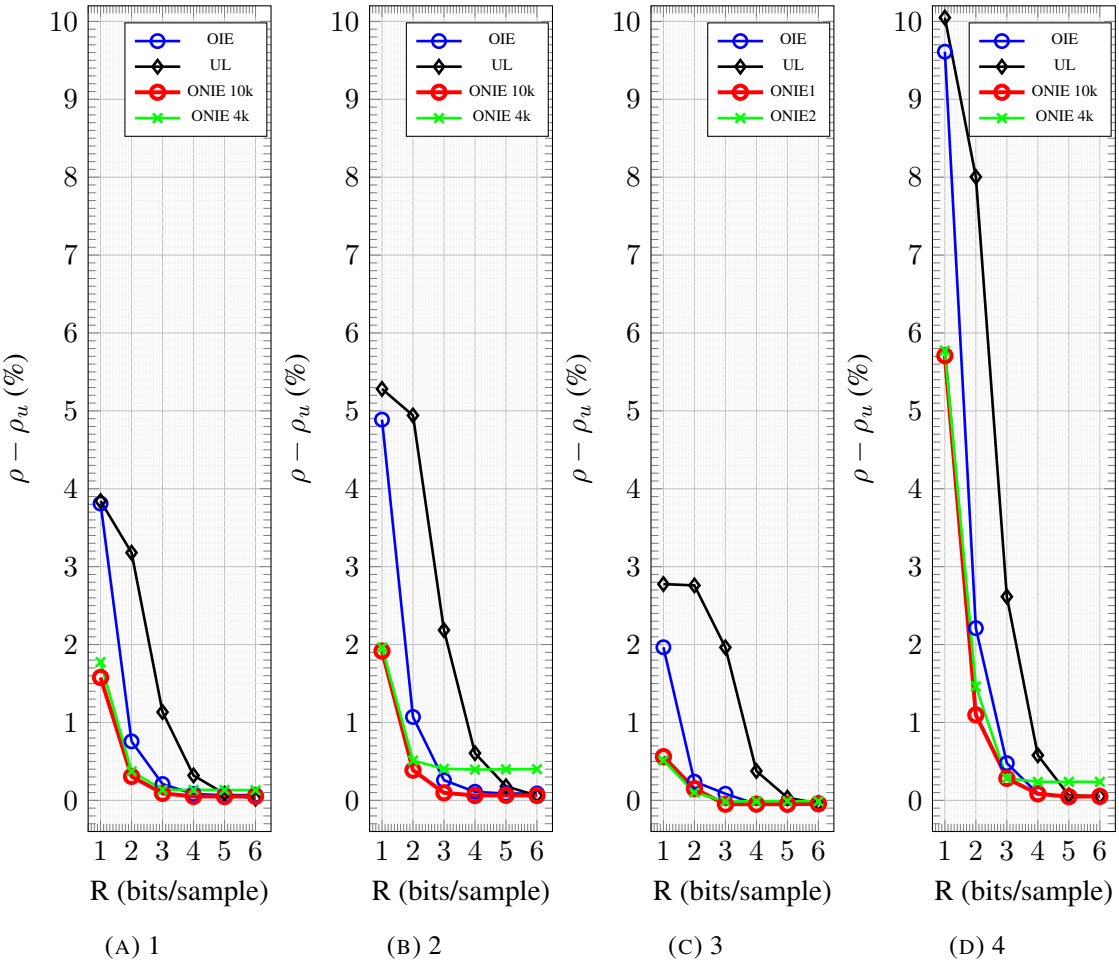
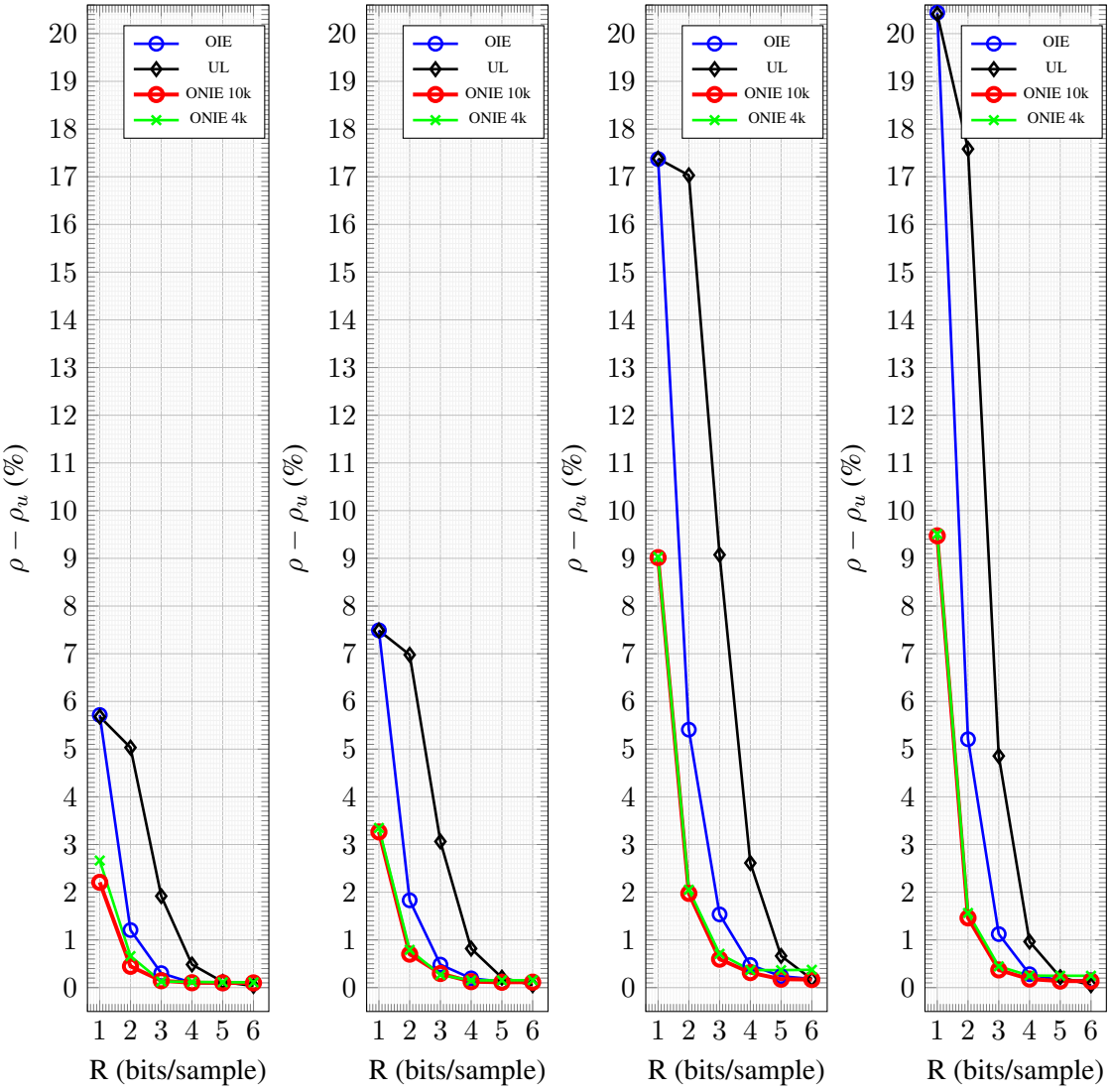


FIGURE 4.4: Comparison of the proposed algorithm for the asymmetric data against OIE and UL. Results of our algorithm trained with 4000 training examples is shown in green.

DSQ minimizing the modified cost for a given distribution. Further, we show that if the linear classifier is the optimal classifier, this algorithm can be further expedited by a factor of source alphabet size, utilizing the Monge property. Moreover, we propose an algorithm for the design of optimal DSQs with identical encoders. Finally, we compare the performance of staggered DSQs with other DSQs with identical encoders, designed using a training sequence. Experimental results for training data derived from both



(A) 1 (B) 2 (C) 3 (D) 4

FIGURE 4.5: Comparison of the proposed algorithm for the symmetric data against OIE and UL. Results of our algorithm trained with 4000 training examples is shown in green.

symmetric and asymmetric distributions indicate the superiority of staggered DSQs over DSQs with identical encoders.

Chapter 5

Application to Decentralized Detection of Vector Sources

In this chapter we show that the theory and algorithms developed in the previous chapters can be applied to certain scenarios of the more general problem of quantizer design for the decentralized detection of two vector sources [52]. In this problem, the classifier at the decoder is not given. The problem is to design the two encoders and the joint decoder such that the classification error to be minimized. We show that the developments of the previous chapters can be applied to this scenario after each vector source is subjected to an appropriate transformation. More specifically, after applying this transformation, each source becomes scalar in the transformed domain and the optimal classifier for the unquantized data in the transformed space is linear.

The main contribution of this chapter is the fastest globally optimal solution to date to the problem of decentralized binary hypothesis testing with the probability of error criterion when the vector sources are conditionally independent given the hypothesis. For this scenario, Tsitsiklis proved that if an optimal encoding strategy exists, then there

is an optimal encoding strategy where each encoder is a "threshold rule" for the corresponding likelihood ratio. In other words, each encoder is a scalar quantizer with convex cells in the likelihood ratio domain. He noted that when the sources are discrete, an exhaustive search over all possible threshold rules solves the problem optimally. This procedure requires $O(N^{K_1+K_2}(N+K_1K_2))$ operations, where N is the size of the largest alphabet of the input vectors [52]. We propose a considerably faster globally optimal solution with time complexity $O(K_1K_2N^3)$, which is a significant improvement in comparison to the exhaustive search algorithm. Furthermore, we show that for the case of equal quantizer rates, by imposing the staggering structure the time complexity of the solution can be decreased to $O(KN)$. To achieve these improvements, we leverage Tsitsiklis's result regarding the optimality of the threshold rule and reduce the design problem to a problem in a transformed domain (related to the likelihood ratio domain), where each encoder is a scalar quantizer with convex cells. Next we show that the problem in the transformed domain is equivalent to the optimization problem addressed in the previous chapters and therefore can be solved using the proposed algorithms.

The chapter has the following structure. We first formulate the general detection problem and show how it relates to the problem solved in the previous chapters in Section 5.1. In Section 5.2, we describe a particular scenario when the detection problem can be reduced to the DSQ design problem of previous chapters. Next, in Section 5.3, we address the case of conditionally independent sources given the class labels. In Section 5.4, we discuss some possible real world applications for the proposed algorithms. Finally, Section 5.5 concludes the chapter.

5.1 Formulation of the General Detection Problem

Consider two distributed sensor nodes as in Fig. 2.1. The data collected by each node is quantized and the message is sent to the server node. The server node jointly decodes the two messages and outputs one of the two possible class labels, 1 or -1. For $k = 1, 2$, the input to the encoder k is a vector denoted by $\mathbf{y}_k$ taking values in a set $\mathcal{Y}_k \subset \mathbb{R}^{d_k}$, where $d_k \geq 1$. The encoder k is denoted by $\varphi_k : \mathbb{R}^{d_k} \to \{1, \cdots, K_k\}$. The joint decoder is $\delta : \{1, \cdots, K_1\} \times \{1, \cdots, K_2\} \to \{-1, 1\}$. This scenario accommodates both cases when the encoder of a node operates on one sample at a time, i.e., $d_k = 1$, (scalar quantizer) or on blocks of samples of size $d_k > 1$ (vector quantizer). Our framework also covers the case when each sensor acquires data from more than one scalar source and the encoder is applied on blocks of samples from all sources. Also note that d_1 and d_2 do not need to be equal.

We will refer to the random vectors that generate $\mathbf{y}_1$ and $\mathbf{y}_2$ as being our vector sources and will assume that the joint distribution of the two vector sources and of the class label is known. The general problem we consider in this chapter is the problem of quantizers design for decentralized detection, i.e., to find the optimal triple $(\varphi_1, \varphi_2, \delta)$ with fixed K_1 and K_2, that minimizes the probability of detection error, i.e.,

$$\min_{(\varphi_1, \varphi_2, \delta)} Pr(\hat{\ell} \neq \ell) \tag{5.1}$$

where $\hat{\ell} = \delta(\varphi_1(\mathbf{y}_1), \varphi_2(\mathbf{y}_2))$

We will consider both the case when the two sources are discrete and the case when the sources are continuous. In the first case, we denote by $P'(\mathbf{y}_1, \mathbf{y}_2, \ell)$ the joint pmf of the sources and the label. In the second case, we denote by $f'(\mathbf{y}_1, \mathbf{y}_2, \ell)$ the joint pdf of

the sources and the label.

Recall that for the problem studied so far in this thesis, the input to the encoder of sensor k is a scalar value x_k, and the joint decoder β maps each pair of indexes (i, j) to a pair of reconstruction values, which is further fed to the fixed linear classifier γ given below.

$$\gamma(x_1, x_2) = \begin{cases} 1 & \text{if } x_1 \leq x_2 \\ -1 & \text{if } x_1 > x_2 \end{cases}. \tag{5.2}$$

The relevant optimization problem corresponding to this scenario is to find the triple $(\alpha_1, \alpha_2, \beta)$ that minimizes the probability of classification error, i.e.,

$$\min_{(\alpha_1, \alpha_2, \beta)} Pr(\gamma(\hat{\mathbf{x}}) \neq \ell) \tag{5.3}$$

where $\hat{\mathbf{x}} = \beta(\alpha_1(x_1), \alpha_2(x_2))$ and $\beta(i, j) \in \alpha_1^{-1}(i) \times \alpha_2^{-1}(j)$ for all (i, j) [1]. Note that the minimization in (5.3) is over all pairs of encoders (α_1, α_2) without any constraint on their structure, while in this thesis we addressed the case when the encoders' partitions are constrained to have convex cells.

When the inputs at the two encoders are scalars, the main difference between the problems (5.1) and (5.3) is that in the latter it is assumed that the given linear classifier γ is applied to the quantized output in order to detect the class label, while the problem (5.1) does not impose such a constraint. We show in this chapter that if γ is the optimal classifier for the unquantized scalar sources, then this constraint does not preclude the

[1] This is an important constraint that we imposed in the previous sections, which helped solving the optimization problem. Without this constraint, problems (5.1) (for scalar sources) and (5.3) are equivalent since for any decision rule δ it is possible to find an appropriate mapping β such that $\delta = \gamma \circ \beta$.

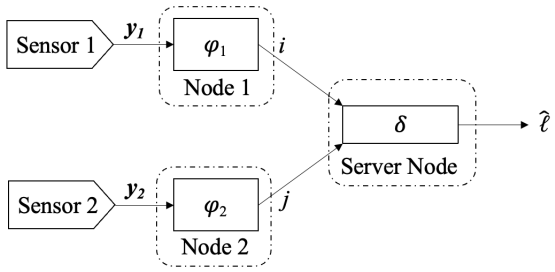


FIGURE 5.1: System block diagram of the detection scenario.

optimality, i.e., if $(\alpha_1, \alpha_2, \beta)$ is a solution to the problem (5.3) then $(\alpha_1, \alpha_2, \gamma \circ \beta)$ is a solution to the problem (5.1). Then the algorithms developed so far to solve the problem (5.3) within the class of encoders with convex cells, can also be used to solve the variant of (5.1) with the corresponding restriction on the encoders' structure. Moreover, the results concerning the structure of the optimal partitions in the equal-rate case derived for the restricted variant of problem (5.3) also hold for the restricted variant of problem (5.1).

Furthermore, we show that when the inputs to the sensors (vectors or scalars) are conditionally independent given the class, the algorithms proposed in previous chapters can be used to solve the problem (5.1) without any constraints. This will lead to a much faster globally optimal solution. In addition, the results about the structure of the equal-rate DSQ at optimality can be used to derive structural optimality properties for problem (5.1).

5.2 Results

Let $\gamma_0 : \mathcal{Y}_1 \times \mathcal{Y}_2 \to \{-1, 1\}$ denote the optimal classifier for the unquantized vector sources. In other words, in the discrete case γ_0 is

$$\gamma_0(\mathbf{y}_1, \mathbf{y}_2) = \begin{cases} 1 & \text{if } P'(\mathbf{y}_1, \mathbf{y}_2, 1) \geq P'(\mathbf{y}_1, \mathbf{y}_2, -1) \\ -1 & \text{if } P'(\mathbf{y}_1, \mathbf{y}_2, 1) < P'(\mathbf{y}_1, \mathbf{y}_2, -1) \end{cases},$$

while for the continuous case P' is replaced by f'. Let Γ_+ denote the decision region for class 1 and let Γ_- denote the decision region for class -1.

Lemma 11. *If $(\varphi_1, \varphi_2, \delta)$ is a solution to problem (5.1), then there is a function $\psi : \{1, \cdots, K_1\} \times \{1, \cdots, K_2\} \to \mathcal{Y}_1 \times \mathcal{Y}_2$ such that*

$$\delta(i, j) = \gamma_0(\psi(i, j)) \tag{5.4}$$

and $\psi(i, j) \in \varphi_1^{-1}(i) \times \varphi_2^{-1}(j)$ for all (i, j).

Proof. It is known that the optimal decision rule δ must satisfy

$$\delta(i, j) = \delta_{opt}(i, j) = \arg\max_{\ell} Pr(\ell | i, j) \tag{5.5}$$

for all (i, j). Assume that the ties in the above relation are resolved in favour of class 1. Consider the case of discrete sources (for continuous sources, the proof is similar). Then

$$\delta_{opt}(i, j) = \arg\max_{\ell} P(i, j, \ell) = \arg\max_{\ell} \sum_{\mathbf{y}_1 \in \varphi_1^{-1}(i)} \sum_{\mathbf{y}_2 \in \varphi_2^{-1}(j)} P'(\mathbf{y}_1, \mathbf{y}_2, \ell). \tag{5.6}$$

If $\delta_{opt}(i,j) = 1$, it follows that $\varphi_1^{-1}(i) \times \varphi_2^{-1}(j) \cap \Gamma_+ \neq \emptyset$. Therefore, if we choose $\psi(i,j) \in \varphi_1^{-1}(i) \times \varphi_2^{-1}(j) \cap \Gamma_+$, relation (5.4) is satisfied. Similarly, when $\delta_{opt}(i,j) = -1$, we have $\varphi_1^{-1}(i) \times \varphi_2^{-1}(j) \cap \Gamma_- \neq \emptyset$. If we choose $\psi(i,j) \in \varphi_1^{-1}(i) \times \varphi_2^{-1}(j) \cap \Gamma_-$, relation (5.4) is satisfied. □

Next we introduce a property that will be used in the sequel.

Property A: There exist functions $T_k : \mathcal{Y}_k \to \subseteq \mathbb{R}$, for $k = 1, 2$, and $h : \mathcal{X}_1 \times \mathcal{X}_2 \to \mathbb{R}$, where $\mathcal{X}_k = T_k(\mathcal{Y}_k)$ such that

- in the discrete case, $P'(\mathbf{y}_1, \mathbf{y}_2, \ell) = h(T_1(\mathbf{y}_1), T_2(\mathbf{y}_2), \ell)$;
- in the continuous case, $f'(\mathbf{y}_1, \mathbf{y}_2, \ell) = h(T_1(\mathbf{y}_1), T_2(\mathbf{y}_2), \ell)$

and

- $h(x_1, x_2, 1) > h(x_1, x_2, -1)$ if and only if $x_1 < x_2$
- $h(x_1, x_2, 1) < h(x_1, x_2, -1)$ if and only if $x_1 > x_2$
- $h(x_1, x_2, 1) = h(x_1, x_2, -1)$ if and only if $x_1 = x_2$.

Proposition 6. *Assume that Property* **A** *is satisfied. Then there is a solution* $(\alpha_1, \alpha_2, \beta)$ *to problem (5.3) in the transformed domain of* $(T_1(\mathbf{y}_1), T_2(\mathbf{y}_2))$ *such that* $(\alpha_1 \circ T_1, \alpha_2 \circ T_2, \gamma \circ \beta)$ *is a solution to problem (5.1) in the domain of* $(\mathbf{y}_1, \mathbf{y}_2)$.

Remark 8. *Property* **A** *means that the classification problem for the pair of unquantized vector sources* $(\mathbf{y}_1, \mathbf{y}_2)$ *can be converted to the classification problem for a pair of scalar sources* (x_1, x_2) *(by applying a separate transformation to each vector) for which the*

optimal classifier is the linear classifier γ. According to Proposition 6, in such a case, solving problem (5.1) in the original domain reduces to solving problem (5.3) in the transformed domain.

Proof of Proposition 6. Let $\mathcal{X}_k = T_k(\mathcal{Y}_k)$, for $k = 1, 2$. We will use the following two claims, which are proved at the end.

Claim 1. There are functions $\alpha_k : \mathcal{X}_k \to \{1, \cdots, K_k\}$, $k = 1, 2$, and an optimal solution $(\varphi_1, \varphi_2, \delta)$ to problem (5.1) such that

$$\varphi_k(\mathbf{y}_k) = \alpha_k(T_k(\mathbf{y}_k)) \tag{5.7}$$

for all $\mathbf{y}_k$, $k = 1, 2$.

Claim 2. If $(\alpha_1 \circ T_1, \alpha_2 \circ T_2, \delta)$ is a solution to problem (5.1), then there is a function $\beta : \{1, \cdots, K_1\} \times \{1, \cdots, K_2\} \to \mathcal{X}_1 \times \mathcal{X}_2$ such that

$$\delta(i, j) = \gamma(\beta(i, j)) \tag{5.8}$$

and $\beta(i, j) \in \alpha_1^{-1}(i) \times \alpha_2^{-1}(j)$ for all (i, j).

According to Claims 1 and 2, there is solution $(\varphi_1, \varphi_2, \delta)$ to problem (5.1) and a triple $(\alpha_1, \alpha_2, \beta)$ satisfying the conditions of problem (5.3) such that

$$(\varphi_1, \varphi_2, \delta) = (\alpha_1 \circ T_1, \alpha_2 \circ T_2, \gamma \circ \beta). \tag{5.9}$$

Note that the transformation (T_1, T_2) incurs a probability distribution in the transformed domain $\mathcal{X} = \mathcal{X}_1 \times \mathcal{X}_2$, where $P(\mathbf{x}, \ell)$, respectively, $f(\mathbf{x}, \ell)$, denotes the joint pmf in the discrete case, respectively pdf in the continuous case, of $\mathbf{x}, \ell$, for $\mathbf{x} = (x_1, x_2) \in \mathcal{X}_1 \times \mathcal{X}_2$ and $\ell \in \{-1, 1\}$. To be more specific

$$P(\mathbf{x}, \ell) = \sum_{\mathbf{y}_1 \in T_1^{-1}(x_1)} \sum_{\mathbf{y}_2 \in T_2^{-1}(x_2)} P'(\mathbf{y}_1, \mathbf{y}_2, \ell)$$
$$f(\mathbf{x}, \ell) = \int_{T_1^{-1}(x_1)} \int_{T_2^{-1}(x_2)} f'(\mathbf{y}_1, \mathbf{y}_2, \ell)\, d\mathbf{y}_2\, d\mathbf{y}_1$$

for all $\mathbf{x} \in \mathcal{X}_1 \times \mathcal{X}_2$ and $\ell \in \{-1, 1\}$.

According to (5.9), the classification error ρ_2, of the DSQ $(\alpha_1, \alpha_2, \beta)$ in the transformed domain $\mathcal{X} = \mathcal{X}_1 \times \mathcal{X}_2$, satisfies

$$\rho_2 = Pr(\gamma(\beta(\alpha_1(x_1), \alpha_2(x_2)) \neq \ell) = Pr(\delta(\psi(\varphi_1(\mathbf{y}_1), \varphi_2(\mathbf{y}_2)))) = \rho_1, \tag{5.10}$$

where ρ_1 denotes the probability of detection error of $(\varphi_1, \varphi_2, \delta)$. Since ρ_1 is minimized, it follows that ρ_2 is minimized too. Therefore, $(\alpha_1, \alpha_2, \beta)$ is a solution to (5.3) in the transformed domain $\mathcal{X} = \mathcal{X}_1 \times \mathcal{X}_2$.

Proof of Claim 1. Consider first the discrete case. Consider an optimal triple $(\varphi_1, \varphi_2, \delta)$ and define

$$cost_1(\mathbf{y}_1, i) = \sum_{j:\delta(i,j)=1} \sum_{\mathbf{y}_2 \in \varphi_2^{-1}(j)} P'(\mathbf{y}_1, \mathbf{y}_2, -1) + \sum_{j:\delta(i,j)=-1} \sum_{\mathbf{y}_2 \in \varphi_2^{-1}(j)} P'(\mathbf{y}_1, \mathbf{y}_2, 1) \tag{5.11}$$

Then the probability of error for $(\varphi_1, \varphi_2, \delta)$ equals

$$\sum_{i=1}^{K_1} \left(\sum_{j:\delta(i,j)=1} \sum_{\mathbf{y}_1 \in \varphi_1^{-1}(i)} \sum_{\mathbf{y}_2 \in \varphi_2^{-1}(j)} P'(\mathbf{y}_1, \mathbf{y}_2, -1) + \sum_{j:\delta(i,j)=-1} \sum_{\mathbf{y}_1 \in \varphi_1^{-1}(i)} \sum_{\mathbf{y}_2 \in \varphi_2^{-1}(j)} P'(\mathbf{y}_1, \mathbf{y}_2, 1) \right). \tag{5.12}$$

By interchanging the order of the summations over j and over $\mathbf{y}_1$, and applying (5.11), we further obtain

$$Pr(\delta(i,j) \neq \ell) = \sum_{i=1}^{K_1} \sum_{\mathbf{y}_1 \in \varphi_1^{-1}(i)} cost_1(\mathbf{y}_1, i). \tag{5.13}$$

Since $(\varphi_1, \varphi_2, \delta)$ is optimal, it follows that φ_1 must satisfy

$$\varphi_1(\mathbf{y}_1) = \arg\min_i cost_1(\mathbf{y}_1, i). \tag{5.14}$$

Since $P'(\mathbf{y}_1, \mathbf{y}_2, \ell) = h(T_1(\mathbf{y}_1), T_2(\mathbf{y}_2), \ell)$, it follows that if $T_1(\mathbf{y}_1) = T_1(\mathbf{y}_1')$ then $cost_1(\mathbf{y}_1, i) = cost_1(\mathbf{y}_1', i)$ for all i. Hence, if the minimum in (5.14) is achieved in a unique value, at optimality we must have $\varphi_1(\mathbf{y}_1) = \varphi_1(\mathbf{y}_1')$. If the minimum in (5.14) is achieved in multiple values, there are multiple encoder functions φ_1 satisfying the optimality condition and there is one for which $\varphi_1(\mathbf{y}_1)) = \varphi_1(\mathbf{y}_1')$ for all $\mathbf{y}_1$ and $\mathbf{y}_1'$ such that $T_1(\mathbf{y}_1) = T_1(\mathbf{y}_1')$. This means that for each $x_1 \in \mathcal{X}_1$, all vectors $\mathbf{y}_1$ that are mapped by T_1 to x_1, are also mapped by φ_1 to the same index i. By defining $\alpha_1(x_1) = i$, we obtain $\varphi_1(\mathbf{y}_1) = \alpha_1(T_1(\mathbf{y}_1))$ for all $\mathbf{y}_1$.

The proof of the fact that there is a function $\alpha_2 : \mathcal{X}_2 \to \{1, \cdots, K_2\}$ such that $\varphi_2(\mathbf{y}_2) = \alpha_2(T_2(\mathbf{y}_2))$ for all $\mathbf{y}_2 \in \mathcal{Y}_2$ is similar. The continuous case can be proved similarly by replacing the summations by integrals and the pmf P' by the pdf f'.

Proof of Claim 2. According to Lemma 11, there is a function $\psi : \{1, \cdots, K_1\} \times \{1, \cdots, K_2\} \to \mathcal{Y}_1 \times \mathcal{Y}_2$ such that

$$\delta(i,j) = \gamma_0(\psi(i,j)) \tag{5.15}$$

and $\psi(i,j) \in T_1^{-1}(\alpha_1^{-1}(i)) \times T_2^{-1}(\alpha_2^{-1}(j))$ for all (i,j). Let $\psi_k(i,j)$ denote the k-th component of $\psi(i,j)$ for $k = 1, 2$. In other words, $\psi(i,j) = (\psi_1(i,j), \psi_2(i,j))$. Since Property **A** holds, it follows that the optimal classifier γ_0 for the unquantized vector sources satisfies

$$\gamma_0(\mathbf{y}_1, \mathbf{y}_2) = \gamma(T_1(\mathbf{y}_1), T_2(\mathbf{y}_2)). \tag{5.16}$$

By combining (5.15) and (5.16), we obtain

$$\delta(i,j) = \gamma_0(\psi_1(i,j), \psi_2(i,j)) = \gamma(T_1(\psi_1(i,j)), T_2(\psi_2(i,j))). \tag{5.17}$$

Define $\beta(i,j) = (T_1(\psi_1(i,j)), T_2(\psi_2(i,j)))$. Then condition (5.8) holds. In addition, $\beta(i,j) \in T_1(T_1^{-1}(\alpha_1^{-1}(i))) \times T_2(T_2^{-1}(\alpha_2^{-1}(j))) = \alpha_1^{-1}(i) \times \alpha_2^{-1}(j)$ for all (i,j) and the proof is completed.

□

5.3 Results for Conditionally Independent Sources Given the Class Label

In this section we assume that the vector sources are conditionally independent given the class label. For $\ell \in \{1, -1\}$, let $P_0(\ell)$ denote the prior probability of class ℓ. For the discrete case, we denote by $P'(\mathbf{y}_1, \mathbf{y}_2, \ell)$ the joint probability mass function of $\mathbf{y}_1, \mathbf{y}_2$ and of the class label ℓ, and by $P_k(\mathbf{y}_k|\ell)$ we denote the conditional probability of $\mathbf{y}_k$ given ℓ. In the continuous case, $f'(\mathbf{y}_1, \mathbf{y}_2, \ell)$ denotes the joint pdf of $\mathbf{y}_1, \mathbf{y}_2$ and ℓ, while by $f_k(\mathbf{y}_k|\ell)$ denotes the conditional pdf of $\mathbf{y}_k$ given ℓ. Then the conditional independence given the class label means that, in the discrete case we have

$$P'(\mathbf{y}_1, \mathbf{y}_2, \ell) = P_1(\mathbf{y}_1|\ell)P_2(\mathbf{y}_2|\ell)P_0(\ell),$$

and in the continuous case we have

$$f'(\mathbf{y}_1, \mathbf{y}_2, \ell) = f_1(\mathbf{y}_1|\ell)f_2(\mathbf{y}_2|\ell)P_0(\ell),$$

For $k = 1, 2$, let $L_k(\mathbf{y}_k)$ denote the likelihood function, i.e., $L_k(\mathbf{y}_k) = \frac{P_k(\mathbf{y}_k|-1)}{P_k(\mathbf{y}_k|1)}$ in the discrete case and $L_k(\mathbf{y}_k) = \frac{f_k(\mathbf{y}_k|-1)}{f_k(\mathbf{y}_k|1)}$ in the continuous case. According to [52], there is an optimal solution $(\varphi_1, \varphi_2, \delta)$ to problem (5.1) where each encoder φ_k is a "threshold rule" for the corresponding likelihood ratio $L_k(\mathbf{y}_k)$. Note that $L_k(\mathbf{y}_k)$ takes values in the interval $[0, \infty]$ of the extended real line, $\mathbb{R} \cup \{-\infty, +\infty\}$. This means that there are two partitions of $[0, \infty]$ consisting of intervals, $\mathcal{P}'_1 = \{U'_i\}_{1 \le i \le K_1}$ and $\mathcal{P}'_2 = \{V'_j\}_{1 \le j \le K_2}$,

such that

$$\varphi_1(\mathbf{y}_1) = i \text{ iff } L_1(\mathbf{y}_1) \in U_i' \tag{5.18}$$

$$\varphi_2(\mathbf{y}_2) = j \text{ iff } L_2(\mathbf{y}_2) \in V_j'. \tag{5.19}$$

Further, we define the transformations $T_k : \mathcal{Y}_k \to \subseteq \mathbb{R}$, for $k = 1, 2$, such that

$$T_1(\mathbf{y}_1) = P_0(-1)L_1(\mathbf{y}_1) \tag{5.20}$$

$$T_2(\mathbf{y}_2) = \frac{P_0(1)}{L_2(\mathbf{y}_2)}. \tag{5.21}$$

Denote $\mathcal{X}_1 = T_1(\mathcal{Y}_1)$ and $\mathcal{X}_2 = T_2(\mathcal{Y}_2)$. The transformation (T_1, T_2) incurs a probability distribution in the transformed domain $\mathcal{X} = \mathcal{X}_1 \times \mathcal{X}_2$, where $P(\mathbf{x}, \ell)$, respectively, $f(\mathbf{x}, \ell)$, denotes the joint pmf, respectively, pdf, of $\mathbf{x}, \ell$, for $\mathbf{x} = (x_1, x_2) \in \mathcal{X}_1 \times \mathcal{X}_2$ and $\ell \in \{-1, 1\}$ defined as in the proof of Proposition 6.

Proposition 7. *Assume that* $\mathbf{y}_1$ *and* $\mathbf{y}_2$ *are conditionally independent given the class label. Then there is a solution* $(\alpha_1, \alpha_2, \beta)$ *to problem (5.3) in the transformed domain of* $(T_1(\mathbf{y}_1), T_2(\mathbf{y}_2))$ *such that for each* $k = 1, 2$ *the partition of* α_k *is formed of convex cells (i.e., intersections of* $\mathcal{X}_k$ *with an interval of the real line) and* $(\alpha_1 \circ T_1, \alpha_2 \circ T_2, \gamma \circ \beta)$ *is a solution to problem (5.1) in the domain of* $(\mathbf{y}_1, \mathbf{y}_2)$.

Proof. According to the aforementioned discussion, there is a solution to (5.1) $(\varphi_1, \varphi_2, \delta)$ such that relations (5.18) and (5.19) hold. Then

$$\varphi_1(\mathbf{y}_1) = i \text{ iff } T_1(\mathbf{y}_1) \in U_i \tag{5.22}$$

$$\varphi_2(\mathbf{y}_2) = j \text{ iff } T_2(\mathbf{y}_2) \in V_j, \tag{5.23}$$

where $U_i = \{P_0(-1)z : z \in U_i'\}$ and $V_j = \{P_0(1)/z : z \in V_j'\}$. Hence, U_i and V_j are also intervals. Let us define the mappings $\alpha_k : \mathcal{X}_k \to \{1, \cdots, K_k\}$, $k = 1, 2$. Then, according to (5.22) and (5.23), we have

$$\varphi_k = \alpha_k \circ T_k, \ k = 1, 2.$$

It is known [51] that the optimal classifier for the unquantized data is

$$\gamma_0(\mathbf{y}_1, \mathbf{y}_2) = \gamma(T_1(\mathbf{y}_1), T_2(\mathbf{y}_2)) \tag{5.24}$$

for all $\mathbf{y}_1, \mathbf{y}_2$. Then Claim 2, which was formulated in the proof of Proposition 6 also holds here. Therefore, there is a function $\beta : \{1, \cdots, K_1\} \times \{1, \cdots, K_2\} \to \mathcal{X}_1 \times \mathcal{X}_2$ such that

$$\delta = \gamma \circ \beta.$$

It follows that $(\varphi_1, \varphi_2, \delta) = (\alpha_1 \circ T_1, \alpha_2 \circ T_2, \gamma \circ \beta)$ and hence $(\alpha_1, \alpha_2, \beta)$ is a solution to problem (5.3) in the transformed domain of $(T_1(\mathbf{y}_1), T_2(\mathbf{y}_2))$. □

According to Proposition 7, problem (5.1)) reduces to the problem studied in the

previous chapters of this thesis. Therefore, all the results established in the previous chapters can be applied here. In conclusion, the following theorem holds.

Theorem 5. *When the two vector sources are conditionally independent given the class label, the problem (5.1) can be solved in* $O(K_1K_2N^3)$*, where* $N = \max\{|\mathcal{Y}_1|, |\mathcal{Y}_2|\}$.

Note that based on (5.24), in this case, the optimal classifier in the transformed domain is linear. Therefore, condition $\mathbf{OC}_d$, respectively $\mathbf{OC}_c$, holds in the case of discrete, respectively continuous, sources.

5.3.1 Equal Rate Case

For the equal rate case, i.e., where $K_1 = K_2 = K$, the following can be concluded in virtue of Theorem 4 and Corollary 5 of Chapter 4.

Theorem 6. *Assume that* $K_1 = K_2 = K$*, and* $\mathbf{y}_1$ *and* $\mathbf{y}_2$ *are conditionally independent discrete vector sources. If condition* $\mathbf{S}_d$ *holds in the transformed domain, then the solution to problem (5.1) can be found in* $O(KN)$ *operations, where* $N = \max\{|\mathcal{Y}_1|, |\mathcal{Y}_2|\}$*. Moreover, the partitions of* α_1 *and* α_2 *in the transformed domain must be staggered.*

Additionally, the following can be concluded in virtue of Theorem 3.

Theorem 7. *Assume that* $K_1 = K_2 = K$*, and* $\mathbf{y}_1$ *and* $\mathbf{y}_2$ *are conditionally independent continuous vector sources. If condition* $\mathbf{S}_c$ *holds in the transformed domain, then the partitions of the optimal* α_1 *and* α_2 *in the transformed domain must be strictly staggered.*

Recall condition $\mathbf{S}_d$.

Condition $\mathbf{S}_d$ in the Transformed Domain: $\mathcal{X}_1 = \mathcal{X}_2$ and the following relation is valid for all $(x_1, x_2) \in \mathcal{X}_1 \times \mathcal{X}_2$:

$$P(x_1, x_2, 1) - P(x_1, x_2, -1) = P(x_2, x_1, -1) - P(x_2, x_1, 1).$$

The above is equivalent to

$$\begin{aligned} &\sum_{\mathbf{y}_1 \in T_1^{-1}(x_1)} \sum_{\mathbf{y}_2 \in T_2^{-1}(x_2)} (P'(\mathbf{y}_1, \mathbf{y}_2, 1) - P'(\mathbf{y}_1, \mathbf{y}_2, -1)) \\ = &\sum_{\mathbf{y}'_1 \in T_1^{-1}(x_2)} \sum_{\mathbf{y}'_2 \in T_2^{-1}(x_1)} (P'(\mathbf{y}'_1, \mathbf{y}'_2, -1) - P'(\mathbf{y}'_1, \mathbf{y}'_2, 1)) \end{aligned}$$

Due to conditionally independence of sources, the above is equivalent to

$$\begin{aligned} &\sum_{\mathbf{y}_1 \in T_1^{-1}(x_1)} \sum_{\mathbf{y}_2 \in T_2^{-1}(x_2)} \{P_1(\mathbf{y}_1|1)P_2(\mathbf{y}_2|1)P_0(1) - P_1(\mathbf{y}_1|-1)P_2(\mathbf{y}_2|-1)P_0(-1)\} \\ = &\sum_{\mathbf{y}'_1 \in T_1^{-1}(x_2)} \sum_{\mathbf{y}'_2 \in T_2^{-1}(x_1)} \{P_1(\mathbf{y}'_1|-1)P_2(\mathbf{y}'_2|-1)P_0(-1) - P_1(\mathbf{y}'_1|1)P_2(\mathbf{y}'_2|1)P_0(1).\} \end{aligned} \tag{5.25}$$

It is of interest to find a simpler form for the above condition. Next we derive a sufficient condition, which has a simpler form and therefore it is easier to verify.

Condition $\mathbf{SCS}_d$: There is a bijective function $g : \mathcal{Y}_1 \to \mathcal{Y}_2$ such that

$$P_2(g(\mathbf{y}_1)|1) = P_1(\mathbf{y}_1|-1) \;\&\; P_2(g(\mathbf{y}_1)|-1) = P_1(\mathbf{y}_1|1), \tag{5.26}$$

and $P_0(1) = P_0(-1)$ for every $\mathbf{y}_1 \in \mathcal{Y}_1$ and $\mathbf{y}_2 \in \mathcal{Y}_2$.

Similarly, for continuous sources, we can derive a sufficient condition for condition $\mathbf{S}_c$ that has a simpler form by replacing pmfs P_1 and P_2 by pdfs f_1 and f_2.

Condition SCS$_c$: There is a bijective function $g : \mathcal{Y}_1 \to \mathcal{Y}_2$ such that

$$f_2(g(\mathbf{y}_1)|1) = f_1(\mathbf{y}_1|-1) \;\&\; f_2(g(\mathbf{y}_1)|-1) = f_1(\mathbf{y}_1|1), \tag{5.27}$$

and $P_0(1) = P_0(-1)$ for every $\mathbf{y}_1 \in \mathcal{Y}_1$ and $\mathbf{y}_2 \in \mathcal{Y}_2$.

Theorem 8. *Assume that $K_1 = K_2 = K$, and $\mathbf{y}_1$ and $\mathbf{y}_2$ are conditionally independent discrete vector sources. If condition* **SCS**$_d$ *holds in the original domain, then the solution to problem (5.1) can be found in $O(KN)$ operations, where $N = \max\{|\mathcal{Y}_1|, |\mathcal{Y}_2|\}$. Moreover, the partitions of α_1 and α_2 in the transformed domain must be staggered.*

Theorem 9. *Assume $K_1 = K_2 = K$, and $\mathbf{y}_1$ and $\mathbf{y}_2$ are conditionally independent continuous vector sources. If condition* **SCS**$_c$ *holds in the original domain, then the partitions of optimal α_1 and α_2 in the transformed domain must be strictly staggered.*

The proof of Theorems 8 and 9 rely on the following lemmas.

Lemma 12. *Assume that $\mathbf{y}_1$ and $\mathbf{y}_2$ are conditionally independent discrete vector sources. If condition* **SCS**$_d$ *is satisfied, then condition* $\mathbf{S}_d$ *is satisfied in the transformed domain of $T_1(\mathbf{y}_1)$ and $T_2(\mathbf{y}_2)$.*

Lemma 13. *Assume $\mathbf{y}_1$ and $\mathbf{y}_2$ are conditionally independent continuous vector sources. If condition* **SCS**$_c$ *is satisfied, then condition* $\mathbf{S}_c$ *is satisfied in the transformed domain of $T_1(\mathbf{y}_1)$ and $T_2(\mathbf{y}_2)$.*

Proof of Theorems 8 and 9. According to Lemma 12, if condition $\mathbf{SCS}_d$ holds, condition $\mathbf{S}_d$ is satisfied in the transform domain. The claim follows in virtue of Theorem 6. Similarly, if condition $\mathbf{SCS}_c$ holds, then condition $\mathbf{S}_c$ is satisfied in the transform domain. Further, the result follows according to Theorem 7.

□

Next , we present the proof of Lemma 12 for discrete sources. The proof of Lemma 13 for continuous sources can be derived with similar steps, by replacing pmfs by pdfs, and summations by integrals.

Proof of Lemma 12. Define the following set for any arbitrary x_1 and x_2.

$$B(x_1, x_2) = T_1^{-1}(x_1) \times T_2^{-1}(x_2)$$

Moreover, denote $A(\mathbf{y}_1, \mathbf{y}_2)$ as follows.

$$A(\mathbf{y}_1, \mathbf{y}_2) = P_1(\mathbf{y}_1|1)P_2(\mathbf{y}_2|1)P_0(1) - P_1(\mathbf{y}_1|-1)P_2(\mathbf{y}_2|-1)P_0(-1) \tag{5.28}$$

To prove the lemma, we need the following claims which are proved at the end.

Claim 1. Let $(x_1, x_2) \in \mathcal{X}_1 \times \mathcal{X}_2$ and $(\mathbf{y}_1, \mathbf{y}_2) \in B(x_1, x_2)$. Then $(g^{-1}(\mathbf{y}_2), g(\mathbf{y}_1)) \in B(x_2, x_1)$ and $A(\mathbf{y}_1, \mathbf{y}_2) = -A(g^{-1}(\mathbf{y}_2), g(\mathbf{y}_1))$.

Claim 2. Let $(x_1, x_2) \in \mathcal{X}_1 \times \mathcal{X}_2$. The mapping $\Phi : B(x_1, x_2) \to B(x_2, x_1)$ defined by $\Phi(\mathbf{y}_1, \mathbf{y}_2) = (g^{-1}(\mathbf{y}_2), g(\mathbf{y}_1))$ is bijective.

Next we prove that (5.25) holds, which implies that condition $\mathbf{S}_d$ holds in the transformed domain. For this, we use Claims 1 and 2 to derive the following sequence of equalities. Denote $\mathbf{y}_1' = g^{-1}(\mathbf{y}_2)$, $\mathbf{y}_2' = g(\mathbf{y}_1)$.

$$\begin{aligned} LHS \text{ of (5.25)} &= \sum_{(\mathbf{y}_1,\mathbf{y}_2)\in B(x_1,x_2)} A(\mathbf{y}_1,\mathbf{y}_2) \\ &= \sum_{(\mathbf{y}_1,\mathbf{y}_2)\in B(x_1,x_2)} -A(\Phi(\mathbf{y}_1,\mathbf{y}_2)) \\ &= \sum_{\Phi^{-1}(\mathbf{y}_1',\mathbf{y}_2')\in B(x_1,x_2)} -A((\mathbf{y}_1',\mathbf{y}_2')) \\ &= \sum_{(\mathbf{y}_1',\mathbf{y}_2')\in B(x_2,x_1)} -A((\mathbf{y}_1',\mathbf{y}_2')) \\ &= RHS \text{ of (5.25)}. \end{aligned}$$

The proof of the lemma is completed. □

Proof of Claim 1. Since $(\mathbf{y}_1,\mathbf{y}_2) \in B(x_1,x_2)$, we have $T_1(\mathbf{y}_1) = x_1$ and $T_2(\mathbf{y}_2) = x_2$. Then condition $\mathbf{SCS}_d$ implies that

$$P_2(\mathbf{y}_2'|1) = P_1(\mathbf{y}_1|-1) \;\&\; P_2(\mathbf{y}_2'|-1) = P_1(\mathbf{y}_1|1), \tag{5.29}$$

which leads to

$$\frac{P_2(\mathbf{y}_2'|1)}{P_2(\mathbf{y}_2'|-1)} = \frac{P_1(\mathbf{y}_1|-1)}{P_1(\mathbf{y}_1|1)}.$$

Since $P_0(1) = P_0(-1)$, we further obtain

$$P_0(1)\frac{P_2(\mathbf{y}_2'|1)}{P_2(\mathbf{y}_2'|-1)} = P_0(-1)\frac{P_1(\mathbf{y}_1|-1)}{P_1(\mathbf{y}_1|1)},$$

which implies that

$$T_2(\mathbf{y}_2') = T_1(\mathbf{y}_1) = x_1. \tag{5.30}$$

Since $\mathbf{y}_1' = g^{-1}(\mathbf{y}_2)$, we have $\mathbf{y}_2 = g(\mathbf{y}_1')$. Then condition $\mathbf{SCS}_d$ leads to

$$P_2(\mathbf{y}_2|1) = P_1(\mathbf{y}_1'| - 1) \;\&\; P_2(\mathbf{y}_2| - 1) = P_1(\mathbf{y}_1'|1) \tag{5.31}$$

and further to

$$T_1(\mathbf{y}_1') = P_0(-1)\frac{P_1(\mathbf{y}_1'| - 1)}{P_1(\mathbf{y}_1'|1)} = P_0(1)\frac{P_2(\mathbf{y}_2|1)}{P_2(\mathbf{y}_2| - 1)} = T_2(\mathbf{y}_2) = x_2. \tag{5.32}$$

Relations (5.30) and (5.32) imply that $(\mathbf{y}_1', \mathbf{y}_2') \in B(x_2, x_1)$ proving the first part of Claim 1. Further, using (5.28), (5.29), (5.31) and the fact that $P_0(1) = P_0(-1)$, we obtain

$$\begin{aligned} -A(\mathbf{y}_1', \mathbf{y}_2') &= P_1(\mathbf{y}_1'| - 1)P_2(\mathbf{y}_2'| - 1)P_0(-1) - P_1(\mathbf{y}_1'|1)P_2(\mathbf{y}_2'|1)P_0(1) \\ &= P_2(\mathbf{y}_2|1)P_1(\mathbf{y}_1|1)P_0(1) - P_2(\mathbf{y}_2| - 1)P_1(\mathbf{y}_1| - 1)P_0(-1) \\ &= A(\mathbf{y}_1, \mathbf{y}_2). \end{aligned}$$

This concludes the proof of Claim 1.

□

Proof of Claim 2. Note that according to Claim 1, the mapping Φ is well defined. Let us prove first that Φ is injective. Consider $(\mathbf{y}_1, \mathbf{y}_2), (\mathbf{y}_1', \mathbf{y}_2') \in B(x_1, x_2)$ such that $(\mathbf{y}_1, \mathbf{y}_2) \neq (\mathbf{y}_1', \mathbf{y}_2')$. If $\mathbf{y}_1 \neq \mathbf{y}_1'$, then since g is bijective, it follows that $g(\mathbf{y}_1) \neq g(\mathbf{y}_1')$

leading to $\Phi(\mathbf{y}_1, \mathbf{y}_2) \neq \Phi(\mathbf{y}'_1, \mathbf{y}'_2)$. If $\mathbf{y}_1 = \mathbf{y}'_1$ then $\mathbf{y}_2 \neq \mathbf{y}'_2$. Since g is bijective, g^{-1} is also bijective, hence $g^{-1}(\mathbf{y}_2) \neq g^{-1}(\mathbf{y}'_2)$, which also implies that $\Phi(\mathbf{y}_1, \mathbf{y}_2) \neq \Phi(\mathbf{y}'_1, \mathbf{y}'_2)$.

Let us prove now that Φ is surjective. Let $(\mathbf{y}''_1, \mathbf{y}''_2) \in B(x_2, x_1)$. According to Claim 1, we have $(g^{-1}(\mathbf{y}''_2), g(\mathbf{y}''_1)) \in B(x_1, x_2)$. Further,

$$\Phi(g^{-1}(\mathbf{y}''_2), g(\mathbf{y}''_1)) = (g^{-1}g(\mathbf{y}''_1), g(g^{-1}(\mathbf{y}''_2))) = (\mathbf{y}''_1, \mathbf{y}''_2),$$

and the proof of Claim 2 is completed. □

Note that the function g could be any one-by-one function. Some examples of such g with scalar y_1 and y_2 that would lead to symmetric distribution in the transformed domain include:

- $g(y_1) = y_1$: In this case, based on condition **SCS**$_d$, $P_2(y_1|1) = P_1(y_1| - 1)$ and $P_2(y_1| - 1) = P_1(y_1|1)$. An example for this case could be Normal distributions $P(y_1, y_2, 1) = c_1 e^{-\frac{q}{2}y_1^2} e^{-\frac{r}{2}y_2^2}$ and $P(y_1, y_2, -1) = c_1 e^{-\frac{r}{2}y_1^2} e^{-\frac{q}{2}y_2^2}$. Another example could be Poisson distributions $P(y_1, y_2, 1) = c_1 \frac{a^{y_1}e^{-a}}{y_1!} \frac{b^{y_2}e^{-b}}{y_2!}$ and $P(y_1, y_2, -1) = c_2 \frac{b^{y_1}e^{-b}}{y_1!} \frac{a^{y_2}e^{-a}}{y_2!}$.

- $g(y_1) = ay_1 + b$ where $a \neq 0$: In this case, based on condition **SCS**$_d$, $P_2(ay_1 + b|1) = P_1(y_1| - 1)$ and $P_2(ay_1 + b| - 1) = P_1(y_1|1)$. One example could be Normal distributions $P(y_1, y_2, 1) = c_1 e^{-\frac{q}{2}y_1^2} e^{-\frac{r}{2}y_2^2}$ and $P(y_1, y_2, -1) = c_2 e^{-\frac{r}{2}(a^2y_1^2 + 2aby_1 + b^2)} e^{-\frac{q}{2a^2}(y_2^2 - 2by_2 + b^2)}$.

- $g(y_1) = y_1^3$: In this case, based on condition **SCS**$_d$, $P_2(y_1^3|1) = P_1(y_1| - 1)$ and $P_2(y_1^3| - 1) = P_1(y_1|1)$. An example could be $P(y_1, y_2, 1) = c_1 e^{-ay_1} e^{-by_2}$ and $P(y_1, y_2, -1) = c_2 e^{-by_1^3} e^{-ay_2^{\frac{1}{3}}}$.

- $g(y_1) = e^{y_1}$: In this case, based on condition **SCS**$_d$, $P_2(e^{y_1}|1) = P_1(y_1| - 1)$ and $P_2(e^{y_1}| - 1) = P_1(y_1|1)$. An example could be $P(y_1, y_2, 1) = c_1 e^{y_1} lny_2$ and $P(y_1, y_2, -1) = c_2 y_1 y_2$.

5.4 Discussion

One possible example application to the proposed algorithms in this thesis is scenarios where multiple sensors can be divided into two distinct groups. These sensors have the ability to communicate internally within their respective groups, but communication between groups is restricted. Each sensor within a group will send their captured signal to a joint unit in close proximity in which a unified message is built from the respective group sensors' transmitted data. This transmission is communication constrained too, but we assume that the encoders within each sensor is fixed, hence, we do not tend to optimize sensor encoders. The unified message from each group then is transmitted to the decoder for classification to be done (Figure 5.2). This later transmission is communication constrained and we can optimize the respective encoders. Using the transformation discussed in this chapter, assuming the two groups are conditionally independent given class labels, the encoders can be optimized. Examples of this scenario may include

- Wireless Sensor Networks (WSNs): In WSNs deployed for environmental monitoring, sensors may be divided into two groups based on their geographic location or the type of data they collect. Each group of sensors communicates among

themselves to aggregate data, and then a single message representing the collective information is needs to be sent to the central decoder for further processing.

- Smart Grids: In smart grid systems, sensors are deployed across the power distribution network to monitor parameters like voltage, current, and power flow. These sensors can be grouped based on the geographical regions they cover or the type of equipment they monitor (e.g., substations, transformers). Communication within each group enables localized monitoring and control, with aggregated data to be sent to the central grid management system for analysis and decision-making.

- Medical Monitoring Systems: In hospitals, medical monitoring systems often consist of numerous sensors monitoring various vital signs of patients. These sensors may be grouped based on the type of measurements they take (e.g., heart rate, blood pressure, oxygen saturation). Each group communicates internally to consolidate data, and a unified message is to be transmitted to the central monitoring system for analysis and diagnosis.

5.5 Conclusion

In this chapter, we have explored the application of previously developed quantizer design theory and algorithms to the decentralized detection problem involving two vector sources. We have demonstrated that in certain situations, by applying an appropriate transformation to each vector source, we can reduce the problem to one involving scalar quantizers in the transformed domain.

The main contribution of this chapter is a computationally efficient globally optimal solution for the decentralized detection problem in the case when the vector sources

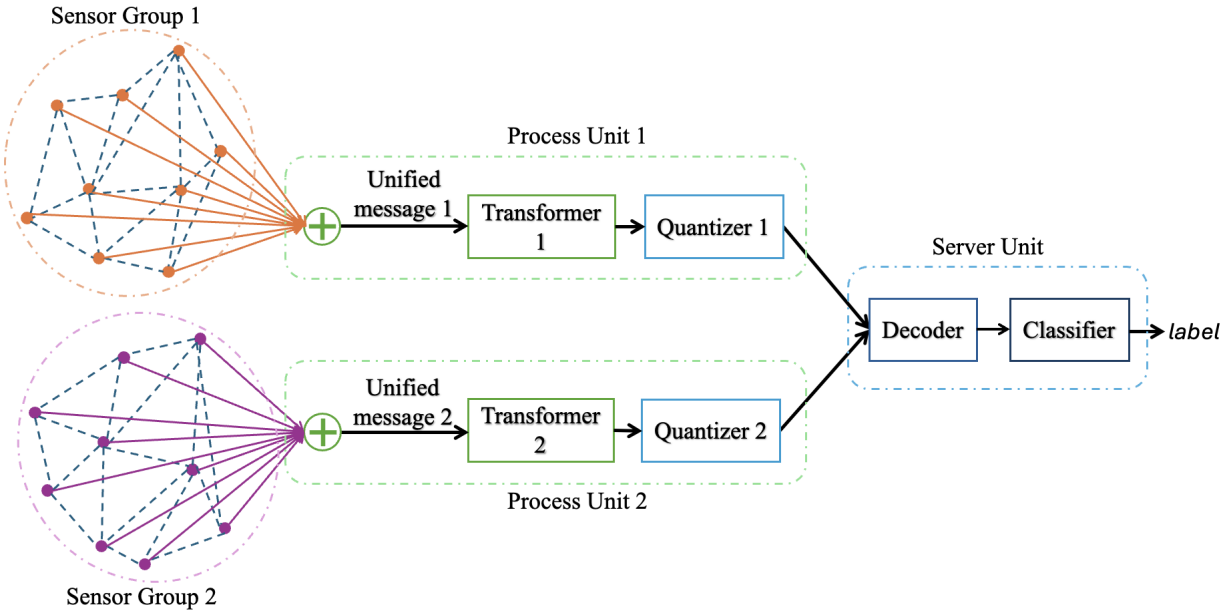


FIGURE 5.2: Example application illustration.

are conditionally independent given the class label. Building upon Tsitsiklis's results regarding the optimality of threshold rules, we have devised an algorithm that significantly improves the time complexity in comparison with the previously known exhaustive search method. Moreover, for the special case of equal quantizer rates, we have shown that imposing a staggering structure can further reduce the time complexity to be linear in the source alphabet size.

Chapter 6

Conclusion and Future Work

This thesis addresses the design of optimal distributed scalar quantizers (DSQ) for two scalar sources tailored to a known binary linear classification task, without assuming any specific data distribution and quantizer boundary arrangements at the encoders. The goal is to design the quantizers such that the misclassification error of the classifier applied to quantized outputs is minimized. Further, the design algorithm is simplified by restricting the rates to be equal for both quantizers and the properties of this scenario are studied. Moreover, we show how the studied setup can be applied to obtaining the optimal DSQ when the sources are vectors and conditionally independent given the class, and without assuming that the classification boundary is known.

in Chapter 2, our focus lies on devising an optimal (K_1, K_2)-level distributed scalar quantizer (DSQ), where K_k denotes the number of quantizer regions of the encoder at node k, for $k = 1, 2$, that minimizes the error of a classifier applied to the quantized output. We make the assumptions that the classifier is binary, linear and known, and the quantizer cells are convex, meaning each cell represents the intersection of a continuous value interval with the source alphabet. Our exploration pertains to situations where

the joint distribution of data within each class is discrete and known. We establish that the problem of optimal DSQ design can be translated into a minimum weight path problem, subject to specific constraints on the number of edges of certain types within a designated weighted directed acyclic graph (WDAG). The proposed solution algorithm exhibits a time complexity of $O(K_1 K_2 M^4)$, where M represents the size of union of the alphabet of the input sources. This is a significant improvement compared to the existing optimal algorithms utilizing an exhaustive search. Further, we demonstrate that the proposed dynamic programming algorithm can be sped up by a factor of M by exploiting the so-called Monge property. Additionally, we show how this algorithm can be applied in scenarios where the source distribution is continuous and known, as well as in cases where the source distribution is unknown but a training sequence is accessible.

Chapter 3 explored the characteristics of DSQ for the equal-rate scenario, where $K_1 = K_2 = K$, with the objective of expediting the design process. Our approach entailed an investigation into strict staggered DSQ (SDSQ) designs, aiming to minimize an upper bound on the error of classifier applied to quantized outputs. We assumed the joint probability distribution of sources per class is known and continuous. To support the consideration such a minimization problem, we analyzed the properties of SDSQs in scenarios where the optimal classifier is linear, demonstrating that the resulting quantizers were optimal SDSQs. Moreover, we showed that the best SDSQ had to be strict, equating minimizing the upper bound on error to minimizing the actual error. Additionally, we established that in cases of symmetric distribution with respect to the classification line, the globally optimal DSQ had to be strictly staggered.

In Chapter 4, with the support of Chapter 3, for the case of equal rates, we constrain

the structure of DSQs to be staggered and assume the joint probability distribution per class is discrete. Consequently, we utilized a modified cost, equivalent to an upper bound on the classification error, for minimization instead of the original cost. This strategic decision led to a significant simplification of the graph model, resulting in a reduction of the solution algorithm's time complexity to $O(KM^2)$. Furthermore, we confirmed that the Monge property holds in instances of optimal classifier, resulting in a reduction in time complexity by an order of M. Experimental results highlighted the superiority of the designed SDSQs with this modified cost over prior work that considered identical encoders, despite our focus on minimizing an upper bound of the error.

Chapter 5 has examined the detection scenario involving vector input sources. For the case of two sensors and two hypothesis with the probability of error criterion, we showed how the problem maps to the DSQ design problem of Chapter 2, when the sources are conditionally independent given the class label. Specifically, we showed that by transforming the input sources into a domain related to the likelihood ratio, each encoder becomes a scalar quantizer with convex cells, and the optimal classifier for the unquantized data in the transformed space is linear. Therefore, the detection problem can be solved using the algorithms designed in ChapterChapter2 with time complexity $O(K_1K_2N^3)$, where N is the size of the largest alphabet of the input vectors. This is a significant improvement in comparison to the only known globally optimal solution of exhaustive search algorithm requiring $O(N^{K_1+K_2}(N + K_1K_2))$ operations. Furthermore, we showed that for the case of encoders with equal rates, results of Chapter 4 hold and by imposing the staggering structure, the time complexity of the solution can be decreased to $O(KN)$.

This thesis addresses the design of DSQs only for two sources and two classes. Therefore, the future work may involve exploring the design for more sources and classes. One such exploration could be to study the feasibility of the staggering structure in higher dimensions for the equal-rate scenario.

In Chapter 5, we noted that the idea of the likelihood transformation is not easily applicable for training sequences. A future work would be to explore ideas on computing an empirical probability for the examples in the training set. For example, divide the space into intervals, and for each interval compute a constant empirical probability for all the points in that interval. Moreover, another possible direction for future work for this chapter is to investigate the optimization of encoders within the sensors along with the two distributed scalar quantizers, for the scenario involving two separate groups of sensors. One other possibility is to explore the idea of [26] to break the dependency of sources by using their technique to share some information between the sources, hence opening the door for the results of Chapter 5 to be applied.

Another possibility could be to combine other techniques, such as source dimension reduction with the quantizer design for the scenarios studied in this thesis.

Another possible future work could be the implementation and further verification of the algorithm for the general-rate scenario which was not fully explored due to insufficient computational resources at the time of the experiment.

Appendix A

Proof of Proposition 1

Proof of Proposition 1. Notice that the encoder of $Q_1 \times Q_2$ partitions $\mathcal{L}$ into segment lines. The separators of these segment lines are the points in the set $\mathcal{W} = \{(a_{u_j}, a_{u_j}) : 0 \leq j \leq K_1\} \cup \{(a_{v_k}, a_{v_k}) : 0 \leq k \leq K_2\}$. Note that the points in $\mathcal{W}$ can be ordered in increasing order of their coordinates. Each segment determined by a pair of consecutive points represents the intersection of $\mathcal{L}$ with a bin of the product quantizer. We will refer to each such segment as an "elementary" segment of $\mathcal{L}$[1].

Let $\mathbf{z} = (z_0, z_1, \cdots, z_{K_1+K_2-1}) \in \mathbb{R}^{K_1+K_2}$, where $z_0 = -\infty$, $z_{K_1+K_2-1} = \infty$ and $(z_1, \cdots, z_{K_1+K_2-2})$ is the sequence of integers obtained by merging the sequences $(u_1, \cdots, u_{K_1-1})$ and $(v_1, \cdots, v_{K_2-1})$ in nondecreasing order such that v_k appears before u_j whenever $u_j = v_k$. For each $\kappa, 0 \leq \kappa \leq K_1 + K_2 - 2$, let $\mathbf{s}(z_\kappa, z_{\kappa+1})$ denote the elementary segment with extremities in $(a_{z_\kappa}, a_{z_\kappa})$ and $(a_{z_{\kappa+1}}, a_{z_{\kappa+1}})$. We consider that $\mathbf{s}(z_\kappa, z_{\kappa+1})$ includes $(a_{z_\kappa}, a_{z_\kappa})$, but does not include $(a_{z_{\kappa+1}}, a_{z_{\kappa+1}})$. Note that, if $z_\kappa = z_{\kappa+1}$, then $\mathbf{s}(z_\kappa, z_{\kappa+1}) = \emptyset$. For the example in Fig. 2.2b, $\mathbf{z} = (v_0, v_1, u_1, u_2, v_2, v_3, v_f)$.

[1] The first and last elementary segments are actually half lines.

Clearly, $\mathbf{s}(z_0, z_1) = U_1 \times V_1 \cap \mathcal{L}$ and $\mathbf{s}(z_{K_1+K_2-2}, z_{K_1+K_2-1}) = U_{K_1} \times V_{K_2} \cap \mathcal{L}$. Further, for $\kappa, 1 \leq \kappa \leq K_1 + K_2 - 3$, based on which quantizer each of the values z_κ and $z_{\kappa+1}$ is a threshold for, one can distinguish four cases for $\mathbf{s}(z_\kappa, z_{\kappa+1})$ as follows.

F1) $z_\kappa = v_k$, $z_{\kappa+1} = u_j$ (ex. $z_1 = v_1$ and $z_2 = u_1$ in Fig. 2.2b); then $u_{j-1} < v_k \leq u_j < v_{k+1}$ and $\mathbf{s}(z_\kappa, z_{\kappa+1}) = \mathcal{L} \cap U_j \times V_{k+1}$.

F2) $z_\kappa = u_j$, $z_{\kappa+1} = v_k$ (ex. $z_3 = u_2$ and $z_4 = v_2$ in Fig. 2.2b); then $v_{k-1} \leq u_j < v_k \leq u_{j+1}$ and $\mathbf{s}(z_\kappa, z_{\kappa+1}) = \mathcal{L} \cap U_{j+1} \times V_k$.

F3) $z_\kappa = u_j$, $z_{\kappa+1} = u_{j+1}$ (ex. $z_2 = u_1$ and $z_3 = u_2$ in Fig. 2.2b); then there is some $k, 1 \leq k \leq K_2$, such that $v_{k-1} \leq u_j < u_{j+1} < v_k$ and $\mathbf{s}(z_\kappa, z_{\kappa+1}) = \mathcal{L} \cap U_{j+1} \times V_k$.

F4) $z_\kappa = v_k$, $z_{\kappa+1} = v_{k+1}$ (ex. $z_4 = v_2$ and $z_5 = v_3$ in Fig. 2.2b); then there is some $j, 1 \leq j \leq K_1$, such that $u_{j-1} < v_k < v_{k+1} \leq u_j$ and $\mathbf{s}(z_\kappa, z_{\kappa+1}) = \mathcal{L} \cap U_j \times V_{k+1}$.

We first show how to construct the mapping $\mathcal{P}$. We start by assigning to each component z_κ of the sequence $\mathbf{z}$ a node in G, denoted by $\mathcal{N}(z_\kappa)$. Then we will show that the obtained sequence of nodes is a path and that path will be $\mathcal{P}(Q_1, Q_2)$.

Let $\mathbf{u} = (u_0, \cdots, u_{K_1})$ be the $(K_1 + 1)$-tuple of thresholds of Q_1 and let $\mathbf{v} = (v_0, \cdots, v_{K_2})$ be the $(K_2 + 1)$-tuple of thresholds of Q_2. Define $\mathcal{N}(z_0) = \nu_0$ and $\mathcal{N}(z_{K_1+K_2-1}) = \nu_f$. For each $j, 1 \leq j \leq K_1 - 1$, there is a unique integer k such that $1 \leq k \leq K_2$ and $v_{k-1} \leq u_j < v_k$. Let us denote this value by $k(j)$. Next, let $\mathcal{N}(u_j) = (v_{k(j)-1}, u_j, v_{k(j)})_v$. Since $v_{k(j)-1}, v_{k(j)} \in \mathcal{M}_2$, $u_j \in \mathcal{M}_1$, and $v_{k(j)-1} \leq u_j < v_{k(j)}$, it follows that $\mathcal{N}(u_j) \in V_v$.

For each $k, 1 \leq k \leq K_2 - 1$, there is a unique integer j such that $1 \leq j \leq K_1$

and $u_{j-1} < v_k \leq u_j$. Let us denote this value by $j(k)$. further, let $\mathcal{N}(v_k) = (u_{j(k)-1}, v_k, u_j(k))_h$. Since $u_{j(k)-1}, u_j(k) \in \mathcal{M}_1$, $v_k \in \mathcal{M}_2$ and $u_{j(k)-1} < v_k \leq u_j(k)$, it follows that $\mathcal{N}(v_k) \in V_h$. Let

$$\bar{\mathcal{N}} = (\mathcal{N}(z_0), \mathcal{N}(z_1), \cdots, \mathcal{N}(z_{K_1+K_2-2})). \tag{A.1}$$

Next we show that, for each $\kappa, 0 \leq \kappa \leq K_1 + K_2 - 2$, there is an edge e from $\mathcal{N}(z_\kappa)$ to $\mathcal{N}(z_{\kappa+1})$ and moreover this is precisely the edge e satisfying $\mathcal{R}(e) \cap \mathcal{L} = \mathbf{s}(z_\kappa, z_{\kappa+1})$.

For this, let us consider first $\kappa = 0$. If $v_1 \leq u_1$ then $z_1 = v_1$ and $\mathcal{N}(z_1) = (0, v_1, u_1)_h$. On the other hand, if $u_1 < v_1$ then $z_1 = u_1$ and $\mathcal{N}(z_1) = (0, u_1, v_1)_v$. Obviously, in both cases, there is an edge e connecting $\mathcal{N}(z_0)$ and $\mathcal{N}(z_1)$ and $\mathcal{R}(e) = U_1 \times V_1$, thus $\mathcal{R}(e) \cap \mathcal{L} = \mathbf{s}(z_0, z_1)$.

Assume now that $\kappa = K_1 + K_2 - 2$. If $v_{K_2-1} \leq u_{K_1-1}$, then $z_{K_1+K_2-2} = u_{K_1-1}$ and $\mathcal{N}(z_{K_1+K_2-2}) = (v_{K_2-1}, u_{K_1-1}, M)_v$. On the other hand, if $u_{K_1-1} < v_{K_2-1}$ then $z_{K_1+K_2-2} = v_{K_2-1}$ and $\mathcal{N}(z_{K_1+K_2-2}) = (u_{K_1-1}, v_{K_2-1}, M)_h$. In both cases, there is an edge e connecting $\mathcal{N}(z_{K_1+K_2-2})$ and $\mathcal{N}(z_{K_1+K_2-1})$ and $\mathcal{R}(e) = U_{K_1} \times V_{K_2}$, thus $\mathcal{R}(e) \cap \mathcal{L} = \mathbf{s}(z_{K_1+K_2-2}, z_{K_1+K_2-1})$.

Let us consider now an arbitrary $\kappa, 1 \leq \kappa \leq K_1 + K_2 - 3$. Next we distinguish between four cases.

C1) $z_\kappa = v_k$ and $z_{\kappa+1} = u_j$ for some $1 \leq j \leq K_1 - 1$ and $1 \leq k \leq K_2 - 1$. Then $u_{j-1} < v_k \leq u_j < v_{k+1}$, which implies that $\mathcal{N}(v_k) = (u_{j-1}, v_k, u_j)_h$ and $\mathcal{N}(u_j) = (v_k, u_j, v_{k+1})_v$. Then $e = (u_{j-1}, v_k, u_j)_h \rightarrow (v_k, u_j, v_{k+1})_v \in E_{hv}$ and

$\mathcal{R}(e) = U_j \times V_{k+1}$. Thus, $w(e) = c(U_j \times V_{k+1})$. In addition, $\mathcal{R}(e) \cap \mathcal{L} = \mathbf{s}(v_k, u_j) = \mathbf{s}(z_\kappa, z_{\kappa+1})$. Note that if $v_k = u_j$ then $\mathcal{R}(e) = \emptyset$.

C2) $z_\kappa = u_j$ and $z_{\kappa+1} = v_k$ for some $1 \leq j \leq K_1 - 1$ and $1 \leq k \leq K_2 - 1$. Then $v_{k-1} \leq u_j < v_k \leq u_{j+1}$, which implies that $\mathcal{N}(u_j) = (v_{k-1}, u_j, v_k)_v$ and $\mathcal{N}(v_k) = (u_j, v_k, u_{j+1})_h$. Then $e = (v_{k-1}, u_j, v_k)_v \rightarrow (u_j, v_k, u_{j+1})_h \in E_{vh}$ and $\mathcal{R}(e) = U_{j+1} \times V_k$. Thus, $w(e) = c(U_{j+1} \times V_k)$. Moreover, $\mathcal{R}(e) \cap \mathcal{L} = \mathbf{s}(u_j, v_k) = \mathbf{s}(z_\kappa, z_{\kappa+1})$. Note that if $u_j = v_k$ then $\mathcal{R}(e) = \emptyset$.

C3) $z_\kappa = u_j$ and $z_{\kappa+1} = u_{j+1}$ for some $1 \leq j \leq K_1 - 2$. Then for some $1 \leq k \leq K_2 - 1$, $v_k \leq u_j < u_{j+1} < v_{k+1}$, which implies that $\mathcal{N}(u_j) = (v_k, u_j, v_{k+1})_v$ and $\mathcal{N}(u_{j+1}) = (v_k, u_{j+1}, v_{k+1})_v$. Then $e = (v_k, u_j, v_{k+1})_v \rightarrow (v_k, u_{j+1}, v_{k+1})_v \in E_{vv}$ and $\mathcal{R}(e) = U_{j+1} \times V_{k+1}$. Thus, $w(e) = c(U_{j+1} \times V_{k+1})$. In addition, $\mathcal{R}(e) \cap \mathcal{L} = \mathbf{s}(u_j, u_{j+1}) = \mathbf{s}(z_\kappa, z_{\kappa+1})$

C4) $z_\kappa = v_k$ and $z_{\kappa+1} = v_{k+1}$ for some $1 \leq k \leq K_2 - 2$. Then for some $1 \leq j \leq K_1 - 1$, $u_j < v_k < v_{k+1} \leq u_{j+1}$, which implies that $\mathcal{N}(v_k) = (u_j, v_k, u_{j+1})_h$ and $\mathcal{N}(v_{k+1}) = (u_j, v_{k+1}, u_{j+1})_h$. Then $e = (u_j, v_k, u_{j+1})_h \rightarrow (u_j, v_{k+1}, u_{j+1})_h \in E_{hh}$ and $\mathcal{R}(e) = U_{j+1} \times V_{k+1}$. Thus, $w(e) = c(U_{j+1} \times V_{k+1})$. Additionally, $\mathcal{R}(e) \cap \mathcal{L} = \mathbf{s}(v_k, v_{k+1}) = \mathbf{s}(z_\kappa, z_{\kappa+1})$.

From the above discussion, it follows that $\bar{\mathcal{N}}$ is a path in G and its weight equals $c(Q_1, Q_2)$. Furthermore, the path contains $K_1 - 1$ vertical nodes and $K_2 - 1$ horizontal nodes.

Now we are left to prove that for any $(K_1 + K_2 - 1)$-edge path P with $K_1 - 1$ vertical nodes and $K_2 - 1$ horizontal nodes from ν_0 to ν_f, there is a DSQ (Q_1, Q_2) such that

$P = \mathcal{P}(Q_1, Q_2)$.

Let us fix such a path P. For $0 \leq \kappa \leq K_1 + K_2 - 1$, let $\nu_\kappa(P)$ denote the κ-th node on the path. Then $\nu_0(P) = \nu_0$, $\nu_{K_1+K_2-1}(P) = \nu_f$ and for $1 \leq \kappa \leq K_1 + K_2 - 2$,

$$\nu_\kappa(P) = (\xi_\kappa, \eta_\kappa, \zeta_\kappa)_{d_\kappa}, \text{ where } d_\kappa \in \{h, v\} \text{ and}$$

$$1)\ \xi_\kappa < \eta_\kappa \leq \zeta_\kappa,\ \xi_\kappa, \zeta_\kappa \in \mathcal{M}_1,\ \eta_\kappa \in \mathcal{M}_2, \text{ if } d_\kappa = h;$$

$$2)\ \xi_\kappa \leq \eta_\kappa < \zeta_\kappa,\ \xi_\kappa, \zeta_\kappa \in \mathcal{M}_2,\ \eta_\kappa \in \mathcal{M}_1, \text{ if } d_\kappa = v.$$

Since for each $1 \leq \kappa \leq K_1 + K_2 - 3$, there is an edge from $\nu_\kappa(P)$ to $\nu_{\kappa+1}(P)$, it follows that

$$\eta_\kappa \leq \eta_{\kappa+1} \text{ with equality only if } d_\kappa = h. \tag{A.2}$$

We are ready now to define the thresholds of Q_1. For $1 \leq j \leq K_1 - 1$, let $u_j = \eta_\kappa$, where κ is the value for which $\nu_\kappa(P)$ is the j-th vertical node on the path P. According to the definition of V_v, $u_j \in \mathcal{M}_1$. Further, according to (A.2), u_j increases as j increases. Then the sequence $\mathbf{u}$ defined in this manner is a valid sequence of thresholds for Q_1.

The thresholds of Q_2 are defined next. Namely, for $1 \leq k \leq K_2 - 1$, let $v_k = \eta_\kappa$, where κ is the value for which $\nu_\kappa(P)$ is the k-th horizontal node on the path P. According to the definition of V_h, $v_k \in \mathcal{M}_2$. Further, according to (A.2), v_k increases as k increases. It follows that the sequence $\mathbf{v}$ defined in this manner is a valid sequence of thresholds for Q_2.

It remains to prove now that $\mathcal{P}(Q_1, Q_2) = P$. Recall that the path $\mathcal{P}(Q_1, Q_2)$ is the sequence of nodes $\bar{\mathcal{N}}$ constructed based on $\mathbf{u}$ and $\mathbf{v}$, as described in (A.1). We will show

that for each j, $1 \leq j \leq K_1 - 1$, $\mathcal{N}(u_j)$ equals the j-th vertical node in P, and for each k, $1 \leq k \leq K_2 - 1$, $\mathcal{N}(v_k)$ equals the k-th horizontal node in P.

Let us fix some j, $1 \leq j \leq K_1 - 1$. Recall that $\mathcal{N}(u_j) = (v_{k(j)-1}, u_j, v_{k(j)})_v$. Let $\kappa_0, \kappa_1, \kappa_2$ be integers such that $\nu_{\kappa_0}(P)$ is the j-th vertical node in P, $\nu_{\kappa_1}(P)$ is the $(k(j) - 1)$-th horizontal node in P and $\nu_{\kappa_2}(P)$ is the $(k(j))$-th horizontal node in P. Then $v_{k(j)-1} = \eta_{\kappa_1}$, $u_j = \eta_{\kappa_0}$ and $v_{k(j)} = \eta_{\kappa_2}$. Combining the above with the fact that $v_{k(j)-1} \leq u_j < v_{k(j)}$ and with (A.2), it follows that $\kappa_1 < \kappa_0 < \kappa_2$.

If the nodes $\nu_{\kappa_1}(P)$, $\nu_{\kappa_0}(P)$ and $\nu_{\kappa_2}(P)$ are consecutive (i.e., $\kappa_2 = \kappa_1 + 2$), then from the definition of E_{hv} and E_{vh} we obtain that $\xi_{\kappa_0} = \eta_{\kappa_1} = v_{k(j)-1}$ and $\zeta_{\kappa_0} = \eta_{\kappa_2} = v_{k(j)}$. This implies that $\mathcal{N}(u_j) = \nu_{\kappa_0}(P)$.

If $\kappa_2 > \kappa_1 + 2$, then there are more than one node between $\nu_{\kappa_1}(P)$ and $\nu_{\kappa_2}(P)$ on P, but all of them are vertical. According to the definition of E_{vv}, in any sequence of vertical nodes connected by edges, the first and the last node have the same value for the first component and the same value for the last component. Combining this observation with the definition of E_{hv} and E_{vh} we obtain again that $\xi_{\kappa_0} = \eta_{\kappa_1} = v_{k(j)-1}$ and $\zeta_{\kappa_0} = \eta_{\kappa_2} = v_{k(j)}$, which leads to $\mathcal{N}(u_j) = \nu_{\kappa_0}(P)$.

The proof of the fact that $\mathcal{N}(v_k)$ equals the k-th horizontal node in P, $\forall k$, $1 \leq k \leq K_2 - 1$, proceeds similarly.

□

Appendix B

Lemmas Needed for the Proof of Theorem 2

We first introduce some terminology. In a non-strict SDSQ, the points on the line $\mathcal{L}$ corresponding to equal thresholds are called *superpoints*. More specifically, a superpoint is a pair $(\tilde{u}_i, \tilde{v}_j)$, $0 \leq i, j \leq K$ satisfying $u_i = v_j$. If i and j are both nonzero and smaller than K, then the superpoint is called an *interior superpoint*.

The relevant cells of an SDSQ are naturally ordered according to the order of their intersections with the classifier line $\mathcal{L}$. Thus, the first relevant cell is $S_{1,1}$ and the last is $S_{K,K}$. A portion of an SDSQ between two consecutive superpoints (i.e., the sequence of relevant cells between the superpoints) is called a *strict portion* of the SDSQ. We say that the *cost region* of a cell $S_{i,j}$ is the region that determines its cost, i.e., it is $T_{i,j}$ when $\mu(T_{i,j}) < \mu(P_{i,j})$ and it is $P_{i,j}$ when $\mu(T_{i,j}) > \mu(P_{i,j})$. When $\mu(T_{i,j}) = \mu(P_{i,j})$ both $T_{i,j}$ and $P_{i,j}$ can be considered cost regions.

Next we present three lemmas there are used in the proof of Theorem 2.

Lemma 14. *Assume that condition* $\mathbf{OC}_c$ *is satisfied. Consider an SDSQ where in a strict SDSQ portion we have a Pcell and either of the following scenarios hold.*

a) The Pcell is followed or preceded by two consecutive Tcells that are included in the same strict SDSQ portion.

b) The Pcell is followed or preceded by a Tcell that is included in the same strict SDSQ portion and is adjacent to a superpoint.

c) The Pcell is between two Tcells that are included in the same strict SDSQ portion.

Then by appropriately changing just one threshold without violating the strictness of the SDSQ portion, the cost of the SDSQ is strictly decreased.

Note that for each interior superpoint there are only two relevant cells adjacent to the superpoint.

Lemma 15. *Assume that condition* $\mathbf{OC}_c$ *is satisfied. Consider a non-strict SDSQ that has an interior superpoint that is adjacent to at least one Tcell. Then by appropriately changing one of the thresholds involved in the superpoint, the tie at the superpoint is broken without violating the rest of inequalities involving the thresholds, while the cost of the SDSQ is strictly reduced.*

Note that in the proofs of the two lemmas we will use extensively Lemma 2. In addition, in order to prove Lemmas 14 and 15, we need the following auxiliary result.

Lemma 16. *Assume that condition* $\mathbf{OC}_c$ *holds. Let* $x_1 \in \mathbb{R}$*, and* $l > 0$*. For each* $\epsilon, 0 < \epsilon < l/2$*, consider the following notations, which are illustrated in Fig. A2.1:*

1) $\mathcal{R}(\epsilon) = [x_1 - \epsilon, x_1] \times [x_1 - l, x_1]$;

2) $\mathcal{R}^l(\epsilon) = [x_1 - l, x_1] \times [x_1 - \epsilon, x_1]$;

3) $\mathcal{R}^u(\epsilon) = [x_1, x_1 + \epsilon] \times [x_1, x_1 + l]$;

4) $\mathcal{R}^r(\epsilon) = [x_1, x_1 + l] \times [x_1, x_1 + \epsilon]$.

Then the following claims hold.

C1) There is $\epsilon_1, 0 < \epsilon_1 < l/2$, *such that*

$$\mu(\mathcal{R}(\epsilon) \cap \mathcal{H}_+) < \mu(\mathcal{R}(\epsilon) \cap \breve{\mathcal{H}}_-), \ \forall \epsilon, 0 < \epsilon < \epsilon_1. \tag{B.1}$$

C2) There is $\epsilon_2, 0 < \epsilon_2 < l/2$, *such that* $\mu(\mathcal{R}^l(\epsilon) \cap \breve{\mathcal{H}}_-) < \mu(\mathcal{R}^l(\epsilon) \cap \mathcal{H}_+), \forall\ \epsilon, 0 < \epsilon < \epsilon_2$.

C3) There is $\epsilon_3, 0 < \epsilon_3 < l/2$, *such that* $\mu(\mathcal{R}^u(\epsilon) \cap \breve{\mathcal{H}}_-) < \mu(\mathcal{R}^u(\epsilon) \cap \mathcal{H}_+), \ \forall \epsilon, 0 < \epsilon < \epsilon_3$.

C4) There is $\epsilon_4, 0 < \epsilon_4 < l/2$, *such that* $\mu(\mathcal{R}^r(\epsilon) \cap \mathcal{H}_+) < \mu(\mathcal{R}^r(\epsilon) \cap \breve{\mathcal{H}}_-), \ \forall \epsilon, 0 < \epsilon < \epsilon_4$.

Proof. We only prove claim C1 since all the other claims follow by symmetry. In the proof, we will use the following notation

$$g(\mathbf{x}) = \begin{cases} f(\mathbf{x}, 1) - f(\mathbf{x}, -1) & \text{if } \mathbf{x} \in \mathcal{H}_+ \\ f(\mathbf{x}, -1) - f(\mathbf{x}, 1) & \text{if } \mathbf{x} \in \breve{\mathcal{H}}_- \end{cases}. \tag{B.2}$$

Since condition $\mathbf{OC}_c$ holds we have $g(\mathbf{x}) = 0$ for $\mathbf{x} \in \mathcal{L}$ and $g(\mathbf{x}) > 0$ for $\mathbf{x} \in \mathbb{R}^2 \setminus \mathcal{L}$.

For each $\epsilon, 0 < \epsilon < l/2$, denote $\mathcal{R}_1(\epsilon) = \mathcal{R}(\epsilon) \cap \mathcal{H}_+$, $\mathcal{R}_2(\epsilon) = [x_1 - \epsilon, x_1] \times [x_1 - l, x_1 - l/2]$ (Fig. A2.2(b)). Additionally, define

$$c_{max} = \max_{\mathbf{x} \in \mathcal{R}_1(\frac{2l}{5})} g(\mathbf{x}),\ c_{min} = \min_{\mathbf{x} \in \mathcal{R}_2(\frac{2l}{5})} g(\mathbf{x}).$$

Note that the minimum and maximum defined above exist since $\mathcal{R}_1(\frac{2l}{5})$ and $\mathcal{R}_2(\frac{2l}{5})$ are compact sets and $g(\mathbf{x})$ is a continuous function. It can be easily seen that $\mathcal{R}_2(\frac{2l}{5}) \subset \breve{\mathcal{H}}_-$, which implies that $g(\mathbf{x}) > 0$ for all $\mathbf{x} \in \mathcal{R}_2(\frac{2l}{5})$, and further that $c_{\min} > 0$. Clearly, $c_{\max} > 0$. Further, for any $\epsilon, 0 < \epsilon < \frac{2l}{5}$, we have $\mathcal{R}_1(\epsilon) \subset \mathcal{R}_1(\frac{2l}{5})$ and $\mathcal{R}_2(\epsilon) \subset \mathcal{R}_2(\frac{2l}{5})$ (Fig. A2.2), which lead to $g(\mathbf{x}) \leq c_{\max}, \forall \mathbf{x} \in \mathcal{R}_1(\epsilon)$ and $g(\mathbf{x}) \geq c_{\min}, \forall \mathbf{x} \in \mathcal{R}_2(\epsilon)$. These relations imply that for all $\epsilon, 0 < \epsilon < \frac{2l}{5}$, we have

$$\mu(\mathcal{R}_1(\epsilon)) = \int_{\mathcal{R}_1(\epsilon)} g(\mathbf{x})\, d\mathbf{x} \leq \int_{\mathcal{R}_1(\epsilon)} c_{\max}\, d\mathbf{x} = c_{\max} \cdot \frac{\epsilon^2}{2}, \tag{B.3}$$

$$\mu(\mathcal{R}_2(\epsilon)) = \int_{\mathcal{R}_2(\epsilon)} g(\mathbf{x})\, d\mathbf{x} \geq \int_{\mathcal{R}_2(\epsilon)} c_{\min}\, d\mathbf{x} = c_{\min} \cdot \frac{\epsilon \cdot l}{2}, \tag{B.4}$$

where we have used the fact that $\mathcal{R}_1(\epsilon)$ is a triangular region. Note that for $\epsilon, 0 < \epsilon < \frac{2l}{5}$, we have $\mathcal{R}_2(\epsilon) \subset \mathcal{R}(\epsilon) \cap \breve{\mathcal{H}}_-$, which implies that $\mu(\mathcal{R}_2(\epsilon)) \leq \mu(\mathcal{R}(\epsilon) \cap \breve{\mathcal{H}}_-)$ in view of Lemma 2. Thus, in order to prove (B.1), it is sufficient to show that $\mu(\mathcal{R}_1(\epsilon)) < \mu(\mathcal{R}_2(\epsilon))$, which, in light of (B.3) and (B.3), holds when $c_{\max} \cdot \frac{\epsilon^2}{2} < c_{\min} \cdot \frac{\epsilon \cdot l}{2}$. The previous inequality is equivalent to $\epsilon < \frac{c_{\min}}{c_{\max}} l$. Thus, by letting $\epsilon_1 = \frac{c_{\min}}{c_{\max}} l$, the conclusion follows. □

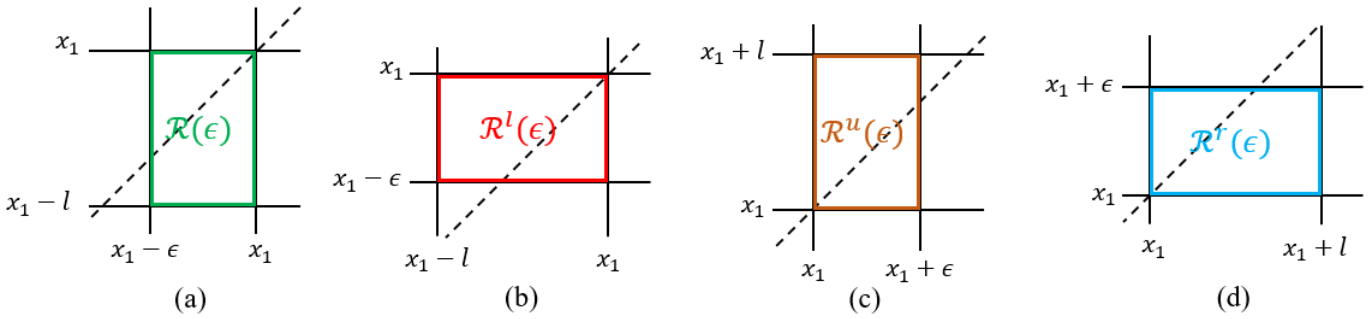


FIGURE A2.1: Illustration of the four cases of Lemma 16.

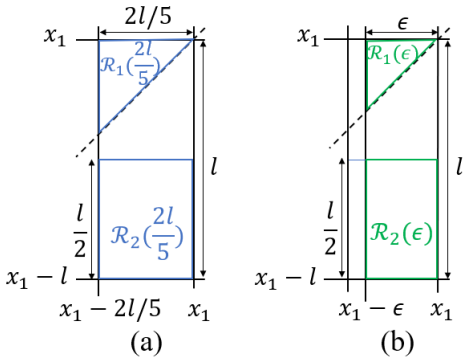


FIGURE A2.2: Illustration of notations used in the proof of claim C1 of Lemma 16.

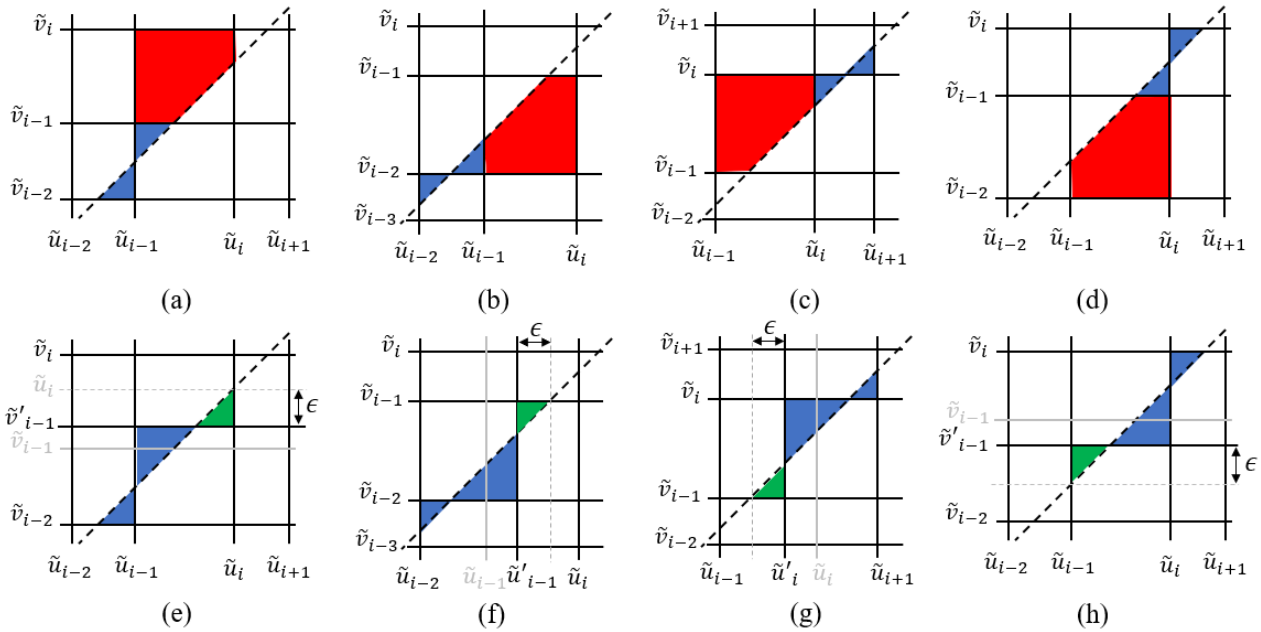


FIGURE A2.3: : (a) initial SDSQ of case a1); (b) initial SDSQ of case a2); (c) initial SDSQ of case a3); (d) initial SDSQ of case a4); the pentagonal cost region of the target Pcell is shown in red and other cost regions are shown in blue; (e-h) new SDSQ for each of the cases a1)-a4), respectively. The pentagonal cost region is reduced to the green triangular cost region after adjusting one threshold.

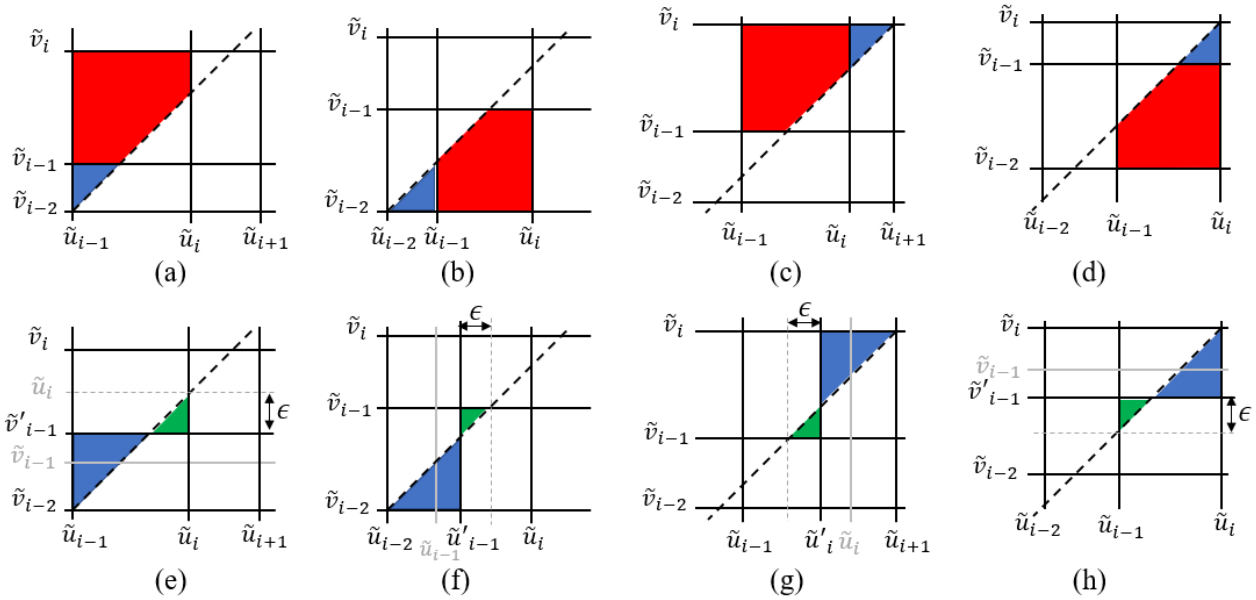


FIGURE A2.4: : (a) initial SDSQ of case b1); (b) initial SDSQ of case b2); (c) initial SDSQ of case b3); (d) initial SDSQ of case b4); the pentagonal cost region of the target Pcell is shown in red and other cost regions are shown in blue; (e-h) new SDSQ for each of the cases b1)-b4), respectively. The pentagonal cost region is reduced to the green triangular cost region after adjusting one threshold.

Proof of Lemma 14

Proof. It is sufficient to prove the lemma for the case when the SDSQ has the u-first orientation. Let $S_{i,j}$ be the Pcell.

We first consider situations a) and b). For case a) we have the following four subcases, which are also illustrated in Fig. A2.3(a-d).

a1) $j = i$ and the Tcells precede the Pcell, i.e., $\tilde{u}_{i-2} \leq \tilde{v}_{i-2} < \tilde{u}_{i-1} < \tilde{v}_{i-1} < \tilde{u}_i \leq \tilde{v}_i \leq \tilde{u}_{i+1}$. The Tcells are $S_{i,i-1}$ and $S_{i-1,i-1}$.

a2) $j = i - 1$ and the Tcells precede the Pcell, i.e., $\tilde{v}_{i-3} \leq \tilde{u}_{i-2} < \tilde{v}_{i-2} < \tilde{u}_{i-1} < \tilde{v}_{i-1} \leq \tilde{u}_i \leq \tilde{v}_i$. The Tcells are $S_{i-1,i-1}$ and $S_{i-1,i-2}$.

a3) $j = i$ and the Tcells follow the Pcell, i.e., $\tilde{v}_{i-2} \leq \tilde{u}_{i-1} \leq \tilde{v}_{i-1} < \tilde{u}_i < \tilde{v}_i < \tilde{u}_{i+1} \leq \tilde{v}_{i+1}$. The Tcells are $S_{i+1,i}$ and $S_{i+1,i+1}$.

a4) $j = i - 1$ and the Tcells follow the Pcell, i.e., $\tilde{u}_{i-2} \leq \tilde{v}_{i-2} \leq \tilde{u}_{i-1} < \tilde{v}_{i-1} < \tilde{u}_i < \tilde{v}_i \leq \tilde{u}_{i+1}$. The Tcells are $S_{i,i}$ and $S_{i+1,i}$.

For the situation b) we have the following four subcases, which are depicted in Fig. A2.4(a-d).

b1) $j = i$ and the Tcell precedes the Pcell, i.e., $\tilde{v}_{i-2} = \tilde{u}_{i-1} < \tilde{v}_{i-1} < \tilde{u}_i \leq \tilde{v}_i \leq \tilde{u}_{i+1}$. The Tcell is $S_{i,i-1}$ and (u_{i-1}, v_{i-2}) is the superpoint.

b2) $j = i - 1$ and the Tcell precedes the Pcell, i.e., $\tilde{u}_{i-2} = \tilde{v}_{i-2} < \tilde{u}_{i-1} < \tilde{v}_{i-1} \leq \tilde{u}_i \leq \tilde{v}_i$. The Tcell is $S_{i-1,i-1}$ and (u_{i-2}, v_{i-2}) is the superpoint.

b3) $j = i$ and the Tcell follows the Pcell, i.e., $\tilde{v}_{i-2} \leq \tilde{u}_{i-1} \leq \tilde{v}_{i-1} < \tilde{u}_i < \tilde{v}_i = \tilde{u}_{i+1}$. The Tcell are $S_{i+1,i}$ and (u_{i+1}, v_i) is the superpoint.

b4) $j = i - 1$ and the Tcell follows the Pcell, i.e., $\tilde{u}_{i-2} \leq \tilde{v}_{i-2} \leq \tilde{u}_{i-1} < \tilde{v}_{i-1} < \tilde{u}_i = \tilde{v}_i$. The Tcell is $S_{i,i}$ and (u_i, v_i) is the superpoint.

For all eight cases we illustrate the changes in Figs. A2.3(e-h) and A2.4(e-h), but only provide the proof for case a1) and show how this proof is adapted for case b1). All the other cases can be treated similarly.

Case a1). This case is depicted in Fig. A2.3(a). We will use Lemma 16 for $x_1 = \tilde{u}_i$ and, $l = \tilde{u}_i - \tilde{u}_{i-1}$. Consider some ϵ smaller than ϵ_2 from Lemma 16 and let $\tilde{v}'_{i-1} = \tilde{u}_i - \epsilon$. Then $\tilde{v}_{i-1} < \tilde{v}'_{i-1} < \tilde{u}_i$ and

$$\mu(\mathcal{R} \cap \breve{\mathcal{H}}_-) < \mu(\mathcal{R} \cap \mathcal{H}_+), \tag{B.5}$$

where $\mathcal{R} = U_i \times [\tilde{v}'_{i-1}, \tilde{u}_i)$. Let us replace $\tilde{v}_{i-1}$ by $\tilde{v}'_{i-1}$ in Q_2, and denote by Q'_2 the new quantizer and by $\mathbf{Q}'$ the new SDSQ. This change only affects the cells V_{i-1} and V_i of Q_2, which will become $V'_{i-1} = [\tilde{v}_{i-2}, \tilde{v}'_{i-1})$ and $V'_i = [\tilde{v}'_{i-1}, \tilde{v}_i)$ (see Fig. A2.3(e)).

The relevant cells of the SDSQ that are affected by this change are $S_{i-1,i-1}$, $S_{i,i-1}$, $S_{i,i}$ and $S_{i+1,i}$. They are converted to $S'_{i-1,i-1} = U_{i-1} \times V'_{i-1}$, $S_{i,i-1} = U_i \times V'_{i-1}$, $S'_{i,i} = U_i \times V'_i$ and $S'_{i+1,i} = U_{i+1} \times V'_i$, respectively.

We will prove that

$$c(S'_{i+1,i}) \leq c(S_{i+1,i}), \tag{B.6}$$

$$c(S'_{i-1,i-1}) \leq c(S_{i-1,i-1}), \tag{B.7}$$

$$c(S'_{i,i-1}) + c(S'_{i,i}) < c(S_{i,i}) + c(S_{i,i-1}), \tag{B.8}$$

which combined lead to the conclusion that $c(\mathbf{Q}') < c(\mathbf{Q})$.

Inequality (B.6) follows since $S'_{i+1,i} \subset S_{i+1,i}$ using Lemma 2. Equality (B.7) is based on the fact that $T'_{i-1,i-1} = T_{i-1,i-1}$ and that $S_{i-1,i-1}$ is a Tcell, which lead to

$$c(S'_{i-1,i-1}) \leq \mu(T'_{i-1,i-1}) = \mu(T_{i-1,i-1}) = c(S_{i-1,i-1}).$$

Let us prove now relation (B.8). Using (B.5) and the fact that $\mathcal{R} \cap \mathcal{H}_+ \subset P'_{i,i} \subset P_{i,i}$ and $\mathcal{R} \cap \breve{\mathcal{H}}_- = T'_{i,i}$, it can be concluded that $\mu(T'_{i,i}) < \mu(P'_{i,i})$, leading further to

$$c(S'_{i,i}) = \mu(T'_{i,i}) = \mu(\mathcal{R} \cap \breve{\mathcal{H}}_-) < \mu(\mathcal{R} \cap \mathcal{H}_+). \tag{B.9}$$

Denote now $\mathcal{R}' = U_i \times [\tilde{v}_{i-1}, \tilde{v}'_{i-1})$. We have the following sequence of relations

$$\begin{aligned} \mu(\mathcal{R} \cap \mathcal{H}_+) + \mu(\mathcal{R}' \cap \mathcal{H}_+) &\overset{(a)}{\leq} \mu(P_{i,i}) \\ &\overset{(b)}{<} \mu(T_{i,i}) \\ &\overset{(c)}{=} \mu(\mathcal{R} \cap \mathcal{H}_-) + \mu(\mathcal{R}' \cap \mathcal{H}_-), \end{aligned} \tag{B.10}$$

where (a) follows from $(\mathcal{R} \cap \mathcal{H}_+) \cup (\mathcal{R}' \cap \mathcal{H}_+) \subset P_{i,i}$ and the fact that $\mathcal{R} \cap \mathcal{H}_+$ and $\mathcal{R}' \cap \mathcal{H}_+$ are disjoint, (b) holds since $S_{i,i}$ is a Pcell, and (c) is due to $T_{i,i} = (\mathcal{R} \cap \mathcal{H}_-) \cup (\mathcal{R}' \cap \mathcal{H}_-)$ and the fact that $\mathcal{R} \cap \mathcal{H}_-$ and $\mathcal{R}' \cap \mathcal{H}_-$ are disjoint. Further, combining (B.10) with (B.5)

leads to

$$\mu(\mathcal{R}' \cap \mathcal{H}_+) < \mu(\mathcal{R}' \cap \mathcal{H}_-). \tag{B.11}$$

Next we obtain the following sequence of relations

$$\begin{aligned} \mu(T'_{i,i-1}) &\overset{(a)}{=} \mu(\mathcal{R}' \cap \mathcal{H}_+) + \mu(T_{i,i-1}) \\ &\overset{(b)}{<} \mu(\mathcal{R}' \cap \mathcal{H}_-) + \mu(P_{i,i-1}) \\ &\overset{(c)}{=} \mu(P'_{i,i-1}), \end{aligned} \tag{B.12}$$

where (a) holds since $\mathcal{R}' \cap \mathcal{H}_+$ and $T_{i,i-1}$ are disjoint and their union equals $T'_{i,i-1}$, (b) follows from (B.11) and the fact that $S_{i,i-1}$ is a Tcell, and (c) is valid since $\mathcal{R}' \cap \mathcal{H}_-$ and $P_{i,i-1}$ are disjoint and their union equals $P'_{i,i-1}$. Relations (B.12) imply that

$$c(S'_{i,i-1}) = \mu(T'_{i,i-1}) = \mu(\mathcal{R}' \cap \mathcal{H}_+) + \mu(T_{i,i-1}).$$

The above together with (B.9) lead to

$$\begin{aligned} c(S'_{i,i-1}) + c(S'_{i,i}) &< \mu(\mathcal{R} \cap \mathcal{H}_+) + \mu(\mathcal{R}' \cap \mathcal{H}_+) + \mu(T_{i,i-1}) \\ &\leq \mu(P_{i,i}) + \mu(T_{i,i-1}) \\ &= c(S_{i,i}) + c(S_{i,i-1}), \end{aligned}$$

where the second inequality follows from relation (a) in (B.10). This concludes the proof of case a1).

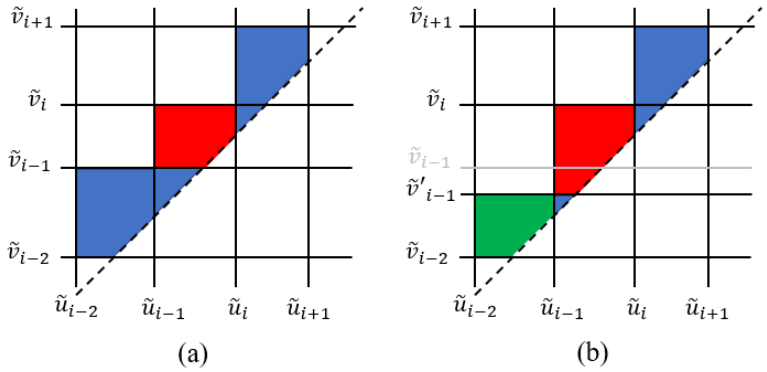


FIGURE A2.5: : (a) initial SDSQ; the pentagonal cost region of the target Pcell is shown in red and other cost regions are shown in blue; (b) new SDSQ after changing one threshold; the pentagonal cost region of $S_{i-1,i-1}$ is reduced to the green region.

The proof for case b1) is very similar with the only difference being that the cells $S_{i-1,i-1}$ and $S'_{i-1,i-1}$ are included in $\mathcal{H}_+$, thus their cost is 0.

Case c) In this case we have a Tcell followed by a Pcell followed by another Tcell inside the same strict SDSQ portion. If the first or last Tcell is adjacent to a superpoint, then we are under case b). If the first Tcell is preceded or the last Tcell is followed by another Tcell, then we are under case a). Thus, it only remains to discuss the situation when the first Tcell is preceded by a Pcell and the last Tcell is followed by a Pcell, which are inside the same strict SDSQ portion. Then we have a sequence of five cells: a Pcell, a Tcell, another Pcell, another Tcell and another Pcell, all in the same strict SDSQ portion. Note that the cost regions of these cells are either all in $\mathcal{H}_+$ or all in $\mathcal{H}_-$. Next we will distinguish between these two subcases.

c1) $P_{i,j}$ is included in $\mathcal{H}_+$ and $j = i$, i.e., $\tilde{u}_{i-2} \leq \tilde{v}_{i-2} < \tilde{u}_{i-1} < \tilde{v}_{i-1} < \tilde{u}_i < \tilde{v}_i < \tilde{u}_{i+1} \leq \tilde{v}_{i+1}$. This case is illustrated in Fig. A2.5(a).

c2) $P_{i,j}$ is included in $\mathcal{H}_-$ and $j = i-1$, i.e., $\tilde{v}_{i-3} \leq \tilde{u}_{i-2} < \tilde{v}_{i-2} < \tilde{u}_{i-1} < \tilde{v}_{i-1} < \tilde{u}_i < \tilde{v}_i \leq \tilde{u}_{i+1}$.

We will prove case c1) since case c2) follows similarly.

Case c1). Choose $\tilde{v}'_{i-1}$ such that $\tilde{u}_{i-1} < \tilde{v}'_{i-1} < \tilde{v}_{i-1}$. Replace $\tilde{v}_{i-1}$ by $\tilde{v}'_{i-1}$, in Q_2, and denote by Q'_2 the new quantizer and by $\mathbf{Q}'$ the new SDSQ. This change only affects the cells V_{i-1} and V_i of Q_2, which will become $V'_{i-1} = [\tilde{v}_{i-2}, \tilde{v}'_{i-1})$ and $V'_i = [\tilde{v}'_{i-1}, \tilde{v}_i)$. Thus, the relevant cells of the SDSQ that are affected by this change are $S_{i-1,i-1}$, $S_{i,i-1}$, $S_{i,i}$ and $S_{i+1,i}$. They are converted to $S'_{i-1,i-1} = U_{i-1} \times V'_{i-1}$, $S_{i,i-1} = U_i \times V'_{i-1}$, $S'_{i,i} = U_i \times V'_i$ and $S'_{i+1,i} = U_{i+1} \times V'_i$, respectively. We will prove the following

$$c(S'_{i+1,i}) \leq c(S_{i+1,i}), \tag{B.13}$$

$$c(S'_{i-1,i-1}) < c(S_{i-1,i-1}), \tag{B.14}$$

$$c(S'_{i,i-1}) + c(S'_{i,i}) \leq c(S_{i,i-1}) + c(S_{i,i}), \tag{B.15}$$

which further imply that $c(\mathbf{Q}') < c(\mathbf{Q})$.

Inequality (B.13) follows from the fact that $S_{i+1,i}$ is a Tcell, while $T'_{i+1,i} = T_{i+1,i}$, which lead to $c(S'_{i+1,i}) \leq \mu(T'_{i+1,i}) = \mu(T_{i+1,i}) = c(S_{i+1,i})$.

In order to prove (B.14), note that since $S_{i-1,i-1}$ is a Pcell, we have $\mu(P_{i-1,i-1}) < \mu(T_{i-1,i-1})$. In addition the following hold: $T'_{i-1,i-1} = T_{i-1,i-1}$, $P'_{i-1,i-1} \subset P_{i-1,i-1}$ and

$area(P_{i-1,i-1} \setminus P'_{i-1,i-1}) > 0$. Therefore, $\mu(P'_{i-1,i-1}) < \mu(P_{i-1,i-1}) < \mu(T_{i-1,i-1}) = \mu(T'_{i-1,i-1})$. It follows that $c(S'_{i-1,i-1}) = \mu(P'_{i-1,i-1}) < \mu(P_{i-1,i-1}) = c(S_{i-1,i-1})$.

Let us prove now (B.15). Notice that the cells $S_{i,i-1}$ and $S_{i,i}$ are decreased and enlarged by $\mathcal{R}$, respectively, where $\mathcal{R} = U_i \times [\tilde{v}'_{i-1}, \tilde{v}_{i-1})$. In other words, $S'_{i,i-1} = S_{i,i-1} \setminus \mathcal{R}$ and $S'_{i,i} = S_{i,i} \cup \mathcal{R}$. Since $S_{i,i-1}$ is a Tcell, we have $c(S_{i,i-1}) = \mu(T_{i,i-1}) \leq \mu(P_{i,i-1})$ and since $S_{i,i}$ is a Pcell, we have $c(S_{i,i}) = \mu(P_{i,i}) < \mu(T_{i,i})$. Note that $T_{i,i-1} = T'_{i,i-1} \cup (\mathcal{R} \cap \mathcal{H}_+)$ and $P'_{i,i} = P_{i,i} \cup (\mathcal{R} \cap \mathcal{H}_+)$ leading further to

$$\mu(T'_{i,i-1}) = \mu(T_{i,i-1}) - \mu(\mathcal{R} \cap \mathcal{H}_+),$$
$$\mu(P'_{i,i}) = \mu(\mathcal{R} \cap \mathcal{H}_+) + \mu(P_{i,i}).$$

According to the above discussion, we conclude that

$$\begin{aligned} c(S'_{i,i-1}) + c(S'_{i,i}) &\leq \mu(T'_{i,i-1}) + \mu(P'_{i,i}) \\ &= \mu(T_{i,i-1}) + \mu(P_{i,i}) \\ &= c(S_{i,i-1}) + c(S_{i,i}). \end{aligned}$$

This completes the proof of the lemma.

□

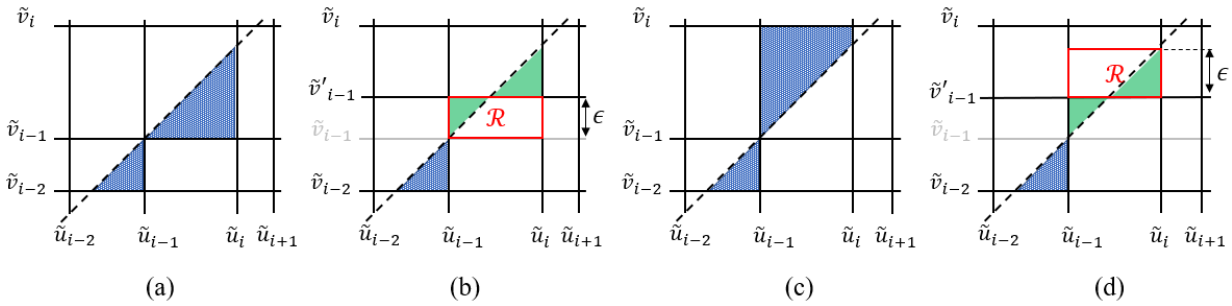


FIGURE A2.6: : (a) initial cells of case d1); (b) the resulting cells after replacing $\tilde{v}_{i-1}$ by $\tilde{v}'_{i-1}$ for case d1); (c) initial cells of case d2); (d) the resulting cells after replacing $\tilde{v}_{i-1}$ by $\tilde{v}'_{i-1}$ for case d2. The initial cost regions and the new cost regions are colored in dashed blue and solid green, respectively.

Proof of Lemma 15

Proof. It is sufficient to provide the proof for an SDSQ with u-first orientation. We have to distinguish between six possible cases based on the nature of the superpoint and the positions of the Tcell and the Pcell. The six cases are listed next.

d1) The superpoint corresponds to $\tilde{u}_{i-1} = \tilde{v}_{i-1}$ and both adjacent cells are Tcells. In other words, $\tilde{v}_{i-2} < \tilde{u}_{i-1} = \tilde{v}_{i-1} < \tilde{u}_i$, and $S_{i-1,i-1}$ and $S_{i,i}$ are Tcells. This case is illustrated in Fig. A2.6(a).

d2) The superpoint corresponds to $\tilde{u}_{i-1} = \tilde{v}_{i-1}$ and it is preceded by a Tcell and followed by a Pcell, i.e., $\tilde{v}_{i-2} < \tilde{u}_{i-1} = \tilde{v}_{i-1} < \tilde{u}_i$, $S_{i-1,i-1}$ is a Tcell and $S_{i,i}$ is a Pcell. This case is depicted in Fig. A2.6(b).

d3) The superpoint corresponds to $\tilde{u}_{i-1} = \tilde{v}_{i-1}$ and it is preceded by a Pcell and followed by a Tcell, i.e., $\tilde{v}_{i-2} < \tilde{u}_{i-1} = \tilde{v}_{i-1} < \tilde{u}_i$, $S_{i-1,i-1}$ is a Pcell and $S_{i,i}$ is a Tcell.

d4) The superpoint corresponds to $\tilde{v}_{i-1} = \tilde{u}_i$ and both adjacent cells are Tcells, i.e., $\tilde{u}_{i-1} < \tilde{v}_{i-1} = \tilde{u}_i < \tilde{v}_i$, $S_{i,i-1}$ and $S_{i,i}$ are Tcells.

d5) The superpoint corresponds to $\tilde{v}_{i-1} = \tilde{u}_i$ and it is preceded by a Tcell and followed by a Pcell, i.e., $\tilde{u}_{i-1} < \tilde{v}_{i-1} = \tilde{u}_i < \tilde{v}_i$, $S_{i,i-1}$ is a Tcell and $S_{i,i}$ is a Pcell.

d6) The superpoint corresponds to $\tilde{v}_{i-1} = \tilde{u}_i$ and it is preceded by a Pcell and followed by a Tcell, i.e., $\tilde{u}_{i-1} < \tilde{v}_{i-1} = \tilde{u}_i < \tilde{v}_i$, $S_{i,i-1}$ is a Pcell and $S_{i+1,i+1}$ is a Tcell.

It is sufficient to prove only d1) and d2) since the other cases follow similarly. The proof for case d2) proceeds as the proof for case a1) of the previous lemma. The only difference is that while in the previous lemma, the cell $S_{i,i-1}$ was a Tcell, in the current scenario, $S_{i,i-1}$ is non-relevant. However, all conditions needed for the argument to hold are still valid.

Next we provide the proof for case d1)

Case d1). By applying claim C4 of Lemma 16 for $x_1 = \tilde{u}_{i-1} = \tilde{v}_{i-1}$ and $l = \tilde{u}_i - \tilde{u}_{i-1}$, we conclude that there is some $\epsilon, 0 < \epsilon < \tilde{u}_i - \tilde{u}_{i-1}$, such that

$$\mu(\mathcal{R} \cap \mathcal{H}_+) < \mu(\mathcal{R} \cap \breve{\mathcal{H}}_-), \tag{B.16}$$

where $\mathcal{R}$ is the rectangular region $[\tilde{u}_{i-1}, \tilde{u}_i] \times [\tilde{v}_{i-1}, \tilde{v}'_{i-1}]$ for $\tilde{v}'_{i-1} = \tilde{v}_{i-1} + \epsilon$ (shown in Fig. A2.6(b)). Clearly, the inequalities $\tilde{v}_{i-1} < \tilde{v}'_{i-1} < \tilde{u}_i \leq \tilde{v}_i$ hold. Let us replace $\tilde{v}_{i-1}$ by $\tilde{v}'_{i-1}$ in Q_2, and denote by Q'_2 the new quantizer and by $\mathbf{Q}'$ the new SDSQ. This change only affects the cells V_{i-1} and V_i of Q_2 that become $V'_{i-1} = [\tilde{v}_{i-2}, \tilde{v}'_{i-1})$ and $V'_i = [\tilde{v}'_{i-1}, \tilde{v}_i)$. Thus, the only cells of the product quantizer that are affected by this

change and could have an impact on the cost are $S_{i-1,i-1}$, $S_{i,i-1}$ and $S_{i,i}$, $S_{i+1,i}$, which become $S'_{i-1,i-1} = U_{i-1} \times V'_{i-1}$, $S'_{i,i-1} = U_i \times V'_{i-1}$ and $S_{i,i} = U_i \times V'_i$, $S'_{i+1,i} = U_{i+1} \times V'_i$, respectively. We will prove the following

$$c(S'_{i+1,i}) \leq c(S_{i+1,i}), \tag{B.17}$$

$$c(S'_{i-1,i-1}) \leq c(S_{i-1,i-1}), \tag{B.18}$$

$$c(S'_{i,i}) + c(S'_{i,i-1}) < c(S_{i,i}), \tag{B.19}$$

which imply that $c(\mathbf{Q}') < c(\mathbf{Q})$.

Inequality (B.17) follows from the fact that $S'_{i+1,i} \subset S_{i+1,i}$. Relation (B.18) holds by the same argument as in the proof of (B.8).

Let us prove now inequality (B.21). Notice that $T'_{i,i-1} = S'_{i,i-1} \cap \mathcal{H}_+ = \mathcal{R} \cap \mathcal{H}_+$ and $\mathcal{R} \cap \breve{\mathcal{H}}_- \subset P'_{i,i-1}$. Using the above relations in conjunction with (B.16) and Lemma 2, we obtain $\mu(T'_{i,i-1}) = \mu(\mathcal{R} \cap \mathcal{H}_+) < \mu(\mathcal{R} \cap \breve{\mathcal{H}}_-) \leq \mu(P'_{i,i-1}$, which lead to

$$c(S'_{i,i-1}) = \mu(\mathcal{R} \cap \mathcal{H}_+) < \mu(\mathcal{R} \cap \breve{\mathcal{H}}_-). \tag{B.20}$$

Next we discuss $S'_{i,i}$ and we use the fact that $\mathcal{R} \cup S'_{i,i} = S_{i,i}$ and the fact that $S_{i,i}$ is a Tcell. It follows that

$$\begin{aligned}\mu(\mathcal{R} \cap \breve{\mathcal{H}}_-) + \mu(T'_{i,i}) &= \mu(T_{i,i}) \\ &\leq \mu(P_{i,i}) \\ &= \mu(\mathcal{R} \cap \mathcal{H}_+) + \mu(P'_{i,i}).\end{aligned}$$

The above relations together with (B.16) lead to $\mu(T'_{i,i}) \leq \mu(P'_{i,i})$, which implies that $c(S'_{i,i}) = \mu(T'_{i,i})$. Corroborating the above with (B.20) and with $\mathcal{R} \cup S'_{i,i} = S_{i,i}$, we obtain

$$\begin{aligned}c(S'_{i,i}) + c(S'_{i,i-1}) &< \mu(T'_{i,i}) + \mu(\mathcal{R} \cap \breve{\mathcal{H}}_-) \\ &= \mu(T_{i,i}) = c(S_{i,i}).\end{aligned} \tag{B.21}$$

With this, the proof of Case d1) is completed. □

Appendix C

Proof of Theorem 2

Proof of Theorem 2. According to Lemma 15, any non-strict SDSQ that has only Tcells cannot be optimum since another SDSQ of smaller cost can be constructed. This is because such an SDSQ must have an interior superpoint adjacent to a Tcell, thus we can apply Lemma 15 to reach the desired conclusion. Therefore, to complete the proof of the theorem it is sufficient to show that any SDSQ that contains at least one Pcell cannot be optimum. Clearly, if either condition in the hypothesis of Lemma 14 is satisfied then the SDSQ is not optimum (since another SDSQ of smaller cost can be constructed). Thus, there are only the following three cases left to consider:

A1) the SDSQ has only Pcells;

A2) the SDSQ has both Pcells and Tcells, and it has a strict portion that contains only Tcells;

A3) the SDSQ has both Pcells and Tcells, any Tcell is between two Pcells that are inside the same strict SDSQ portion and no Pcell is between two Tcells that are inside the same strict SDSQ portion.

A1) In this case we swap the thresholds of Q_1 and Q_2. The new DSQ $\mathbf{Q}'$ is still strictly stagerred, but its orientation has changed. We will show that all its relevant cells are Tcells.

The product quantizers of the two SDSQs are illustrated in A3.1. Note that the new sequences of thresholds are $\tilde{\mathbf{u}}' = \tilde{\mathbf{v}}$ and $\tilde{\mathbf{v}}' = \tilde{\mathbf{u}}$. Further, $S'_{j,i}$ (i.e., $U'_j \times V'_i$) is the reflection of $S_{i,j}$ (i.e., $U_i \times V_j$) across $\mathcal{L}$. This implies that

$$T_{i,j} \subseteq P'_{j,i}, \text{ and } T'_{j,i} \subseteq P_{i,j}. \tag{C.1}$$

Since each relevant cell $S_{i,j}$ is a Pcell, we have $\mu(P_{i,j}) = c(S_{i,j})$, and further

$$\mu(T'_{j,i}) \leq \mu(P_{i,j}) = c(S_{i,j}) \leq \mu(T_{i,j}) \leq \mu(P'_{j,i}),$$

which implies that

$$c(S'_{j,i}) = \mu(T'_{j,i}) \leq c(S_{i,j}). \tag{C.2}$$

Further, consider the case when there is some i_0 such that $u_{i_0} \neq v_{i_0}$. Then for $(i,j) = (i_0, j_0)$ the inequality in (C.2) is strict. This is because $area(P_{i_0,j_0} \setminus T'_{i_0,j_0}) > 0$, therefore $\mu(T'_{i_0,j_0}) < \mu(P_{i_0,j_0})$ according to Lemma 2, which leads to $c(S'_{i_0,j_0}) < c(S_{i_0,j_0})$. In this case, the new SDSQ has smaller cost.

On the other hand, when $u_i = v_i$ for all i, we have $c(\mathbf{Q}) = c(\mathbf{Q}')$. However, $\mathbf{Q}'$ is a non-strict SDSQ with only Tcells, therefore it cannot be optimal (by Lemma 15), which implies that $\mathbf{Q}$ is not optimal either.

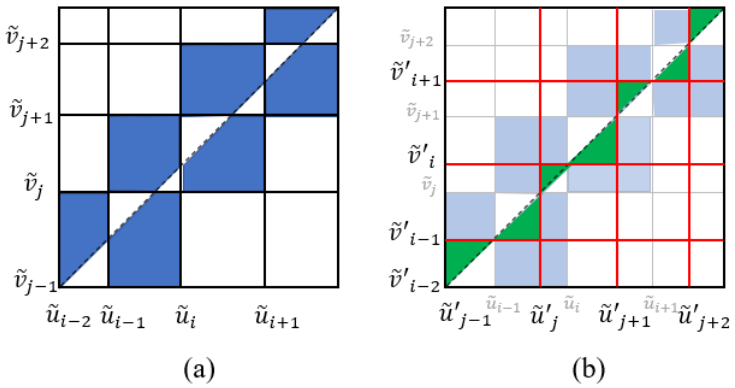


FIGURE A3.1: : (a) initial cells; (b) the resulting cells after swapping all the thresholds of Q_1 and Q_2. The initial cost regions and the new cost regions, are colored in blue and green, respectively.

A2) In this case, the SDSQ has at least one interior superpoint adjacent to a Tcell. Therefore, we can apply Lemma 15 and the conclusion follows.

A3) In this case, the necessary construction comprises two stages. In the first stage we swap the thresholds sequences of Q_1 and Q_2 and obtain the SDSQ $\mathbf{Q}'$. In the second stage, we perform some adjustments to the thresholds of $\mathbf{Q}'$.

The SDSQ $\mathbf{Q}'$ obtained after the first stage has threshold sequences $\tilde{\mathbf{u}}' = \tilde{\mathbf{v}}$ and $\tilde{\mathbf{v}}' = \tilde{\mathbf{u}}$. This way, any cell $S_{i,j}$ in $\mathbf{Q}$ is converted to $S'_{j,i} = U'_j \times V'_i$ in $\mathbf{Q}'$. As in case A1, if $S_{i,j}$ is a Pcell, then $S'_{j,i}$ is a Tcell and $c(S'_{j,i}) \leq c(S_{i,j})$. On the other hand, if $S_{i,j}$ is a Tcell, we have two cases: 1) $\mu(T_{i,j}) \geq \mu(T'_{j,i})$ and 2) $\mu(T_{i,j}) < \mu(T'_{j,i})$. If the first case holds, we will say that the Tcell $S_{i,j}$ falls in category I, otherwise we say that it falls in category II.

If $S_{i,j}$ is in Category I, then $S'_{j,i}$ is also Tcell since $\mu(T'_{j,i}) \leq \mu(T_{i,j}) \leq \mu(P'_{j,i})$ and hence $c(S'_{j,i}) \leq c(S_{i,j})$. If $S_{i,j}$ is in category II, we will modify one threshold of $S'_{j,i}$ (i.e., a threshold involved in the definition of $S'_{j,i}$) such that the new cell $S''_{j,i}$ is a Tcell,

while the status of all other cells is not affected and the sum of the costs of the cells in the modified region is smaller than in the initial SDSQ **Q**. Clearly, if all Tcells are in category I, then $\mathbf{Q}'$ has only Tcells and the same argument as in case **A1)** can be used to conclude the proof.

We continue the proof by distinguishing between two cases: **A3a)** there is only one Tcell in category II; **A3b)** there are more than one Tcell in category II.

A3a) Let $S_{i+1,j+1}$ be the only Tcell in category II. Thus we have

$$\mu(T_{i+1,j+1}) < \mu(T'_{j+1,i+1}) \tag{C.3}$$

We further consider two subcases of **A3a)**: **A3a1)** $T_{i+1,j+1} \subset \mathcal{H}_+$; **A3a21)** $T_{i+1,j+1} \subset \breve{\mathcal{H}}_-$;

A3a1) This case is illustrated in Fig. A3.2. According to the premise of case A3, the cells that are adjacent to $S_{i+1,j+1}$, i.e., $S_{i+1,j+2}$ and $S_{i,j+1}$, are Pcells and are inside the same strict SDSQ portion. Thus $S_{i+1,j+1}$ is not adjacent to any superpoint and we have

$$\tilde{u}_{i-1} \leq \tilde{v}_j < \tilde{u}_i < \tilde{v}_{j+1} < \tilde{u}_{i+1} \leq \tilde{v}_{j+2}. \tag{C.4}$$

Note that if **Q** has u-first orientation then $i = j + 1$. Otherwise, $i = j$. The above implies that

$$\tilde{v}'_{i-1} \leq \tilde{u}'_j < \tilde{v}'_i < \tilde{u}'_{j+1} < \tilde{v}'_{i+1} \leq \tilde{u}'_{j+2}. \tag{C.5}$$

Now we apply the claim C2 of Lemma 16 for $x_1 = \tilde{u}'_{j+1}$ and $l = \tilde{u}'_{j+1} - \tilde{v}'_i$, and let $\tilde{v}''_i = \tilde{u}'_{j+1} - \epsilon$ for some ϵ smaller than than the value of ϵ_2 from the lemma. Then we

have $\tilde{v}'_i < \tilde{v}''_i < \tilde{u}'_{j+1}$ and

$$\mu(\mathcal{R} \cap \breve{\mathcal{H}}_-) < \mu(\mathcal{R} \cap \mathcal{H}_+) \tag{C.6}$$

where $\mathcal{R} = [\tilde{u}_i, \tilde{u}'_{j+1}) \times [\tilde{v}''_i, \tilde{v}_{j+1})$. Now we will replace the threshold $\tilde{v}'_i$ by $\tilde{v}''_i$. The only relevant cells of $\mathbf{Q}'$ affected by this change are $S'_{j,i}$, $S'_{j+1,i}$, $S'_{j+1,i+1}$, and $S'_{j+2,i+1}$. They become respectively $S''_{j,i} = U'_j \times V''_i$, $S''_{j+1,i} = U'_{j+1} \times V''_i$, $S''_{j+1,i+1} = U'_{j+1} \times V''_{i+1}$, and $S''_{j+2,i+1} = U'_{j+2} \times V''_{i+1}$, where $V''_i = [\tilde{v}'_{i-1}, \tilde{v}''_i)$ and $V''_{i+1} = [\tilde{v}''_i, \tilde{v}'_{i+1})$. Let us analyze the costs of the new cells.

Let us start with $S''_{j,i}$. Note that $T''_{j,i} = S''_{j,i} \cap \breve{\mathcal{H}}_- = S'_{j,i} \cap \breve{\mathcal{H}}_- = T'_{j,i}$, while $P''_{j,i} \supset P'_{j,i}$. According to the specification of case **A3)**, $S_{i,j}$ cannot be a Tcell. It is either non-relevant (when $\tilde{u}_{i-1} = \tilde{v}_j$) or it is a Pcell. In the first case, $S''_{j,i}$ is non-relevant too and $c(S''_{j,i}) = 0 = c(S''_{j,i})$. In the latter case, it follows that $S'_{j,i}$ is a Tcell (as proved in case **A1**). Therefore, $\mu(T'_{j,i}) \leq \mu(P'_{j,i})$. We conclude that

$$c(S''_{j,i}) = \mu(T''_{j,i}) = c(S'_{j,i}) \leq c(S_{i,j}). \tag{C.7}$$

Consider now the cell $S''_{j+2,i+1}$. Note that $T''_{j+2,i+1} = S''_{j+2,i+1} \cap \mathcal{H}_+ \subset S_{i+1,j+2} \cap \mathcal{H}_+ = P_{i+1,j+2}$ and $P''_{j+2,i+1} = S''_{j+2,i+1} \cap \breve{\mathcal{H}}_- \supset S_{i+1,j+2} \cap \breve{\mathcal{H}}_- = T_{i+1,j+2}$. Combining the above with the fact that $S_{i+1,j+2}$ is a Pcell leads to

$$\mu(T''_{j+2,i+1}) \leq \mu(P_{i+1,j+2}) \leq \mu(T_{i+1,j+2}) \leq \mu(P''_{j+2,i+1}),$$

which further leads to the conclusion that $S''_{j+2,i+1}$ is a Tcell and

$$c(S''_{j+2,i+1}) \leq c(S_{i+1,j+2}). \tag{C.8}$$

Let us discuss now $S''_{j+1,i}$. For this denote $\mathcal{R}_1 = [\tilde{u}'_j, \tilde{u}_i) \times [\tilde{v}_j, \tilde{v}''_i) \cap \mathcal{H}_+$ and $\mathcal{R}_2 = [\tilde{u}_i, \tilde{u}'_{j+1}) \times [\tilde{v}'_i, \tilde{v}''_i)$. We have $\mu(T_{i+1,j+1}) = \mu(\mathcal{R} \cap \breve{\mathcal{H}}_+) + \mu(\mathcal{R}_2 \cap \mathcal{H}_+)$ and $\mu(T'_{j+1,i+1}) = \mu(\mathcal{R} \cap \breve{\mathcal{H}}_-) + \mu(\mathcal{R}_2 \cap \breve{\mathcal{H}}_-)$. The above combined with (C.3) and (C.6) imply that

$$\mu(\mathcal{R}_2 \cap \mathcal{H}_+) < \mu(\mathcal{R}_2 \cap \breve{\mathcal{H}}_-). \tag{C.9}$$

Further, note that $T''_{j+1,i} = S''_{j+1,i} \cap \mathcal{H}_+ = \mathcal{R}_1 \cup (\mathcal{R}_2 \cap \mathcal{H}_+)$. Since $\mathcal{R}_1$ and $\mathcal{R}_2 \cap \mathcal{H}_+$ are disjoint, we further obtain

$$\begin{aligned} \mu(T''_{j+1,i}) &= \mu(\mathcal{R}_1) + \mu(\mathcal{R}_2 \cap \mathcal{H}_+) \\ &\overset{(a)}{<} \mu(P_{i,j+1}) + \mu(\mathcal{R}_2 \cap \breve{\mathcal{H}}_-) \\ &\overset{(b)}{\leq} \mu(T_{i,j+1}) + \mu(\mathcal{R}_2 \cap \breve{\mathcal{H}}_-) \\ &\overset{(c)}{\leq} \mu(S''_{j+1,i} \cap \breve{\mathcal{H}}_-) = \mu(P''_{j+1,i}), \end{aligned} \tag{C.10}$$

where (a) follows from $\mathcal{R}_1 \subset P_{i,j+1}$ and from (C.9), (b) holds since $S_{i,j+1}$ is a Pcell, and (c) is based on the fact that $T_{i,j+1}$ and $\mathcal{R}_2 \cap \breve{\mathcal{H}}_-$ are disjoint and their union is included in $P''_{j+1,i}$. We conclude that

$$c(S''_{j+1,i}) = \mu(T''_{j+1,i}). \tag{C.11}$$

Let us analyze now the cost of $S''_{j+1,i+1}$. we have

$$\mu(T''_{j+1,i+1}) = \mu(S''_{j+1,i+1} \cap \breve{\mathcal{H}}_-) = \mu(\mathcal{R} \cap \breve{\mathcal{H}}_-) < \mu(\mathcal{R} \cap \mathcal{H}_+) < \mu(P''_{j+1,i+1}), \quad \text{(C.12)}$$

where the last inequality follows from $\mathcal{R} \cap \mathcal{H}_+ \subset S''_{j+1,i+1} \cap \mathcal{H}_+ = P''_{j+1,i+1}$. We conclude that

$$c(S''_{j+1,i+1}) = \mu(T''_{j+1,i+1}) = \mu(\mathcal{R} \cap \breve{\mathcal{H}}_-). \quad \text{(C.13)}$$

Note that $T_{i+1,j+1} = S_{i+1,j+1} \cap \mathcal{H}_+ = (\mathcal{R} \cap \mathcal{H}_+) \cup (\mathcal{R}_2 \cap \mathcal{H}_+)$ and $\mathcal{R} \cap \mathcal{H}_+$ and $\mathcal{R}_2 \cap \mathcal{H}_+$ are disjoint. Using (C.6), we have

$$\mu(\mathcal{R} \cap \breve{\mathcal{H}}_-) + \mu(\mathcal{R}_2 \cap \mathcal{H}_+) < \mu(\mathcal{R} \cap \mathcal{H}_+) + \mu(\mathcal{R}_2 \cap \mathcal{H}_+) = c(S_{i+1,j+1}). \quad \text{(C.14)}$$

We obtain

$$\begin{aligned} c(S''_{j+1,i}) + c(S''_{j+1,i+1}) &= \mu(\mathcal{R}_1) + \mu(\mathcal{R}_2 \cap \mathcal{H}_+) + \mu(\mathcal{R} \cap \breve{\mathcal{H}}_-) \\ &< \mu(P_{i,j+1}) + c(S_{i+1,j+1}) \\ &= c(S_{i,j+1}) + c(S_{i+1,j+1}). \end{aligned}$$

The above relations together with (C.7) and (C.8) imply that

$$c(S''_{j,i}) + c(S''_{j+1,i}) + c(S''_{j+1,i+1}) + c(S''_{j+2,i+1}) < c(S_{i,j}) + c(S_{i,j+1}) + c(S_{i+1,j+1}) + c(S_{i+1,j+2}).$$

Since for every other cell of $\mathbf{Q}'$ that was not affected by the change, its cost is no

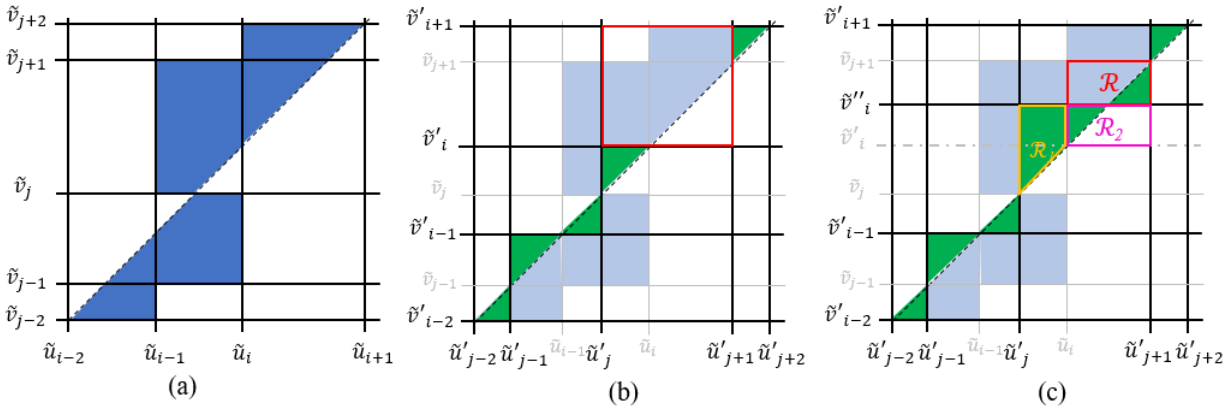


FIGURE A3.2: : (a) initial SDSQ with cost regions colored in blue; Tcell $S_{i+1,j+1}$ is of category II and Tcell $S_{i-1,j}$ is of category I; (b) resulting SDSQ after swapping the thresholds of Q_1 and Q_2; the new cost regions are shown in green; the Tcell in category II is converted to the cell shown with red outline; (c) the resulting SDSQ after replacing $\tilde{v}'_i$ with $\tilde{v}''_i$; the cost regions are shown in green.

larger than the cost of its counterpart in $\mathbf{Q}$, we conclude that the cost of the new SDSQ is strictly smaller than the cost of $\mathbf{Q}$. This case is depicted in Fig. A3.2.Thus, the proof of case **A3a1)** is completed.

A3a2) This situation is depicted in Fig. A3.3. For convenience, let us replace i by $i-2$ and j by $j-1$. In other words, the Tcell in category II is $S_{i-1,j}$. Thus we have $T_{i-1,j} \subset \breve{\mathcal{H}}_-$. Note that this change does not restrict the generality. It will be clear in the proof of case **A3b)** why we made this change.

Then we have

$$\tilde{v}_{j-2} \leq \tilde{u}_{i-2} < \tilde{v}_{j-1} < \tilde{u}_{i-1} < \tilde{v}_j \leq \tilde{u}_i. \tag{C.15}$$

The above implies that

$$\tilde{u}'_{j-2} \leq \tilde{v}'_{i-2} < \tilde{u}'_{j-1} < \tilde{v}'_{i-1} < \tilde{u}'_{j} \leq \tilde{v}'_{i}. \tag{C.16}$$

The modification we perform is to replace $\tilde{u}'_{j-1}$ by $\tilde{u}''_{j-1} = \tilde{v}'_{i-1} - \epsilon$, where $\epsilon > 0$ is some value smaller than ϵ_1 obtained by applying the claim C1 of Lemma 16 for $x_1 = \tilde{v}'_{i-1}$ and $l = \tilde{v}'_{i-1} - \tilde{u}'_{j-2}$. This implies that $\tilde{u}'_{j-1} < \tilde{u}''_{j-1} < \tilde{v}'_{i-1}$.

The only relevant cells of $\mathbf{Q}'$ affected by this change are $S'_{j-1,i-2}$, $S'_{j-1,i-1}$, $S'_{j,i-1}$, and $S'_{j,i}$. They become respectively $S''_{j-1,i-2}$, $S''_{j-1,i-1}$, $S''_{j,i-1}$, and $S''_{j,i}$.

With a similar reasoning as in case **A3a1)** we obtain

$$c(S''_{j-1,i-1}) + c(S''_{j,i-1}) < c(S_{i-1,j-1}) + c(S_{i-1,j}), \tag{C.17}$$

$$c(S''_{j-1,i-2}) \leq c(S_{i-2,j-1}) \tag{C.18}$$

$$c(S''_{j,i}) \leq c(S_{i,j}). \tag{C.19}$$

We conclude that the cost of the new SDSQ is strictly smaller than the cost of $\mathbf{Q}$ and the proof of case **A3a2)** is completed.

A3b) In this case, for each Tcell $S_{i,j}$ in n II we can change a threshold of $S'_{j,i}$ as described in the proof of case **A3a)**. We will prove that after performing all these changes, the new SDSQ has a strictly smaller cost that $\mathbf{Q}$.

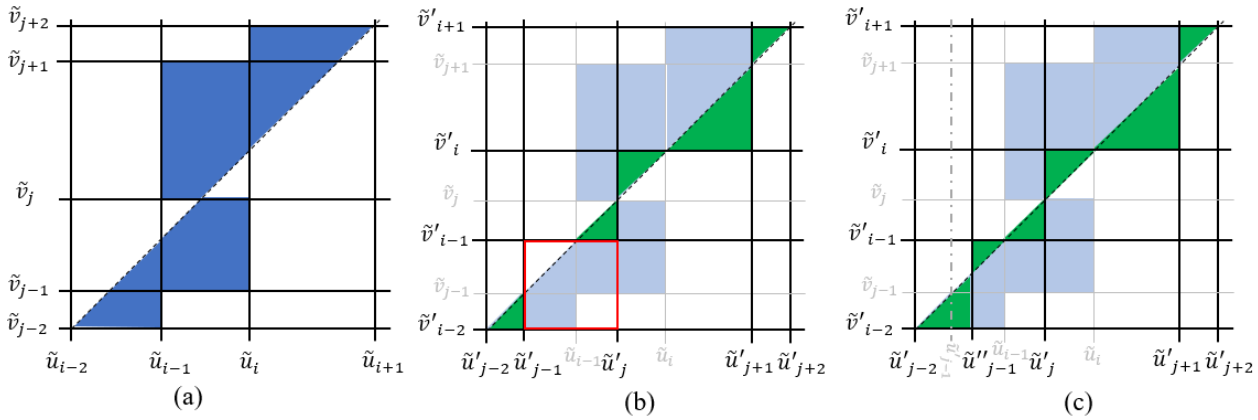


FIGURE A3.3: : (a) initial SDSQ with the cost regions colored in blue; the Tcell $S_{i+1,j+1}$ is of category I and the Tcell $S_{i-1,j}$ is of category II; (b) the resulting SDSQ after swapping the thresholds of Q_1 and Q_2; the new cost regions are shown in green; the Tcell in category II is converted to the cell shown with red outline; (c) the resulting SDSQ after replacing $\tilde{u}'_{j-1}$ with $\tilde{u}'_{j-1}$; the cost regions are colored in green.

According to the discussion in the proof of case **A3a)** if $S_{i,j}$ is a Tcell in n II, the corresponding change in **Q** affects only four relevant cells, namely $S'_{j,i}$, the two cells preceding $S'_{j,i}$ and the cell following $S'_{j,i}$. We will say that the sequence formed of these four cells is the region affected by the change.

It follows that if two Tcells in n II are separated by at least three other cells, then the regions affected by their corresponding changes do not overlap. If all the Tcells in n II satisfy the above condition then the impact of each change on the cost can be analyzed independently of other changes, as in the case **A3a)** and the claim follows.

If there are two Tcells in n II that are closer, they still must be separated by two Pcells in which case the regions affected by the two changes have an overlap, but this overlap consists of only one cell. Let S' denote this cell. Then S' is the last cell in one region and the first cell in the other region. Then similar arguments to those used in in the proof of case **A3a)** can be used to conclude that $c(S'') \leq c(S)$, where S'' is the cell

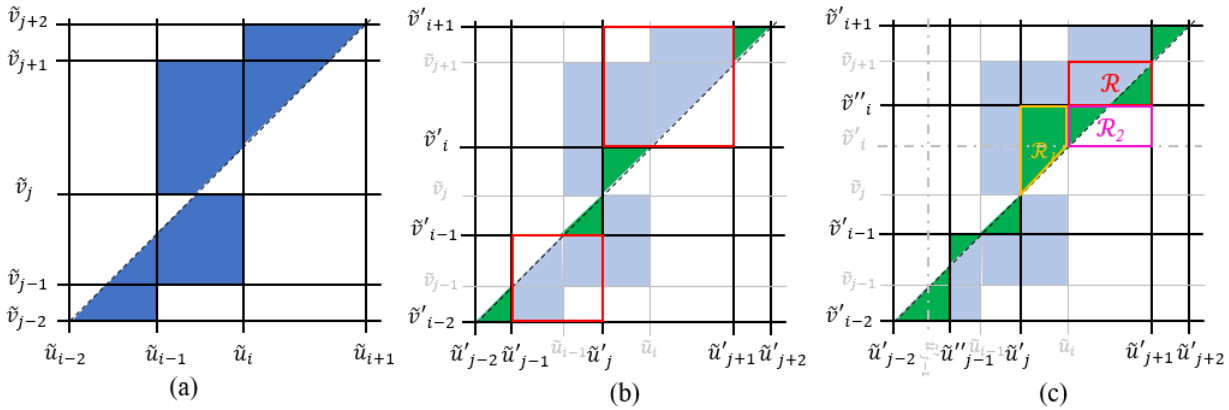


FIGURE A3.4: : (a) initial SDSQ with cost regions colored in blue; both Tcells are of category II; (b) the resulting SDSQ after flipping the thresholds of Q_1 and Q_2; the new cost regions are shown in green; the two Tcells of category II are converted to the cells shown with red outline; (c) the resulting SDSQ after replacing $\tilde{v}'_i$ with $\tilde{v}''_i$ and $\tilde{u}'_{j-1}$ with $\tilde{u}''_{j-1}$. The new cost regions are colored in green.

obtained after both changes ae performed and S is the counterpart in $\mathbf{Q}$. An example is when the two Tcells are $T_{i+1,j+1}$ discussed in case **A3a1)** and the other Tcell is $T_{i-1,j}$ discussed in case**A3a2)** . Then $S' = S'_{j,i}$. Figure A3.4 illustrates this situation.

For the cells with non-overlapping regions the same analysis as in case **A3a)** can be performed. According to the above discussion, the new SDSQ will have strictly smaller cost.

□

Appendix D

Proof of Theorem 3

First we state two lemmas that are used in the proof of Theorem 3. After that we prove the theorem and proceed to proving the lemmas.

Lemma 17. *Assume that conditions* $\mathbf{OC}_c$ *and* $\mathbf{S}_c$ *hold. Consider a* (K, K)*-level DSQ* (Q_1, Q_2) *with threshold sequences* $\tilde{\mathbf{u}}$ *and* $\tilde{\mathbf{v}}$ *such that for some* $j_0, 1 < j_0 \leq K - 1$

$$\tilde{u}_i \leq \tilde{v}_i, \text{ for all } i, 1 \leq i \leq j_0 - 1 \tag{D.1}$$

$$\tilde{v}_i \leq \tilde{u}_{i+1} \text{ for all } i, 1 \leq i \leq j_0 - 1 \tag{D.2}$$

$$\tilde{v}_{j_0} < \tilde{u}_{j_0}. \tag{D.3}$$

Then we can construct a DSQ (Q_1', Q_2') *with the cost no larger than* (Q_1, Q_2) *with threshold sequences* $\tilde{\mathbf{u}}'$ *and* $\tilde{\mathbf{v}}'$ *such that*

$$\tilde{u}_i' \leq \tilde{v}_i', \text{ for all } i, 1 \leq i \leq j_0 \tag{D.4}$$

$$\tilde{v}_i' \leq \tilde{u}_{i+1}' \text{ for all } i, 1 \leq i \leq j_0 - 1. \tag{D.5}$$

Lemma 18. *Assume that conditions* $\mathbf{OC}_c$ *and* $\mathbf{S}_c$ *hold. Consider a* (K, K)*-level DSQ* (Q_1, Q_2) *with threshold sequences* $\tilde{\mathbf{u}}$ *and* $\tilde{\mathbf{v}}$ *such that for some* $j_0, 1 < j_0 \leq K-1$*,*

$$\tilde{u}_i \leq \tilde{v}_i, \text{ for all } i, 1 \leq i \leq j_0, \tag{D.6}$$

$$\tilde{v}_i \leq \tilde{u}_{i+1} \text{ for all } i, 1 \leq i \leq j_0 - 1, \tag{D.7}$$

$$\tilde{u}_{j_0+1} < \tilde{v}_{j_0}. \tag{D.8}$$

Then we can construct a DSQ (Q_1', Q_2') *with the cost no larger than* (Q_1, Q_2) *with threshold sequences* $\tilde{\mathbf{u}}'$ *and* $\tilde{\mathbf{v}}'$ *such that*

$$\tilde{u}_i' \leq \tilde{v}_i', \text{ for all } i, 1 \leq i \leq j_0, \tag{D.9}$$

$$\tilde{v}_i' \leq \tilde{u}_{i+1}' \text{ for all } i, 1 \leq i \leq j_0. \tag{D.10}$$

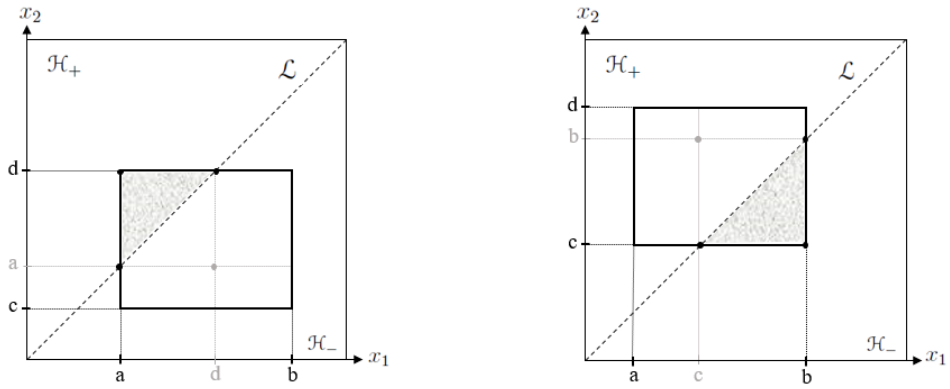
Proof of Theorem 3. Starting from any (K, K)-level DSQ that is not a u-first staggered DSQ, by applying repeatedly Lemmas 17 and 18, we obtain a (K, K)-level u-first staggered DSQ with a cost that is no larger than that of the initial DSQ. By applying further Theorem 2, the conclusion follows. □

Next we present a lemma that will be used in the proofs of Lemmas 17 and 18.

Lemma 19. *Assume that conditions* $\mathbf{OC}_c$ *and* $\mathbf{S}_c$ *hold. Let* $S = [a, b) \times [c, d)$*, where* $a < b$ *and* $c < d$*.*

a) If $c \leq a \leq d \leq b$*, then* $c(S) = \mu(S \cap \mathcal{H}_+) = c([a, d) \times [a, d))$.

b) If $a \leq c \leq b \leq d$*, then* $c(S) = \mu(S \cap \breve{\mathcal{H}}_-) = c([c, b) \times [c, b))$.



(A) First case of Lemma 19. (B) Second case of Lemma 19.

FIGURE A4.1: The region used to calculate the cost is shown in grey.

Proof. a) Fig. A4.1(a) illustrates this case. First notice that $S \cap \mathcal{H}_+$ is a triangular region whose vertexes are the points (a, a), (a, d) and (d, d). The reflection of $\mathcal{S} \cap \mathcal{H}_+$ is the triangular region with vertexes (a, a), (d, a) and (d, d), which is included in $\mathcal{S} \cap \mathcal{H}_-$. In other words $\sigma(\mathcal{S} \cap \mathcal{H}_+) \subseteq \mathcal{S} \cap \mathcal{H}_-$. In view of Lemmas 4 and 2, it follows that $\mu(\mathcal{S} \cap \mathcal{H}_+) = \mu(\sigma(\mathcal{S} \cap \mathcal{H}_+)) \leq \mu(\mathcal{S} \cap \mathcal{H}_-) = \mu(\mathcal{S} \cap \breve{\mathcal{H}}_-)$. The proof for this case is complete.

b) Fig. A4.1(b) illustrates this case. The proof is similar to the proof for the previous case.

□

Proof of Lemma 17. First let us note that we have $\tilde{u}_{j_0-1} \leq \tilde{v}_{j_0-1} < \tilde{v}_{j_0} < \tilde{u}_{j_0}$. This can be seen in Fig. A4.2-(a). Let us construct $\tilde{\mathbf{u}}'$ and $\tilde{\mathbf{v}}'$ from $\tilde{\mathbf{u}}$ and $\tilde{\mathbf{v}}$ by exchanging $\tilde{u}_i$ and

$\tilde{v}_i$ for all $i \geq j_0$. In other words, we have

$$\tilde{u}'_j = \begin{cases} \tilde{u}_j & \text{if } j \leq j_0 - 1 \\ \tilde{v}_j & \text{if } j \geq j_0 \end{cases} \tag{D.11}$$

$$\tilde{v}'_k = \begin{cases} \tilde{v}_k & \text{if } k \leq j_0 - 1 \\ \tilde{u}_k & \text{if } k \geq j_0 \end{cases} \tag{D.12}$$

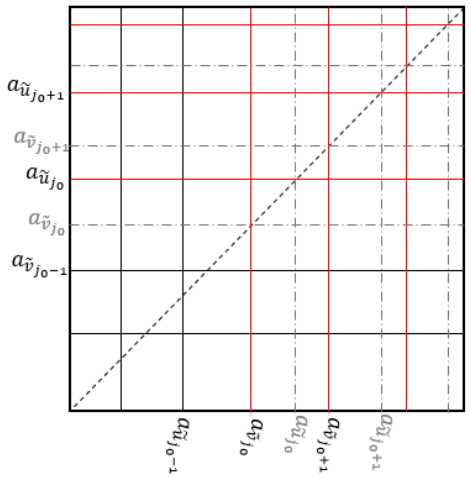
Let the cells of Q'_1 be denoted $U'_j = [a_{\tilde{u}'_{j-1}}, a_{\tilde{u}'_j})$, for $1 \leq j \leq K$, and the cells of Q'_2 be denoted $V'_k = [a_{\tilde{v}'_{k-1}}, a_{\tilde{v}'_k})$, for $1 \leq k \leq K$. Then we have

$$U'_j = \begin{cases} U_j & \text{if } j \leq j_0 - 1 \\ [a_{\tilde{u}_{j_0-1}}, a_{\tilde{v}_{j_0}}) & \text{if } j = j_0 \\ V_j & \text{if } j \geq j_0 + 1 \end{cases} \tag{D.13}$$

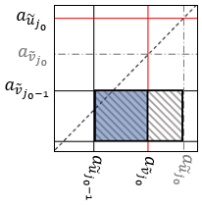
$$V'_k = \begin{cases} V_k & \text{if } k \leq j_0 - 1 \\ [a_{\tilde{v}_{j_0-1}}, a_{\tilde{u}_{j_0}}) & \text{if } k = j_0 \\ U_k & \text{if } k \geq j_0 + 1 \end{cases} \tag{D.14}$$

The new arrangement of the cells can be seen in Fig. A4.2-(a).

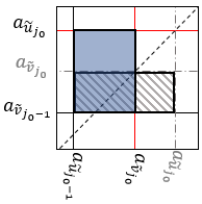
We will evaluate the cost of each bin $U'_j \times V'_k$ in the product quantizer of the new DSQ in comparison with the cost of the old bins. Let us use the simplified notation $\alpha(j, k) =$



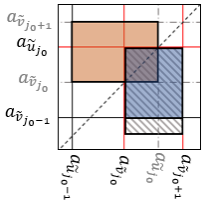
(A) New and previous arrangement



(B) Case 4: $U'_j \times V'_k$ in blue, $U_j \times V_k$ in striped gray.



(C) Case 5: $U'_j \times V'_k$ in blue, $U_j \times V_k$ in striped gray.



(D) Case 8: $U'_j \times V'_k$ in blue, $V_j \times U_k$ in striped gray, $U_k \times V_j$ in orange.

FIGURE A4.2: the initial thresholds that changed are shown with dashed lines. The solid black lines indicate the thresholds that did not change. Solid red lines indicate new thresholds.

$c(U_j \times V_k)$ and $\alpha'(j,k) = c(U'_j \times V'_k)$, for all $1 \leq j \leq K-1$ and $1 \leq k \leq K-1$. Based on (D.13) and (D.14), we distinguish between ten cases for the pairs (j,k). Next, we will treat each case separately. We mention that for all integers $1 \leq m < n \leq M$, we will use the notation $Sq(m,n)$ for the square region $[a_m, a_n) \times [a_m, a_n)$. In addition, for any set $b \subset \mathbb{R}$, we denote by $inf(B)$ the infimum of B and by $sup(B)$ the supremum of B.

1) $j \leq j_0 - 1, k \leq j_0 - 1$. Then $U'_j = U_j$ and $V'_k = V_k$ leading to $\alpha'(j,k) = \alpha(j,k)$.

2) $j \leq j_0 - 1$, $k = j_0$. Then $U'_j = U_j = [a_{\tilde{u}_{j-1}}, a_{\tilde{u}_j})$ and $V'_{j_0} = [a_{\tilde{v}_{j_0-1}}, a_{\tilde{u}_{j_0}})$. Since $sup(U'_j) = a_{\tilde{u}_j} \leq a_{\tilde{u}_{j_0-1}} \leq a_{\tilde{v}_{j_0-1}} = inf(V'_{j_0})$, it follows that $x_1 < x_2$ for all $(x_1, x_2) \in U'_j \times V'_V$. Thus, we have $U'_j \times V'_{j_0} \subseteq \mathcal{H}_+$, which implies that $\alpha'(j, j_0) = 0 \leq \alpha(j, j_0)$.

3) $j \leq j_0 - 1, k \geq j_0 + 1$. Then $U'_j = U_j = [a_{\tilde{u}_{j-1}}, a_{\tilde{u}_j})$ and $V'_k = U_k = [a_{\tilde{u}_{k-1}}, a_{\tilde{u}_k})$. Since $sup(U'_j) = a_{\tilde{u}_j} \leq a_{\tilde{u}_{j_0-1}} < a_{\tilde{u}_{j_0}} \leq a_{\tilde{u}_{k-1}} = inf(V'_k)$, it follows that $U'_j \times V'_k \subseteq \mathcal{H}_+$, and further that $\alpha'(j,k) = 0 \leq \alpha(j,k)$.

4) $j = j_0, k \leq j_0 - 1$. Then $U'_{j_0} = [a_{\tilde{u}_{j_0-1}}, a_{\tilde{v}_{j_0}}) \subseteq U_{j_0}$ (since $\tilde{v}_{j_0} < \tilde{u}_{j_0}$) and $V'_k = V_k = [a_{\tilde{v}_{k-1}}, a_{\tilde{v}_k})$. It follows that $U'_{j_0} \times V'_k \subseteq U_{j_0} \times V_k$, leading to $\alpha'(j_0,k) \leq \alpha(j_0,k)$ in virtue of Lemma 2. This case is illustrated in Fig. A4.2(b) for $k = j_0 - 1$, and in Fig. A4.2(f) for $k \leq j_0 - 2$.

5) $j = j_0, k = j_0$. Then $U'_{j_0} = [a_{\tilde{u}_{j_0-1}}, a_{\tilde{v}_{j_0}})$ and $V'_{j_0} = [a_{\tilde{v}_{j_0-1}}, a_{\tilde{u}_{j_0}})$ (Fig. A4.2(c)). Since $\tilde{u}_{j_0-1} \leq \tilde{v}_{j_0-1} < \tilde{v}_{j_0} < \tilde{u}_{j_0}$, by applying Lemma 19, we obtain that $\alpha'(j_0,j_0) = c(Sq(\tilde{v}_{j_0-1}, \tilde{v}_{j_0}))$. Further, note that $Sq(\tilde{v}_{j_0-1}, \tilde{v}_{j_0}) \subseteq U_{j_0} \times V_{j_0}$ and by Lemma 2, we obtain that $c(Sq(\tilde{v}_{j_0-1}, \tilde{v}_{j_0})) \leq \alpha(j_0,j_0)$. Combining the above two inequalities leads to $\alpha'(j_0,j_0) \leq \alpha(j_0,j_0)$.

6) $j = j_0, k \geq j_0 + 1$. Then $U'_{j_0} = [a_{\tilde{u}_{j_0-1}}, a_{\tilde{v}_{j_0}})$ and $V'_k = U_k = [a_{\tilde{u}_{k-1}}, a_{\tilde{u}_k})$. Since $sup(U'_{j_0}) = a_{\tilde{v}_{j_0}} < a_{\tilde{u}_{j_0}} \leq a_{\tilde{u}_{k-1}} = inf(V'_k)$, it follows that $U'_{j_0} \times V'_k \subseteq \mathcal{H}_+$, which implies that $\alpha'(j_0,k) = 0$.

7) $j \geq j_0 + 1, k \leq j_0 - 1$. Then $U'_j = V_j = [a_{\tilde{v}_{j-1}}, a_{\tilde{v}_j})$ and $V'_k = V_k = [a_{\tilde{v}_{k-1}}, a_{\tilde{v}_k})$. Since $sup(V'_k) = a_{\tilde{v}_k} \leq a_{\tilde{v}_{j_0-1}} < a_{\tilde{v}_{j_0}} \leq a_{\tilde{v}_{j-1}} = inf(U'_j)$, it follows that $U'_j \times V'_k \subseteq \breve{\mathcal{H}}_-$, yielding $\alpha'(j,k) = 0 \leq \alpha(j,k)$.

8) $j = j_0+1, k = j_0$. Then $U'_{j_0+1} = V_{j_0+1} = [a_{\tilde{v}_{j_0}}, a_{\tilde{v}_{j_0+1}})$ and $V'_{j_0} = [a_{\tilde{v}_{j_0-1}}, a_{\tilde{u}_{j_0}}) \subseteq U_{j_0}$ (since $\tilde{u}_{j_0-1} \leq \tilde{v}_{j_0-1}$) (Fig. A4.2(d)). It follows that $U'_{j_0+1} \times V'_{j_0} \subseteq V_{j_0+1} \times U_{j_0} = \sigma(U_{j_0} \times V_{j_0+1})$. By applying Lemma 2 and Lemma 4, we obtain that $\alpha'(j_0+1, j_0) \leq \alpha(j_0, j_0+1)$. Further, recall that according to case 6 we have

$\alpha'(j_0, j_0+1) = 0$. It follows that $\alpha'(j_0+1, j_0) + \alpha'(j_0, j_0+1) \leq \alpha(j_0+1, j_0) + \alpha(j_0, j_0+1)$.

9) $j \geq j_0+2$, $k = j_0$. Then $U'_j = V_j = [a_{\tilde{v}_{j-1}}, a_{\tilde{v}_j})$ and $V'_{j_0} = [a_{\tilde{v}_{j_0-1}}, a_{\tilde{u}_{j_0}})$. Note that $sup(V'_{j_0}) = a_{\tilde{u}_{j_0}} < a_{\tilde{u}_{j_0+1}} \leq a_{\tilde{v}_{j-1}} = inf(U'_j)$, which leads to $U'_j \times V'_{j_0} \subseteq \breve{\mathcal{H}}_-$, and further to $\alpha'(j, j_0) = 0 \leq \alpha(j, j_0)$.

10) $j \geq j_0+1$, $k \geq j_0+1$. Then $U'_j = V_j = [a_{\tilde{v}_{j-1}}, a_{\tilde{v}_j})$ and $V'_k = U_k = [a_{\tilde{u}_{k-1}}, a_{\tilde{u}_k})$. Then $U'_j \times V'_k = V_j \times U_k = \sigma(U_k \times V_k)$, leading to $\alpha'(j,k) = \alpha(k,j)$ in virtue of Lemma 4. By adding the costs for all bins falling in this case, we obtain

$$\sum_{j=j_0+1}^{K} \sum_{k=j_0+1}^{K} \alpha'(j,k) = \sum_{k=j_0+1}^{K} \sum_{j=j_0+1}^{K} \alpha(k,j) = \sum_{j=j_0+1}^{K} \sum_{k=j_0+1}^{K} \alpha(j,k). \quad \text{(D.15)}$$

Finally, by combining the results from all aforementioned ten cases, leads to

$$c(Q'_1, Q'_2) = \sum_{j=1}^{K} \sum_{k=1}^{K} \alpha'(j,k) \leq \sum_{j=1}^{K} \sum_{k=1}^{K} \alpha(j,k) = c(Q_1, Q_2). \quad \text{(D.16)}$$

and the proof is complete.

□

Proof of Lemma 18. Note that $\tilde{u}_{j_0-1} \leq \tilde{v}_{j_0-1} \leq \tilde{u}_{j_0} < \tilde{u}_{j_0+1} < \tilde{v}_{j_0}$. Let j_1 be the smallest integer such that $j_0 < j_1 \leq K-1$ and $\tilde{v}_{j_1} \leq \tilde{u}_{j_1+1}$. Note that such an integer must exist since $\tilde{v}_{K-1} < \tilde{u}_K = M+1$. Then $\tilde{u}_{j_1} < \tilde{v}_{j_1-1} < \tilde{v}_{j_1} \leq \tilde{u}_{j_1+1}$. We construct $\tilde{\mathbf{u}}'$ and $\tilde{\mathbf{v}}'$ by starting from $\tilde{\mathbf{u}}$ and $\tilde{\mathbf{v}}$ and exchanging $\tilde{u}_{j_0+1}, \cdots, \tilde{u}_{j_1}$ with $\tilde{v}_{j_0}, \cdots, \tilde{u}_{j_1-1}$,

respectively. In other words,

$$\tilde{u}'_j = \begin{cases} \tilde{u}_j & \text{if } j \leq j_0 \text{ or } j \geq j_1 + 1 \\ \tilde{v}_{j-1} & \text{if } j_0 + 1 \leq j \leq j_1 \end{cases}, \tag{D.17}$$

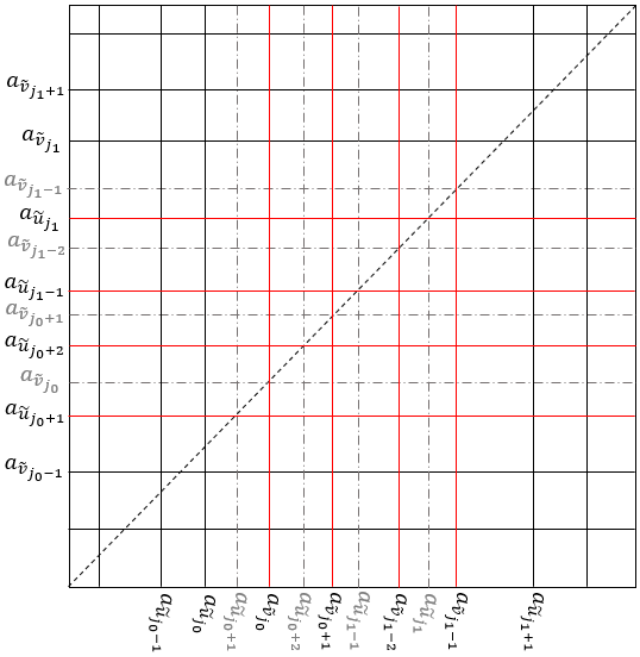
$$\tilde{v}'_k = \begin{cases} \tilde{v}_k & \text{if } k \leq j_0 - 1 \text{ or } k \geq j_1 \\ \tilde{u}_{k+1} & \text{if } j_0 \leq k \leq j_1 - 1 \end{cases}. \tag{D.18}$$

It follows that

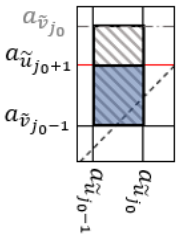
$$U'_j = \begin{cases} U_j & \text{if } j \leq j_0 \text{ or } j \geq j_1 + 2 \\ V_{j-1} & \text{if } j_0 + 2 \leq j \leq j_1 \\ [a_{\tilde{u}_{j_0}}, a_{\tilde{v}_{j_0}}) & \text{if } j = j_0 + 1 \\ [a_{\tilde{v}_{j_1-1}}, a_{\tilde{u}_{j_1+1}}) & \text{if } j = j_1 + 1 \end{cases}, \tag{D.19}$$

$$V'_k = \begin{cases} V_k & \text{if } k \leq j_0 - 1 \text{ or } k \geq j_1 + 1 \\ U_{k+1} & \text{if } j_0 + 1 \leq k \leq j_1 - 1 \\ [a_{\tilde{v}_{j_0-1}}, a_{\tilde{u}_{j_0+1}}) & \text{if } k = j_0 \\ [a_{\tilde{u}_{j_1}}, a_{\tilde{v}_{j_1}}) & \text{if } k = j_1 \end{cases}. \tag{D.20}$$

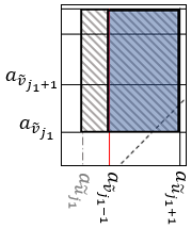
An example of this situation and the new thresholds can be seen in Fig. A4.3-(a). We will distinguish between 20 cases for the pairs (j, k). We will consider each of them separately.



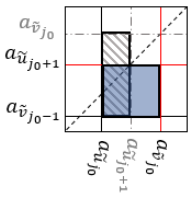
(A) New and previous arrangement



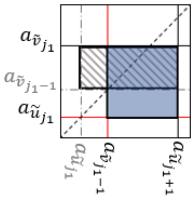
(B) Case 4: $U'_j \times V'_k$ in blue, $U_j \times V_k$ in striped gray.



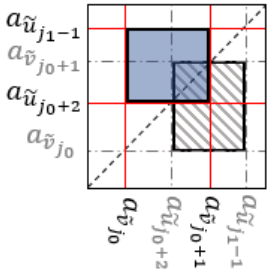
(C) Case 19: $U'_j \times V'_k$ in blue, $U_j \times V_k$ in striped gray.



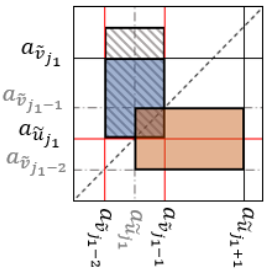
(D) Case 16: $U'_j \times V'_k$ in blue, $U_j \times V_k$ in striped gray.



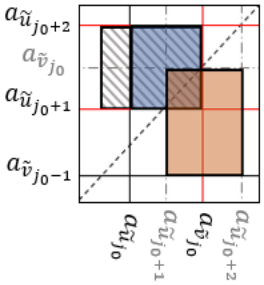
(E) Case 20: $U'_j \times V'_k$ in blue, $U_j \times V_k$ in striped gray.



(F) Case 9: $U'_j \times V'_k$ in blue, $U_{k+1} \times V_{j-1}$ in striped gray.



(G) Case 11: $U'_j \times V'_k$ in blue, $V_{j-1} \times U_{k+1}$ in striped gray, and $U_{k+1} \times V_{j-1}$ in orange.



(H) Case 15: $U'_j \times V'_k$ in blue, $V_{j-1} \times U_{k+1}$ in striped gray, and $U_{k+1} \times V_{j-1}$ in orange

FIGURE A4.3: the initial thresholds that changed are shown with dashed lines. The solid black lines indicate the thresholds that did not change. Solid red lines indicate new thresholds.

1) $j \leq j_0$ or $j \geq j_1 + 2$, while $k \leq j_0 - 1$ or $k \geq j_1 + 1$. Then $U'_j = U_j$ and $V'_k = V_k$ leading to $\alpha'(j,k) = \alpha(j,k)$.

2) $j \leq j_0$, $j_0+1 \leq k \leq j_1-1$. Then $U'_j = U_j = [a_{\tilde{u}_{j-1}}, a_{\tilde{u}_j})$ and $V'_k = U_{k+1} = [a_{\tilde{u}_k}, a_{\tilde{u}_{k+1}})$. Since $sup(U'_j) = a_{u_j} \leq a_{\tilde{u}_{j_0}} < a_{\tilde{u}_{j_0+1}} \leq a_{\tilde{u}_k} = inf(V'_k)$, it follows that $U'_j \times V'_k \subseteq \mathcal{H}_+$, which leads to $\alpha'(j,k) = 0$.

3) $j \geq j_1+2$, $j_0+1 \leq k \leq j_1-1$. Then $U'_j = U_j = [a_{\tilde{u}_{j-1}}, a_{\tilde{u}_j})$ and $V'_k = U_{k+1} = [a_{\tilde{u}_k}, a_{\tilde{u}_{k+1}})$. We have $sup(V'_k) = a_{\tilde{u}_{k+1}} \leq a_{\tilde{u}_{j_1}} < a_{\tilde{u}_{j_1+1}} \leq a_{\tilde{u}_{j-1}} = inf(U'_j)$, leading to $U'_j \times V'_k \subseteq \breve{\mathcal{H}}_-$, and further to $\alpha'(j,k) = 0$.

4) $j \leq j_0$ or $j \geq j_1+2$, $k = j_0$. Then $U'_j = U_j = [a_{\tilde{u}_{j-1}}, a_{\tilde{u}_j})$ and $V'_{j_0} = [a_{\tilde{v}_{j_0-1}}, a_{\tilde{u}_{j_0+1}}) \subseteq V_{j_0}$ (since $\tilde{u}_{j_0+1} < \tilde{v}_{j_0}$). It follows that $U'_j \times V'_{j_0} \subseteq U_j \times V_{j_0}$, which implies that $\alpha'(j,j_0) \leq \alpha(j,j_0)$ by Lemma 2. An illustration of this case can be seen in Fig. A4.3-(b).

5) $j \leq j_0$, $k = j_1$. Then $U'_j = U_j = [a_{\tilde{u}_{j-1}}, a_{\tilde{u}_j})$ and $V'_{j_1} = [a_{\tilde{u}_{j_1}}, a_{\tilde{v}_{j_1}})$. Note that $sup(U'_j) = a_{\tilde{u}_j} \leq a_{\tilde{u}_{j_0}} < a_{\tilde{u}_{j_1}} = inf(V'_k)$. Thus, we have $U'_j \times V'_k \subseteq \mathcal{H}_+$, yielding $\alpha'(j,k) = 0$.

6) $j \geq j_1+2$, $k = j_1$. Then $U'_j = U_j = [a_{\tilde{u}_{j-1}}, a_{\tilde{u}_j})$ and $V'_{j_1} = [a_{\tilde{u}_{j_1}}, a_{\tilde{v}_{j_1}})$. Since $sup(V'_{j_1}) = a_{\tilde{v}_{j_1}} < a_{\tilde{v}_{j_1+1}} \leq a_{\tilde{u}_{j-1}} = inf(U'_j)$, it follows that $U'_j \times V'_k \subseteq \breve{\mathcal{H}}_-$, which implies that $\alpha'(j,k) = 0$.

7) $j_0+2 \leq j \leq j_1$, $k \leq j_0-1$. Then $U'_j = V_{j-1} = [a_{\tilde{v}_{j-2}}, a_{\tilde{v}_{j-1}})$ and $V'_k = V_k = [a_{\tilde{v}_{k-1}}, a_{\tilde{v}_k})$. Note that $sup(V'_k) = a_{\tilde{v}_k} \leq a_{\tilde{v}_{j_0-1}} < a_{\tilde{v}_{j_0}} \leq a_{\tilde{v}_{j-2}} = inf(U'_j)$, which implies that $U'_j \times V'_k \subseteq \breve{\mathcal{H}}_-$, yielding $\alpha'(j,k) = 0$.

8) $j_0+2 \leq j \leq j_1$, $k \geq j_1+1$. Then $U'_j = V_{j-1} = [a_{\tilde{v}_{j-2}}, a_{\tilde{v}_{j-1}})$ and $V'_k = V_k = [a_{\tilde{v}_{k-1}}, a_{\tilde{v}_k})$. We have $sup(U'_j) = a_{\tilde{v}_{j-1}} \leq a_{\tilde{v}_{j_1-1}} < a_{\tilde{v}_{j_1}} \leq a_{\tilde{v}_{k-1}} = inf(V'_k)$, which leads to $U'_j \times V'_k \subseteq \mathcal{H}_+$, and further to $\alpha'(j,k) = 0$.

9) $j_0+2 \leq j \leq j_1$, $j_0+1 \leq k \leq j_1-1$. Then $U'_j = V_{j-1} = [a_{\tilde{v}_{j-2}}, a_{\tilde{v}_{j-1}})$ and $V'_k = U_{k+1} = [a_{\tilde{u}_k}, a_{\tilde{u}_{k+1}})$. An example of this case for $j = k = j_0+2$ is shown in Fig. A4.3-(f). It follows that $U'_j \times V'_k = \sigma(U_{k+1} \times V_{j-1})$, which implies that $\alpha'(j,k) = \alpha(k+1, j-1)$ by Lemma 4. By summing over all j and k corresponding to this case, we obtain that

$$\sum_{j=j_0+2}^{j_1} \sum_{k=j_0+1}^{j_1-1} \alpha'(j,k) = \sum_{j=j_0+2}^{j_1} \sum_{k=j_0+1}^{j_1-1} \alpha(k+1, j-1) = \sum_{j'=j_0+2}^{j_1} \sum_{k'=j_0+1}^{j_1-1} \alpha(j', k'),$$

where the last equality is obtained by using the change of variables $j' = k+1$ and $k' = j-1$ and by changing the order of the two summations.

10) $j_0+2 \leq j \leq j_1$, $k = j_0$. Then $U'_j = V_{j-1} = [a_{\tilde{v}_{j-2}}, a_{\tilde{v}_{j-1}})$ and $V'_{j_0} = [a_{\tilde{v}_{j_0-1}}, a_{\tilde{u}_{j_0+1}})$. The fact that $sup(V'_{j_0}) = a_{\tilde{u}_{j_0+1}} < a_{\tilde{v}_{j_0}} \leq a_{\tilde{v}_{j-2}} = inf(U'_j)$ implies that $U'_j \times V'_{j_0} \subseteq \breve{\mathcal{H}}_-$, further leading to $\alpha'(j, j_0) = 0$.

11) $j_0+2 \leq j \leq j_1$, $k = j_1$. Then $U'_j = V_{j-1} = [a_{\tilde{v}_{j-2}}, a_{\tilde{v}_{j-1}})$ and $V'_{j_1} = [a_{\tilde{u}_{j_1}}, a_{\tilde{v}_{j_1}}) \subseteq U_{j_1+1}$ (since $\tilde{v}_{j_1} \leq \tilde{u}_{j_1+1}$). An example of this case for $j = k = j_1$ is shown in Fig. A4.3-(g). Then we have $U'_j \times V'_{j_1} \subseteq V_{j-1} \times U_{j_1+1} = \sigma(U_{j_1+1} \times V_{j-1})$. In virtue of Lemma 2 and Lemma 4, we obtain that $\alpha'(j, j_1) \leq \alpha(j_1+1, j-1)$. By summing over all j's, we obtain

$$\sum_{j=j_0+2}^{j_1} \alpha'(j, j_1) \leq \sum_{j=j_0+2}^{j_1} \alpha(j_1+1, j-1). \tag{D.21}$$

12) $j = j_1+1$, $j_0+1 \leq k \leq j_1-1$. Then $U'_{j_1+1} = [a_{\tilde{v}_{j_1-1}}, a_{\tilde{u}_{j_1+1}})$ and $V'_k = U_{k+1} = [a_{\tilde{u}_k}, a_{\tilde{u}_{k+1}})$. Since $sup(V'_k) = a_{\tilde{u}_{k+1}} \leq a_{\tilde{u}_{j_1}} < a_{\tilde{v}_{j_1-1}} = inf(U'_j)$, it follows that $U'_j \times V'_k \subseteq \breve{\mathcal{H}}_-$, and further that $\alpha'(j_1+1, k) = 0$. By summing over all k's, we

obtain

$$\sum_{k=j_0+1}^{j_1-1} \alpha'(j_1+1,k) = 0. \tag{D.22}$$

After replacing k by $j-1$ in (D.22), and combining with (D.21), we obtain that

$$\sum_{j=j_0+2}^{j_1} (\alpha'(j,j_1) + \alpha'(j_1+1,j-1)) \leq \sum_{j=j_0+2}^{j_1} (\alpha(j,j_1) + \alpha(j_1+1,j-1)). \tag{D.23}$$

13) $j = j_0+1$, $k \leq j_0-1$. Then $U'_{j_0+1} = [a_{\tilde{u}_{j_0}}, a_{\tilde{v}_{j_0}})$ and $V'_k = V_k = [a_{\tilde{v}_{k-1}}, a_{\tilde{v}_k})$. Then $sup(V'_k) = a_{\tilde{v}_k} \leq a_{\tilde{v}_{j_0-1}} \leq a_{\tilde{u}_{j_0}} = inf(U'_{j_0+1})$, leading to $U'_j \times V'_k \subseteq \breve{\mathcal{H}}_-$, and further to $\alpha'(j,k) = 0$.

14) $j = j_0+1$, $k \geq j_1+1$. Then $U'_{j_0+1} = [a_{\tilde{u}_{j_0}}, a_{\tilde{v}_{j_0}})$ and $V'_k = V_k = [a_{\tilde{v}_{k-1}}, a_{\tilde{v}_k})$. We have $sup(U'_{j_0+1}) = a_{\tilde{v}_{j_0}} < a_{\tilde{v}_{j_1}} \leq a_{\tilde{v}_{k-1}} = inf(V'_k)$, yielding $U'_j \times V'_k \subseteq \mathcal{H}_+$, and further $\alpha'(j,k) = 0$.

15) $j = j_0+1$, $j_0+1 \leq k \leq j_1-1$. Then $U'_{j_0+1} = [a_{\tilde{u}_{j_0}}, a_{\tilde{v}_{j_0}}) \subseteq V_{j_0}$ (since $\tilde{v}_{j_0-1} \leq \tilde{u}_{j_0}$) and $V'_k = U_{k+1} = [a_{\tilde{u}_k}, a_{\tilde{u}_{k+1}})$. These imply that $U'_{j_0+1} \times V'_k \subseteq V_{j_0} \times U_{k+1} = \sigma(U_{k+1} \times V_{j_0})$. An example of this case for $j = k = j_0+1$ is shown in Fig. A4.3-(h). According to Lemma 2 and Lemma 4, it follows that $\alpha'(j_0+1,k) \leq \alpha(k+1,j_0)$. Recall the result of case 10, namely that $\alpha'(k+1,j_0) = 0$ for all $j_0+1 \leq k \leq j_1-1$ (obtained after replacing j by $k+1$). By combining the two results and summing over all k's, leads to

$$\sum_{k=j_0+1}^{j_1-1} (\alpha'(j_0+1,k) + \alpha'(k+1,j_0)) \leq \sum_{k=j_0+1}^{j_1-1} (\alpha(j_0+1,k) + \alpha(k+1,j_0)).$$

16) $j = j_0 + 1$, $k = j_0$. Then $U'_{j_0+1} = [a_{\tilde{u}_{j_0}}, a_{\tilde{v}_{j_0}})$ and $V'_{j_0} = [a_{\tilde{v}_{j_0-1}}, a_{\tilde{u}_{j_0+1}})$. An illustration of this case is given in Fig. A4.3-(d). Since $\tilde{v}_{j_0-1} \leq \tilde{u}_{j_0} < \tilde{u}_{j_0+1} < \tilde{v}_{j_0}$, it follows based on Lemma 19 that $\alpha'(j_0+1, j_0) = c(Sq(\tilde{u}_{j_0}, \tilde{u}_{j_0+1}))$. Further note that $Sq(\tilde{u}_{j_0}, \tilde{u}_{j_0+1}) \subseteq U_{j_0+1} \times V_{j_0}$, which implies that $c(Sq(\tilde{u}_{j_0}, \tilde{u}_{j_0+1})) \leq \alpha(j_0+1, j_0)$. The above two inequalities lead to $\alpha'(j_0+1, j_0) \leq \alpha(j_0+1, j_0)$.

17) $j = j_1+1$, $k = j_0$. Then $U'_{j_1+1} = [a_{\tilde{v}_{j_1-1}}, a_{\tilde{u}_{j_1+1}})$ and $V'_{j_0} = [a_{\tilde{v}_{j_0-1}}, a_{\tilde{u}_{j_0+1}})$. Since $sup(V'_{j_0}) = a_{\tilde{u}_{j_0+1}} < a_{\tilde{v}_{j_0}} \leq a_{\tilde{v}_{j_1-1}} = inf(U'_{j_1+1})$, it follows that $U'_{j_1+1} \times V'_{j_0} \subseteq \breve{\mathcal{H}}_-$, which implies that $\alpha'(j_1+1, j_0) = 0$.

18) $j = j_0 + 1$, $k = j_1$. Then $U'_{j_0+1} = [a_{\tilde{u}_{j_0}}, a_{\tilde{v}_{j_0}}) \subseteq V_{j_0}$ (since $\tilde{v}_{j_0-1} \leq \tilde{u}_{j_0}$) and $V'_{j_1} = [a_{\tilde{u}_{j_1}}, a_{\tilde{v}_{j_1}}) \subseteq U_{j_1+1}$ (since $\tilde{v}_{j_1} \leq \tilde{u}_{j_1+1}$). Thus, we have $U'_{j_0+1} \times V'_{j_1} \subseteq V_{j_0} \times U_{j_1+1} = \sigma(U_{j_1+1} \times V_{j_0})$, which implies that $\alpha'(j_0+1, j_1) \leq \alpha(j_1+1, j_0)$. Combining with the result of the previous case, we obtain $\alpha'(j_0+1, j_1) + \alpha'(j_1+1, j_0) \leq \alpha(j_0+1, j_1) + \alpha(j_1+1, j_0)$.

19) $j = j_1+1$, $k \leq j_0-1$ or $k \geq j_1+1$. Then $U'_{j_1+1} = [a_{\tilde{v}_{j_1-1}}, a_{\tilde{u}_{j_1+1}}) \subseteq U_{j_1+1}$ (since $\tilde{u}_{j_1} < \tilde{v}_{j_1-1}$) and $V'_k = V_k = [a_{\tilde{v}_{k-1}}, a_{\tilde{v}_k})$. An illustration of this case is given in Fig. A4.3-(c). It follows $U'_{j_1+1} \times V'_k \subseteq U_{j_1+1} \times V_k$, yielding $\alpha'(j_1+1, k) \leq \alpha(j_1+1, k)$.

20) $j = j_1 + 1$, $k = j_1$. $U'_{j_1+1} = [a_{\tilde{v}_{j_1-1}}, a_{\tilde{u}_{j_1+1}})$ and $V'_{j_1} = [a_{\tilde{u}_{j_1}}, a_{\tilde{v}_{j_1}})$. An illustration of this case is given in Fig. A4.3-(e).. Recall that $\tilde{u}_{j_1} < \tilde{v}_{j_1-1} < \tilde{v}_{j_1} \leq \tilde{u}_{j_1+1}$. By applying Lemma 19, we obtain that $\alpha'(j_1+1, j_1) = c(Sq(v_{j_1-1}, v_{j_1}))$. Additionally, since $V_{j_1} = [a_{\tilde{v}_{j_1-1}}, a_{\tilde{v}_{j_1}}) \subseteq U_{j_1+1}$, it follows that $Sq(\tilde{v}_{j_1-1}, \tilde{v}_{j_1}) \subseteq U_{j_1+1} \times V_{j_1}$, which further implies that $c(Sq(\tilde{v}_{j_1-1}, \tilde{v}_{j_1})) \leq \alpha(j_1+1, j_1)$ in virtue of Lemma 2. We conclude that $\alpha'(j_1+1, j_1) \leq \alpha(j_1+1, j_1)$.

By summarizing the results from all the cases, we obtain that

$$c(Q_1', Q_2') = \sum_{j=1}^{K} \sum_{k=1}^{K} \alpha'(j,k) \leq \sum_{j=1}^{K} \sum_{k=1}^{K} \alpha(j,k) = c(Q_1, Q_2),$$

and the proof is complete.

□

Bibliography

[1] A. Aggarval, M. Klave, S. Moran, P. Shor, and R. Wilber. Geometric applications of a matrix searching algorithm. *Algorithmica* (2) (1987), 195–208.

[2] S. Aldosari and J. Moura. Saddlepoint approximation for sensor network optimization. In: *Proceedings of International Conference on Acoustics, Speech, Signal Processing*. Vol. 4. Mar. 2005, 741–744.

[3] T. Berger. Multiterminal source coding. *The Information Theory Approach to Communications* CISM Courses and Lectures(229) (1978), 171–231.

[4] T. Berger and R. W. Yeung. Multiterminal source encoding with one distortion criterion. *IEEE Transactions on Information Theory* 35(2) (Mar. 1989), 228–236.

[5] R. E. Burkard, B. Klinz, and R. Rudolf. Perspectives of Monge properties in optimization. *Discrete Applied Mathematics* 70(2) (Sept. 1996), 95–161.

[6] J. F. Chamberland and V. V. Veeravalli. Wireless sensors in distributed detection applications. *IEEE Signal Processing Magazine* 24(3) (May 2007), 16–25.

[7] I. Diakonikolas, E. Grigorescu, J. Li, A. Natarajan, K. Onak, and L. Schmidt. Communication-efficient distributed learning of discrete distributions. *Advances in Neural Information Processing Systems* 30 (2017).

[8] V. Doshi, D. Shah, M. Médard, and M. Effros. Functional compression through graph coloring. *IEEE Transactions on Information Theory* 56(8) (July 2010), 3901–3917.

[9] S. Du, Y. Xu, H. Zhang, C. Li, P. Grover, and A. Singh. Novel quantization strategies for linear prediction with guarantees. In: *In Proceedings of ICML 2016 Workshop on On-Device Intelligence*. 2016.

[10] S. Dumitrescu. On the design of optimal noisy channel scalar quantizer with random index assignment. *IEEE Transactions on Information Theory* 62(2) (Feb. 2016), 724–735.

[11] S. Dumitrescu and X. Wu. Optimal multiresolution quantization for scalable multimedia coding. In: *Proceedings of IEEE Information Theory Workshop (ITW 2002)*. Bangalore, India, Oct. 2002, 139–142.

[12] S. Dumitrescu and X. Wu. Algorithms for optimal multi-resolution quantization. *Algorithms* 50(1) (Jan. 2004), 1–22.

[13] S. Dumitrescu and X. Wu. Fast algorithms for optimal two-description scalar quantizer design. *Algorithmica* 41(4) (Feb. 2005), 269–287.

[14] S. Dumitrescu and S. Zendehboodi. Globally Optimal Design of a Distributed Scalar Quantizer for Linear Classification. In: *2021 IEEE ISIT*. July 2021, 3167–3172.

[15] D. Elzouki, S. Dumitrescu, and J. Chen. Lattice-based robust distributed source coding. *IEEE Transactions on Information Theory* 65(3) (Mar. 2019), 1764–78.

[16] Y. H. Ezzeldin, C. Fragouli, and S. Diggavi. Quantizing signals for linear classification. In: *Proceedings of IEEE ISIT*. Paris, France, July 2019, 912–916.

[17] J. Fang and H. Li. Hyperplane-based vector quantization for distributed estimation in wireless sensor networks. *IEEE Transactions on Information Theory* 55(12) (Nov. 2009), 5682–5699.

[18] M. Fleming, Q. Zhao, and M. Effros. Network vector quantization. *IEEE Transactions on Information Theory* 50(8) (Aug. 2004), 1584–1604.

[19] T. J. Flynn and R. M. Gray. Encoding of correlated observations. *IEEE Transactions on Information Theory* IT-33(6) (Nov. 1987), 773–787.

[20] J. A. Gubner. Distributed estimation and quantization. *IEEE Transactions on Information Theory* 39 (July 1993), 1456–1459.

[21] A. Gyorgy, T. Linder, P. A. Chou, and B. J. Betts. Do optimal entropyconstrained quantizers have finite or infinite number of codewords? *IEEE Transactions on Information Theory* 49(11) (Nov. 2003), 3031–3037.

[22] T. S. Han. Hypothesis testing with multiterminal data compression. *IEEE Transactions on Information Theory* IT-33(6) (Nov. 1987), 759–772.

[23] T. S. Han and S. Amari. Statistical inference under multiterminal data compression. *IEEE Transactions on Information Theory* 44(6) (Oct. 1998), 2300–2324.

[24] Y. Han, A. Özgür, and T. Weissman. Geometric lower bounds for distributed parameter estimation under communication constraints. *IEEE Transactions on Information Theory* 67(12) (Aug. 2021), 8248–8263.

[25] O. A. Hanna, Y. H. Ezzeldin, T. Sadjadpour, C. Fragouli, and S. Diggavi. On distributed quantization for classification. *IEEE Journal of Selected Areas in Information Theory* 1(1) (May 2020), 237–249.

[26] O. A. Hanna, X. Li, C. Fragouli, and S. Diggavi. Can we break the dependency in distributed detection? In: *2022 IEEE ISIT*. IEEE, 2022, 2720–2725.

[27] X. He, K. Cai, W. Song, and Z. Mei. Dynamic Programming for Sequential Deterministic Quantization of Discrete Memoryless Channels. *IEEE Transactions on Communications* 69(6) (Mar. 2021), 3638–3651.

[28] P. Ishwar, R. Puri, K. Ramchandran, and S. S. Pradhan. On rate-constrained distributed estimation in unreliable sensor networks. *IEEE Journal of Selected Areas in Communication* 23(4) (Apr. 2005), 765–775.

[29] D. Krithivasan and S. S. Pradhan. Lattices for distributed source coding: Jointly Gaussian sources and reconstruction of a linear function. *IEEE Transactions on Information Theory* 55(12) (Dec. 2009), 5628–5651.

[30] W. M. Lam and A. R. Reibman. Design of quantizers for decentralized estimation systems. *IEEE Transactions on Communications* 41(11) (Nov. 1993), 1602–1605.

[31] T. Linder and S. Yuksel. On optimal zero-delay quantization of vector Markov sources. *IEEE Transactions on Information Theory* 60(10) (Oct. 2014), 6126–6131.

[32] M. Longo, T. D. Lookabaugh, and R. M. Gray. Quantization for decentralized hypothesis testing under communication constraints. *IEEE Transactions on Information Theory* 36(2) (Mar. 1990), 241–255.

[33] V. Megalooikonomou and Y. Yesha. Quantizer design for distributed estimation with communication constraints and unknown observation statistics. *IEEE Transactions on Communications* 48(2) (Feb. 2000), 181–184.

[34] V. Misra, V. K. Goyal, and L. R. Varshney. Distributed scalar quantization for computing: High-resolution analysis and extensions. *IEEE Transactions on Information Theory* 57(8) (Aug. 2011), 5298–5325.

[35] D. Muresan and M. Effros. Quantization as histogram segmentation: optimal scalar quantizer design in network systems. *IEEE Transactions on Information Theory* 54(1) (Jan. 2008), 344–366.

[36] X. Nguyen, M. J. Wainwright, and M. I. Jordan. Nonparametric Decentralized Detection Using Kernel Methods. *IEEE Transactions on Signal Processing* 53(11) (Nov. 2005), 4053–4066.

[37] Y. Oohama. Gaussian multiterminal source coding. *IEEE Transactions on Information Theory* 43(6) (Nov. 1997), 1912–1923.

[38] a. L. T. P. Venkitasubramaniam and A. Swami. Quantization for maximin ARE in distributed estimation. *IEEE Transactions on Signal Processing* 55(7) (June 2007), 3596–3605.

[39] Y. Pu, M. N. Zeilinger, and C. N. Jones. Quantization design for distributed optimization. *IEEE Transactions on Automation and Control* 62(5) (Aug. 2016), 2107–2120.

[40] D. Rebollo-Monedero, R. Zhang, and B. Girod. Design of optimal quantizers for distributed source coding. *IEEE DCC* (Mar. 2003).

[41] X. W. S. Dumitrescu. Lagrangian optimization of two-description scalar quantizers. *IEEE Transactions on Information Theory* 53(11) (Nov. 2007), 3990–4012.

[42] A. Saxena, J. Nayak, and K. Rose. On efficient quantizer design for robust distributed source coding. In: *Proceedings of IEEE Data Compression Conference*. Snowbird, UT, 2006, 63–72.

[43] A. Saxena and K. Rose. On scalable distributed coding of correlated sources. *IEEE Transactions on Signal Processing* 58(5) (May 2010), 2875–2883.

[44] D. K. Sharma. Design of absolutely optimal quantizers for a wide class of distortion measures. *IEEE Transactions on Information Theory* 24(6) (Nov. 1978), 693–702.

[45] F. Shirani and S. S. Pradhan. A new achievable rate-distortion region for distributed source coding. *IEEE Transactions on Information Theory* 67(7) (July 2021), 4485–4503.

[46] N. Shlezinger and Y. Eldar. Deep task-based quantization. *Entropy* 23(1) (Jan. 2021), 104.

[47] N. Shlezinger, Y. Eldar, and M. Rodrigues. Hardware-limited task-based quantization. *IEEE Transactions on Signal Processing* 67(20) (2019), 5223–5238.

[48] M. Shohat, G. Tsintsadze, N. Shlezinger, and Y. Eldar. quantization for MIMO channel estimation. In: *2019 IEEE International Conference on Acoustics, Speech and Signal Processing*. ICASSP 2019, May 2019, 3912–3916.

[49] J. Z. Sun and V. K. Goyal. Intersensor collaboration in distributed quantization networks. *IEEE Transactions on Communications* 61(9) (July 2013), 3931–3942.

[50] J. Z. Sun, V. Misra, and V. K. Goyal. Distributed functional scalar quantization simplified. *IEEE Transactions on Signal Processing* 61(14) (July 2013), 3495–3508.

[51] R. R. Tenney and N. R. Sandell. Detection with distributed sensors. *IEEE Transactions on Aerospace and Electronic systems* 17(4) (July 1981), 501–510.

[52] J. Tsitsiklis. Decentralized detection. *Advanced Statistical Signal Processing* 2 (1993), 297–344.

[53] S. Y. Tung. Multiterminal Source Coding. PhD thesis. Ithaca, NY: School of Electrical Engineering, Cornell University, May 1978.

[54] V. A. Vaishampayan. Classification in a large network. In: *Proceedings of IEEE ISIT*. Paris, France, July 2019, 1807–1811.

[55] R. Viswanathan and P. Varshney. Distributed detection with multiple sensors: Part I—fundamentals. In: *Proceedings of IEEE*. Vol. 85. 1. Jan. 1997, 54–63.

[56] A. B. Wagner. On distributed compression of linear functions. *IEEE Transactions on Information Theory* 57(1) (Dec. 2010), 79–94.

[57] A. B. Wagner, S. Tavildar, and P. Viswanath. Rate region of the quadratic Gaussian two-encoder source-coding problem. *IEEE Transactions on Information Theory* 54(5) (May 2008), 1938–1961.

[58] J. Wang and J. Chen. Vector Gaussian multiterminal source coding. *IEEE Transactions on Information Theory* 60(9) (Sept. 2014), 5533–5552.

[59] J. Wang, J. Chen, and X. Wu. On the sum rate of Gaussian multiterminal source coding: New proofs and results. *IEEE Transactions on Information Theory* 56(8) (Aug. 2010), 3946–3960.

[60] N. Wernersson, J. Karlsson, and M. Skoglund. Distributed quantization over noisy channels. *IEEE Transactions on Communications* 57(6) (June 2009), 1693–1700.

[61] H. Wu and S. Dumitrescu. Design of Optimal Scalar Quantizer for Sequential Coding of Correlated Sources. *IEEE Transactions on Communications* 67(1) (Sept. 2018), 693–707.

[62] H. Wu and S. Dumitrescu. Design of Successively Refinable Unrestricted Polar Quantizer. *IEEE Transactions on Communications* 67(5) (May 2019), 3525–3539.

[63] H. Wu and S. Dumitrescu. Design of general entropy-constrained successively refinable unrestricted polar quantizer. *IEEE Transactions on Communications* 68(6) (June 2020), 3369–3385.

[64] X. Wu. Optimal quantization by matrix searching. *Algorithms* 12(4) (Dec. 1991), 663–673.

[65] X. Wu, A. Bais, and N. Sarshar. Quantization for robust distributed coding. *International Journal of Distributed Sensor Networks* 12(5) (May 2016), 1–6.

[66] X. Wu and K. Zhang. Quantizer monotonicities and globally optimal scalar quantizer design. *IEEE Transactions on Information Theory* 39(3) (May 1993), 1049–1053.

[67] R. W. Yeung and Z. Zhang. Distributed source coding for satellite communications. *IEEE Transactions on Information Theory* 45(4) (May 1999), 1111–1120.

[68] S. Zendehboodi and S. Dumitrescu. Optimal Distributed Quantizer Design for Binary Classification of Conditionally Independent Vector Sources. In: *2024 IEEE ISIT*. 2024.

[69] Q. Zheng and S. Dumitrescu. Optimal Design of A Two-stage Wyner-Ziv Scalar Quantizer with Forwardly/Reversely Degraded Side Information. *IEEE Transactions on Communications* 67(2) (Feb. 2019), 1437–1451.