Title

Optimal Codebook Generation and Adaptation in Compression, Communications and Machine Learning

Author
Elshafiy, Ahmed

Publication Date
2022

University of California
Santa Barbara

Optimal Codebook Generation and Adaptation in Compression, Communications and Machine Learning

A dissertation submitted in partial satisfaction
of the requirements for the degree

Doctor of Philosophy
in
Electrical and Computer Engineering

by

Ahmed Elshafiy

Committee in charge:

Professor Kenneth Rose, Chair
Professor Ramtin Pedarsani
Professor Joao Hespanha
Professor Upamanyu Madhow
Professor Rami Zamir, Tel Aviv University

December 2022

The Dissertation of Ahmed Elshafiy is approved.

Professor Ramtin Pedarsani

Professor Joao Hespanha

Professor Upamanyu Madhow

Professor Rami Zamir, Tel Aviv University

Professor Kenneth Rose, Committee Chair

December 2022

Optimal Codebook Generation and Adaptation in Compression, Communications and Machine Learning

by

Ahmed Elshafiy

I dedicate this work to my family: *i*) my brilliant and caring **father**, who has always aspired me to reach my full potential, may his soul rest in peace, *ii*) my compassionate **mother**, who has always taught me that any good work or deed will be rewarded, *iii*) my backbone **sister**, who has always supported me with her amazing personality, *iv*), my **wife** and my soulmate, who has always inspired me and believed in me even when the odds were stacked against me, and finally, *v*) my true source of joy, my beloved **son**.

Acknowledgements

I sincerely acknowledge all of my signal compression lab colleagues who have supported me, in any way, towards fulfilling my PhD degree requirements and completing my research work.

I would like to also acknowledge the ECE department at UCSB as they never hesitated to offer me financial support through my graduate program journey, in the form of paid tuition, research fellowships, TA ships, etc.

Furthermore, part of the work done here has been supported financially by the National Science Foundation (NSF) under grant CCF-1909423.

Curriculum Vitæ
Ahmed Elshafiy

Education

2022 Ph.D. in Electrical Engineering (Expected), University of California, Santa Barbara.

2016 M.Sc. in Electronics and Electrical Communications Engineering, Cairo University, Egypt.

2014 B.Sc. in Communication and Computer Engineering, Cairo University, Egypt.

Publications

1. **A. ElShafiy**, K. Rose, **On Stochastic Codebook Generation for Markov Sources**, submitted to Data Compression Conference (DCC) 2023.
2. **A. ElShafiy**, M. Namazi, R. Zamir, K. Rose, **Towards A Practically Tractable and Asymptotically Optimal Stochastic Codebook Regeneration for Lossy Coding**, submitted to IEEE transactions on information theory, 2022.
3. **A. ElShafiy**, M. Namazi, R. Zamir, K. Rose, **Stochastic Codebook Regeneration for Sequential Compression of Continuous Alphabet Sources**, in IEEE ISIT 2021.
4. **A. ElShafiy**, M. Namazi, R. Zamir, K. Rose, "**On-The-Fly Stochastic Codebook Re-generation for Sources with Memory**," in IEEE International Theory Workshop, 2021.
5. **A. ElShafiy**, A. Sampath, K. Rose, "**A Clustering Approach to Optimizing Beam Steering Directions in Wireless Systems**," in IEEE Wireless Commun. and Net. Conf., 2021.
6. **A. ElShafiy**, M. Namazi, K. Rose, "**On Effective Stochastic Mechanisms for On-The-Fly Codebook Regeneration**," in IEEE International Symposium on Information Theory, 2020.
7. **A. ElShafiy**, K. Rose, A. Sampath, "**On Optimal Beam Steering Directions in Millimeter Wave Systems**," in IEEE Int. Conf. on Acoustics, Speech and Signal Processing, 2019.
8. **A. ElShafiy**, A. Sampath, "**Beam Broadening for 5G Millimeter Wave Systems**," in IEEE Wireless Commun. and Networking Conf., April 2019.
9. **A. ElShafiy**, A. Sampath, "**System Performance of Indoor Office Millimeter Wave Communications**," in IEEE Wireless Commun. and Networking Conf., April 2019.
10. **A. ElShafiy** *et al.*, "**On Optimization of Mixed-Radix FFT: A Signal Processing Approach**," in IEEE Wireless Commun. and Networking Conf., April 2019.
11. **A. ElShafiy**, T. Nanjundaswamy, S. Zamani, K. Rose, "**On Error Resilient Design of Predictive Scalable Coding Systems**," in IEEE Int. Conf. on Acoustics, Speech and Signal Processing, 2018.
12. **A. ElShafiy**, M. Farag, M. ElMotaz, O. Nasr and H. Fahmy, "**Two-Stage Optimization of CORDIC-Friendly FFT**," in IEEE Int. Conf. on Electronics, Circuits, and Systems, Dec 2015.

Abstract

Optimal Codebook Generation and Adaptation in Compression, Communications and Machine Learning

by

Ahmed Elshafiy

Codebook design, generation, and adaptation, based on matching to stochastic source examples or prior knowledge of source distribution, has played a central role in many applications of source coding. The original iterative "natural type selection" (NTS) algorithm performs stochastic codebook generation of memoryless sources, and achieves the rate-distortion bound, as it asymptotically converges to the optimal codebook reproduction distribution, Q^*. However, these optimality results are subject to significant limitations that compromise the practical applicability of NTS, namely: i) the string length L is required to go to infinity at the outset, before NTS iterations begin, whereas the iteration complexity is exponential in L, and ii) it is only applicable to discrete and memoryless sources, thus precluding a vast portion of important lossy coding applications. This thesis offers means to eliminate or circumvent these critical shortcomings by proposing new and enhanced NTS algorithms, complemented by optimality proofs that are not subject to the above limitations. To circumvent the need to start with asymptotically large string length, L, the approach leverages a maximum likelihood framework to estimate, at each NTS iteration n, the reproduction distribution most likely to generate the sequence of K length-L codewords that respectively and independently "d-match" (i.e., are within distortion d from) a sequence of K length-L source words. The reproduction distribution estimated at iteration n is used to regenerate the codebook for iteration $n + 1$. The sequence of reproduction distributions estimated by NTS is shown

to converge, asymptotically in K, n, and L (in this order), to the optimal distribution that achieves the rate-distortion bound. Thus, the string length L is the last parameter to be sent to infinity. Moreover, it is established that, for finite length L, the new NTS algorithm converges to the best achievable distribution, i.e., as constrained by the string length L, and details are provided for various types of sources, where numerical simulations show that the algorithm rate of convergence in n for finite length L is at least as fast as convergence in n with infinite L. To handle sources with memory, NTS is further generalized by considering source sub-vectors or "super-symbols", of memory depth M, during d-match search in the codebook, maximum likelihood estimation of reproduction distribution, and codebook regeneration. Asymptotic convergence, in L and M, to the optimal reproduction distribution is also established for sources with memory. As for, perhaps the more challenging, sources over continuous alphabet spaces, which are inconsistent with the traditional concept of "type" or "typical sequence", in the proposed asymptotically optimal approach, we employ empirical probability measures for codebook reproduction distribution estimation.

Methodologies for optimal codebook generation and adaptation are further developed and employed in two promising example applications in the areas of i) wireless communications and ii) machine learning. In particular, for 5G cellular systems and next generation wireless local area networks, directional beamforming with large antenna arrays is key to mitigating the substantial signal loss experienced at the millimeter wave frequency band, where it entails a significant increase in the number of beams required to maintain cell coverage, and hence an increase in the beam management overhead necessary to maintain link with mobile users. This observation, in turn, suggests that the underlying problem of finding the optimal set of beam steering directions will benefit from fundamental signal processing and codebook design methodologies, and specifically from basic principles and algorithms for cluster analysis. This part of the thesis establishes

and exploits the equivalence between the problem of optimizing a set of beam steering directions and the classical problems of clustering in pattern recognition and codebook design in data compression, albeit with an unusual distortion measure. Subsequently, a global optimization approach within the deterministic annealing framework is derived, to circumvent poor local optima that may riddle the cost surface under the classical gradient descent clustering techniques. System simulation results show that the proposed approaches deliver considerable gains, over the baseline beam steering techniques, in terms of average signal-to-noise ratio.

The third part of the thesis is concerned with codebook design and adaptation for machine learning or artificial intelligence. Machine learning applications have exploded in recent years due to the availability of huge data sets, as well as advances in computational and storage capabilities. Although successful methods have been proposed to reduce learning system complexity while maintaining required accuracy levels, theoretical understanding of the underlying trade-offs remains elusive. In this work, the classical supervised learning problem is reformulated within a rate-distortion framework. It provides insights into the underlying accuracy-complexity trade-offs, by considering the overall learning system as consisting of two components. The first is tasked with extracting (learning) from the source the minimal number of information bits necessary to ultimately achieve the prescribed output accuracy. The learned bits are then used to retrieve the desired output from the second component, an appropriately designed *codebook*. The premise here is that an optimal system is characterized by having to learn the minimum amount of information from the source, just sufficient to yield the system output at the desired precision, which implies efficiency in terms of system complexity, generalization and training data requirements. The design and training of such a reformulated system is detailed, and asymptotically optimal performance that achieves the rate-distortion bound is established.

Contents

Chapter 1

Introduction

Codebook design, or equivalently quantizer design, has played a central role with different flavors in numerous applications in the areas of source coding, communications, machine learning, etc. In the communications or information-theory literature, an early clustering method was suggested for scalar quantization, variants of which are known as the Lloyd algorithm [1] or the Max quantizer [2]. This method was later generalized to Vector Quantization (VQ), and to a large family of distortion measures [3], and the resulting algorithm is commonly referred to as the Generalized Lloyd Algorithm (GLA). In the pattern-recognition literature, similar algorithms have been introduced, including the ISODATA [4] and the K-means [5] algorithms. All the above iterative methods alternate between two complementary steps (often referred to as the Lloyd iteration): optimization of the partition into clusters given the current codebook entries, and optimization of the codebook entries for their respective clusters. It is easy to show that such an iterative procedure is monotone non-increasing in the distortion, and convergence to a local minimum of the distortion is guaranteed. The Deterministic Annealing (DA) approach, for conventional distortion measures, has been proposed as a powerful algorithm for avoiding poor local minima that may riddle the cost function [6]. The optimal solution can be tracked in a deterministic annealing framework, starting at the global optimum for high distortion (where the cluster means all coincide at a single point, i.e., we have a single effective mean for the entire training set) and tracking the minimum as the temperature (the Lagrangian parameter controlling the tradeoff between distortion and entropy) is lowered. During this annealing process, the system undergoes a sequence of "phase transitions" whereby the cardinality of the set of effective means increases. Inspired by principles of statistical physics and derived in terms of information theory, DA was proposed as a powerful non-convex optimization tool for compression, clustering, classification and related problems.

Crucial to all of the above iterative codebook design algorithms is the prior knowledge of source statistics, however in many applications, the source statistics are not known or are varying with time. Stochastic codebook generation and adaptation, based on codeword statistics and source string matching, without prior knowledge of source distributions, offered major contributions to lossless and lossy source coding applications. Particularly influential were the seminal contributions of Lempel and Ziv, as evidenced by the numerous prevalent variants of the LZ77 and LZ78 algorithms [7, 8] which showed stochastic codebook generation/adaptation, where noticeably the generated codebooks do not require exponential d-match search complexity, to be a powerful tool for lossless coding. As an example of how stochastic codebook generation is employed, consider LZ78, where on-the-fly compression is performed by creating a dictionary or tree of codewords, as source strings are encoded. This tree of codewords is created, without recourse to prior knowledge of source statistics, in a manner that ensures that the relative frequency of typical source sequences in the tree, asymptotically approaches one [8]. Stochastic codebook generation mechanisms have been proposed for lossy coding as well, e.g., the gold-washing [9] and natural type selection [10, 11] algorithms. It is important to emphasize that optimizing the codebook reproduction distribution is fundamentally more difficult in the lossy coding setting. The lossless coding problem is “simpler” not only because perfect matching is less complex than matching with distortion, but also because the optimal codebook generating distribution, which achieves the minimum coding rate, is exactly the source distribution. Hence, in lossless coding, the ultimate goal of stochastic codebook generating algorithms is simply to *learn* the source distribution from source examples. However, in lossy coding, the problem is vastly more difficult as the source distribution P and optimal codebook generating distribution Q^* are generally different, even if the alphabets are the same, and more so in the high distortion constraint regime [12, 13, 14, 10]. For example, in the case of continuous alphabet sources with the squared error distortion measure, at very small distortion (high resolution) $Q^* \approx P$, but as the distortion constraint is relaxed, i.e., d increases, Q^* diverges from P, it shrinks, often becomes discrete, and eventually collapses to a single point when $d = d_{\max}$ [15]. Hence, it is not enough to simply “mimic” the source, and finding the optimal codebook reproduction distribution represents a significant challenge. The key insight, as articulated in [10] for discrete alphabet sources, is that “type selection”, rather than learning and matching source statistics, is the appropriate approach to codebook adaptation in the lossy coding setting. Ultimately, the optimal codebook *reproduction type* for the source is estimated by the codebook adaptation algorithm at a given distortion constraint d.

This idea of “type selection” for codebook generation or adaptation, which is the most relevant to the work done in this thesis, was first introduced in the stochastic codebook generation and adaptation algorithm, known as “Natural Type Selection” (NTS) [10, 11]. In this algorithm, at each time step or iteration n, a source word of length L is encoded. Given the source word, the type of the first codeword (in a randomly generated codebook drawn from a generating distribution Q_n) to satisfy the specified distortion constraint d, is used as the distribution from which to regenerate the codebook in the next iteration, $n+1$. In other words, the codebook reproduction types are naturally selected in response to source examples, and evolve through a sequence of “d-match” operations, hence the name natural type selection, with a nod to Darwin’s theory of evolution. Consequently, it was shown that asymptotically in, first, the string length L, and then the number of iterations n, the sequence of codebook generating types $Q_1, Q_2, \ldots$ converges to the optimal reproduction distribution $Q^*_{P,d}$ that achieves the rate-distortion bound $R(P, d)$. While the early NTS algorithm was shown to achieve asymptotic optimality, the results nevertheless are constrained by limitations that significantly compromise the algorithm’s general applicability. In order to converge to the optimal codebook generating distribution, first the string length L has to be sent to infinity, and only then can the codebook be iteratively regenerated. Unfortunately, the codebook size must grow exponentially in L, in order to ensure that a d-match to a source example is found with probability one. Such intractable d-search and match complexity, which is the central step in the NTS iteration, severely limits the algorithm usefulness in practical implementations. Clearly, the reversed order of limits, where L is the last parameter to go to infinity, is much more desirable in practice. It is preferable to regenerate the codebook at manageable d-search complexity, and only then gradually increase the string length L. Moreover, the original NTS algorithm was only shown to be asymptotically optimal for memoryless discrete alphabet sources, while in practice, many sources of interest are sources with memory, and/or continuous alphabet sources. These fundamental shortcomings, coupled with the observation that stochastic codebook generation has had a phenomenal impact on practical lossless source coding, provide strong motivation to take a principled look at the original NTS algorithm, and develop a tractable yet asymptotically optimal alternative. The central part of this thesis proposes a tractable codebook generation and adaptation algorithm, with the desired reversed order of limits, for which we establish the asymptotic optimality for discrete alphabet memoryless sources. Next, we extend the stochastic codebook adaptation and generation mechanism to discrete alphabet sources with memory, where we consider vector sources or sources with finite memory depth, e.g., sources with Markovian property. We also establish the asymptotic optimality of the algorithm variant in these settings. Consequently, we generalize the framework to accommodate sources over continuous alphabet spaces, which in turn, include the vast majority of sources seen in practice.

In order to assess the effectiveness of the proposed codebook design and adaptation techniques, methodologies for optimal codebook generation and adaptation are further developed and employed in two promising example applications, that can greatly benefit of such algorithms, in the areas of *i*) wireless communications and *ii*) machine learning. In particular, for 5G cellular systems and next generation wireless local area networks, Multiple-Input Multiple-Output (MIMO) systems in conjunction with millimeter-wave frequencies have been recognized as a promising tool in the effort to satisfy the ever-growing demand for higher data-rates. Given that physical layer technologies already operate at, or close to, Shannon capacity, the main focus must be on the system bandwidth [16, 17]. Studies have shown that considerable rate gains can be achieved through millimeter-wave communications by exploiting the substantial bandwidth available at these frequencies. However, a number of significant challenges arise as well [18, 19], including increased path-loss, shadowing losses, signal attenuation, and atmospheric absorption at some frequencies, which cause considerable decrease in link budget and result in considerable reduction in cell coverage area. To meet this challenge, larger transmit/receive arrays, and hence increased array factors, are employed to boost the link budget. Considering a transmit linear-array of length N_{tx}, the increase in Effective Isotropic Radiated Power (EIRP) due to beamforming is proportional to N_{tx} [20], yielding a corresponding increase in the receiver signal-to-noise ratio (SNR). However, the Half-Power Beam-Width (HPBW) is inversely proportional to N_{tx}. Thus, large arrays offer EIRP gains in the steering direction, but at the cost of narrower beams, which in-turn result in an increase in the number of beams required to maintain acceptable spatial coverage. Both transmitter and receiver typically operate with predefined "*codebooks*" of beamforming vectors, wherein each codebook entry corresponds to a beam steering direction. An increase in codebook size hinders beam tracking and beam alignment due to the inherent increase in beam measurement time (sweep time) and thus compromises the system responsiveness to user and environment dynamics.

Beam broadening was proposed as a countermeasure to allow for a tradeoff between the requirements of high EIRP and low beam management complexity, especially in conjunction with user tracking and initial access [21, 20, 22, 23]. Additionally, enhanced robustness to user dynamics can be achieved by employing a more efficient beam search or beam alignment algorithms for a given codebook of beam steering directions, as has been pursued in [24, 25]. However, regardless of the beam width or the beam alignment algorithm, the overall performance of the system can be improved by optimal design of the beam steering directions, to match the observed or estimated user statistics. It is intuitively obvious that an optimal design of beam steering directions will jointly consider the distribution of users as well as the direction-dependent beam width. The objective of this work is to develop a sound methodology, from basic signal processing principles, for finding the optimal set of beam steering directions, i.e., designing the beam steering codebook, given a codebook size budget. The problem of finding the optimal beam steering angles can, in fact, be viewed as a clustering problem, where the two-dimensional angular space (azimuth and elevation angles) is partitioned into N_b sub-cells each represented by a pointing angle [26]. As the number of pointing angles is increased, the average link performance over the angular space increases, but so does the rate of beam updates, and the system becomes less robust to dynamics. This tradeoff is analogous to the classical rate-distortion tradeoff considered in quantizer design for data compression. Consequently, this work derives an approach within a powerful optimization framework, namely, deterministic annealing to circumvent poor local optima (that might result from the k-means or similar algorithm in [26]), solve the clustering problem at hand, and achieve significant performance gains.

The third part of the thesis is concerned with codebook design and adaptation for machine learning or artificial intelligence. A considerable surge in machine learning applications has been observed in recent years, due to the explosion of available data sets, computation and storage capabilities, as well as the advances in machine learning techniques. Machine learning algorithms are currently employed in a broad spectrum of day-to-day applications, including speech recognition and synthesis, computer vision, virtual personal assistants, medical image analysis, autonomous cars, social media and marketing services, etc. Machine learning encompasses a variety of adaptive methods to optimize the parameters of a system of processing units based on past observations, i.e., examples or training sequences, such that it approximates the desired behavior. The underlying assumption is that both training and testing sets are drawn from the same stochastic model. In supervised learning, a teacher or annotator is available to provide the desired outputs (or labels) for the training examples. Hence, system parameters can be fine-tuned to minimize the error between desired and estimated outputs. In contrast, unsupervised learning is needed when the desired outputs are unknown, and the algorithm seeks to cluster the input instances into useful classes by discerning the statistical structure underlying the training set.

Deep learning and Deep Neural Networks (DNN)s refer to a class of machine learning methods where the system architecture is composed of a network of perceptrons (neurons) arranged in multiple information-processing layers (also called hidden layers). In early the 1960s, it was shown in [27, 28] that a single perceptron can learn to classify any linearly separable set of inputs with an ensured convergence to the optimal separating hyperplane. Later, multi-layer perceptrons were proposed as a generalization for non-linear partition of the input space. Many methods have been proposed and implemented thus far for reducing the size and computational complexity of DNNs, while attempting to maintain accuracy or performance constraints. These methods include weight sharing [29], quantization [30], network pruning [31][32][33], and low-rank approximations [34]. While successful, these methods involve substantial changes to the DNN in question and give little control over the inherent accuracy-complexity trade-offs.

This part of the work proposes a reformulation of the classical supervised learning problem within a (practical) rate-distortion or information theoretic framework, thereby providing insights into the crucial accuracy-complexity trade-offs. This is accomplished by dividing the problem into two parts. The first part of the proposed framework extracts (i.e., learns) from the source the minimal necessary number of information bits, by employing a learning system (e.g., a deep network). The learned bits are used to retrieve the desired output from the second part, i.e., a designed codebook, which satisfies a distortion or accuracy requirement. The premise here is that an optimal system is the one characterized by extracting the minimum amount of information from the source, required to yield the system output at the desired precision, which implies efficiency of the learning task in terms of system complexity, generalization and training data requirements. The asymptotic optimality results of the proposed NTS algorithm for large spectrum of sources supports the promise of the recursive NTS framework as an optimal means to generate or update codebooks, from source examples, for the proposed rate-distortion based learning system.

The remainder of the thesis is organized as follows: The relevant background on stochastic and deterministic codebook design methodologies is introduced in Chapter 2. The tractable and asymptotically optimal stochastic codebook design approach, i.e., the NTS algorithm, is proposed in Chapter 3. The asymptotic optimality is established for the vast majority of sources, and mathematical proofs are provided in Appendix A.

Derivation and employment of deterministic codebook design methods in wireless communications systems is shown in Chapter 4, where considerable gains in the average signal-to-noise ratios are observed in numerical simulations. Subsequently, reformulation of the classical supervised learning system into a rate-distortion framework, in conjunction with the derivation of a variant of the NTS algorithm for such settings, is proposed in Chapter 5. Finally, conclusion are drawn in Chapter 6.

Chapter 2

Relevant Background

2.1 Conceptual Signal Compression System Model

Throughout this chapter denote $\mathcal{X}$ and $\mathcal{Y}$ as the source and reproduction alphabet spaces. We assume that the alphabet $\mathcal{X}$ is either a discrete space or (more generally) a complete separable metric space (often called Polish space), equipped with its associated Borel σ-field $\mathcal{X}'$. Similarly, we assume that the reproduction alphabet $\mathcal{Y}$ is either a discrete space or (more generally) also a Polish space equipped with its associated Borel σ-field $\mathcal{Y}'$. Furthermore, let $\{X_u\}_{u=1}^{\infty}$ be a stationary ergodic source, where the source realization is denoted as $x_u \in \mathcal{X}$, and similarly, the reproduction realization is denoted as y_u. Next, let $\{\tilde{\mathbf{X}}_i\}_{i=1}^{\infty}$ be a sequence of independent and identically distributed (i.i.d.) M-tuples (or source "super-symbols"), each obtained by drawing M successive symbols from the distribution underlying source $\{X_u\}$. Hence, let P_M be the vector source distribution of $\tilde{\mathbf{x}}$ on $\mathcal{X}^M$. Define a source block (source word) that contain L source vectors as $\mathbf{X} = \left(\tilde{\mathbf{X}}_1, \tilde{\mathbf{X}}_2, \ldots, \tilde{\mathbf{X}}_L\right)$, and source block realization as $\mathbf{x}$. Next, we define an arbitrary non-negative (measurable) scalar-valued distortion function $\rho : \mathcal{X} \times \mathcal{Y} \to [0, \infty)$. The distortion between a realization of the source block $\mathbf{x}$ and a realization of the code block

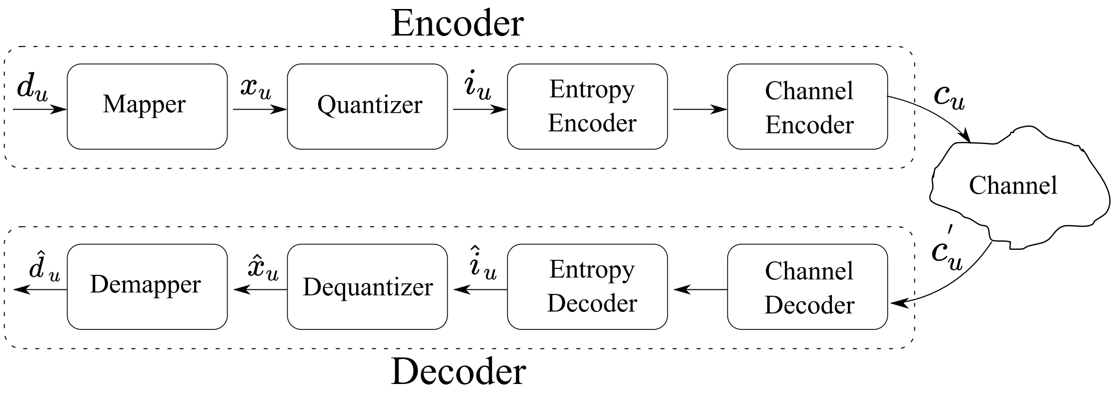


Figure 2.1: The conceptual signal compression system model.

(codeword or reproduction word) $\mathbf{y}=(\tilde{\mathbf{y}}_1\ldots,\tilde{\mathbf{y}}_L)$, with $\tilde{\mathbf{y}}_i\in\mathcal{Y}^M$, is assumed additive, and is specifically, the average distortion over super-symbols in the block:

$$\rho\left(\mathbf{x},\mathbf{y}\right)=\frac{1}{L}\sum_{\ell=1}^{L}\rho\left(\tilde{\mathbf{x}}_\ell,\tilde{\mathbf{y}}_\ell\right)=\frac{1}{L}\sum_{\ell=1}^{L}\left(\frac{1}{M}\sum_{m=1}^{M}\rho(x_{\ell,m},y_{\ell,m})\right), \tag{2.1}$$

where $x_{\ell,m}$ and $y_{\ell,m}$ are the m-th letters in $\tilde{\mathbf{x}}_\ell$ and $\tilde{\mathbf{y}}_\ell$, respectively.

The conceptual compression system model is shown in Fig. 2.1. Notice that the operations that are performed by the encoder are reversed by the decoder in order to recover the original data $\{d_u\}_{u=1}^{\infty}$. The first conceptual encoder block is the “Mapper”. The main operation of the mapper is to remove any redundancy in the source message that can be present in the form of correlation. For example, if the data is temporally correlated, a mapper can perform prediction from the previous reconstructed data that is available at the decoder, i.e., $\{\hat{d}_u\}$, and only the prediction errors are required to be encoded and transmitted to the receiver. The data can also be made uncorrelated by simply performing a transform operation such that the output of the transform is uncorrelated or nearly uncorrelated. The majority of practical compression systems include both predictors and transforms to decorrelate the source data and hence remove any redundancies. For example, in video compression standards, discrete cosine transforms are used to spatially decorrelate the image pixels, and then linear prediction is performed to exploit the temporal correlation [35, 36]. The function of the next block in Fig. 2.1, i.e. the quantizer or the codebook, is to remove "irrelevant" data. In other words, the quantizer main goal is to minimize the amount of data that is required to be transmitted in a way that would optimally introduce little to no distortion. There are various ways to design a codebook for a given system, depending on whether the system has prior knowledge of source statistics, whether the source statistics are fixed or time-varying, whether the data to be quantized is grouped into blocks of samples, and many other factors. Consequently, the third block in the encoder conceptual diagram is the entropy encoder. The function of the entropy encoder, like the mapper, is to remove any redundancy in the source message. By Shannon source coding theorem, the minimum bit-rate that can be used to encode a given source while ensuring an asymptotically vanishing probability of error is the source entropy, where the source entropy is simply defined as the average information rate of the source. The formal and mathematical definition of source entropy and other definitions of information-theoretic quantities are given in subsequent sections. Hence, the entropy encoder simply assigns different codes (possibly with different lengths) to the quantizer levels, with the aim to minimize the average encoding rate or code lengths. Finally, the last block in the system model is the channel encoder, which is present to equip the transmit message with additional redundancies in order to combat the distortions introduced by the channel. The simplest form of channel coding is repetition coding, where the code bits are repeated to allow for a chance to recover the original message at the decoder if few bits were corrupted by the channel. The codebook or the quantizer block is the main focus of this thesis. In the next sections, we investigate the most relevant methods of codebook design.

2.2 Stochastic Codebook Design: Random Coding

In this section, we introduce the mechanisms of stochastic codebook design or random coding. First, for a scalar-valued fidelity constraint d, define a "d-match" event as the event that $\rho\left(\mathbf{x},\mathbf{y}\right) \leq d$. Suppose a random codebook $\mathcal{C}_L$ of infinite number of length-ML codewords ($\mathbf{Y}(j)$, with $j \geq 1$) is generated such that, each codeword consists of L i.i.d. vectors as $Q_M = \{Q_M(\tilde{\mathbf{y}}) : \tilde{\mathbf{y}} \in \mathcal{Y}^M)\}$. We call Q_M the codebook reproduction distribution. Let $N_{M,L}$ be the index of the first codeword in $\mathcal{C}_L$ that d-matches the source word realization $\mathbf{x}$, i.e.,

$$N_{M,L} = \inf\left\{j \geq 1 : \rho\left(\mathbf{x},\mathbf{y}(j)\right) \leq d\right\}, \tag{2.2}$$

with the convention that the infimum of an empty set is $+\infty$. Given a codebook reproduction distribution Q_M over $\mathcal{Y}^M$, we define,

$$D_{\min} \triangleq \mathbb{E}_{P_M}\left[\operatorname*{ess\,inf}_{\tilde{\mathbf{y}}\sim Q_M} \rho(\tilde{\mathbf{X}},\tilde{\mathbf{Y}})\right], \tag{2.3}$$

$$D_{\text{av}} \triangleq \mathbb{E}_{P_M\times Q_M}\left[\rho(\tilde{\mathbf{X}},\tilde{\mathbf{Y}})\right], \tag{2.4}$$

where $\operatorname{ess\,inf}_{\tilde{\mathbf{y}}\sim Q_M}(\cdot)$ denotes the essential infimum of a function, i.e.,

$$\operatorname*{ess\,inf}_{\mathbf{Y}\sim Q_M} \rho(\tilde{\mathbf{x}},\tilde{\mathbf{Y}}) = \sup\{t\in\mathbb{R} : Q_M(\rho(\tilde{\mathbf{x}},\tilde{\mathbf{Y}}) > t) = 1\}, \quad \text{for any } \tilde{\mathbf{x}} \in \mathcal{X}^M. \tag{2.5}$$

We will assume throughout this chapter that D_{av} is finite, and $D_{\min} < D_{\text{av}} < \infty$. We will also restrict our attention to the non-trivial range of distortion levels $d \in (D_{\min}, D_{\text{av}})$. Shannon's theorem of lossless coding states: if we generate a random codebook of length $\exp(L(H(P_M)+\epsilon))$ using the source distribution P_M, then the probability of finding yet another independently generated source word in the codebook goes to one asymptotically in L, wherein $H(P_M)$ is the source entropy, defined as follows for discrete alphabet spaces,

$$H(P_M) = -\sum_{\tilde{\mathbf{x}} \in \mathcal{X}^M} P_M(\tilde{\mathbf{x}}) \log(P_M(\tilde{\mathbf{x}})). \tag{2.6}$$

Unless otherwise mentioned, all familiar information-theoretic quantities (entropy, mutual information, and so on) are defined in terms of logarithms taken to base e, and are therefore expressed in nats. On the other hand, Shannon's lossy coding theorem for scalar-valued distortion measures states: if a random codebook of length $\exp(L(R(P_M, d) + \epsilon))$ is generated using an optimal reproduction distribution $Q^*_{P_M,d}$, the probability of finding a codeword that d-matches an independently generated source word, drawn from the source distribution P_M, goes to one as L goes to infinity, wherein $R(P_M, d)$ is the *joint* (or M-th order) rate-distortion function, i.e., [37, 38, 39]

$$R(P_M, d) = \inf_{\substack{V:[V]_x = P_M, \\ \mathbb{E}_V(\rho(\tilde{\mathbf{X}}, \tilde{\mathbf{Y}})) \leq d}} I(\tilde{\mathbf{X}}, \tilde{\mathbf{Y}}). \tag{2.7}$$

Here, $I(\tilde{\mathbf{X}}, \tilde{\mathbf{Y}})$ is the mutual information between the m-tuples random vectors $\tilde{\mathbf{X}}$ and $\tilde{\mathbf{Y}}$, and the infimum is taken over all joint probability measure V such that the x-marginal of V, denoted $[V]_x$, is P_M and the expected distortion $\mathbb{E}_V(\rho(\tilde{\mathbf{X}}, \tilde{\mathbf{Y}})) \leq d$. Let $V^*_{P_M,d}$ be the optimal joint distribution that realizes the infimum in (2.7), then the optimal codebook reproduction distribution $Q^*_{P_M,d}$ is the y-marginal of the optimal joint distribution $V^*_{P_M,d}$. However, if a random codebook is generated from distribution $Q_M \neq Q^*_{P_M,d}$, then the minimum encoding rate to guarantee a d-match in probability, as L goes to infinity, was effectively shown in [40], and extended to memoryless sources over abstract alphabets in

[41], to be

$$R(P_M, Q_M, d) = \inf_{\substack{V:[V]_x = P_M,\\ \mathbb{E}_V(\rho(\tilde{\mathbf{X}}, \tilde{\mathbf{Y}})) \le d}} \mathcal{D}(V || P_M \times Q_M), \tag{2.8}$$

$$R(P_M, Q_M, d) = \inf_{Q'_M} \{ I_{\min}(P_M || Q'_M, d) + \mathcal{D}(Q'_M || Q_M) \}, \tag{2.9}$$

where $\mathcal{D}(\cdot||\cdot)$ is the Kullback-Leibler (KL) divergence, and $I_{\min}(P_M||Q'_M, d)$ is the usual minimum mutual information but with an additional constraint on the output distribution, i.e.,

$$I_{\min}(P_M || Q'_M, d) = \inf_{\substack{V:[V]_x = P_M,\ [V]_y = Q'_M,\\ \mathbb{E}_V(\rho(\tilde{\mathbf{X}}, \tilde{\mathbf{Y}})) \le d}} I(\tilde{\mathbf{X}}, \tilde{\mathbf{Y}}). \tag{2.10}$$

Here the infimum is taken over all joint distributions V of the random vectors $(\tilde{\mathbf{X}}, \tilde{\mathbf{Y}})$, whose x-marginal, denoted by $[V]_x$, is P_M, and y-marginal, denoted by $[V]_y$, is Q'_M, and such that the expected distortion does not exceed d. In [42, Th. 2], it was shown that, under these assumptions for the memoryless case (for which extension to the sources with memory case is straight forward), $R(P_M, Q_M, d)$ is finite, strictly positive, and that the infimum in its definition in (2.8) is always achieved by some joint distribution $V^*_{P_M, Q_M, d}$. Moreover, since the set of V over which the infimum is taken is convex, from [43] it can be concluded that $V^*_{P_M, Q_M, d}$ is the unique minimizer. Hence, a unique minimizer to (2.9) also exists, i.e.,

$$Q^*_{P_M, Q_M, d} = \arg\min_{Q'_M} \{ I_{\min}(P_M || Q'_M, d) + \mathcal{D}(Q'_M || Q_M) \}. \tag{2.11}$$

Next, we define the minimum coding rate per letter for stationary ergodic sources with memory required to guarantee a d-match with probability one asymptotically in L as [38, 39],

$$R(d) = \lim_{M \to \infty} M^{-1} R(P_M, d). \tag{2.12}$$

The limit in (2.12) exists for stationary ergodic sources, and for any M, $R(P_M, d)$ is an upper bound to $R(d)$ [44, Th. 9.8.1]. Consequently, the optimal codebook reproduction distribution that achieves $R(d)$ is obtained as,

$$Q_d^* = \lim_{M \to \infty} Q_{P_M,d}^*. \tag{2.13}$$

For a source with discrete input and reproduction alphabets, define a '*type*' of source or code vector as the fraction of occurrence of every letter in the alphabet as seen in the vector [45]. To accommodate sources with memory, and account for memory depth of M, we proceed as follows [46]: define $Q_{n,M,L}\left(\tilde{\mathbf{y}}(j)\right) = \left\{Q(\mathbf{y}) : Q(\mathbf{y}) = \frac{1}{L}N(\mathbf{y}|\tilde{\mathbf{y}}(j)), \mathbf{y} \in \mathcal{Y}^M\right\}$ as the *M-th order type* of codeword $\tilde{\mathbf{y}}(j)$, where $N(\mathbf{y}|\tilde{\mathbf{y}}(j))$ is the number of occurrences of the sub-vector (or super symbol) $\mathbf{y}$ in the codeword. This is simply the type for the source considered as over a "super alphabet" of super-symbols. Now suppose a finite source block length L is considered, then for a given finite codebook $\tilde{\mathcal{C}}_L$ of C codewords, i.e., $\tilde{\mathcal{C}}_L = \{\tilde{\mathbf{Y}}(j),\ \text{with}\ j = 1, \ldots, C\}$, we define an (L, M, C, d, ϵ) code as a pair of mappings $f : \mathcal{X}^{ML} \to \{1, \ldots, C\}$, and $g : \{1, \ldots, C\} \to \mathcal{Y}^{ML}$, such that,

$$f(\tilde{\mathbf{x}}) = \inf\left\{j \in \{1, \ldots, C\} : \rho\left(\tilde{\mathbf{x}}, \tilde{\mathbf{y}}(j)\right) \leq d\right\}, \tag{2.14}$$

and the probability of exceeding the distortion constraint d between a source word and its reproduction is,

$$\mathbb{P}\left[f\left(\tilde{\mathbf{x}}\right) = +\infty\right] \leq \epsilon. \tag{2.15}$$

The rate of the (L, M, C, d, ϵ) code is denoted by $R_L(P_M, d, \epsilon) = \frac{1}{L}\log(C)$. By Shannon's theorem of lossy coding, if the codebook generating distribution is $Q_{P_M,d}^*$, it can be shown that [47],

$$\lim_{L \to \infty} R_L(P_M, d, \epsilon) \to R(P_M, d), \quad \forall \epsilon \in [0, 1]. \tag{2.16}$$

2.2.1 Lossless Random Coding: Lempel-Ziv Algorithms

Seminal contributions were achieved by the lossless coding algorithms developed by Lempel and Ziv, as evidenced by the numerous prevalent variants of the LZ77 and LZ78 algorithms [7, 8], where noticeably the generated codebooks do not require exponential d-match search complexity.

As an example of how stochastic codebook generation is employed, consider LZ78, where on-the-fly compression is performed by creating a dictionary or tree of codewords, as source strings are encoded. This tree of codewords is created, without recourse to prior knowledge of source statistics, in a manner that ensures that the relative frequency of typical source sequences in the tree, asymptotically approaches one [8]. The main idea behind LZ algorithms, is that if we consider a non-uniformly distributed source, a sub string that has already been generated and seen by the source is more likely to be generated again than a sub-string that have not been seen yet. The LZ78 algorithm works by constructing a dictionary of sub-strings, which we will call “phrases”, that have been generated by the source. The LZ78 algorithm constructs its dictionary on the fly, only going through the data once. For examples, suppose that the source alphabet is given by $\mathcal{X} = \{A, B\}$, and suppose the source generated the following sequence of letters, which is required to be encoded and transmitted, i.e.,

$$AAABABBABABABAAAABABABA \tag{2.17}$$

The algorithm starts by encoding the shortest phrase that has not been seen in the dictionary. In the beginning of the encoding process, the shortest phrase is always the left most single letter in the source sequence. Hence, the first phrase in the dictionary is

Output Message	Dictionary	
	Index	Phrase
$(0, A)$	1	A
$(1, A)$	2	AA
$(0, B)$	3	B
$(1, B)$	4	AB
$(3, A)$	5	BA
$(5, B)$	6	BAB
$(4, A)$	7	ABA
$(2, A)$	8	AAA
$(6, A)$	9	$BABA$

Table 2.1: The dictionary table constructed by LZ78 algorithm for the given example in Section 2.2.1

$'A'$.

$$A|AABABBABABABAAAABABABA \tag{2.18}$$

Now we proceed to the next phrase in the sequence that has not been seen in the dictionary, hence the second phrase imported in the dictionary is $'AA'$ because the phrase $'A'$ is already available in the dictionary.

$$A|AA|BABBABABABAAAABABABA \tag{2.19}$$

The algorithm continues to process the source string on-the-fly, and consequently, the source sequence will be divided into phrases as follows:

$$A|AA|B|AB|BA|BAB|ABA|AAA|BABA|BA \tag{2.20}$$

Finally, the dictionary is constructed from the observed phrases and an output message is constructed for every phrase. The output message can be one of three possible options: $i)$ $(0, x)$ if the one character phrase $x \in \mathcal{X}$ is not in the dictionary, $ii)$ (dictionary index, x) if the multi-character phrase ending with the letter $x \in \mathcal{X}$ is not in the dictionary, and *iii*) (dictionary index, $\emptyset$) if the last sequence letter or the last phrase is entirely in the dictionary. In the above examples, the corresponding dictionary as well as the output messages are given in Table 2.1. Finally, the LZ78 encoding and dictionary generation algorithm is summarized in Algorithm 1.

Algorithm 1 : LZ78 Algorithm in [8]

```
procedure LZ78($x_1, x_2, \ldots, x_L$)
    Dictionary $\mathcal{D} \leftarrow \{\emptyset\}$, dictionary index $i \leftarrow 1.$, string index $n \leftarrow 1$.
    while $n \leq L$ do
        Find the shortest phrase $\mathbf{y} = \{x_n, \ldots, x_{n+N-1}\} \notin \mathcal{D}$.
        $n \leftarrow n + N$.
        if $N$ is equal to 1 then.
            Send the message $(0, x_{n+N-1})$.
            Insert $\mathbf{y}$ to $\mathcal{D}$ at index $i$.
            $i \leftarrow i + 1$.
        else if $\mathbf{y} \notin \mathcal{D}$ then
            Send the message $(j, x_{n+N-1})$, where $\{x_n, \ldots, x_{n+N-2}\} \in \mathcal{D}$ at index $j$.
            Insert $\mathbf{y}$ to $\mathcal{D}$ at index $i$.
            $i \leftarrow i + 1$.
        else
            Send the message $(j, \emptyset)$, where $\{x_n, \ldots, x_{n+N-1}\} \in \mathcal{D}$ at index $j$.
        end if
    end while
    return $\mathcal{D}$.
end procedure
```

It can be shown that, asymptotically in the string length L, the LZ78 algorithm average encoding length approaches the entropy of the source, and the probability of finding an independently generated source string in the dictionary goes to one, thus establishing the asymptotic optimality of LZ78 algorithm. The crucial impact attained by LZ algorithms in lossless source coding has provided a strong motivation to develop and derive alternative algorithms for lossy source coding, in which the source phrase is allowed to be reconstructed within some distortion level d. Note that the optimal codebook generating distribution in lossless coding is simply the source distribution, so the stochastic mechanism's essential objective is to learn this distribution from observation of source strings. However, in lossy coding, the optimal codebook generating, or reproduction distribution Q^* differs from the source distribution P, as it depends on the distortion constraint d. This represents a non-trivial learning challenge, especially in the *non*-high resolution regime, where Q^* deviates significantly from P [12, 13, 14, 10]. For example, in the case of continuous alphabet sources with the squared error distortion measure, at small distortion (high resolution) $Q^* \approx P$, but as the distortion constraint is relaxed, i.e., d increases, Q^* increasingly differs from P, it shrinks, often becomes discrete, and eventually collapses to a single point when $d = d_{\max}$ [15]. In the next subsection, we introduce a stochastic codebook design algorithm based on source examples for lossy coding, namely Natural Type Selection (NTS).

2.2.2 Lossy Random Coding: Natural Type Selection Algorithm

In [10], a novel codebook regeneration algorithm was developed and shown to achieve asymptotically optimal performance, in the rate-distortion sense, for discrete memoryless sources. Consider the memoryless case for which M is set to one, and the source letters are generated according to $P_1 = \{P_1(x) : x \in \mathcal{X}\}$, where $\mathcal{X}$ is a discrete alphabet space. The subscript "1" here and below stands for $M = 1$. Additionally, let the memoryless codebook reproduction distribution over discrete alphabet space $\mathcal{Y}$, be $Q_1 = \{Q_1(y) : y \in \mathcal{Y}\}$. It was shown in [10] that the empirical type of the codeword that d-matches an independently generated source word, converges in probability to $Q^*_{P_1,Q_1,d}$ as the string length $L \to \infty$. Note that $Q^*_{P_1,Q_1,d}$ is more efficient in coding the source than the initial Q_1, i.e., $R(P_1, Q^*_{P_1,Q_1,d}, d) < R(P_1, Q_1, d)$. This immediately suggests a recursive algorithm. Let n be the iteration index, $N_{1,L}$ be the index of the first d-matching codeword, whose type is denoted as $Q_{n,1,L}^{N_{1,L}}$. Starting with a strictly positive initial codebook reproduction distribution denoted $Q_{0,1,L}$, the type of the d-matching codeword at the current iteration is used to generate the codebook of the next iteration. In other words, the next iteration's codebook reproduction distribution is naturally selected by the source through a d-match event, hence the name "natural type selection". The original NTS algorithm is summarized in Algorithm 2.

Algorithm 2 : Original NTS Algorithm for memoryless Sources in [10]

```
procedure NTS_Original($N, L, d, Q_0, \mathbf{x}(1), \ldots, \mathbf{x}(N)$)
    $Q_{1,1,L} \leftarrow Q_0$.
    for $n = 1 : N$ do
        $i \leftarrow n$.
        $j \leftarrow 0$.
        while $d' < d$ do
            $j \leftarrow j + 1$.
            Generate $j$-th codeword $\mathbf{y}(j)$ using $Q_{n,1,L}$.
            $d' \leftarrow \rho\left(\mathbf{x}(i), \mathbf{y}(j)\right)$.
        end while
        $Q_{n+1,1,L} \leftarrow Q_{n,1,L}^{N_{1,L}}$.
    end for
    return $Q_{N+1,1,L}$.
end procedure
```

This algorithm results in a sequence of codebook reproduction distributions,

$$Q_{n,1,L} = Q_{n-1,1,L}^{N_{1,L}}, \tag{2.21}$$

$$Q_{n,1} = \lim_{L \to \infty} Q_{n,1,L} = Q^*_{P_1, Q_{n-1,1}, d}, \ n = 1, 2, \ldots \tag{2.22}$$

It was shown in [10], that the sequence of distributions in (2.22), i.e., $Q_{0,1}, Q_{1,1}, Q_{2,1}, \ldots$, coincides with the recursion in the fixed distortion version of the Blahut algorithm [48] for computation of the rate-distortion function. In other words, the NTS procedure stochastically simulates the Blahut algorithm, where the next distribution at each iteration step emerges "on the fly" through the coding process. Hence, it was shown that the

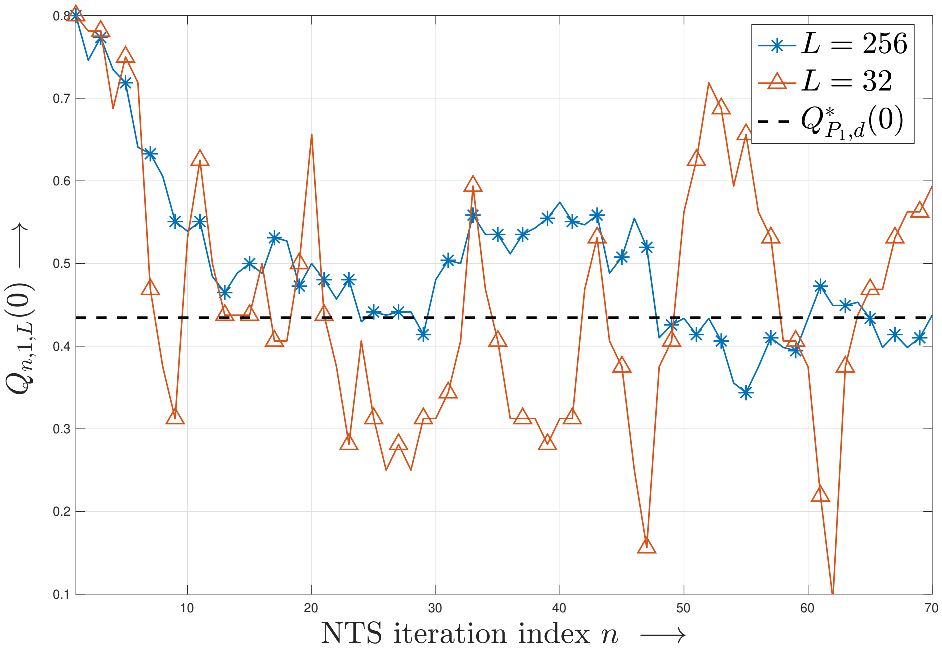


Figure 2.2: Unstable evolution of the codebook reproduction distribution by the original NTS algorithm for finite-lengths $L = 32$ and $L = 256$. Binary source is considered with $P_1 = \{0.48, 0.52\}$, and Hamming distortion measure with $d = 0.35$.

recursion in (2.22) converges to the optimal codebook distribution $Q^*_{P_1,d}$ that achieves the rate-distortion bound $R(P_1, d)$ in (2.7) for $M = 1$, i.e.,

$$Q^*_{P_1,d} = \lim_{n\to\infty} \lim_{L\to\infty} Q_{n,1,L}, \tag{2.23}$$

$$R(P_1, d) = \lim_{n\to\infty} \lim_{L\to\infty} R(P_1, Q_{n,1,L}, d). \tag{2.24}$$

The asymptotic optimality result, established for the original NTS algorithm, suffers from several fundamental shortcomings the impact its practical implementation. The first shortcoming pertains to complexity and hinges on the order of limits that requires that string length L be sent to infinity first, and only then can NTS iterations be performed ($n \to \infty$). In other words, NTS iterations must be performed on very large strings. Unfortunately, the probability of finding a d-match decreases exponentially with the string length, or alternatively, the codebook size must grow exponentially with L, which implies intractable d-search complexity, even in early NTS iterations. Clearly, it is the reversed order of limits that would be desirable in practice. To see the difficulties encountered when attempting to run NTS at finite L, consider a toy example assuming binary input and reproduction alphabets, i.e., $\mathcal{X} = \mathcal{Y} = \{0, 1\}$, asymmetric memoryless ($M = 1$) source, with $P_1 = \{0.48, 0.52\}$, under the Hamming distortion measure. Fig. 2.2 depicts the instability, in terms of codebook reproduction distribution, exhibited by the original NTS algorithm at finite string lengths $L < \infty$, with obvious fluctuations. Note that even though the codebook reproduction distributions tend to approach the neighborhood around $Q^*_{P_1,d}$ as the iteration index n increases, these fluctuations ultimately render the resulting codebook reproduction distribution useless. It is also worth noting that as the source block length L increases, the intensity of codebook reproduction distribution fluctuation decreases, where it vanishes asymptotically in L. In [11], a parametric set of codebook reproduction distributions $\mathcal{Q}_\Theta$ was considered, wherein the codebook reproduction distributions were constrained to a pre-specified family of distributions, $Q \in \mathcal{Q}_\Theta$, spanned by a parameter vector $\theta \in \Theta$. A smoothing block on the parameter vector θ was proposed to reduce the fluctuations of the codebook reproduction distributions around the optimal solution, at finite string length L, which nevertheless exhibited some significant instability (see Figure 2 in [11]). An additional limitation of the original NTS algorithm was that optimality results were only available for discrete alphabet memoryless sources. In practice, however, lossy coding is applied to a much broader class of sources, and sources with memory and/or continuous alphabet sources are common. The above shortcomings represent a significant obstacle on the way to achieve major impact on lossy coding. The phenomenal impact of stochastic codebook generating algorithms (e.g., LZ78 in [8]), in many lossless coding applications, suggests that overcoming these shortcomings may deliver considerable benefits. This provides a strong motivation to develop a tractable NTS algorithm that is asymptotically optimal for a wide spectrum of sources. In the next section, we turn our attention to non-stochastic codebook design methods, i.e., methods for which the source statistics are available, and hence, the design steps are carried deterministically based on the available statistics rather than the stochastic source examples. Consequently, the codebook is generated deterministically from the available training set, unlike the stochastic codebook design methods.

2.3 Relevant Non-stochastic Codebook Design

As the deterministic quantizer design or, more generally, the clustering problem, appears with various flavors in many diverse applications, solution methods have been developed in multiple disciplines, e.g. the Generalized Lloyd algorithm [3] or the Max quantizer [2] in communications or information theory literature, or the ISODATA [4] and the k-means [5] algorithms in the pattern recognition literature. All the aforementioned iterative methods alternate between two complementary steps (often referred to as the Lloyd iteration): optimization of the partition into clusters given the current codebook entries, and optimization of the codebook entries for their respective clusters. The details of the Lloyd iteration will be illustrated in the next subsection.

Even though these algorithms ensure convergence, they only guarantee convergence to a locally optimal solution, while in many cases of interest the cost surface is riddled with poor local minima. A variety of heuristic approaches have been proposed to tackle this difficulty, and they range from repeated optimization with different initialization, and heuristics to obtain good initialization, to heuristic rules for cluster splits and merges, etc. Nevertheless, there is a substantial gain to be recouped by a principled attack on the problem. This motivates the use of powerful optimization tools. Deterministic annealing has been demonstrated to be highly effective in avoiding poor local minima, when conventional distortion measures are used, and has become the method of choice in numerous disciplines [6]. DA is motivated by the annealing process in physical chemistry, where certain chemical systems are driven to their low energy states by annealing, i.e., via gradual cooling of their temperature. Additional non-convex optimization tools have also been inspired by the annealing process of chemical systems, such as stochastic relaxation [49] or simulated annealing [50]. However, these optimization methods can only reach the global minimum if the rate of lowering the temperature follows $T \propto 1/\log(n)$, where n is the iteration index [49]. This slow annealing schedule is often unrealistic in many practical applications. As its name suggests, DA tries to enjoy the best of two scenarios. On the one hand it is deterministic, meaning that random motion on the energy surface while making incremental progress on the average, as is the case for stochastic relaxation, is discouraged due to its slow convergence. On the other hand, it is still an annealing method and aims at the global minimum, instead of getting greedily attracted to a nearby local minimum.

DA introduces a controlled amount of randomness in the optimization, measured by the Shannon entropy, and controlled by a Lagrange multiplier T, analogous to “temperature” in the physical system. The resulting Lagrangian, an expectation function accounting for the tradeoff between distortion and entropy, is in fact exactly the Helmholtz free energy in physics, and is *deterministically* minimized at successive temperatures, thus circumventing the high computational complexity of stochastic simulated annealing. In the next subsections we will introduce two relevant clustering approaches in depth, i.e., the Generalized Lloyd algorithm and Deterministic Annealing

2.3.1 Generalized Lloyd Algorithm

In this subsection, we deeply illustrate the inner-workings of the generalized Lloyd algorithm or its relatives. of The two steps in Lloyd iteration can be formally stated as:

1. Fix the codebook entries $\{\mathbf{y}_j\}$, and assign each data point $\mathbf{x}_i$ to the codeword incurring the least distortion. Let $\mathcal{S}_j$ be the set of data points assigned to codeword y_j, also called the jth cluster. The clustering partition is given by the (generalized) nearest neighbor rule. Specifically, cluster j is given by:

$$\mathcal{S}_j = \{i \colon \rho(\mathbf{x}_i, \mathbf{y}_j) \leq \rho(\mathbf{x}_i, \mathbf{y}_k),\ \forall k \neq j\}\,. \tag{2.25}$$

2. Fix the clustering partition $\{\mathcal{S}_j\}$ and optimize the entries in the codebook to minimize the average distortion. Specifically, adjust each codeword $\mathbf{y}_j$ so that it minimizes its cluster's average distortion:

$$\mathbf{y}_j = \arg\min_{\mathbf{y}} \frac{1}{|\mathcal{S}_j|} \sum_{i \in \mathcal{S}_j} \rho(\mathbf{x}_i, \mathbf{y}), \quad j = 1, 2, \ldots, N_c, \tag{2.26}$$

where $|\cdot|$ denotes the set cardinality, and N_c is the number of codewords (or the number of centroids) in the codebook. A necessary condition for optimality can be obtained by setting the gradient with respect to $\mathbf{y}$ to zero:

$$\frac{1}{|\mathcal{S}_j|} \sum_{i \in \mathcal{S}_j} \frac{\partial}{\partial \mathbf{y}} \rho(\mathbf{x}_i, \mathbf{y}) = 0 \quad, j = 1, 2, \ldots, N_c, \tag{2.27}$$

Note that the traditional K-means "centroid" rule which computes each codebook entry as the cluster sample average,

$$\mathbf{y}_j = \frac{1}{|\mathcal{S}_j|} \sum_{i \in \mathcal{S}_j} \mathbf{x}_i, \tag{2.28}$$

is only valid for the squared error distortion measure, where (2.27) simplifies to (2.28).

In every "Lloyd iteration", one can evaluate the average distortion as,

$$D = \frac{1}{|\mathcal{S}_1 \cup \mathcal{S}_2 \cdots \cup \mathcal{S}_{N_c}|} \sum_{j=1}^{N_c} \sum_{i \in \mathcal{S}_j} \rho(\mathbf{x}_i, \mathbf{y}_j), \tag{2.29}$$

where $\cup$ denotes the set union operation.

It is straightforward to show that the two steps of the main iteration guarantee that D is monotonically non-increasing, and in fact monotonically decreasing until convergence (under mild assumptions regarding treatment of ties in the nearest neighbor step). Additionally, note that as $N_c \to \infty$, the codebook average distortion asymptotically vanishes, i.e., $D \to 0$, which is consistent with standard requirements of distortion measures and represents the "ideal setting" at the limit of high resolution. Finally, the GLA algorithm is summarized in Algorithm 3.

2.3.2 Deterministic Annealing

Unlike the GLA algorithm, DA considers a probabilistic assignment between the data points $\{\mathbf{x}_i\}$ and codebook entries or cluster centroids $\{\mathbf{y}_j\}$. Let the cluster association probabilities be denoted as $p(j|i)$. In this case, the overall average distortion in the

Algorithm 3 : Generalized Lloyd Algorithm in [3]

```
procedure GLA($\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N, \mathcal{C}_0, N_c$)
    Initialize the codebook $\mathcal{C} = \{\mathbf{y}_1, \ldots, \mathbf{y}_{N_c}\} \leftarrow \mathcal{C}_0$.
    Initialize clusters $\mathcal{S}_j = \{\emptyset\}, \forall j \in \{1, \ldots, N_c\}$.
    while Convergence has not been achieved, i.e., $\frac{\Delta D}{D} >$ threshold do
        for $i = 1 : N$ do
            Adjust clusters by assigning $\mathbf{x}_i$ to $\mathcal{S}_j$ if $\mathbf{y}_j = \arg\min_{\mathbf{y} \in \mathcal{C}} \rho(\mathbf{x}_i, \mathbf{y})$.
        end for
        for $j = 1 : N_c$ do
            Adjust centroids according to $\mathbf{y}_j = \arg\min_{\mathbf{y}} \frac{1}{|\mathcal{S}_j|} \sum_{i \in \mathcal{S}_j} \rho(\mathbf{x}_i, \mathbf{y})$.
        end for
        Calculate the total average distortion $D = \frac{1}{|\mathcal{S}_1 \cup \mathcal{S}_2 \cdots \cup \mathcal{S}_{N_c}|} \sum_{j=1}^{N_c} \sum_{i \in \mathcal{S}_j} \rho(\mathbf{x}_i, \mathbf{y}_j)$.
    end while
    return $\mathcal{C}$.
end procedure
```

system due to quantization of data points is given by the expectation,

$$D = \sum_i \sum_j p(j|i)p(i)\rho(\mathbf{x}_i, \mathbf{y}_j), \tag{2.30}$$

where $p(i)$ is the prior probability of a data vector $\mathbf{x}_i$. Note that minimizing the distortion with respect to the free parameters $\{\mathbf{y}_j, p(j|i)\}$ would immediately lead to hard association between the data point and the nearest codebook entry, where the term "nearest" is used in the sense of the distortion measure. Instead, the distortion is minimized subject to an imposed level of randomness, which is naturally measured by Shannon's entropy H. Hence, the Lagrangian function to be minimized can be written as,

$$\mathcal{L} = D - TH, \tag{2.31}$$

where,

$$H = -\sum_i \sum_j p(j|i)p(i) \log \left(p(j|i)p(i)\right), \tag{2.32}$$

and T ("temperature") is the Lagrangian parameter. Next, an iterative approach, which is an appropriately designed random relative of the GLA algorithm, is employed to minimize the Lagrangian function:

1. Initialize temperature, $T = T_{\max}$ and the entries of the codebook $\{\mathbf{y}_j\}$.

2. Fix the codebook $\{\mathbf{y}_j\}$ and find the random clustering partition (i.e., probabilistic assignment of data points to centroids) which minimizes the Lagrangian cost:

   $$\{p(j|i)\} = \arg\min_{\{p(j|i)\}} \mathcal{L}, \qquad \forall i, \forall j \tag{2.33}$$

   Note that the solution must further impose the constraint $\sum_j p(j|i) = 1, \forall i$, which directly yields a random relative of the nearest neighbor rule, given by the Gibbs distribution:

   $$p(j|i) = \frac{\exp\left(-\frac{\rho(\mathbf{x}_i,\mathbf{y}_j)}{T}\right)}{Z_i}, \tag{2.34}$$

   where the normalization constant is

   $$Z_i = \sum_j \exp\left(-\frac{\rho(\mathbf{x}_i, \mathbf{y}_j)}{T}\right), \tag{2.35}$$

   sometimes called the partition function in physics.

3. Fix the random clustering partition, $\{p(j|i)\}$ and optimize the entries of the codebook to minimize the Lagrangian cost. Specifically,

   $$\{\mathbf{y}_j\} = \arg\min_{\{\mathbf{y}_j\}} \mathcal{L} = \arg\min_{\{\mathbf{y}_j\}} D, \tag{2.36}$$

where we used the fact that the entropy is determined by the (fixed) clustering partition, and hence can be discarded from $\mathcal{L}$ in this step. Noting further that D is additive in the contributions of individual codebook entries, we obtain:

$$\mathbf{y}_j = \arg\min_{\mathbf{y}} \sum_i p(j|i)p(i)\rho(\mathbf{x}_i, \mathbf{y}), \tag{2.37}$$

or as necessary condition for optimality, the random relative of the centroid rule:

$$\sum_i p(j|i)p(i)\frac{\partial}{\partial \mathbf{y}}\rho(\mathbf{x}_i, \mathbf{y}) = 0 \quad , j = 1, 2, \ldots, N_b, \tag{2.38}$$

4. Check if convergence condition satisfied, else go to step 2.

5. Cool the system, e.g., $T = \alpha T$, with $\alpha < 1$. If the prescribed minimum temperature is reached, then terminate the algorithm.

6. Perturb the codebook entries to check for possible splitting of codebook centroids, also known as phase transition, then go to step 2.

At $T = 0$, the DA algorithm degenerates to the GLA algorithm, however the annealing process until then eliminates the sensitivity to initialization. In step 4, convergence can be checked by comparing $\frac{\Delta\mathcal{L}}{\mathcal{L}}$ to a convergence threshold. It is important to note that by gradual cooling, the system undergoes a series of phase transitions at corresponding "critical temperatures", in analogy to physical systems, wherein the cardinality of the codebook grows. The critical temperatures can be derived using tools of variational calculus. Let us consider first the case of very high temperature, $T \to \infty$, where the Gibbs distribution of (2.34) becomes a uniform distribution, and all the codebook entries

$\{\mathbf{y}_j\}$ merge to a single codebook entry, $\mathbf{y}^*$, and the condition in (2.38) simplifies to,

$$\sum_i p(i) \frac{\partial}{\partial \mathbf{y}} \rho(\mathbf{x}_i, \mathbf{y}^*) = 0. \tag{2.39}$$

Hence, at high temperature, there is effectively one codebook entry and one cluster - the entire training set. Next let $\mathcal{L}^*$ denote the minimum of the Lagrangian function $\mathcal{L}$ over the cluster association probabilities, i.e.,

$$\begin{aligned} \mathcal{L}^*(\{\mathbf{y}_j\}, T) &= \min_{\{p(j|i)\}} \mathcal{L}, \\ &= -T \sum_i p(i) \log \sum_j \exp\left(-\frac{\rho(\mathbf{x}_i, \mathbf{y}_j)}{T}\right), \end{aligned} \tag{2.40}$$

which must still be minimized with respect to the codebook. The bifurcation or splitting of codebook entries occurs when, as the temperature is lowered, the existing codebook entries that satisfy the necessary condition for optimality, no longer correspond to a minimum of $\mathcal{L}^*$, i.e., we now observe a saddle point or a maximum of the cost $\mathcal{L}^*$. The necessary condition for optimality of a codebook $\{\mathbf{y}_j\}$, is

$$\frac{d}{d\epsilon} \mathcal{L}^*(\{\mathbf{y}_j + \epsilon \eta_j\}, T)|_{\epsilon=0} = 0, \quad \forall \{\eta_j\}, \tag{2.41}$$

where $\{\mathbf{y}_j + \epsilon \eta_j\}$ is a perturbed codebook with finite perturbation vectors $\{\eta_j\}$. The solution represents a minimum of the cost, as long as the second-order derivative is positive for all finite perturbations,

$$\frac{d^2}{d\epsilon^2} \mathcal{L}^*(\{\mathbf{y}_j + \epsilon \eta_j\}, T)|_{\epsilon=0} > 0, \quad \forall \{\eta_j\}. \tag{2.42}$$

Consequently, bifurcation occurs when the gradually lowered temperature yields equality in (2.42), hence the solution is no longer stable. (Intuitively, there are perturbation directions along which if we split the codebook entries, we will be able to decrease the cost.) This temperature is what is called critical temperature in statistical physics. See [6] for extensive analysis of DA's sequence of phase transitions through which the cardinality of the codebook grows, as well as for demonstration that the algorithm is invariant to initialization. Finally, the DA algorithm is summarized in Algorithm 4.

Algorithm 4 : Deterministic Annealing in [6]

```
procedure DA($\mathbf{x}_1, \ldots, \mathbf{x}_N, p(1), \ldots, p(N), \mathcal{C}_0, N_c$)
    Initialize the codebook $\mathcal{C} = \{\mathbf{y}_1, \ldots, \mathbf{y}_{N_c}\} \leftarrow \mathcal{C}_0$.
    Initialize temperature $T = T_{\max}$.
    while $T > T_{\min}$ do
        while Convergence has not been achieved, i.e., $\frac{\Delta\mathcal{L}}{\mathcal{L}} >$ threshold do
            for $i = 1 : N$ do
                Adjust cluster association probabilities of $\mathbf{x}_i$ as $p(j|i) = \frac{\exp\left(-\frac{\rho(\mathbf{x}_i, \mathbf{y}_j)}{T}\right)}{Z_i}$.
            end for
            for $j = 1 : N_c$ do
                Adjust centroids according to $\mathbf{y}_j = \arg\min_{\mathbf{y}} \sum_i p(j|i)p(i)\rho(\mathbf{x}_i, \mathbf{y})$.
            end for
            Calculate the total average Lagrangian function $\mathcal{L} = D - TH$.
        end while
        Cool the system, i.e., $T = \alpha T$, with $\alpha < 1$.
        Perturb centroids, i.e., $\mathbf{y}_j = \mathbf{y}_j + \mathbf{w}_j$, where $\mathbf{w}_j$ is the perturbation noise.
    end while
    return $\mathcal{C}$.
end procedure
```

Chapter 3

A Tractable Natural Type Selection Codebook Design Algorithm

In this chapter, we develop a tractable version of the NTS algorithm for a larger spectrum of sources. The results included in this chapter are published in [51, 46, 52, 53]. We adopt the same notation (for source words, codewords, distortion function, alphabet spaces, etc.) as defined in Chapter 2. In other words, denote $\mathcal{X}$ and $\mathcal{Y}$ as the source and reproduction alphabet spaces. We assume that the alphabet $\mathcal{X}$ is either a discrete space or (more generally) a complete separable metric space (often called Polish space), equipped with its associated Borel σ-field $\mathcal{X}'$. Similarly, we assume that the reproduction alphabet $\mathcal{Y}$ is either a discrete space or (more generally) also a Polish space equipped with its associated Borel σ-field $\mathcal{Y}'$. Furthermore, let $\{X_u\}_{u=1}^{\infty}$ be a stationary ergodic source, where the source realization is denoted as $x_u \in \mathcal{X}$, and similarly, the reproduction realization is denoted as y_u. Next, let $\{\tilde{\mathbf{X}}_i\}_{i=1}^{\infty}$ be a sequence of independent and identically distributed (i.i.d.) M-tuples (or source "super-symbols"), each obtained by drawing M successive symbols from the distribution underlying source $\{X_u\}$. Hence, let P_M be the vector source distribution of $\tilde{\mathbf{x}}$ on $\mathcal{X}^M$. Define a source block (source word) that contain L source vectors as $\mathbf{X} = \left(\tilde{\mathbf{X}}_1, \tilde{\mathbf{X}}_2, \ldots, \tilde{\mathbf{X}}_L\right)$, and source block realization as $\mathbf{x}$. Next, we define an arbitrary non-negative (measurable) scalar-valued distortion function $\rho\colon \mathcal{X}^M \times \mathcal{Y}^M \rightarrow [0, \infty)$. The distortion between a realization of the source block $\mathbf{x}$ and a realization of the code block (codeword or reproduction word) $\mathbf{y} = (\tilde{\mathbf{y}}_1 \ldots, \tilde{\mathbf{y}}_L)$, with $\tilde{\mathbf{y}}_i \in \mathcal{Y}^M$, is assumed additive, and is specifically, the average distortion over super-symbols in the block:

$$\rho\left(\mathbf{x}, \mathbf{y}\right) = \frac{1}{L} \sum_{\ell=1}^{L} \rho\left(\tilde{\mathbf{x}}_\ell, \tilde{\mathbf{y}}_\ell\right) = \frac{1}{L} \sum_{\ell=1}^{L} \left(\frac{1}{M} \sum_{m=1}^{M} \rho(x_{\ell,m}, y_{\ell,m}) \right), \tag{3.1}$$

where $x_{\ell,m}$ and $y_{\ell,m}$ are the m-th letters in $\tilde{\mathbf{x}}_\ell$ and $\tilde{\mathbf{y}}_\ell$, respectively. First, we turn our attention to sources over discrete alphabet spaces. Generalization of such tractable algorithm to sources over abstract alphabets is introduced next in section 3.3.

3.1 Discrete Alphabet Sources

We start by posing a natural and important question: can a more effective algorithm be devised such that converging behavior is achieved in a non-asymptotic parameter settings? Can, at least, the convergence to the optimal reproduction distribution $Q^*_{P_M,d}$ be achieved but through a reversed order of limits? I.e., can we achieve $Q^*_{P_M,d}$ by first sending n to infinity, while maintaining finite L, and then sending L to infinity? Obviously, if string length L is finite, then the type of the d-matching codeword (of the same length) is restricted in resolution to $1/L$, as the relative frequency of a letter or super symbol, in the codeword, is a multiple of $1/L$. Such a low resolution of types may cause difficulties for an iterative algorithm that advances by potentially very small adjustments to the distribution. In order to circumvent this shortcoming, we propose to update the estimate of a general codebook reproduction distribution (not restricted to type resolution of

$1/L$), after observing many d-matching events. In other words, we find the *maximum likelihood* estimate of the distribution that would have generated the observed sequence of d-matching codewords in response to a sequence of independently generated source words. Note that this approach is closely connected to finding the maximum likelihood codebook reproduction distribution, within a family of distributions, that maximize the probability of d-match events, which has been investigated in [54, 55]. However, in our method, we find the reproduction distribution that maximizes the probability of generating a set of observed d-matching codewords, which is shown to provide asymptotically optimal rate-distortion results as will be proved in Theorems 1-7. Let K be the number of the d-matching events considered before performing maximum likelihood estimation and updating the codebook reproduction distribution.

Lemma 1: The Maximum Likelihood (ML) estimate of the codebook reproduction distribution that generates i.i.d. M-length super symbols, given a set of K d-matching codewords, is the *average* of the d-matching codeword types, i.e.,

$$Q_{n+1,M,L,K} = \hat{Q}^{\mathrm{ML}} = \frac{1}{K}\sum_{k=1}^{K} Q_k, \tag{3.2}$$

where k enumerates the d-matching events, and Q_k is the M-order type of the k-th d-matching codeword $\tilde{\mathbf{y}}(j_k)$, whose index in the random codebook is j_k. In these settings, the codeword (and source word) consists of L sub-vectors (or super symbols) each of size M. A simplified version of this lemma, with simplified notations, has appeared in [51] for discrete alphabet memoryless ($M = 1$) sources.

The proof of Lemma 1 is given in Appendix A.1. The sketch of proof is, first, to establish that the probability of generating a codeword from a given distribution only depends on its type. Next, we realize that the maximum likelihood distribution that generates a set of a d-matching codewords, is the one that minimize the sum of KL divergences seen versus all d-matching codewords. The average type of K d-matching length-L codewords is exactly equal to the type of length-KL code block formed by concatenating the K d-matching codewords. Note that the resolution of the maximum likelihood codebook reproduction distribution $\hat{Q}^{\mathrm{ML}}$ is $1/(LK)$. However, the complexity of computing $\hat{Q}^{\mathrm{ML}}$ only grows linearly with K, while the d-matching complexity is exponential in L. This result immediately suggests a modified and considerably more tractable variant of the NTS recursive algorithm. Starting with an arbitrary and strictly positive initial reproduction distribution $Q_{0,M,L}$ (the subscripts '0' denotes $n = 0$), the *average* M-th order type of K d-matching codewords is used to generate a new codebook in the next NTS iteration. The modified NTS algorithm is summarized in Algorithm 5.

Algorithm 5 : Modified NTS Algorithm for Discrete Alphabets

```
procedure NTS_Modified_Discrete($N, M, L, K, d, Q_0, \mathbf{x}(1), \ldots, \mathbf{x}(KN)$)
    $Q_{1,M,L,K} \leftarrow Q_0$.
    for $n = 1 : N$ do
        for $k = 1 : K$ do
            $i \leftarrow (n-1)K + k$.
            $j \leftarrow 0$.
            while $d' < d$ do
                $j \leftarrow j + 1$.
                Generate $j$-th codeword $\mathbf{y}(j)$ using $Q_{n,M,L,K}$.
                $d' \leftarrow \rho\left(\mathbf{x}(i), \mathbf{y}(j)\right)$.
            end while
            $Q_k \leftarrow$ $M$-th order type of $\mathbf{y}(j)$.
        end for
        $Q_{n+1,M,L,K} \leftarrow \frac{1}{K} \sum_{k=1}^{K} Q_k$.
    end for
    return $Q_{N+1,M,L,K}$.
end procedure
```

This modified algorithm yields a sequence of reproduction distributions, i.e.,

$$Q_{n,M,L,K} = \frac{1}{K} \sum_{k=1}^{K} Q_k, \tag{3.3}$$

Table 3.1: Summary of the NTS Algorithm parameters' definitions.

n	NTS iteration index.
M	Memory depth or size of "super-symbol".
L	Number of (length M) super symbols encoded together.
K	Statistical depth for ML codebook distribution estimation.

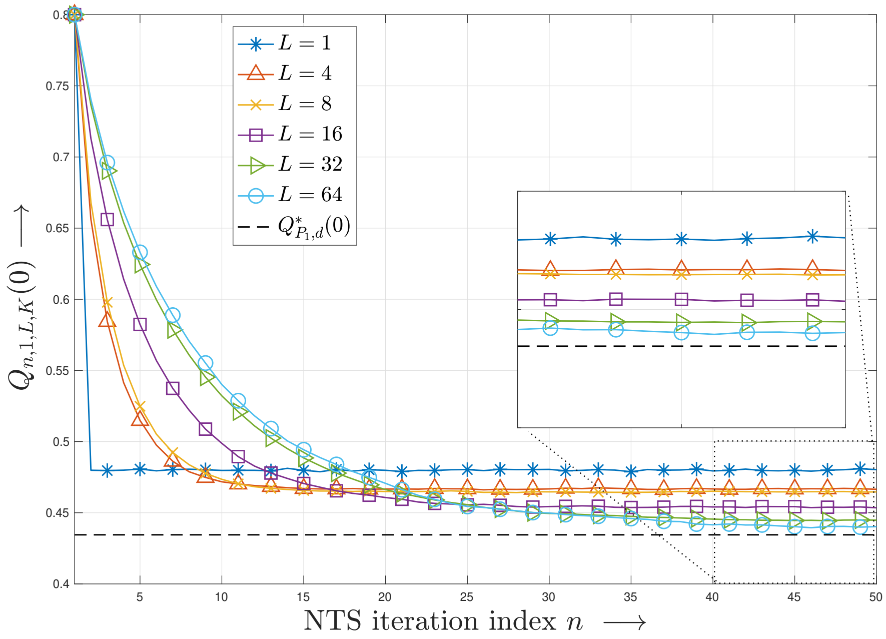


Figure 3.1: Evolution of the codebook reproduction distribution by the tractable NTS algorithm for different finite source lengths L and statistical depth $K = 10^5$. Binary memoryless source is considered with $P_1 = \{0.48, 0.52\}$, and Hamming distortion measure with $d = 0.35$. Reprinted, with permission, from [51] © 2020 IEEE.

$$Q_{n,M,L} = \lim_{K\to\infty} Q_{n,M,L,K}, \quad n = 1, 2, \ldots \tag{3.4}$$

For convenience, the main NTS parameters are summarized in Table 3.1. Before we dive into the convergence analysis of the modified NTS algorithm, we immediately verify the stability of the proposed modified NTS algorithm using the same example that was considered in Fig. 2.2, i.e., a discrete binary input and reproduction alphabet spaces ($\mathcal{X} = \mathcal{Y} = \{0, 1\}$). The source is assumed memoryless ($M = 1$) asymmetric with $P_1 = \{0.48, 0.52\}$. A Hamming distortion function is employed, i.e., the distortion between source word and codeword is the frequency of positions at which the corresponding source and code letters differ. The distortion constraint is set as $d = 0.35$. The statistical depth is set in the simulation environment to $K = 10^5$. For these settings, the optimal codebook reproduction distribution is $Q^*_{P_1,d} \approx \{0.44, 0.56\}$. Fig. 3.1 shows the evolution of the codebook reproduction distribution, i.e., $Q_{n,1,L}$, across the NTS iteration index n for different values of finite source word lengths L. As we see in Fig. 3.1 (for the same source and distortion as in Fig. 2.2), the use of $K > 1$ smooths out the large fluctuations present in the codebook reproduction distributions of the original NTS algorithm, and enhances the speed of convergence.

In the following analysis, we establish that the sequence of reproduction distributions of the modified NTS algorithm, despite the fact that it maintains a fixed and finite string length L, converges asymptotically, in probability, as $n \to \infty$ and $K \to \infty$ to the optimal *achievable* reproduction distribution $Q^*_{M,L}(P_M, d)$, where "achievable" reflects the limitations due to the fixed string length L, as will formally be stated in Theorem 2. For a more clear delivery of our main results in Theorems 1 and Theorem 2, we will consider only the memoryless case, i.e., $M = 1$. Generalization of these theorems to sources with memory will be stated in the Corollary 1 and Corollary 2.

First, define $\mathcal{U}_L(d)$ as the set of all possible pairs of L-length source words and codewords that can d-match, i.e.,

$$\mathcal{U}_L(d) \triangleq \left\{(\mathbf{x}, \mathbf{y}) : \mathbf{x} \in \mathcal{X}^L, \mathbf{y} \in \mathcal{Y}^L, \rho(\mathbf{x}, \mathbf{y}) \leq d\right\}. \tag{3.5}$$

Next, for memoryless source distribution $P = P_1$, define the set $E_L(P, d)$ as,

$$E_L(P,d) \triangleq \Big\{V : V(\mathbf{x},\mathbf{y}) \geq 0, V\left(\mathcal{X}^L, \mathcal{Y}^L\right) = 1, V(\mathbf{x},\mathbf{y}) = 0 \ \forall (\mathbf{x},\mathbf{y}) \notin \mathcal{U}_L(d), [V]_x = P^L\Big\}, \tag{3.6}$$

where P^L is the L-dimensional product distributions of P on $\mathcal{X}^L$. Note that the joint distributions in $E_L(P,d)$ are defined on the Cartesian product of L-fold product source and reproduction alphabet spaces, i.e., $\mathcal{X}^L \times \mathcal{Y}^L$, and hence, the joint distributions in $E_L(P,d)$ live in an r-dimensional simplex space $\mathcal{S}_r$ (with $r = |\mathcal{X}^L| \times |\mathcal{Y}^L|$), defined as,

$$\mathcal{S}_r = \left\{ \mathbf{z} = (z_1, \ldots, z_r) \in \mathbb{R}^r : \sum_i^r z_i = 1,\ z_i \geq 0, \forall i \in \{1, \ldots, r\} \right\}. \tag{3.7}$$

where $\mathbb{R}$ denotes the real line. We can now state our first Theorem.

Theorem 1 *For an initial codebook reproduction distribution Q_0 that is strictly positive everywhere over the discrete alphabet $\mathcal{Y}$, i.e., $Q(y) > 0, \forall y \in \mathcal{Y}$, and for distortion measure satisfying $0 \leq D_{\min} < D_{\text{av}} < \infty$, the reproduction distribution of the generalized recursive NTS algorithm in (3.3 converges asymptotically, as $K \to \infty$, given a fixed source word length L, a fixed super-symbol length $M = 1$, and iteration index $n-1$, to the marginal distribution of the L-dimensional distribution $Q^*_L(P, Q_{n-1,1,L}, d)$ in probability, i.e.,*

$$i)\quad Q_{n,1,L} = \lim_{K\to\infty} Q_{n,1,L,K} \to \mathbb{E}[Q_k], \tag{3.8}$$

where k enumerates the d-matching events ($k = 1, \ldots, K$), and Q_k is the M-th order type of the k-th d-matching codeword $\tilde{\mathbf{y}}(j_k)$, whose index in the random codebook is j_k. Furthermore, we show,

$$ii)\quad \mathbb{E}[Q_k] = Q^*_L(P, Q_{n-1,1,L}, d)^{\text{Marg.}}, \tag{3.9}$$

where the super script "Marg." denotes the marginal distribution of the L-dimensional distribution on the reproduction space $\mathcal{Y}$ (or more generally $\mathcal{Y}^M$ if $M > 1$ memory depth *is considered by the algorithm), and $Q_L^*(P, Q, D)$ is defined as follows,*

$$
\begin{aligned}
&V_L^*(P, Q, d) \triangleq \arg \min_{V \in E_L(P,d)} \mathcal{D}\left(V \middle|\middle| P^L \times Q^L\right), \\
&Q_L^*(P, Q, d) = \left[V_L^*(P, Q, d)\right]_y,
\end{aligned}
\tag{3.10}
$$

where the notation $[\cdot]_y$ denotes the y-marginal distribution of the argument.

It should be noted that, similar to P^L, Q^L is the L-dimensional product distributions of Q over $\mathcal{Y}^L$. The proof of Theorem 1 is provided in Appendix A.4. A brief proof sketch is that, first, it establishes that the sequence of d-matching codewords' types are independent and identically distributed, and thus by weak Law of Large Numbers (LLN), the ML type in (3.2) or the average type converges to the expected value in probability. Next, to show the second part of Theorem 1, we employ a variant of the conditional limit theorem in [45] to establish that, conditioned on the rare event that the joint input-output distribution of a block of K concatenated respective source and codewords $\left(\overline{\mathbf{X}}, \overline{\mathbf{Y}}\right)$ belongs to a convex set of distributions that generated d-matching source and code pairs with probability one, then the joint distribution of this block converges in probability, as $K \to \infty$, to the distribution $V_L^*(P, Q_{n-1,1,L}, d)$. This theorem has an intimate relationship with the *Gibbs Conditioning Principle* of statistical mechanics (see [56] and the references therein). The Gibbs conditioning principle roughly states: suppose that $\{X_1, \ldots, X_N\}$ are i.i.d. random variables distributed over a Polish space with marginal distribution P_X and a measurable function $f : \mathcal{X} \to \mathbb{R}$. Hence, under suitable conditions on P_X and $f(\cdot)$, and conditioned on the rare event that $\left\{\frac{1}{N}\sum_i f(X_i) \in [a - \delta, a + \delta]\right\}$, where $a \in \mathbb{R}$ and $\delta > 0$, the distribution of X_i converges in probability, as $N \to \infty$, to the distribution that minimizes the divergence $\mathcal{D}(\cdot||P_X)$ over all distributions that satisfy the constraint, which is very closely related to the arguments used to prove Theorem 1.

Theorem 1 implies that at every NTS iteration, the codebook reproduction distribution converges to a marginal of $Q_L^*(P, Q_{n-1,1,L}, d)$, as $K \to \infty$. Note that the codebook reproduction distribution in the next NTS iteration $Q_L^*(P_M, Q_{n-1,1,L}, d)^{\text{Marg.}}$ is more efficient in encoding the source than $Q_{n-1,M,L}$, the reproduction distribution in the current NTS iteration. This immediately suggests that the reproduction distributions improve by NTS iterations. Next we extend this result to sources with memory, for which the source and code words are now viewed as an L-length sequence of i.i.d. M-length "super symbols" on the alphabets $\mathcal{X}^{ML}$ and $\mathcal{Y}^{ML}$, respectively, where M is the considered memory depth.

Corollary 1 *For an initial codebook reproduction distribution Q_0 that is strictly positive everywhere over the discrete alphabet $\mathcal{Y}^M$, i.e., $Q(\mathbf{y}) > 0, \forall \mathbf{y} \in \mathcal{Y}^M$, and for distortion measure satisfying $0 \leq D_{\min} < D_{\text{av}} < \infty$, the reproduction distribution of the generalized recursive NTS algorithm in (3.3) converges asymptotically, as $K \to \infty$, given a fixed source word length L, a fixed super-symbol length $M \geq 1$, and iteration index $n-1$, to the marginal of the ML-dimensional distribution $Q_{M,L}^*(P, Q_{n-1,M,L}, d)$ in probability, i.e.,*

$$Q_{n,M,L} = Q_{M,L}^*(P_M, Q_{n-1,M,L}, d)^{\text{Marg.}}, \tag{3.11}$$

where $Q_{M,L}^*(P_M, Q_M, d)$ is defined similar to Theorem 1, i.e.,

$$\begin{aligned} V_{M,L}^*(P_M, Q_M, d) &\triangleq \arg \min_{V \in E_{M,L}(P_M, d)} \mathcal{D}\left(V \middle|\middle| P_M^L \times Q_M^L\right), \\ Q_{M,L}^*(P_M, Q_M, d) &= \left[V_{M,L}^*(P_M, Q_M, d)\right]_y, \end{aligned} \tag{3.12}$$

The distributions P_M^L and Q_M^L are the L-dimensional product distributions of P_M and Q_M over $\mathcal{X}^{ML}$ and $\mathcal{Y}^{ML}$, respectively. Additionally, the set $E_{M,L}(P_M, d)$ is the direct generalization of $E_L(P,d)$ for $M \geq 1$, i.e.,

$$\begin{aligned} E_{M,L}(d) \triangleq \Big\{ & V : V(\mathbf{x},\mathbf{y}) \geq 0, V\left(\mathcal{X}^{ML}, \mathcal{Y}^{ML}\right) = 1, \\ & V(\mathbf{x},\mathbf{y}) = 0 \ \forall (\mathbf{x},\mathbf{y}) \notin \mathcal{U}_{M,L}(d), [V]_x = P_M^L \Big\}, \end{aligned} \tag{3.13}$$

$$\mathcal{U}_{M,L}(d) \triangleq \left\{ (\mathbf{x},\mathbf{y}) : \mathbf{x} \in \mathcal{X}^{ML}, \mathbf{y} \in \mathcal{Y}^{ML}, \rho(\mathbf{x},\mathbf{y}) \leq d \right\}. \tag{3.14}$$

Proof: The proof of Corollary 1 follows directly from Theorem 1 by instead considering the source word as a sequence of i.i.d. M-length vectors, each distributed according to P_M. Hence, the new vector source will consists of "super-symbols" on the super-alphabet $\mathcal{X}^M$. Consequently, to accommodate such source containing a sequence of i.i.d. vectors, the code words are similarly constructed as a sequence of i.i.d. M-length vectors, each distributed according to Q_M over the reproduction super-alphabet space $\mathcal{Y}^M$. The proof follows similar to the proof of Theorem 1 under the new source and code words settings.

■

Next we establish the asymptotic optimality of the tractable NTS algorithm as the number of NTS iterations n, and the source words length L go to infinity. Similar to Theorem 1, for clarity of presentation, we first consider memoryless sources in Theorem 2, and then, we generalize our results to sources with memory in Corollary 2.

Theorem 2 *Given a memoryless source and a strictly positive initial codebook reproduction distribution Q_0 over $\mathcal{Y}$, i.e., $Q_0(y) > 0$, $\forall y \in \mathcal{Y}$ and for distortion measure satisfying $0 \leq D_{\min} < D_{\text{av}} < \infty$, the recursion in Algorithm 5, with $M = 1$, achieves,*

$$i) \qquad Q_{n,1,L} \to Q_L^*(P,d)^{\text{Marg.}}, \quad \text{as } n \to \infty \ , \tag{3.15}$$

$$ii) \qquad \begin{array}{c} \frac{1}{L} R_L(P,d) \to R(P,d) \\ Q_L^*(P,d)^{\text{Marg.}} \to Q_{P,d}^* \end{array}, \quad \text{as } L \to \infty \ , \tag{3.16}$$

where the subscript '1' stands for $M = 1$, $Q^*_{P,d}$ is the optimum reproduction distribution that achieves the rate distortion function $R(P, d)$, and $Q^*_L(P, d)^{\text{Marg.}}$ is the marginal distribution of the optimum achievable L-dimension reproduction distribution $Q^*_L(P, d)$ for finite source word length L that would asymptotically achieve the rate $R_L(P, d)$, i.e.,

$$\begin{aligned} &R_L(P, Q, d) \triangleq \min_{V \in E_L(P,d)} \mathcal{D}\left(V \;\middle|\middle|\; P^L \times Q^L\right), \\ &\mathcal{W}_L(P, d) \triangleq \left\{W\colon P^L \circ W = V, V \in E_L(P, d)\right\}, \\ &W^*_L(P, d) = W^*_L \triangleq \arg \min_{W \in \mathcal{W}_L(P,d)} I\left(P^L, W\right), \\ &R_L(P, d) \triangleq \min_Q R_L(P, Q, d) = I\left(P^L, W^*_L\right), \\ &Q^*_L(P, d) \triangleq \left[P^L \circ W^*_L(P, d)\right]_y, \end{aligned} \tag{3.17}$$

where $I\left(P^L, W\right)$ is the mutual information defined over the distributions, i.e.,

$$I\left(P^L, W\right) = \sum_{\tilde{\mathbf{x}} \in \mathcal{X}^L} \sum_{\tilde{\mathbf{y}} \in \mathcal{Y}^L} P^L(\tilde{\mathbf{x}})\ W(\tilde{\mathbf{y}}|\tilde{\mathbf{x}}) \log \frac{W(\tilde{\mathbf{y}}|\tilde{\mathbf{x}})}{Q(\tilde{\mathbf{y}})}, \tag{3.18}$$

$$Q(\tilde{\mathbf{y}}) = \sum_{\tilde{\mathbf{x}}' \in \mathcal{X}^L} P^L(\tilde{\mathbf{x}}')\ W(\tilde{\mathbf{y}}|\tilde{\mathbf{x}}'). \tag{3.19}$$

Note that the only difference between $R_L(P, Q, d)$ in (3.17) and $R(P, Q, d)$ in (2.8), with $M = 1$, is the set over which the min operation is taken as well as the dimension of the distributions L. The convex set $E_L(P, d)$, which is constrained by the finite source word length L, is a subset of the set $\{V : [V]_x = P^L,\ \mathbb{E}_V(\rho(\mathbf{X}, \mathbf{Y})) \leq d\}$. Additionally, we show in Theorem 2 that these two sets are equivalent asymptotically as $L \to \infty$. The proof of Theorem 2 is provided in Appendix A.5.

Now we turn our attention to the case for which the source is with memory. Consequently, in order to capture the correlation between source samples, the source words and code words are constructed by concatenating a sequence of L i.i.d. M-length sub vectors according to P_M on $\mathcal{X}^M$ and Q_M on $\mathcal{Y}^M$, respectively, where M is the memory depth considered.

Corollary 2 *For an initial codebook reproduction distribution Q_0 that is strictly positive everywhere over the discrete alphabet $\mathcal{Y}^M$, i.e., $Q(\mathbf{y}) > 0, \forall \mathbf{y} \in \mathcal{Y}^M$, and for a distortion measure satisfying $0 \leq D_{\min} < D_{\text{av}} < \infty$, the recursion in Algorithm (5), with $M \geq 1$, achieves,*

$$i) \qquad Q_{n,M,L} \to Q^*_{M,L}(P_M, d)^{\text{Marg.}}, \quad \text{as } n \to \infty \ , \tag{3.20}$$

$$ii) \qquad \begin{array}{l} \frac{1}{L} R_{M,L}(P_M, d) \to R(P_M, d) \\ Q^*_{M,L}(P_M, d)^{\text{Marg.}} \to Q^*_{P_M,d} \end{array}, \quad \text{as } L \to \infty \ , \tag{3.21}$$

where $Q^*_{P_M,d}$ is the optimum reproduction distribution that achieves the joint M-th order rate distortion function $R(P_M, d)$, and $Q^*_{M,L}(P_M, d)^{\text{Marg.}}$ is the marginal distribution of the optimum achievable ML-dimension reproduction distribution $Q^*_{M,L}(P_M, d)$, for finite source word length L and sub-vector length M, that would asymptotically achieve the rate $R_{M,L}(P_M, d)$, i.e.,

$$\begin{aligned} &R_{M,L}(P_M, Q_M, d) \triangleq \min_{V \in E_{M,L}(P_M,d)} \mathcal{D}\left(V \mid\mid P_M^L \times Q_M^L\right), \\ &\mathcal{W}_{M,L}(P_M, d) \triangleq \left\{W \colon P_M^L \circ W = V, V \in E_{M,L}(P_M, d)\right\}, \\ &W^*_{M,L}(P_M, d) = W^*_{M,L} \triangleq \arg \min_{W \in \mathcal{W}_{M,L}(P_M,d)} I\left(P_M^L, W\right), \\ &R_{M,L}(P_M, d) \triangleq \min_{Q} R_{M,L}(P_M, Q_M, d) = I\left(P_M^L, W^*_{M,L}\right), \\ &Q^*_{M,L}(P_M, d) \triangleq \left[P_M^L \circ W^*_{M,L}(P_M, d)\right]_y , \end{aligned} \tag{3.22}$$

where $I\left(P_M^L, W\right)$ is the mutual information defined over the distributions.

Proof: Similar to Corollary 1, the proof of Corollary 2 follows directly from Theorem 2 by instead considering the source word as a sequence of i.i.d. M-length sub-vectors or super-symbols, each distributed according to P_M. Hence, the new "vector" source will consists of "super-symbols" on the super-alphabet $\mathcal{X}^M$. Consequently, to accommodate such source containing a sequence of i.i.d. vectors, the code words are similarly constructed as a sequence of i.i.d. M-length sub-vectors or super-symbols, each distributed according to Q_M over the reproduction super-alphabet space $\mathcal{Y}^M$. Hence, the proof follows similar to the proof of Theorem 2 under the new source and code words settings. ■

Additionally, by (2.12) and (2.13), for sources with memory, the proposed NTS algorithm can further find Q_d^*, that achieves $R(d)$, if the memory depth M (or the size of the "super symbol") is sent to infinity.

Corollary 3 *Among all possible finite-length codebook reproduction distributions that generate i.i.d. M-length super symbols, and that induce a joint distribution generating d-matching source and code pairs with probability one, the tractable NTS algorithm finds the codebook reproduction distribution that minimizes the divergence between the input-output joint distribution, and the product of their marginal distributions, asymptotically in K and n, i.e.,*

$$V^*_{M,L}(P_M, d) = \arg \min_{V \in E_{M,L}(P_M,d)} I(V, P_M^L \times Q_M^L), \tag{3.23}$$

$$I(V, P_M^L \times Q_M^L) = \sum_{\tilde{\mathbf{x}} \in \mathcal{X}^{ML}} \sum_{\tilde{\mathbf{y}} \in \mathcal{Y}^{ML}} V(\tilde{\mathbf{x}}, \tilde{\mathbf{y}}) \log \frac{V(\tilde{\mathbf{x}}, \tilde{\mathbf{y}})}{P_M^L(\tilde{\mathbf{x}}) Q_M^L(\tilde{\mathbf{y}})}, \tag{3.24}$$

$$Q_M(\mathbf{y}) = \sum_{\tilde{\mathbf{y}}' \in \mathcal{Y}^{ML}} N(\mathbf{y}|\tilde{\mathbf{y}}') \sum_{\tilde{\mathbf{x}}' \in \mathcal{X}^{ML}} V(\tilde{\mathbf{x}}', \tilde{\mathbf{y}}), \quad \forall \mathbf{y} \in \mathcal{Y}^M, \tag{3.25}$$

where $N(\mathbf{y}|\tilde{\mathbf{y}})$ is the frequency of occurrence of $\mathbf{y}$ as seen in $\tilde{\mathbf{y}}$, hence, Q_M is the y-marginal of V on $\mathcal{Y}^M$. In other words, out of all the possible finite-length codebook reproduction distributions that induce joint distributions generating d-matching ML-length *source and code pairs with probability one, the NTS algorithm finds the distribution that minimizes the encoding rate when the codebook reproduction distribution is used with asymptotic-length encoding settings. This also implies that the NTS algorithm with finite length settings finds the distribution that achieves the rate-distortion function, albeit for a max-distortion constraint d over every ML-length segment.*

Proof: An optimal finite-length codebook $\mathcal{C}^*(P_M, d)$, given a distortion constraint d, will only generate the codewords that can possibly d-match finite-length source examples. Hence, the optimal codebook reproduction distribution belongs to the convex hull of all possible codewords that can exist in $\mathcal{C}^*(P_M, d)$. This set of distributions is exactly $\{Q_M^L : [V]_y = Q_M, V \in E_{M,L}(P_M, d)\}$, and the NTS algorithm finds the minimizing joint distribution, in $E_{M,L}(P_M, d)$, to the mutual information by (3.17) and (3.23). ■

While ensuring optimality, the NTS algorithm for source with memory in [46] suffers from fundamental practical flaws. In order to converge to the optimal distribution that achieves the rate-distortion bound for sources with memory, the algorithm needs to encode source words that are composed of i.i.d. M-length vectors according to the M-th order source joint distribution P_M, while sending M to infinity. This obviously requires the prior knowledge of the source statistics in order to artificially generate such vectors with i.i.d. constraints, which is elusive in practical cases. Furthermore, even for finite-memory sources such as sources with finite order Markovian property, the algorithm requires sending the length of i.i.d. source and code vectors M, within the L-length codeword, to infinity in order to converge to the optimal. It is important to note that sending M to infinity implies that the cardinalities of the source and code super alphabet spaces increase beyond any practical computational power available to perform d-search, ML estimation and codebook regeneration operations, thus rendering the system unfeasible in practical scenarios. The requirement of sending M to infinity is also counter-intuitive, specifically for sources with finite memory, i.e., sources for which the current sample distribution only depends on a subset of the past samples. In the next section, we propose to modify the NTS algorithm for Markovian sources, such that the algorithm converges to its optimal distribution without sending M to infinity. More specifically, we restrict the generating codebook distribution to distributions with M-th order Markovian property, which is the same Markovian property order as the source. Then, asymptotic convergence to the optimal constrained distribution, i.e., the distribution that achieves the minimum per letter encoding rate over all codebook distributions within the constrained family of Markovian distributions, is shown.

3.2 Sources with Markovian Property

In this section, we restrict our attention to stationary and ergodic sources with memory, described by the Markovian property, over discrete alphabet space $\mathcal{X}$. Denote the M-th order Markov source as a source with M-length Markov property, i.e., for any time index $n \geq M > 0$,

$$\begin{aligned}\mathbb{P}(X_n = x_n \mid X_{n-1} = x_{n-1}, X_{n-2} = x_{n-2}, \ldots, X_0 = x_0) = \\ \mathbb{P}(X_n = x_n \mid X_{n-1} = x_{n-1}, X_{n-2} = x_{n-2}, \ldots, X_{n-M} = x_{n-M}).\end{aligned} \tag{3.26}$$

This implies that the current source sample distribution only depends on the previous M-samples. This Markov source can be described by a state transition diagram containing exactly $|\mathcal{X}|^M$ states. Let $P_{j|i}$ be the homogenous source state transition probability from state i to state j, where $i, j \in \mathcal{R} = \mathcal{X}^M$. Hence, let $\mathbf{P}$ be the state transition probability matrix for which the entry in the i-th row and j-th column is $P_{j|i}$. Furthermore, let $P(X|\mathbf{x}) = \{P(x|\mathbf{x}) : x \in \mathcal{X}\}$ be the stationary source letter distribution conditioned on the M previous samples being $\mathbf{x}$. Note that there exists a one-to-one mapping between the set $\{P_{j|i}, \forall (i,j) \in \mathcal{R}^2\}$ and the set $\{P(x|\mathbf{x}), \forall x \in \mathcal{X}, \forall \mathbf{x} \in \mathcal{X}^M\}$. By the stationary assumption of the Markov chain, the stationary distribution is computed as, $\mathbf{\Pi} = \mathbf{\Pi}\ \mathbf{P}$, where $\mathbf{\Pi} = [\pi(1), \ldots, \pi(|\mathcal{R}|)]$ is a row vector. In order to take into account the Markovian property of the source, we restrict the codebook reproduction distribution to distributions with M-th order Markov property. Let $Q_{j|i}$ be the codebook distribution state transition probability from state i to state j, where $i \in \mathcal{S}, j \in \mathcal{S}$, and $\mathcal{S} = \mathcal{Y}^M$. Hence, let $\mathbf{Q}$ be the state transition probability matrix for which the entry in the i-th row and j-th column is $Q_{j|i}$. Let the random L-tuples source words and codewords $\mathbf{X} = [X_1, \ldots, X_L]$, and $\mathbf{Y} = [Y_1, \ldots, Y_L]$, be generated according to state transition matrices $\mathbf{P}$ and $\mathbf{Q}$, respectively. First, we introduce a variant of NTS algorithm for the above setup. At every NTS iteration with index n, the algorithm finds a set of d-matching codewords in the random codebook to a set of K independently generated source words. Let the realizations of the d-matching source and code sets be denoted as $\{\mathbf{x}(i_1), \ldots, \mathbf{x}(i_K)\}$, and $\{\mathbf{y}(j_1), \ldots, \mathbf{y}(j_K)\}$, where j_k is the index of the codeword that d-match the k-th source word in the codebook. Next, similar to before, the NTS algorithm finds the most likely constrained reproduction distribution to produce the set of d-matching codewords, where the distribution is constrained to have M-th order Markov property.

Lemma 3 [57]: The ML estimate of the M-th order Markov process state transition probabilities underlying the codebook reproduction distribution, given a set of K d-matching codewords, is the *average* of the d-matching codewords' transitions, i.e.,

$$\mathbf{Q}_{n+1,M,L,K} = \mathbf{Q}^{\mathrm{ML}} = \left\{ Q_{j|i} : Q_{j|i} = \frac{\sum\limits_{k=1}^{K} N(i \to j|\mathbf{y}(k))}{\sum\limits_{k=1}^{K} \sum\limits_{j' \in \mathcal{S}} N(i \to j'|\mathbf{y}(k))}, \quad \forall (i,j) \in \mathcal{S}^2 \right\}, \tag{3.27}$$

where k enumerates the d-matching events, and $N(j \to i|\mathbf{y}(j_k))$ is the number of transitions from state i to state j as seen in the k-th d-matching codeword $\mathbf{y}(j_k)$, whose index in the random codebook is j_k. The proof of Lemma 3, originally written in [57], is given in A.3. Thus, this algorithm yields a sequence of state transition matrices as in (3.27), or equivalently, a sequence of conditional distributions $Q_{n+1,M,L,K}(Y|\mathbf{y})$. Hence, the iterative NTS algorithm for Markovian sources over discrete alphabet spaces can be summarized in Algorithm 6.

Algorithm 6 : Modified NTS Algorithm for Markovian Sources over Discrete Alphabets

```
procedure NTS_MARKOV_DISCRETE($N, M, L, K, d, \mathbf{Q}_0, \mathbf{x}(1), \ldots, \mathbf{x}(KN)$)
    $\mathbf{Q}_{1,M,L,K} \leftarrow \mathbf{Q}_0$.
    for $n = 1 : N$ do
        for $k = 1 : K$ do
            $i \leftarrow (n-1)K + k$.
            $j \leftarrow 0$.
            while $d' < d$ do
                $j \leftarrow j + 1$.
                Generate $j$-th codeword $\mathbf{y}(j)$ using state transition matrix $\mathbf{Q}_{n,M,L,K}$.
                $d' \leftarrow \rho\left(\mathbf{x}(i), \mathbf{y}(j)\right)$.
            end while
            Record $N_k^{(i \to j)} \leftarrow N(i \to j|\mathbf{y}(j)), \quad \forall (i,j) \in \mathcal{S}^2$.
        end for
        $\mathbf{Q}_{n+1,M,L,K} \leftarrow \left\{ Q_{j|i} : Q_{j|i} = \frac{\sum_{k=1}^{K} N_k^{(i \to j)}}{\sum_{k=1}^{K} \sum_{j' \in \mathcal{S}} N_k^{(i \to j')}}, \quad \forall (i,j) \in \mathcal{S}^2 \right\}$.
    end for
    return $\mathbf{Q}_{N+1,M,L,K}$.
end procedure
```

In the next discussion, we quantify the asymptotic performance of the NTS algorithm specialized for Markov sources. Let the random codebook be generated according to a Markov process with conditional probabilities $Q(Y|\mathbf{y})$, $\forall \mathbf{y} \in \mathcal{Y}^M$, i.e., $Q(Y|\mathbf{y})$ is the row in the matrix $\mathbf{Q}$ that corresponds to transitions from state $s = \mathbf{y} \in \mathcal{S}$. Before we characterize the asymptotic performance of such algorithm, we start by transforming this variant of NTS algorithm into a dual set of NTS algorithms for *memoryless sources.* Let the sets of K d-matching source words and codewords be concatenated into KL-length source and code blocks denoted as, $\mathbf{s} = [\mathbf{x}(i_1), \ldots, \mathbf{x}(i_K)]$, and $\mathbf{c} = [\mathbf{y}(j_1), \ldots, \mathbf{y}(j_K)]$, respectively. Furthermore, let the source and code blocks be independently divided into sub-streams based on the previous source and code M-tuples, denoted as $\{\mathbf{s}_{\mathbf{x}}, \forall \mathbf{x} \in \mathcal{X}^M\}$, and $\{\mathbf{c}_{\mathbf{y}}, \forall \mathbf{y} \in \mathcal{Y}^M\}$. Note that by the homogenous assumption of the Markovian source and code distributions, the sequence of symbols of the sub-streams $\{\mathbf{s}_{\mathbf{x}}\}$ are i.i.d. according to $P(X|\mathbf{x})$. The set of d-match event $\{\rho(\mathbf{x}(i_k), \mathbf{y}(i_k)) \leq d, \;\; \forall k\}$, implies that $\rho(\mathbf{s}, \mathbf{c}) \leq d$, which is equivalent to a set of size $|\mathcal{X}^M \times \mathcal{Y}^M|$ events of distortion matches between the sub-streams $\{\mathbf{s}_{\mathbf{x}}\}$ and $\{\mathbf{c}_{\mathbf{y}}\}$, each with distortion level denoted as $d_{\mathbf{x},\mathbf{y}}$, such that, $\sum\limits_{\mathbf{x},\mathbf{y}} \mathbb{M}_n(\mathbf{x}, \mathbf{y}) d_{\mathbf{x},\mathbf{y}} \leq d$. Here $\mathbb{M}_n(\mathbf{x}, \mathbf{y})$, is the empirical probability of reproducing a letter in sub stream $\mathbf{s}_{\mathbf{x}}$, by a letter in sub stream $\mathbf{c}_{\mathbf{y}}$, at NTS iteration n, as seen by the code and source blocks $\mathbf{s}$, and $\mathbf{c}$, respectively. To illustrate this idea further, consider Fig 3.2 showing binary source and code blocks, which are formed by concatenating three d-matching source and code words. The source and code generating distributions exhibit a first order Markovian property, hence the number of Markov states is $|\mathcal{X}| = |\mathcal{Y}| = 2$, furthermore, a Hamming distortion measure is employed at distortion level $d = 1/3$. The letters that are divided into different i.i.d. sub-streams are assigned different colors. For example, the samples that follow '0' are assigned black color, and the samples following '1' are assigned blue color. The i.i.d sub-streams $\mathbf{s}_{\mathbf{x}}$ and $\mathbf{c}_{\mathbf{y}}$ are formed by concatenating all the samples following the same letter together as shown in Fig. 3.2.

Theorem 3 *For an initial codebook generating Markov chain with strictly positive transition probabilities $Q(Y|\mathbf{y}) > 0$, $\forall \mathbf{y} \in \mathcal{S} = \mathcal{Y}^M$, and distortion measure satisfying $0 \leq D_{\min} < D_{\mathrm{av}} < \infty$, the transition probabilities $Q(Y|\mathbf{y})$, of the recursive NTS algorithm for Markov sources, where each recursion involves collecting K d-matches, converge in probability and asymptotically, as $L \to \infty$, as follows,*

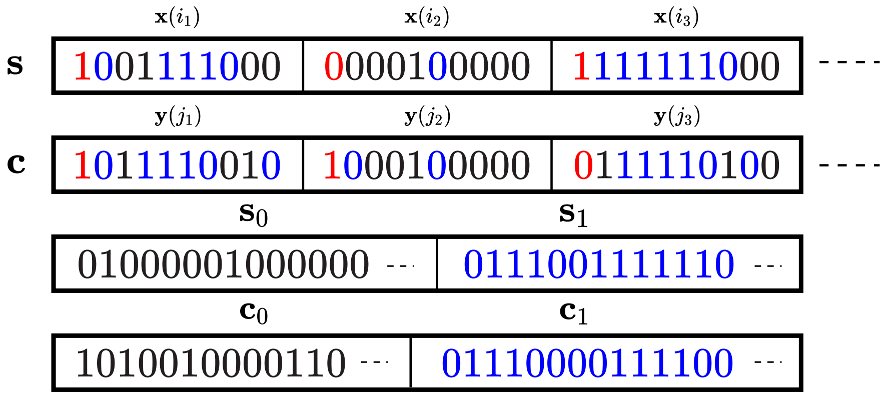


Figure 3.2: Division of the d-matching source and code blocks into i.i.d. sub-streams based on the previous sample. Reprinted, with permission, from [53] © 2023 IEEE.

$$\begin{aligned}
&Q_{n+1,M,K}(Y|\mathbf{y}) \rightarrow \sum_{\mathbf{x}\in\mathcal{X}^M} \mathbb{M}_n^*(\mathbf{x}|\mathbf{y}) Q^*\left(P(X|\mathbf{x}), Q_{n,M,K}(Y|\mathbf{y}), d^*_{\mathbf{x},\mathbf{y}}\right),\\
&V^*\left(P(X|\mathbf{x}), Q(Y|\mathbf{y}), d^*_{\mathbf{x},\mathbf{y}}\right) \triangleq \arg \min_{V\in E_{\mathbf{x},\mathbf{y}}(d^*_{\mathbf{x},\mathbf{y}})} \mathcal{D}\left(V \middle|\middle| P(X|\mathbf{x})\times Q(Y|\mathbf{y})\right),\\
&Q^*\left(P(X|\mathbf{x}), Q(Y|\mathbf{y}), d^*_{\mathbf{x},\mathbf{y}}\right) = \left[V^*\left(P(X|\mathbf{x}), Q(Y|\mathbf{y}), d^*_{\mathbf{x},\mathbf{y}}\right)\right]_y,
\end{aligned} \tag{3.28}$$

where $Q_{n+1,M,K}(Y|\mathbf{y}) = \lim_{L\to\infty} Q_{n+1,M,L,K}(Y|\mathbf{y})$*, and the set* $E_{\mathbf{x},\mathbf{y}}(d_{\mathbf{x},\mathbf{y}})$ *is defined as,*

$$E_{\mathbf{x},\mathbf{y}}(d^*_{\mathbf{x},\mathbf{y}}) = \left\{V : V = P'\circ W', P' = P(X|\mathbf{x}),\ \rho(P',W') \le d^*_{\mathbf{x},\mathbf{y}}\right\}. \tag{3.29}$$

Here $\rho(P',W')$ *is the average distortion computed over distributions, and the set of distortion levels* $\{d^*_{\mathbf{x},\mathbf{y}}, \forall(\mathbf{x},\mathbf{y}) \in \mathcal{X}^M \times \mathcal{Y}^M\}$*, satisfies,*

$$\frac{\partial}{\partial\delta} R(P(X|\mathbf{x}), Q(Y|\mathbf{y}), \delta)\Big|_{\delta=d^*_{\mathbf{x},\mathbf{y}}} = R'_{P,Q,d},\ \forall(\mathbf{x},\mathbf{y}), \quad \sum_{\mathbf{x},\mathbf{y}} \mathbb{M}_n^*(\mathbf{x},\mathbf{y}) d^*_{\mathbf{x},\mathbf{y}} \le d, \tag{3.30}$$

where $R'_{P,Q,d}$ *is independent of the sub-stream pair* $(\mathbf{x},\mathbf{y})$*.*

In other words, the distortion allocation to sub-stream pairs, $d^*_{\mathbf{x},\mathbf{y}}$, ensures they all maintain the same rate-distortion slope, given codebook generating distributions $\{Q(Y|\mathbf{y})\}$, while satisfying the overall average distortion constraint d. The proof of Theorem 3 is given in Appendix A.6. Next, we look at the asymptotic convergence of the codebook reproduction conditional distributions as the number of iterations n goes to infinity.

Theorem 4 *Given an initial codebook that is generated using a Markov process with strictly positive conditional distributions $Q(Y|\mathbf{y})$ for any state $\mathbf{y} \in \mathcal{S} = \mathcal{Y}^M$, the recursion in (3.27) achieves the minimum average coding rate over the cross product of all source-code sub streams, denoted as $\overline{R}(d)$, i.e.,*

$$\overline{R}(d) = \min_{Q(Y|\mathbf{y})} \min_{\substack{\mathbb{M}(\mathbf{y}|\mathbf{x}) \\ d_{\mathbf{x},\mathbf{y}}, V_{\mathbf{x},\mathbf{y}}}} \sum_{\mathbf{x},\mathbf{y}} \mathbb{M}(\mathbf{x})\mathbb{M}(\mathbf{y}|\mathbf{x})\ \mathcal{D}\left(V_{\mathbf{x},\mathbf{y}}\ \big|\big|\ P(X|\mathbf{x}) \times Q(Y|\mathbf{y})\right). \tag{3.31}$$

and the set of optimization variables that achieves the minimum in (3.31), satisfies,

$$\frac{\partial}{\partial \delta} R\left(P(X|\mathbf{x}), Q^*(Y|\mathbf{y}), \delta\right)\Big|_{\delta = d^*_{\mathbf{x},\mathbf{y}}} = R'_{P,Q^*,d}, \ \sum_{\mathbf{x},\mathbf{y}} \mathbb{M}^*(\mathbf{x},\mathbf{y}) d^*_{\mathbf{x},\mathbf{y}} \leq d, \tag{3.32}$$

where $R'_{P,Q^*,d}$ is independent of the sub-stream pair $(\mathbf{x}, \mathbf{y})$.

The proof of Theorem 4 is provided in Appendix A.7. This establishes that the NTS algorithm finds the conditional distributions that minimize the average encoding rate over all i.i.d. source and code cross sub-streams $\{\mathbf{s}_{\mathbf{x}} \times \mathbf{c}_{\mathbf{y}}\}$ while maintaining the distortion level d, hence implying asymptotic optimality.

In the next section, we generalize the proposed NTS algorithm to sources with abstract alphabet spaces, which are more prominent in practical applications. The preliminary results of this generalization have first appeared in [52].

3.3 Abstract Alphabet Spaces

While earlier sections focused on discrete alphabet sources, the prevalence of continuous alphabet sources in practical compression applications provides strong motivation for generalization of the NTS algorithm to accommodate these sources. It is important to emphasize that the standard concept of types, which was the cornerstone of the aforementioned NTS work on discrete alphabet sources, and was specifically instrumental to showing asymptotic convergence to the reconstruction distribution that achieves the rate-distortion bound, does not apply to continuous alphabet sources. Hence, the generalization of the NTS algorithm to continuous alphabet sources is not straightforward and is, in fact, fundamentally more challenging. In order to circumvent this issue, we start working with probability measures over abstract alphabet spaces rather than the method of types. Important advances were made in [41], which studied abstract alphabet spaces in the random codebook coding context of plain and entropy-constrained quantization, and further generalized the conditional limit theorem, which is at the heart of the NTS algorithm, to stationary ergodic sources with abstract alphabet spaces. We assume that the alphabet $\mathcal{X}$ is a complete separable metric space (often called Polish space), equipped with its associated Borel σ-field $\mathcal{X}'$. Similarly, we assume that the reproduction alphabet $\mathcal{Y}$ is also a Polish space equipped with its associated Borel σ-field $\mathcal{Y}'$. Similar to the tractable NTS algorithm for discrete sources in Section 3.1, a sequence of K d-matching events between independent source examples, $\{\mathbf{x}(i_1), \dots, \mathbf{x}(i_K)\}$, and codewords $\{\mathbf{y}(j_1), \dots, \mathbf{y}(j_K)\}$ is observed before estimating the maximum likelihood codebook reproduction distribution. Note that matching codewords are indexed in the codebook, as reflected in the notation j_k for the index of the k-th matching codeword. We turn our attention to the source block and the code block that are formed by concatenating the source words and the d-matching codewords, i.e., $\overline{\mathbf{x}} = (\mathbf{x}(i_1), \dots, \mathbf{x}(i_K))$, and $\overline{\mathbf{y}} = (\mathbf{y}(j_1), \dots, \mathbf{y}(j_K))$. Note that, obviously, $\forall k \in \{1, \dots, K\}, \rho(\mathbf{x}(i_k), \mathbf{y}(j_k)) \leq d$. Hence, the source block $\overline{\mathbf{x}}$ and code block $\overline{\mathbf{y}}$ satisfies a stricter distortion requirement, due to the inherent maximum distortion constraint over sub-blocks. In order to capture such stricter distortion requirement, we define a scalar-valued auxiliary distortion function as follows: $\left(\rho^{(d)} : \mathcal{X}^{ML} \times \mathcal{Y}^{ML} \to \{0, 1\}\right)$, which is additive across the K ML-length sub-blocks, i.e.,

$$\rho^{(d)}\left(\mathbf{x}(i_k),\mathbf{y}(j_k)\right)=\begin{cases}0 & \text{if } \rho\left(\mathbf{x}(i_k),\mathbf{y}(j_k)\right)\leq d,\\ 1 & \text{otherwise}\end{cases} \tag{3.33}$$

$$\rho^{(d)}(\overline{\mathbf{x}},\overline{\mathbf{y}})=\frac{1}{K}\sum_{k=1}^{K}\rho^{(d)}\left(\mathbf{x}(i_k),\mathbf{y}(j_k)\right), \tag{3.34}$$

It should be noted that by setting $\rho^{(d)}(\overline{\mathbf{x}},\overline{\mathbf{y}})=0$, we impose a requirement of maximum distortion d per sub-block, over the K sub-blocks. Thus, the auxiliary distortion measure $\rho^{(d)}$ is a subterfuge to impose maximum distortion while maintaining the additive property over the K sub-blocks. Next, in view of (3.2), the next iteration codebook reproduction distribution is computed as the average of the codeword *empirical distributions*, i.e.,

$$Q_{n+1,M,L,K}=\frac{1}{K}\sum_{k=1}^{K}Q_{\mathbf{y}(j_k)}, \tag{3.35}$$

$$Q_{n+1,M,L}=\lim_{K\to\infty}Q_{n+1,M,L,K}, \tag{3.36}$$

$$Q_{n+1,M}=\lim_{L\to\infty}Q_{n+1,M,L}, \tag{3.37}$$

where $Q_{\mathbf{y}(j_k)}$ is the empirical distribution of the k-th d-matching codeword $\mathbf{y}(j_k)$ on $\mathcal{Y}^{ML}$,i.e., (3.35) can be rewritten as

$$Q_{n+1,M,L,K}=\frac{1}{K}\sum_{k=1}^{K}\delta_{\mathbf{y}(j_k)}, \tag{3.38}$$

with $\delta_{\mathbf{y}(j_k)}$ denoting a Dirac measure located at $\mathbf{y}(j_k)$. The modified NTS algorithm for sources with abstract alphabets is summarized in Algorithm 7.

Theorem 5 *Let a random codebook be generated with $Q_M = Q_{n-1,M,L}$ having strictly positive density everywhere on $\mathcal{Y}^{ML}$, and assume the auxiliary distortion measure $\rho^{(d)}$ satisfies $0 \leq D_{\min} < D_{\text{av}} < \infty$, then the probability measure $Q_{n,M,L,K}$ on $\mathcal{Y}^{ML}$ converges weakly almost surely, as K goes to infinity, to the optimal distribution $Q^*_{P^L_M,Q^L_M,\gamma}$ that*

Algorithm 7 : Modified NTS Algorithm for Abstract Alphabets

```
procedure NTS_Modified_Abstract($N, M, L, K, d, Q_0, \mathbf{x}(1), \dots, \mathbf{x}(KN)$)
    $Q_{1,M,L,K} \leftarrow Q_0$.
    for $n = 1 : N$ do
        for $k = 1 : K$ do
            $i \leftarrow (n-1)K + k$.
            $j \leftarrow 0$.
            while $d' < d$ do
                $j \leftarrow j + 1$.
                Generate $j$-th codeword $\mathbf{y}(j)$ using $Q_{n,M,L,K}$.
                $d' \leftarrow \rho\left(\mathbf{x}(i), \mathbf{y}(j)\right)$.
            end while
            $j_k \leftarrow j$.
        end for
        $Q_{n+1,M,L,K} \leftarrow \frac{1}{K} \sum_{k=1}^{K} \delta_{\mathbf{y}(j_k)}$.
    end for
    return $Q_{N+1,M,L,K}$.
end procedure
```

achieves the bound $R(P_M^L, Q_M^L, \gamma)$, for the auxiliary distortion measure $\rho^{(d)}(\cdot)$, with the extreme distortion constraint $\gamma = 0$, i.e.,

$$Q_{n,M,L,K} \Longrightarrow Q^*_{P_M^L, Q_M^L, \gamma}, \ \gamma = 0, \quad \text{as} \ \ K \to \infty, \tag{3.39}$$

$$Q^*_{P_M^L, Q_M^L, \gamma} = \arg\min_{Q_M^{L'}} \left\{ I_{\min}\left(P_M^L || Q_M^{L'}, \gamma\right) + \mathcal{D}\left(Q_M^{L'} || Q_M^L\right) \right\}, \tag{3.40}$$

$$I_{\min}\left(P_M^L || Q_M^{L'}, \gamma\right) = \inf_{\substack{V:[V]_x = P_M^L, \ [V]_y = Q_M^{L'}, \\ \mathbb{E}_V\left(\rho^{(d)}(\tilde{\mathbf{X}}, \tilde{\mathbf{Y}})\right) \leq \gamma}} I\left(\tilde{\mathbf{X}}, \tilde{\mathbf{Y}}\right), \tag{3.41}$$

where "$\Longrightarrow$" denotes weak convergence of random probability measures almost surely, and the probability measures P_M^L, and Q_M^L denotes the L-product probability measures of P_M and Q_M, respectively, on $\mathcal{X}^{ML}$ and $\mathcal{Y}^{ML}$.

The proof of Theorem 5 is provided in Appendix A.8. Note that $Q^*_{P_M^L, Q_M^L, \gamma}$ is more efficient in encoding than $Q_{n-1,M,L}$, which immediately suggests that the algorithm is improving in the rate-distortion sense at every NTS iteration. Alternating minimization over convex sets arguments can be further invoked to show asymptotic optimality of the NTS algorithm as n goes to infinity, as has been shown in Theorem 2 for discrete alphabet sources.

Theorem 6 *For an initial distribution Q_0 having strictly positive density everywhere over $\mathcal{Y}^{ML}$ and for a distortion measure satisfying $0 \leq D_{\min} < D_{\text{av}} < \infty$, the recursion in Algorithm 7 achieves,*

$$Q_{n,M,L} \Longrightarrow Q^*_{P^L_M,\gamma}, \quad \textit{for } \gamma = 0, \quad \textit{as } n \to \infty \ , \tag{3.42}$$

where $Q^*_{P^L_M,\gamma}$ is the optimal reproduction distribution that achieves the rate distortion function $R(P^L_M, \gamma)$ for the auxiliary distortion measure $\rho^{(d)}(\cdot)$, i.e.,

$$R\left(P^L_M, \gamma\right) = \inf_{Q^L_M} \inf_{\substack{V:[V]_x = P^L_M, \\ \mathbb{E}_V\left(\rho^{(d)}(\tilde{\mathbf{X}},\tilde{\mathbf{Y}})\right) \leq \gamma}} \mathcal{D}\left(V || P^L_M \times Q^L_M\right), \tag{3.43}$$

such that the inner infimum is taken over all joint distributions V of the random vectors $(\tilde{\mathbf{X}}, \tilde{\mathbf{Y}})$ such that the x-marginal of V is P^L_M, and the expected distortion $\mathbb{E}_V \left(\rho^{(d)}(\tilde{\mathbf{X}}, \tilde{\mathbf{Y}})\right) \leq \gamma$.

The proof of Theorem 6 is provided in Appendix A.9. Next, we show the asymptotic performance of the tractable NTS algorithm for asymptotic length source words over abstract alphabet spaces.

Theorem 7 *Assuming the same conditions as in Theorem 5 and Theorem 6, the marginal probability measure of $Q_{n,M,L}$ on $\mathcal{Y}^M$, denoted by $Q^{\text{Marg.}}_{n,M,L}$, converges in the weak convergence sense to the optimal probability measure $Q^*_{P_M,d}$ that achieves the M-th order rate-distortion function $R(P_M, d)$ as L and n go to infinity for the original distortion measure*

$\rho(\cdot)$.

It is worth noting that by (2.12) and (2.13), for sources with memory, the proposed NTS algorithm can further find Q_d^*, which achieves $R(d)$, if the memory depth M (or the size of the "super symbol") is sent to infinity. Appendix A.10 provides the detailed proof of Theorem 7.

3.4 Rate of Convergence

In order to paint a better picture of the proposed NTS algorithm in practical cases, where the parameters n, M, L and K are finite, one can assess the speed of convergence of such algorithm. Hence, in this section we provide arguments that quantify the convergence rate of the proposed tractable NTS algorithm with respect to *i*) the number of NTS iterations n, *ii*) the statistical depth K, and *iii*) the source word length L.

3.4.1 Number of NTS Iterations n

As has been mentioned earlier, in the limit as L goes to infinity, the original NTS algorithm simulates the Blahut algorithm in [48] for computation of the rate-distortion function $R(P_M, d)$ [10], where the next distribution at each iteration step emerges "on the fly" through the coding process. Earlier work [58] has shown an upper bound on the speed of convergence of Blahut algorithm for rate-distortion function computation with respect to the number of iterations n, where it has been shown that the approximation error of the rate-distortion function is, at most, inversely proportional to the number of iterations n. For finite length L, and by the results of theorems 3 and 4, the NTS algorithm yet again simulates Blahut algorithm for computation of the rate-distortion function $R(P_M^L, \gamma)$, asymptotically in K, albeit for the auxiliary distortion measure $\rho^{(d)}$

defined in (3.34) and the extreme distortion level $\gamma = 0$. Hence, the approximation error of $R(P_M^L, \gamma)$ is, at most, inversely proportional to the number of iterations n. The numerical results depicted in Fig 3.1 provide a compelling evidence that the modified NTS algorithm converges faster with respect to n for smaller values of finite length L.

3.4.2 Statistical Depth K

Here we establish the behavior of the modified NTS algorithm at every iteration for finite statistical depth K in terms of the speed of convergence. We present the result for discrete alphabet spaces, with the note that generalization to abstract alphabet spaces is straight forward.

Lemma 2: The ML estimator of the codebook reproduction distribution that is restricted to generate i.i.d. M-length super symbols is an unbiased estimator with variance decaying proportional to $1/K$ around its mean, i.e.,

$$\mathbb{VAR}[Q_{n,M,L,K}(\mathbf{y})] = \frac{1}{K}\mathbb{VAR}[Q_k(\mathbf{y})], \quad \forall \mathbf{y} \in \mathcal{Y}^M, \tag{3.44}$$

where $\mathbb{VAR}[\cdot]$ denotes the variance of the argument. The proof of Lemma 2 is found in Appendix A.2. This establishes the speed of convergence of the ML estimator with finite statistical depth K, i.e., the variance of the random ML distribution around the limiting distribution decays proportional to $1/K$, where K is the number of d-match events observed before estimating the ML distribution.

3.4.3 Source-word Length L

In order to assess the convergence rate of the NTS algorithm with respect to the source word length L, one should equivalently assess the convergence rate of the conditional limit theorem, which is intimately connected to the convergence rate of Sanov's theorem or Gibbs conditioning principle [59, 56] [60, Sec. 3.3]. While theoretical quantification of such convergence rates has been elusive over the decades in the literature, a significant speed of convergence result has been introduced in [61] using Nummelin's conditional weak law of large numbers defined as follows: Let $(\mathcal{B}, || \cdot ||)$ be a real separable Banach space of dimension $1 \leq u \leq \infty$, and assume $\mathbf{Y}_1, \mathbf{Y}_2, \ldots$ are i.i.d. $\mathcal{B}$ valued random vectors with probability measure μ and mean $m = \int_{\mathcal{B}} \mathbf{y} d_{\mu}$. Nummelin's conditional weak law of large numbers establishes that under suitable conditions on $(D \subset \mathcal{B}, \mu)$ and for every $\epsilon > 0$,

$$\lim_{L \to \infty} \mathbb{P}(||S_L/L - a_0|| < \epsilon \mid S_L/L \in D) = 1, \tag{3.45}$$

with a_0 the dominating point of D and $S_L = \sum_{i=1}^{L} \mathbf{Y}_i$. In [61], authors studied the rates of convergence of such law, i.e., they examined $\lim_{L \to \infty} \mathbb{P}(||S_L/L - a_0|| < t/L^r \mid S_L/L \in D)$ as r, t and D vary, where a connection to Gibbs conditioning principle was also investigated. More specifically, it was shown that, for any Borel set $\mathcal{A}$ of $\mathcal{B}$,

$$\lim_{L \to \infty} \left| \mathbb{P}\left(\mathbf{Y} \in \mathcal{A} \;\middle|\; \frac{S_L}{L} \in D_{L,r,t}\right) - \mathbb{P}(\mathbf{Y}^* \in \mathcal{A})) \right| = 0, \text{ for } r < 0.5, \tag{3.46}$$

where $\mathbf{Y}^*$ is generated according to the limiting distribution μ^*, and $D_{n,r,t} = \{\mathbf{y} : ||\mathbf{y} - a_0|| < t/L^r\} \cap D$, with the mean m not belonging to D or its closure $\overline{D}$, in addition to other constraints on μ and $\mathcal{B}$ (see [61] for the full list of constraints in Theorem 4 and Theorem 7).

Additionally, we emphasize that the sequence of K d-matching ML-length codewords is memoryless in K, because each codeword must satisfy the distortion constraint with an independently generated source word separately (independently of the other codewords). In contrast, the L super-symbols inside each codeword are dependent, because the distortion constraint ties them together, as they satisfy it only on the average. It seems quite clear that this induced "memory" is most likely to slow down the convergence speed, hence we expect the algorithm to be converging in L slower than K. However, these convergence rates should also be studied in terms of computational steps where each step is a d-match event resulting in a produced d-matching codeword in the sequence that quantifies the convergence rates. In that sense, the step size is computationally constant when we increase K (on average it takes the same number of codewords generated to find a d-match to a given source word), but it grows exponentially with increase in L as it requires exponentially more codewords to find a d-match.

Furthermore, it is worth noting that the exact rate-distortion performance of a random code with finite block length L and memory depth M, which is required to guarantee d-match with probability $1-\epsilon$, (with $0<\epsilon<1$), to independently generated source words, has been detailed in [47] for different types of source distributions. In other words, for a given (L, M, C, d, ϵ) random code having an optimal codebook generating distribution $Q^*_{M,L}$, the expected probability of excess distortion is given by,

$$\mathbb{E}[\epsilon] = \mathbb{E}\left[1 - Q^*_{M,L}\left(\mathcal{B}_{M,L}\left(\tilde{\mathbf{X}}, d\right)\right)\right]^C, \tag{3.47}$$

where C is the number of codewords in the codebook, $\mathcal{B}_{M,L}(\cdot, d)$ is the d-distortion ball around source words, i.e.,

$$\mathcal{B}_{M,L}\left(\tilde{\mathbf{x}}, d\right) = \left\{\tilde{\mathbf{y}} : \tilde{\mathbf{y}} \in \mathcal{Y}^{ML}, \rho\left(\tilde{\mathbf{x}}, \tilde{\mathbf{y}}\right) \leq d\right\}, \tag{3.48}$$

and $Q^*_{M,L}\left(\mathcal{B}_{M,L}(\cdot, d)\right)$ denotes the probability of generating codewords that falls in the distortion ball $\mathcal{B}_{M,L}(\cdot, d)$ via $Q^*_{M,L}$ codebook reproduction distribution. The finite-length rate can be calculated from (3.47) as,

$$R_L(P_M, d, \epsilon) = \frac{1}{L}\log(C) > R_L(P_M, d), \quad \forall L < \infty, \text{ and for sufficiently small } \epsilon. \tag{3.49}$$

The d-ball set $\mathcal{B}_{M,L}(\tilde{\mathbf{X}}, d)$ and the finite length rate $R_L(P_M, d, \epsilon)$ are detailed for popular sources and distortion measures in practice, e.g., binary memoryless sources with hamming distortion measure, discrete memoryless sources with hamming distortion measure, and Gaussian sources with mean squared error distortion measure in [47]. For identical source and reproduction alphabet spaces, i.e., $\mathcal{X} = \mathcal{Y}$, the constrained set of joint distributions $E_{M,L}(P_M, d)$, regardless of d and L, will always contain the joint distribution for which the channel is given as,

$$\widehat{W}(\tilde{\mathbf{y}}|\tilde{\mathbf{x}}) = \begin{cases} 1 & \text{if } \tilde{\mathbf{y}} = \tilde{\mathbf{x}} \\ 0 & \text{otherwise} \end{cases} \quad \forall \tilde{\mathbf{x}} \in \mathcal{X}^{ML}. \tag{3.50}$$

$$\widehat{V} = P_M^L \circ \widehat{W}. \tag{3.51}$$

This will result in a codebook reproduction distribution identical to the source distribution P_M. Hence, this in turn suggests an upper bound on the NTS performance, i.e.,

$$R_L(P_M, d) \leq R_L^{\text{UB}}(P_M, d) = \min_{Q_M} \mathcal{D}\left(\widehat{V} || P_M^L \times Q_M^L\right). \tag{3.52}$$

$$R_L(P_M, d, \epsilon) \leq R_L^{\text{UB}}(P_M, d, \epsilon), \tag{3.53}$$

where $R_L^{\text{UB}}(P_M, d, \epsilon) = \frac{1}{L}\log\left(\widehat{C}\right)$ and $\widehat{C}$ satisfies,

$$\mathbb{E}[\epsilon] = \mathbb{E}\left[1 - P_M\left(\mathcal{B}_{M,L}\left(\tilde{\mathbf{X}}, d\right)\right)\right]^{\widehat{C}}. \tag{3.54}$$

Finally, in [47], the authors derived a lower bound on the codebook size required to realize an (L, M, C, d, ϵ) code. This lower bound is based on binary hypothesis testing. Suppose $\mathcal{X} = \mathcal{Y}$, the optimal performance achievable among all randomized tests $P_{W|X} : \mathcal{X}^{ML} \to \{0, 1\}$ between L-product probability distributions P_M^L and Q_M^L on $\mathcal{X}^{ML}$ is denoted by (1 indicates that the test chooses P_M^L)

$$\beta_\alpha \left(P_M^L, Q_M^L\right) = \min_{\substack{P_{W|X}: \\ \mathbb{P}_{P_M^L}(W=1) \geq \alpha}} \mathbb{P}_{Q_M^L}(W = 1), \tag{3.55}$$

where $\mathbb{P}_{P_M^L}$ and $\mathbb{P}_{Q_M^L}$ are used to denote the probabilities of events on the underlying probability spaces induced by the distributions P_M^L and Q_M^L. Hence, it was shown that the codebook size C required to realize an (L, M, C, d, ϵ) code must satisfy,

$$C \geq \sup_{Q} \inf_{y \in \mathcal{Y}^{ML}} \frac{\beta_{1-\epsilon}\left(P_M^L, Q_M^L\right)}{\mathbb{P}_{Q_M^L}[\rho(\tilde{\mathbf{X}}, \tilde{\mathbf{y}})]}. \tag{3.56}$$

This equivalently results in a lower bound on the encoding rate, denoted as $R_L^{\mathrm{LB}}(P_M, d, \epsilon)$, required to guarantee finding a d-match in the codebook of size C with expected probability $1 - \epsilon$, to any independently generated source word that contains L independent M-length super-symbols. In the next section, we show the stable evolution of the codebook reproduction distribution, obtained by the NTS algorithm, for binary source toy examples to further illustrate the results derived in Theorems 1-7.

3.5 Toy Examples

First, we refer the readers back to the memoryless binary asymmetric source considered in Fig. 2.2 for the original NTS algorithm and Fig. 3.1 for the tractable NTS algorithm. It can be seen in Fig. 3.1 that the codebook reproduction distribution approaches the optimal reproduction distribution as $L \to \infty$. Nevertheless, the algorithm always converges, for any finite length L, to the optimal achievable distribution $Q^*_{1,L}(P_1, d)$. For the settings in Fig. 3.1, and for $L = 1$, the only codeword that can possibly d-match the source word is one identical to it. Hence, the set of L-constrained joint distributions $E_{M,L,K}$ contains exactly one element, asymptotically in K, which consequently explains why the modified NTS algorithm converges after only one iteration for $L = 1$. As the finite length L increases, the set of joint distributions $E_{M,L,K}$ expands, which demonstrates a slower convergence of the codebook reproduction distribution to the optimal achievable distribution $Q^*_{1,L}(P_1, d)$. Additionally, in order to show a fair comparison between original and modified NTS algorithm, it is important to emphasize that the complexity of both algorithms is proportional to $nK \exp(L)$. Hence, we show in Fig. 3.3 the unstable performance of the original NTS algorithm for fixed string length $L = 64$, and for large number of NTS iterations $n = 5 \times 10^6$ to compensate for $K = 1$ inherent in the original NTS algorithm. Thus, the complexity of the original NTS algorithm with these parameters is roughly similar to the complexity of the modified NTS algorithm with parameters $L = 64, n = 50$, and $K = 10^5$, for which the performance curve is circle-marked and shown in Fig. 3.1. Hence, it can be concluded, at least for the cases examined in our simulations, that the tractable NTS algorithm, proposed in this work, is substantially more stable when compared to the original NTS algorithm in [10], which renders the proposed codebook generating algorithm very attractive to practical implementations. As, even for finite and small values of the string lengths L, convergence is guaranteed. Moreover, we note that the modified NTS algorithm, for finite length L, achieves the codebook reproduction distribution that minimize the encoding rate, while maintaining average distortion level d per L-segment source-code pairs, when this distribution is used in asymptotic-length encoding settings, as illustrated in Corollary 1.

Next, we depict the effect of the statistical depth K on the evolution of the code-

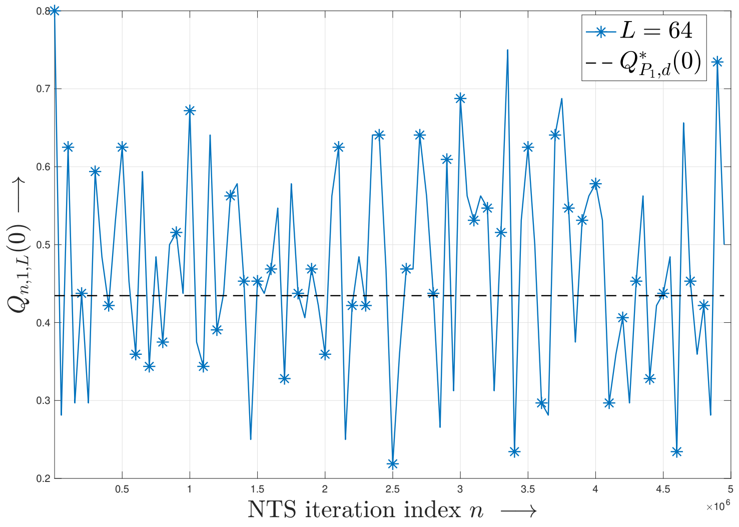


Figure 3.3: Evolution of the codebook reproduction distribution by the original NTS algorithm for finite source length $L = 64$. Binary memoryless source is considered with $P_1 = \{0.48, 0.52\}$, and Hamming distortion measure with $d = 0.35$.

book reproduction distribution for the aforementioned binary asymmetric source in Fig 3.4. It is important to note that the modified NTS algorithm estimate the next iteration codebook reproduction distribution, using ML estimator, from a sequence of K independent d-matching codewords, which exactly simplifies to the K-average (or sample average) of the M-th order d-matching codeword types. Thus, this estimator is an unbiased estimator with a variance decaying proportional to $1/K$. Fig. 3.4 further verifies that the amplitudes of fluctuations around the per-iteration asymptotic estimate, i.e., $Q^*_{M,L}(P_M, Q_{n,M,L,K}, d)^{\text{Marg.}}$, roughly decays proportional to $\sqrt{1/K}$ as illustrated in Lemma 2. It is worth noting that the approximate delay in the system, i.e., the total number of source super-symbols seen before the algorithm terminates, is proportional to K and is specifically equal to LKN.

To further assess the rate-distortion performance of the NTS algorithm for the given source example, we depict the finite-length rate in [47] denoted as $R_L(P_1, d, \epsilon)$, rate-distortion function $R(P_1, d)$, the upper and lower bounds on the finite length rate $R_L^{\text{UB}}(P_1, d, \epsilon)$,

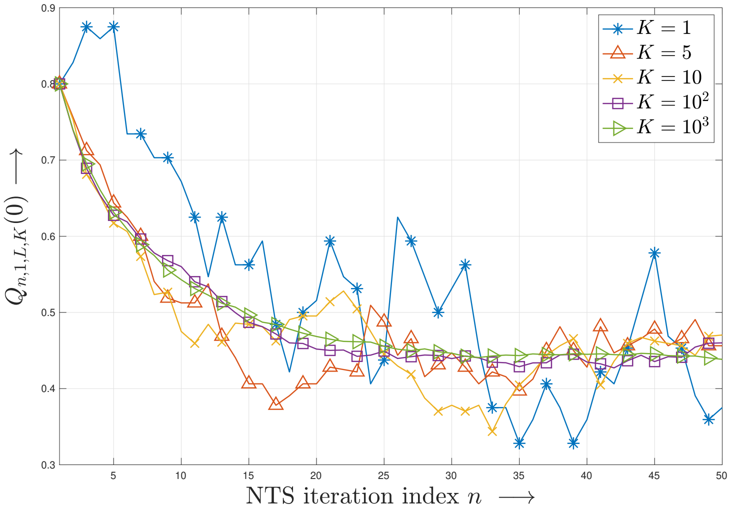


Figure 3.4: Effect of the statistical depth K on the evolution of codebook reproduction distribution. A binary memoryless source is considered with $P_1 = \{0.48, 0.52\}$, Hamming distortion measure at $d = 0.35$, and source length $L = 64$.

and $R_L^{\text{LB}}(P_1, d, \epsilon)$ for different values of source lengths L in Fig. 3.5. A Hamming distortion function is assumed with distortion level $d = 0.35$, and the probability of reproducing a source-word with a codeword, for which the distortion level d is not met, is $\mathbb{P}\left[f\left(\tilde{\mathbf{x}}\right) = +\infty\right] \leq \epsilon = 0.01$. For every source length L, $R_L(P_1, d, \epsilon)$ is the rate required, by the random codebook which is generated via $Q^*_{1,L}(P_1, d)^{\text{Marg.}}$, to guarantee that a d-match is found to a random finite-length source-word in the codebook with expected probability greater than or equal to $(1 - \epsilon)$. Hence, $R_L(P_1, d, \epsilon)$ captures the rate penalty due to both finite source length $L < \infty$ and the asymptotically non-optimal codebook reproduction distribution $Q^*_{1,L}(P_1, d)^{\text{Marg.}} \neq Q^*_{P_M,d}$. Notice that, as expected, $R_L(P_1, d, \epsilon) > R(P_1, d)$, $\forall L < \infty$. The gap between $R_L(P_1, d, \epsilon)$ and $R(P_1, d)$ continues to shrink as L approaches ∞.

Next, we turn our attention to sources with memory. A binary-alphabet stationary Markov chain source is considered, i.e., $\mathcal{X} = \mathcal{Y} = \{0, 1\}$. Let the source transition probabilities be $P(0|1) = 0.6, P(0|0) = 0.7, P(1|0) = 0.3$, and $P(1|1) = 0.4$ (This is also

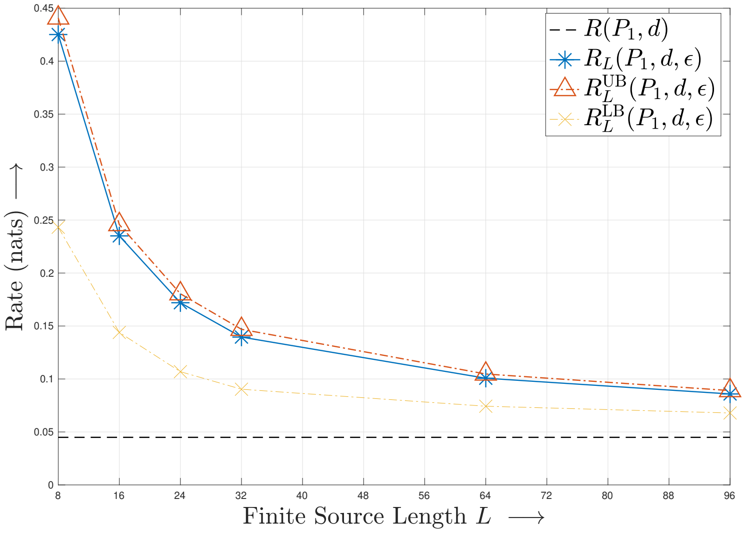


Figure 3.5: Rate performance for finite source lengths $L < \infty$. A binary source is considered with $P_1 = \{0.48, 0.52\}$, Hamming distortion measure at $d = 0.35$, statistical depth $K = 10^5$, and the probability of reproducing a source-word with a codeword for which the distortion level is not met is $\mathbb{P}\left[f\left(\tilde{\mathbf{x}}\right) = +\infty\right] \leq \epsilon = 0.01$.

a simple Gilbert-Elliott model). First, We consider the vector source that is resulted from such Markov source, hence we turn our attention to Algorithm 5 with $M > 1$. In the simulation environment, the vector length is set to two, i.e., $M = 2$, which corresponds to source distribution $P_2 = \{7/15, 3/15, 3/15, 2/15\}$. In order words, the source words are formed of i.i.d. vectors according to P_2. Obviously, such artificially generated source words require the prior knowledge of source joint distribution. For these settings, the Hamming distortion measure is considered. Without loss of generality, the distortion constraint is assumed as follows: $d = d_{\max} = 1/3$. Note that $Q^*_{P_2, d_{\max}} = \{1, 0, 0, 0\}$, i.e., the optimal reproduction distribution collapses onto one point that corresponds to the super-symbol $\mathbf{y} = (0, 0)$. The generalized NTS algorithm is run with $K = 10^5$, and different values of L. The evolution of the codebook reproduction distribution is plotted across the NTS iteration index n in Fig 3.6. For $L = 1$, and the considered distortion constraints, the only codeword that can d-match the source word is identical to it. Hence, the codebook reproduction distribution in (3.4) immediately converges to the source distribution P_2. As L increases, the set of all possible d-matching joint types $\mathcal{U}_{2,L}(d)$ expands, and it can be observed that the codebook reproduction distribution $Q_{n,2,L,K}$ approaches $Q^*_{P_2,d}$, asymptotically in n, as derived in Theorem 2.

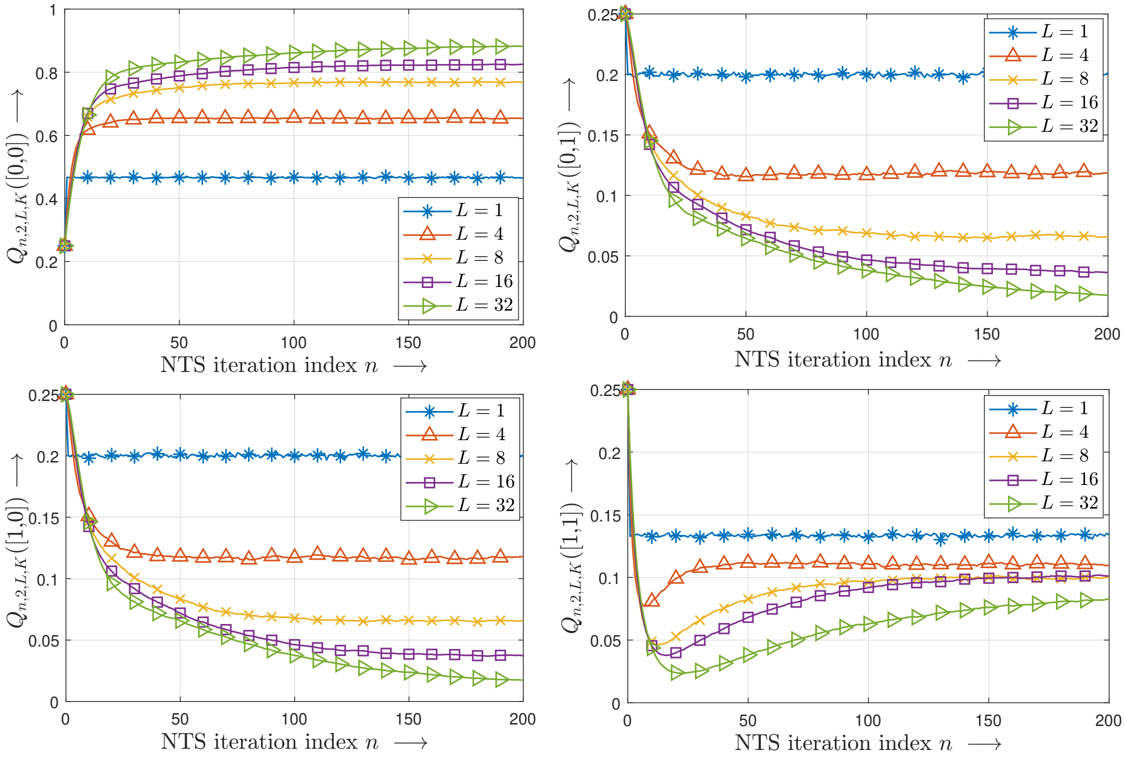


Figure 3.6: Evolution of the codebook reproduction distribution by the tractable NTS algorithm for different finite source lengths L and statistical depth $K = 10^5$. Binary Markov source is considered with Hamming distortion measure at $d = d_{\max} = 1/3$. Reprinted, with permission, from [46] © 2021 IEEE.

Finally, we illustrate the convergence behavior of the NTS algorithm variant, which is tailored for Markovian sources, i.e., Algorithm 6. We consider a more distinctive first-order Markov source with the following transition probabilities: $P(0|0) = 0.8, P(1|0) = 0.2, P(0|1) = 0.4$, and $P(1|1) = 0.6$. Similarly, for these settings, the Hamming distortion measure is considered. Without loss of generality, the distortion constraint is assumed as follows: $d = d_{\max} = 1/3$. We depict the evolution of transition probabilities of the codebook reproduction distributions as the number of NTS iterations n increases for

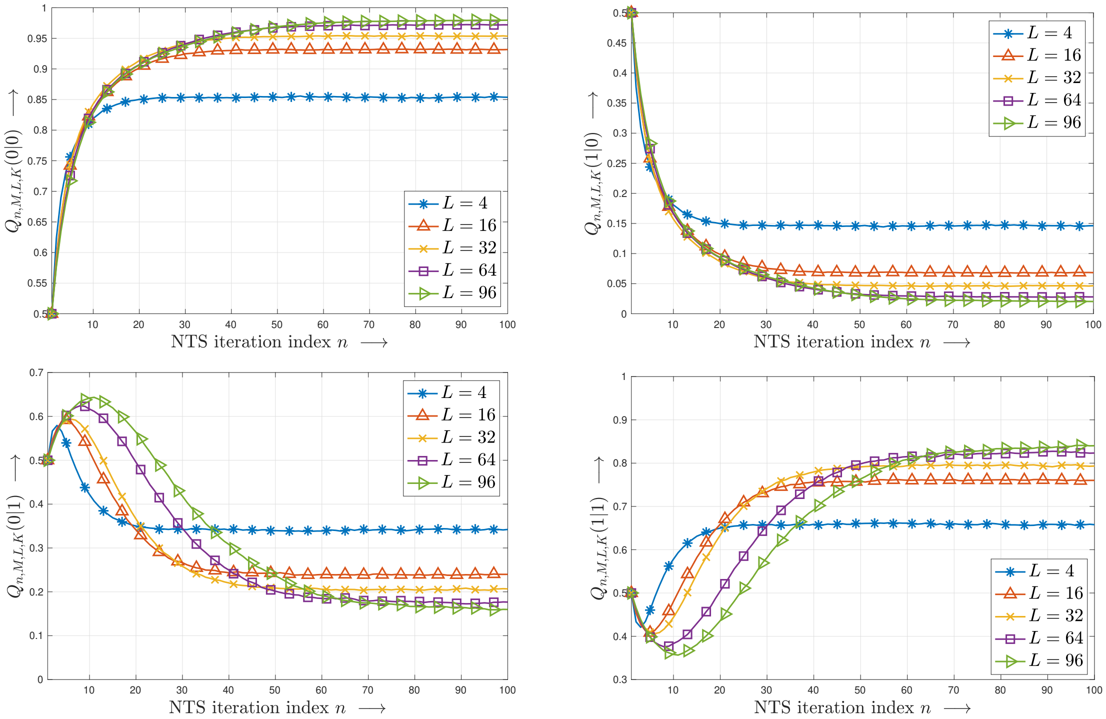


Figure 3.7: Evolution of the conditional codebook reproduction distribution by the tractable NTS algorithm Markov sources variant, for different finite source lengths L and statistical depth $K = 10^5$. Binary Markov source is considered with Hamming distortion measure at $d = d_{\max} = 1/3$.

different values of finite source word length L in Fig. 3.7. It is worth noting that as L increases, and for the given distortion level d, the transition probabilities of the codebook reproduction distribution approaches $Q^*(0|0) = 1$, and $Q^*(1|1) = 1$, which represents the highest source probable symbol conditioned on the previous sample as expected.

Chapter 4

An Optimal Codebook Design Approach for Beam Steering Directions in Wireless Systems

In this chapter, we develop and employ novel codebook design algorithms in millimeter wave wireless systems, where we think that this example application will greatly benefit from codebook design tools and methods. The results included in this chapter are published in [23, 62, 26, 63]. First, the wireless system model is introduced in Section 4.1. Then the relevant beamforming techniques are introduced in Section 4.2. The proposed codebook design algorithms for beam steering directions are developed in Section 4.3. Finally, the experimental results of the proposed design techniques are shown in Section 4.4.

4.1 System Model

Consider the downlink transmission direction. For outdoor settings, the 5G base station (also called gNB) is equipped with a planar array consisting of N_{tx} antennas, while

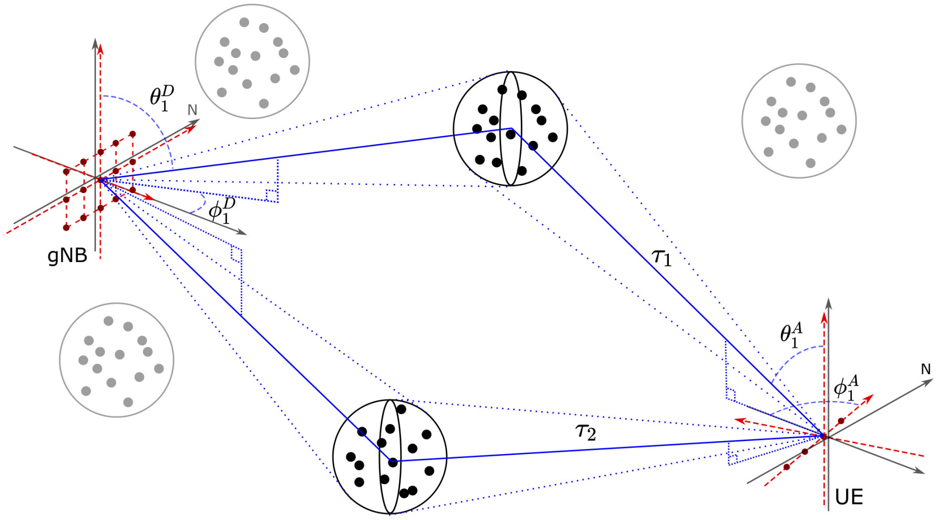


Figure 4.1: Snapshot of the 3D cluster delay line channel model in [64]. Reprinted, with permission, from [23] © 2019 IEEE.

the user equipment (UE) comprises a linear array consisting of N_{rx} antennas. Let $s \in \{1, 2, \ldots, N_{\mathrm{tx}}\}$ and $u \in \{1, 2, \ldots, N_{\mathrm{rx}}\}$ denote the transmit and receive antenna indices, respectively. The downlink channel, we consider, is modeled as the 3GPP Cluster Delay Line (CDL) channel [64], which is depicted in Fig. 4.1. Let N_{c} denote the number of detected clusters, and M_{r} the number of rays within a single cluster. Let $m \in \{1, 2, \ldots, M_{\mathrm{r}}\}$ be the ray index, and $n \in \{1, 2, \ldots, N_{\mathrm{c}}\}$ be the cluster index. The $(N_{\mathrm{rx}} \times N_{\mathrm{tx}})$ channel matrix is denoted by $\mathbf{H}_{n,m}(t)$, where t is the time index. Next, the unit-norm phase-control $(N_{\mathrm{tx}} \times 1)$ transmit beamforming vector and, similarly, the $(N_{\mathrm{rx}} \times 1)$ receive beamforming vector are denoted by $\mathbf{b}_{\mathrm{tx}}(\boldsymbol{\varphi})$ and $\mathbf{b}_{\mathrm{rx}}(\boldsymbol{\vartheta})$, respectively, where $\boldsymbol{\varphi}$, and $\boldsymbol{\vartheta}$ are the transmit and receive vectors of the beamforming phases.

Beamforming can be attained using either amplitude control, phase control, or both. For maximum power efficiency and maximum total transmit power, it is desirable to operate the power amplifier associated with each antenna as close to its saturation point as possible. Typically, to avoid non-linear effects, the operating point is selected to be a few dB below the saturation point (also called back off), to allow for some peak-to-average power margin. Amplitude-based beamforming is suboptimal due to the drop seen in the EIRP when power amplifiers are either switched off or operating well below the optimal efficiency point. Beam broadening achieved by switching off $(\ell - 1)N_{\text{tx}}/\ell$ power amplifiers, with $\ell \in \{2^1, 2^2, \ldots, N_{\text{tx}}\}$ is a special case of amplitude-based beamforming. For example, turning off half the power amplifiers results in 6 dB drop in EIRP at the steering direction. This is because the total transmit power decreases by 3 dB and the power array factor drops by 3 dB as well. Correspondingly, the beam width increases by a factor of two. Due to the severe loss in EIRP when using amplitude-based beam beamforming or beam broadening, throughout the remainder of this chapter, only phase-control based beamforming will be considered. The received signal in this setting is given by,

$$\begin{aligned} y(t, f_r) = (\mathbf{b}_{\text{rx}}(\boldsymbol{\vartheta}))^{\text{H}} \sum_{n=1}^{N_{\text{c}}} \sum_{m=1}^{M_{\text{r}}} \Big\{ & \left(\mathbf{H}_{n,m}(t) e^{-j2\pi f_r \tau_n(t)}\right) \mathbf{b}_{\text{tx}}(\boldsymbol{\varphi}) \\ & x(t, f_r) \Big\} + (\mathbf{b}_{\text{rx}}(\boldsymbol{\vartheta}))^{\text{H}} \, \mathbf{n}(t, f_r), \end{aligned} \tag{4.1}$$

where f_r is the rth sub-carrier frequency, $\tau_n(t)$ is the nth cluster delay, $x(t, f_r)$ is the complex frequency domain transmit symbol with $\mathbb{E}\left[|x(t, f_r)|^2\right] = 1$, and $\mathbf{n}(t, f_r) \sim \mathcal{CN}(0, \sigma_n^2)$ is the complex Additive White Gaussian Noise (AWGN) vector, with $\sigma_n^2 = k_{\text{B}} T B$, where k_{B} is the Boltzmann constant, T is the temperature in Kelvin, and B is the transmission bandwidth. We employ the standard notation $(\cdot)^{\text{T}}$ and $(\cdot)^{\text{H}}$ to denote transposition and the conjugate transposition operations, respectively. The (u, s) element of the channel matrix $\mathbf{H}_{n,m}(t)$ is denoted by $h_{n,m}^{u,s}(t)$, and is given by,

$$\begin{aligned} h_{n,m}^{u,s}(t) = \sqrt{\frac{P_n}{M_{\text{r}}}} F_{\text{rx}}(\theta_{n,m}^A, \phi_{n,m}^A) e^{j2\pi \frac{\left(\hat{\theta}_{n,m}^A\right)^{\text{T}} \cdot \bar{d}_u}{\lambda_c}} e^{j\zeta_{n,m}} & \\ F_{\text{tx}}(\theta_{n,m}^D, \phi_{n,m}^D) e^{j2\pi \frac{\left(\hat{\theta}_{n,m}^D\right)^{\text{T}} \cdot \bar{d}_s}{\lambda_c}} e^{-2\pi j \frac{\left(\hat{\theta}_{n,m}^A\right)^{\text{T}} \cdot \bar{v}}{\lambda_c} t}, & \end{aligned} \tag{4.2}$$

where $\left(\theta_{n,m}^D, \phi_{n,m}^D\right)$ are the elevation and azimuth departure angles for the ray m in cluster n, while $\left(\theta^A_{n,m}, \phi^A_{n,m}\right)$ are the elevation and azimuth arrival angles. Furthermore, $F_{\mathrm{tx}}(\theta^D_{n,m}, \phi^D_{n,m})$, and $F_{\mathrm{rx}}(\theta^D_{n,m}, \phi^D_{n,m})$ are the transmit and receive normalized field patterns for vertical polarization as a function of the elevation and azimuth angles. The field patterns are assumed identical for all antenna elements on either of the transmit or receive sides. Throughout this chapter, no cross polarization is considered. The n-th cluster power is denoted as P_n, λ_c is the carrier wavelength, and $\zeta_{n,m}$ models the ray's random initial phase. The UE velocity vector is denoted as $\overline{v}$. The vectors $\overline{d}_s$ and $\overline{d}_u$ are the location vectors of the individual antenna element relative to the transmit and receive origins, respectively. Additionally, the spherical unit vectors $\hat{\theta}^A_{n,m}$ and $\hat{\theta}^D_{n,m}$ are defined as,

$$\hat{\theta}^A_{n,m} \triangleq \begin{bmatrix} \sin(\theta^A_{n,m})\cos(\phi^A_{n,m}) \\ \sin(\theta^A_{n,m})\sin(\phi^A_{n,m}) \\ \cos(\theta^A_{n,m}) \end{bmatrix}, \tag{4.3}$$

$$\hat{\theta}^D_{n,m} \triangleq \begin{bmatrix} \sin(\theta^D_{n,m})\cos(\phi^D_{n,m}) \\ \sin(\theta^D_{n,m})\sin(\phi^D_{n,m}) \\ \cos(\theta^D_{n,m}) \end{bmatrix}. \tag{4.4}$$

Next, define the transmit and receive array factors for ray m in cluster n as,

$$A_{\mathrm{tx}}(\theta^D_{n,m}, \phi^D_{n,m}, \boldsymbol{\varphi}) \triangleq \left[e^{j2\pi \frac{\left(\hat{\theta}^D_{n,m}\right)^{\mathrm{T}} \cdot \bar{d}_1}{\lambda_c}} \dots e^{j2\pi \frac{\left(\hat{\theta}^D_{n,m}\right)^{\mathrm{T}} \cdot \bar{d}_{N_{\mathrm{tx}}}}{\lambda_c}} \right] \mathbf{b}_{\mathrm{tx}}(\boldsymbol{\varphi}), \tag{4.5}$$

$$A_{\mathrm{rx}}(\theta^A_{n,m}, \phi^A_{n,m}, \boldsymbol{\vartheta}) \triangleq \left(\mathbf{b}_{\mathrm{rx}}(\boldsymbol{\vartheta})\right)^{\mathrm{H}} \left[e^{j2\pi \frac{\left(\hat{\theta}^A_{n,m}\right)^{\mathrm{T}} \cdot \bar{d}_1}{\lambda_c}} \dots e^{j2\pi \frac{\left(\hat{\theta}^A_{n,m}\right)^{\mathrm{T}} \cdot \bar{d}_{N_{\mathrm{rx}}}}{\lambda_c}} \right]^{\mathrm{T}}. \tag{4.6}$$

Therefore, the perceived channel coefficients, upon applying the beamforming vectors, are obtained as,

$$\begin{aligned} h_{n,m}(t,f_r) &= \left(\mathbf{b}_{\mathrm{rx}}(\boldsymbol{\vartheta})\right)^{\mathrm{H}} \mathbf{H}_{n,m}(t) e^{-j2\pi f_r \tau_n(t)} \mathbf{b}_{\mathrm{tx}}(\boldsymbol{\varphi}), \\ h_{n,m}(t,f_r) &= \sqrt{\frac{P_n}{M_{\mathrm{r}}}} F_{\mathrm{rx}}(\theta^D_{n,m},\phi^D_{n,m}) A_{\mathrm{rx}}(\theta^A_{n,m},\phi^A_{n,m},\boldsymbol{\vartheta}) \\ &\qquad F_{\mathrm{tx}}(\theta^A_{n,m},\phi^A_{n,m}) A_{\mathrm{tx}}(\theta^D_{n,m},\phi^D_{n,m},\boldsymbol{\varphi}) \\ &\qquad e^{j\zeta_{n,m}} e^{-2\pi j \frac{\left(\hat{\theta}^A_{n,m}\right)^{\mathrm{T}} \cdot \overline{v}}{\lambda_c} t} e^{-j2\pi f_r \tau_n(t)}. \end{aligned} \tag{4.7}$$

Finally the aggregate channel transfer function, due to all clusters and rays, is according to,

$$h(t,f_r) = \sum_{n=1}^{N_{\mathrm{c}}} \sum_{m=1}^{M_{\mathrm{r}}} h_{n,m}(t,f_r). \tag{4.8}$$

The Signal-to-Noise Ratio (SNR) seen at rth sub-carrier with frequency f_r is given by,

$$\gamma_r(t) = \frac{P_{\mathrm{tx}} G_{\mathrm{tx}} |h(t,f_r)|^2 G_{\mathrm{rx}}}{P_{\mathrm{L}}(t) F_n \ \sigma_n^2}, \tag{4.9}$$

where P_{tx} is the average transmit power, $P_{\mathrm{L}}(t)$ is the path-loss, F_n is the receiver noise factor, and where G_{tx} and G_{rx} are the maximum gains of the transmit and receive antenna elements relative to an isotropic antenna element, respectively.

4.2 Beamforming Techniques

This subsection provides an analysis for phase-control transmit beamforming, noting that the corresponding analysis for receive beamforming is similarly obtained in a straightforward manner. Consider a planar antenna array with uniform spacing between horizontal and vertical elements, i.e., d_x=d_y=$\frac{\lambda_c}{2}$, where λ_c is the carrier wavelength. Define the beam-space transformation on the x-axis and y-axis as Ω_x=$kd_x \sin(\theta)\cos(\phi)$=$\pi\sin(\theta)\cos(\phi)$, and Ω_y=$kd_y \sin(\theta)\sin(\phi)$=$\pi\sin(\theta)\sin(\phi)$, where $k = \frac{2\pi}{\lambda_c}$ is the wave number, θ is the elevation angle, and ϕ is the azimuth angle. The conventional planar array setup is shown

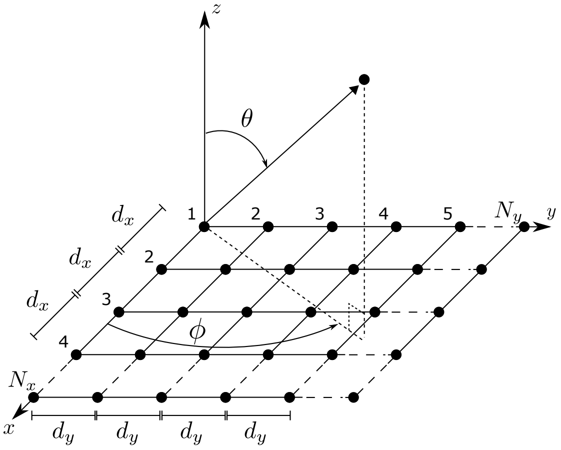


Figure 4.2: Co-ordinate system of planar array with uniform spacing. Reprinted, with permission, from [23] © 2019 IEEE.

in Fig. 4.2. Hence, the transmit power array factor simplifies to [65],

$$\begin{aligned}\mathbf{a}_{\text{tx}}\left(\Omega_x, N\right) &\triangleq \begin{bmatrix}1 \; e^{-j\Omega_x} \; \dots \; e^{-j\Omega_x(N-1)}\end{bmatrix}^{\text{T}}, \\ \mathbf{a}_{\text{tx}}\left(\Omega_y, N\right) &\triangleq \begin{bmatrix}1 \; e^{-j\Omega_y} \; \dots \; e^{-j\Omega_y(N-1)}\end{bmatrix}^{\text{T}}\end{aligned} \tag{4.10}$$

$$\mathbf{b}_{\text{tx}}\left(\boldsymbol{\varphi}\right) \triangleq \mathbf{b}_{\text{tx}}^{(x)}\left(\boldsymbol{\varphi}_x\right) \otimes \mathbf{b}_{\text{tx}}^{(y)}\left(\boldsymbol{\varphi}_y\right), \tag{4.11}$$

$$\begin{aligned}A_{\text{tx}}(\Omega_x, \Omega_y, \boldsymbol{\varphi}_x, \boldsymbol{\varphi}_y) &= \left(\mathbf{a}_{\text{tx}}\left(\Omega_x\right)^{\text{H}}\mathbf{b}_{\text{tx}}^{(x)}(\boldsymbol{\varphi}_x)\right) \cdot \left(\mathbf{a}_{\text{tx}}\left(\Omega_y\right)^{\text{H}}\mathbf{b}_{\text{tx}}^{(y)}(\boldsymbol{\varphi}_y)\right), \\ A_{\text{tx}}(\Omega_x, \Omega_y, \boldsymbol{\varphi}_x, \boldsymbol{\varphi}_y) &= A_{\text{tx}}^{(x)}(\Omega_x, \boldsymbol{\varphi}_x) A_{\text{tx}}^{(y)}(\Omega_y, \boldsymbol{\varphi}_y),\end{aligned} \tag{4.12}$$

where $\mathbf{b}_{\text{tx}}^{(x)}(\boldsymbol{\varphi}_x)$ and $\mathbf{b}_{\text{tx}}^{(y)}(\boldsymbol{\varphi}_y)$ are the beamforming vectors along the x and y coordinates of the planar array in [26, Fig. 1], respectively. The Kronecker product operation is denoted by ⊗. The array factor can be maximized at a given steering direction by using the conventional Constant Phase Offset (CPO) beamforming technique [23, 22, 20, 65], yielding the beamforming vectors:

$$\mathbf{b}_{\text{tx}}^{(x)}(\omega_x) = \frac{1}{\sqrt{N_{\text{x}}}}\begin{bmatrix}1 \quad e^{-j\omega_x} \quad \dots \quad e^{-j\omega_x(N_{\text{x}}-1)}\end{bmatrix}^{\text{T}}, \tag{4.13}$$

$$\mathbf{b}_{\mathrm{tx}}^{(y)}(\omega_y) = \frac{1}{\sqrt{N_{\mathrm{y}}}} \begin{bmatrix} 1 & e^{-j\omega_y} & \ldots & e^{-j\omega_y(N_{\mathrm{y}}-1)} \end{bmatrix}^{\mathrm{T}}, \tag{4.14}$$

where $\omega_x = \pi \sin(\theta_0)\cos(\phi_0)$ and $\omega_y = \pi \sin(\theta_0)\sin(\phi_0)$ are the beam space transformation of the elevation and azimuth steering angles θ_0 and ϕ_0, respectively. In this setting, the highest possible array factor, $10\log_{10}(N_{\mathrm{tx}})$ dB, is guaranteed at the steering direction. It is worthwhile to note in passing that if a single beam is scheduled by the base station to serve a user, then maximizing the array factor at the dominant channel direction between the UE and the gNB will consequently boost the perceived user SNR, defined in (4.9). Hence, the average beamforming array factor across users is the objective function of choice for the beam steering design approaches introduced later on in this chapter. Additionally, in [22], the authors showed that the low-complexity dominant directional beamforming scheme suffers only a minimal SNR loss (less than a dB loss for over 50% of the users in channels with up to $N_c = 5$ clusters) relative to even the best beamforming scheme. Consequently, single serving beam per user with CPO beamforming at the channel dominant direction has been widely employed in practice [20, 22, 66].

Both transmitter and receiver typically operate with predefined "*codebooks*" of beamforming vectors, wherein each codebook entry corresponds to a beam steering direction. An increase in codebook size hinders beam tracking and beam alignment due to the inherent increase in beam measurement time (sweep time) and thus compromises the system responsiveness to user and environment dynamics.

4.3 Optimal Beam Steering Directions

This section covers our main contributions, namely, the development of codebook design methods to approach beam steering optimality. Note that we only focus on designing the pointing angles of the codebook for CPO beams, without recourse to other design aspects such as beam shape, side lobes level, etc. The beamforming vectors are stored as codebook entries, such that each entry corresponds to an angular direction. Specifically, each codebook entry corresponds to an elevation and azimuth angle pair. The simplest (and most common) beam steering approach is to quantize the elevation and azimuth field-of-view *uniformly* into N_b pointing directions, similar to [21, 67], where N_b is the number of beams (entries) in the codebook. A somewhat more sophisticated approach quantizes the beam-space field-of-view $\overline{\Omega}_x$, and $\overline{\Omega}_y$ uniformly, which is known as the Discrete Fourier Transform (DFT) codebook [68, 66].

It is important to note that the beam shape is direction-dependent, i.e., different beam steering angles result in wider or narrower beams, as depicted in Fig. 4.3. Moreover, a common simplifying assumption is that the UE positions are uniformly distributed on the horizontal plane [64], which nevertheless results in a non-uniform distribution of user angles ϕ_i and θ_i across the angular space, where i is the user index. Uniform distribution of steering angles implies that the beams' density across the angular space remains unchanged in the regions of space at which the beams are wider or in the regions of space at which there is low or no user density. Hence, we conclude that uniform distribution of beam steering angles across the field-of-view is virtually always suboptimal, even under simplistic assumptions such as uniform user distribution on a plane.

4.3.1 Heuristic Non-uniform Beam Steering

First, we propose a heuristic non-uniform approach to overcome some shortcomings of the uniform approach. The central idea of this heuristic approach is to adapt the density of steering directions to the direction-dependent beam width, i.e., the density of pointing angles decreases when the beams become wider and vice versa. Let $\overline{\phi} \leq 360$, and $\overline{\theta} \leq 90$ be the field-of-view in degrees, and let $\overline{\Omega}_x \leq 2\pi$, and $\overline{\Omega}_y \leq \pi$ be the corresponding beam-

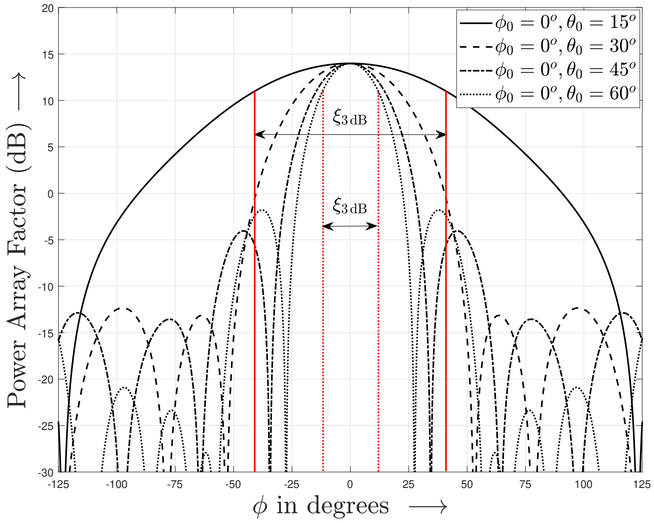


Figure 4.3: The power array factor slice for 5×5 planar array using CPO beamforming. For each steering direction, the slice is taken at $\theta = \theta_0$. Reprinted, with permission, from [62] © 2019 IEEE.

space field-of-view. Define the number of elevation beams as N_b^θ. Furthermore, define the X dB azimuth beam width for CPO beamforming $\xi_X(\theta_0)$ as the beam width in degrees of power array factor slice at $\theta = \theta_0$, such that the power array factor drops by X dB from its maximum. The 3-dB beam width, $\xi_{3\,\mathrm{dB}}(\theta_0)$, is shown in Fig. 4.3 for $\theta_0 = 15^o$ and $\theta_0 = 60^o$. Consequently, the heuristic non-uniform beam placement algorithm is proposed as:

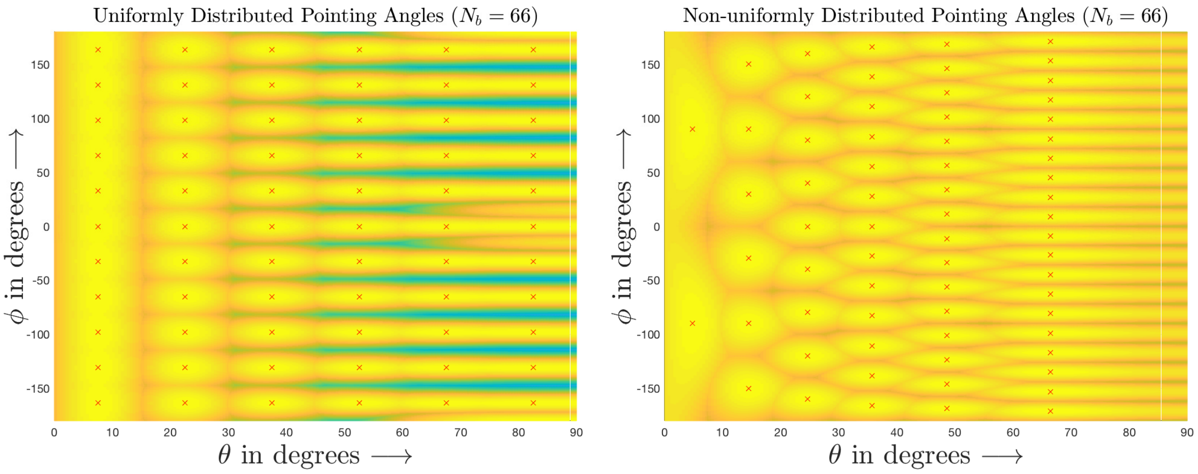
1. Initialize the beam steering set to an empty set, $\mathcal{A} = \emptyset$.

2. Quantize $\overline{\Omega}_y$ uniformly into N_b^θ samples as:

$$\zeta_k = \frac{\overline{\Omega}_y \cdot k}{N_b^\theta} + \frac{\overline{\Omega}_y}{2N_b^\theta}, \quad k \in \left\{0, 1, \ldots, N_b^\theta - 1\right\} \tag{4.15}$$

Note that this will result in a non-uniform quantization of the elevation angles, i.e., the elevation angles are quantized uniformly only in the sin angle domain.

3. **For** every elevation pointing angle, $[k = 0, k \leq N_b^\theta - 1, k++]$ **do**



Uniformly Distributed Pointing Angles ($N_b = 66$)

Non-uniformly Distributed Pointing Angles ($N_b = 66$)

Figure 4.4: Heat map of the absolute array factor, denoted by $|A_{\text{tx}}^{(x)}(\Omega_x, \boldsymbol{\varphi}_x) A_{\text{tx}}^{(y)}(\Omega_y, \boldsymbol{\varphi}_y)|$, across the overall coverage area for an indoor system scenario considering access points mounted on the ceiling. The array size is (5×5), and the field-of-view is $\left(\overline{\phi} = 360^o, \overline{\theta} = 90^o\right)$. The steering directions are marked with cross signs. Reprinted, with permission, from [62] © 2019 IEEE.

(a) Calculate the X dB beam width $\xi_X(\theta_0)$ for $\phi_0 = 0$, and $\theta_0 = \sin^{-1}\left(\frac{\zeta_k}{\pi}\right)$, this beam width is used to determine the corresponding beam density at the current iterate elevation direction.

(b) Calculate the kth elevation pointing angles set (which contains one element): $\Theta_b^k = \left\{\sin^{-1}\left(\frac{\zeta_k}{\pi}\right)\right\}$.

(c) Calculate the number of azimuth beams at the kth iterate:

$$N_{b,k}^{\phi} = \left\lceil \frac{\overline{\phi}}{(\beta_k \xi_X(\theta_0))} \right\rceil, \tag{4.16}$$

where $\lceil \cdot \rceil$ is the ceiling operation, and β_k is the beam density factor. Hence, the number of beams at a given elevation angle is proportional to the azimuth field-of-view divided by the X dB beam width, where the parameters X and β_k, which control the degree of beam overlap in the azimuth direction, and provide the necessary degree of freedom in order to achieve the required codebook size.

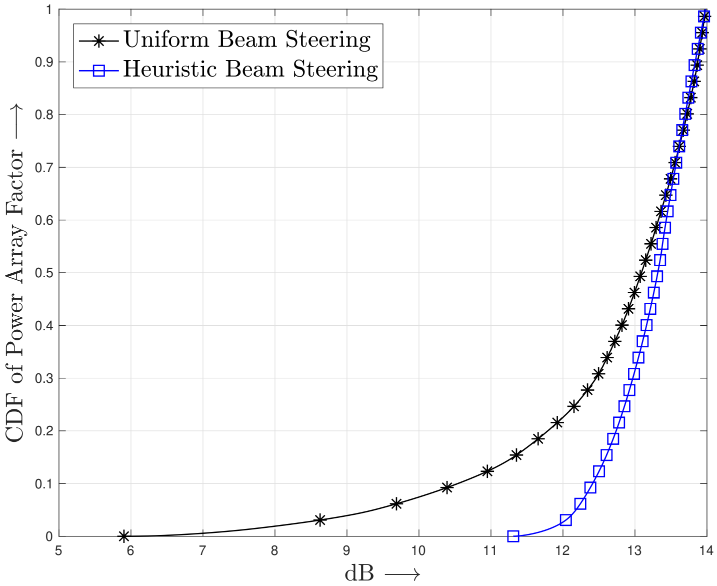


Figure 4.5: CDF of the power array factor across the overall coverage area for an indoor system scenario considering access points mounted on the ceiling. The array size is (5×5), and the field-of-view is $\left(\overline{\phi} = 360^o, \overline{\theta} = 90^o\right)$.

(d) Calculate the kth azimuth pointing angles set by uniformly quantizing the azimuth field-of-view in the angle domain into $N^{\phi}_{b,k}$ points:

$$\Phi^k_b = \left\{ \frac{\overline{\phi} \cdot \ell}{N^{\phi}_{b,k}} + \frac{\overline{\phi}}{2N^{\phi}_{b,k}} \right\}, \quad \ell \in \left\{0, \ldots, N^{\phi}_{b,k} - 1\right\}. \tag{4.17}$$

(e) The beam steering codebook expands to include the Cartesian product of Θ^k_b and Φ^k_b as:

$$\mathcal{A} = \mathcal{A} \cup \left\{\Theta^k_b \times \Phi^k_b\right\}, \tag{4.18}$$

where ∪ and × denotes the union and the Cartesian product of sets, respectively.

4. The codebook size is calculated as, $N_b = \sum_k N^{\phi}_{b,k}$.

Although this technique accounts for the non-uniform beam width, it does not account for users location distribution, which is potentially time-varying. However, this heuristic non-uniform beam steering technique aims to maximize the beam coverage across the field of view as shown in Fig. 4.4, and is useful in cases where user statistics are unknown or hard to obtain. The considerable improvements in the power array factor can be quantitatively observed in Fig. 4.5, where the 10th percentile of the power array factor across the angular space, for an indoor system scenario with access points mounted on the ceiling, is improved by about 2 dB in comparison with the conventional uniform distribution of steering directions scheme.

4.3.2 K-means-based Beam Steering

Our second approach to this problem is to pursue an iterative framework that guarantees convergence to (at least locally) optimal performance. The first key realization is that the beam steering problem at hand is effectively equivalent to a generalized clustering problem (albeit with an unusual distortion measure). The space to be divided into regions is the 2-dimensional angular space, with boundaries specified by the transmitter field-of-view. The data vectors to be clustered are the users' angle vectors as seen from the transmitter local coordinate system, which are denoted as $\boldsymbol{\psi}_i = [\phi_i \;\; \theta_i]^{\mathrm{T}}$. For each cluster, a single beam steering direction, which we will also refer to as the cluster centroid, is chosen to serve any of the users in the cluster. We now turn to the distortion measure that determines the assignment of a user to any cluster. An obvious option is to employ the traditional Mean-Squared Error (MSE) distortion measure. In other words, assign the user, i.e., the user's angle vector, to the nearest cluster representative (or cluster centroid), in the MSE distortion sense. This option is suboptimal, due to the fact that the absolute array factor beam width is direction-dependent, and the MSE distortion measure only takes into account the angular distance between the user angular vector and the centroid. In other words, the beam steering direction nearest to the user angular vector in the MSE sense does not necessarily yield the largest transmit array factor. Thus, the MSE distortion measure is mismatched with the true objective. In the ideal setting (asymptotically high resolution of codebook entries), the maximum attainable absolute array factor $\sqrt{N_x N_y}$ could be achieved at any user position. However, this is not realizable in practice, where practical considerations limit the codebook size. Motivated by the above considerations, we define a new distortion measure between the ith vector $\boldsymbol{\psi}_i$ and the jth codebook entry $\boldsymbol{\chi}_j$:

$$d(\boldsymbol{\psi}_i, \boldsymbol{\chi}_j) = \sqrt{N_x N_y} - |A_{\text{tx}}^{(x)}(\boldsymbol{\psi}_i, \boldsymbol{\chi}_j)||A_{\text{tx}}^{(y)}(\boldsymbol{\psi}_i, \boldsymbol{\chi}_j)|, \tag{4.19}$$

where $|A_{\text{tx}}^{(x)}(\boldsymbol{\psi}_i, \boldsymbol{\chi}_j)|$ and $|A_{\text{tx}}^{(y)}(\boldsymbol{\psi}_i, \boldsymbol{\chi}_j)|$ are the per-dimension absolute array factors. For example, if the CPO technique (directional beam) is employed, the per-dimension array factors are

$$\begin{aligned}
\left|A_{\text{tx}}^{(x)}(\boldsymbol{\psi}, \boldsymbol{\chi})\right| &= \left|A_{\text{tx}}^{(x)}([\phi\ \theta]^{\text{T}}, [\phi_0\ \theta_0]^{\text{T}})\right| \\
&= \frac{1}{\sqrt{N_x}}\left[\frac{\sin\left(\frac{N_x\pi}{2}\left(\cos(\phi)\sin(\theta) - \cos(\phi_0)\sin(\theta_0)\right)\right)}{\sin\left(\frac{\pi}{2}\left(\cos(\phi)\sin(\theta) - \cos(\phi_0)\sin(\theta_0)\right)\right)}\right], \\
\left|A_{\text{tx}}^{(y)}(\boldsymbol{\psi}, \boldsymbol{\chi})\right| &= \left|A_{\text{tx}}^{(y)}([\phi\ \theta]^{\text{T}}, [\phi_0\ \theta_0]^{\text{T}})\right| \\
&= \frac{1}{\sqrt{N_y}}\left[\frac{\sin\left(\frac{N_y\pi}{2}\left(\sin(\phi)\sin(\theta) - \sin(\phi_0)\sin(\theta_0)\right)\right)}{\sin\left(\frac{\pi}{2}\left(\sin(\phi)\sin(\theta) - \sin(\phi_0)\sin(\theta_0)\right)\right)}\right].
\end{aligned} \tag{4.20}$$

In other words, the distortion between ith user and jth beam steering angle is defined as *the decrease in absolute array factor*, relative to the maximum achievable value (in the ideal setting). This will subsequently take into account the direction-dependent beam width, and thus users are assigned to clusters at which the transmit array factor is maximized.

Next, a variant of the K-means algorithm is derived to optimize the codebook of beam steering angles. Each algorithm iteration (analogous to the Lloyd iteration in the compression setting) consists of the following two main steps:

1. Fix the beam steering angles codebook $\{\boldsymbol{\chi}_j\}$, and assign each user to the steering angle incurring the least distortion. Let $\mathcal{S}_j$ be the set of users assigned to steering angle $\boldsymbol{\chi}_j$, also called the jth cluster. The clustering partition is given by the (generalized) nearest neighbor rule. Specifically, cluster j is given by:

$$\mathcal{S}_j = \{i \colon d(\boldsymbol{\psi}_i, \boldsymbol{\chi}_j) \leq d(\boldsymbol{\psi}_i, \boldsymbol{\chi}_k),\ \forall k \neq j\}\,. \tag{4.21}$$

2. Fix the clustering partition $\{\mathcal{S}_j\}$ and optimize the steering angles codebook to minimize the average distortion. Specifically, adjust each steering angle $\boldsymbol{\chi}_j$ so that it minimizes its cluster's average distortion:

$$\boldsymbol{\chi}_j = \arg\min_{\boldsymbol{\chi}} \frac{1}{|\mathcal{S}_j|} \sum_{i \in \mathcal{S}_j} d(\boldsymbol{\psi}_i, \boldsymbol{\chi}), \quad j = 1, 2, \ldots, N_b, \tag{4.22}$$

where $|\cdot|$ denotes the set cardinality. A necessary condition for optimality can be obtained by setting the gradient with respect to $\boldsymbol{\chi}$ to zero:

$$\frac{1}{|\mathcal{S}_j|} \sum_{i \in \mathcal{S}_j} \frac{\partial}{\partial \boldsymbol{\chi}} d(\boldsymbol{\psi}_i, \boldsymbol{\chi}) = 0 \quad, j = 1, 2, \ldots, N_b, \tag{4.23}$$

Numerical search with finite resolution in the 2D angular space or gradient descent algorithms with multiple initialization points or both can be employed to solve the minimization problem of (4.22). Note that the traditional K-means "centroid" rule which computes each codebook entry as the cluster sample average,

$$\boldsymbol{\chi}_j = \frac{1}{|\mathcal{S}_j|} \sum_{i \in \mathcal{S}_j} \boldsymbol{\psi}_i, \tag{4.24}$$

is valid for the squared error distortion measure, where (4.23) simplifies to (4.24), but that is not the case for our distortion measure.

In every "Lloyd iteration", one can evaluate the average distortion as,

$$D = \frac{1}{|\mathcal{S}_1 \cup \mathcal{S}_2 \cdots \cup \mathcal{S}_{N_b}|} \sum_{j=1}^{N_b} \sum_{i \in \mathcal{S}_j} d(\boldsymbol{\psi}_i, \boldsymbol{\chi}_j), \tag{4.25}$$

where $\cup$ denotes the set union operation.

It is straightforward to show that the two steps of the main iteration guarantee that D is monotonically non-increasing, and in fact monotonically decreasing until convergence (under mild assumptions regarding treatment of ties in the nearest neighbor step). Additionally, note that as $N_b \to \infty$, the codebook average distortion asymptotically vanishes, i.e., $D \to 0$, which is consistent with standard requirements of distortion measures and represents the "ideal setting" at the limit of high resolution.

4.3.3 DA-based Beam Steering

One major drawback of the classical K-means clustering algorithm, is that it only guarantees convergence to a locally optimal solution, while in many cases of interest the cost surface is riddled with poor local minima. Deterministic annealing has been demonstrated to be highly effective in avoiding poor local minima, when conventional distortion measures are used, and has become the method of choice in numerous disciplines [6].

Unlike the K-means algorithm, DA considers a probabilistic assignment between the users' angular vectors $\{\boldsymbol{\psi}_i\}$ and codebook entries or cluster centroids $\{\boldsymbol{\chi}_j\}$. Let the association probabilities be denoted as $p(j|i)$. In this case, the overall average distortion in the system due to quantization of beam pointing angle is given by the expectation,

$$D = \sum_i \sum_j p(j|i)p(i)d(\boldsymbol{\psi}_i, \boldsymbol{\chi}_j), \tag{4.26}$$

where $p(i)$ is the prior probability of a user positioned at the angular vector $\boldsymbol{\psi}_i$. Note that minimizing the distortion with respect to the free parameters $\{\boldsymbol{\chi}_j, p(j|i)\}$ would immediately lead to hard association between the user and the nearest codebook entry, where the term "nearest" is used in the sense of the distortion measure. Instead, the distortion is minimized subject to an imposed level of randomness, which is naturally measured by Shannon's entropy H. Hence, the Lagrangian function to be minimized can be written as,

$$\mathcal{L} = D - TH, \tag{4.27}$$

where,

$$H = -\sum_i \sum_j p(j|i)p(i) \log \left(p(j|i)p(i)\right), \tag{4.28}$$

and T ("temperature") is the Lagrangian parameter. Next, an iterative approach, which is an appropriately designed random relative of the K-means algorithm, is employed to minimize the Lagrangian function:

1. Initialize temperature, $T = T_{\max}$ and beam steering angles' codebook $\{\boldsymbol{\chi}_j\}$.

2. Fix the codebook $\{\boldsymbol{\chi}_j\}$ and find the random clustering partition (i.e., probabilistic assignment of users to steering angles) which minimizes the Lagrangian cost:

$$\{p(j|i)\} = \underset{\{p(j|i)\}}{\arg\min}\, \mathcal{L}, \qquad \forall i, \forall j \tag{4.29}$$

Note that the solution must further impose the constraint $\sum_j p(j|i) = 1, \forall i$, which directly yields a random relative of the nearest neighbor rule, given by the Gibbs distribution:

$$p(j|i) = \frac{\exp\left(-\frac{d(\boldsymbol{\psi}_i,\boldsymbol{\chi}_j)}{T}\right)}{Z_i}, \tag{4.30}$$

where the normalization constant is

$$Z_i = \sum_j \exp\left(-\frac{d(\boldsymbol{\psi}_i,\boldsymbol{\chi}_j)}{T}\right), \tag{4.31}$$

sometimes called the partition function in physics.

3. Fix the random clustering partition, $\{p(j|i)\}$ and optimize the steering angles codebook to minimize the Lagrangian cost. Specifically,

$$\{\boldsymbol{\chi}_j\} = \arg\min_{\{\boldsymbol{\chi}_j\}} \mathcal{L} = \arg\min_{\{\boldsymbol{\chi}_j\}} D, \tag{4.32}$$

where we used the fact that the entropy is determined by the (fixed) clustering partition, and hence can be discarded from $\mathcal{L}$ in this step. Noting further that D is additive in the contributions of individual steering angles we obtain:

$$\boldsymbol{\chi}_j = \arg\min_{\boldsymbol{\chi}} \sum_i p(j|i)p(i)d(\boldsymbol{\psi}_i,\boldsymbol{\chi}), \tag{4.33}$$

or as necessary condition for optimality, the random relative of the centroid rule:

$$\sum_i p(j|i)p(i)\frac{\partial}{\partial\boldsymbol{\chi}}d(\boldsymbol{\psi}_i,\boldsymbol{\chi}) = 0 \quad, j = 1, 2, \ldots, N_b, \tag{4.34}$$

Numerical search with finite resolution in the 2D angular space or gradient descent algorithms with multiple initialization points or both can be employed to solve the minimization problem of (4.33).

4. Check if convergence condition satisfied, else go to step 2.

5. Cool the system, e.g., $T = \alpha T$, with $\alpha < 1$. If the prescribed minimum temperature is reached, then terminate the algorithm.

6. Perturb the codebook entries to check for possible splitting of codebook centroids, also known as phase transition, then go to step 2.

At $T = 0$, the DA algorithm degenerates to the K-means algorithm, however the annealing process until then eliminates the sensitivity to initialization. In step 4, convergence can be checked by comparing $\frac{\Delta\mathcal{L}}{\mathcal{L}}$ to a convergence threshold. It is important to note that by gradual cooling, the system undergoes a series of phase transitions at corresponding "critical temperatures", in analogy to physical systems, wherein the cardinality of the codebook grows. See [6] for extensive analysis of DA's sequence of phase transitions through which the cardinality of the codebook grows, as well as for demonstration that the algorithm is invariant to initialization. The derived DA algorithm, for the optimization of beam steering directions problem, is outlined in Fig. 4.6.

4.4 Experimental Results

The beam steering angles optimization algorithms are first evaluated in terms of the average and the 10th percentile of the array factor seen across all users. The competing beam placement schemes are: *i*) DFT-based beam steering as defined in [68, 66] *ii*) Uniform beam steering as employed in [21, 67], *iii*) Heuristic non-uniform beam steering for which preliminary results are published in [62], *iv*) k-means-based beam steering for which preliminary results are published in [26], and finally *v*) DA-based beam steering proposed in Section 4.3. The former two (DFT-based and uniform beam steering) serve as baseline reference for the comparison, and the latter three are the proposed schemes

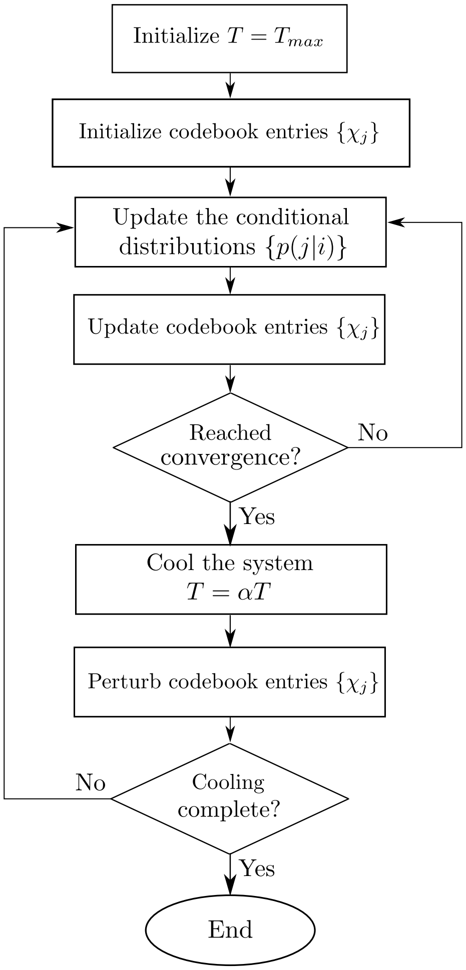

Initialize $T = T_{max}$
Initialize codebook entries $\{\chi_j\}$
Update the conditional distributions $\{p(j|i)\}$
Update codebook entries $\{\chi_j\}$
Reached convergence?
No
Yes
Cool the system $T = \alpha T$
Perturb codebook entries $\{\chi_j\}$
No
Cooling complete?
Yes
End

Figure 4.6: Flow chart of the proposed DA-based beam steering algorithm. Reprinted, with permission, from [63] © 2021 IEEE.

presented in this chapter. For uniform beam steering and heuristic non-uniform beam steering algorithms, the number of elevation beams $N_b^\theta \in \{1, 2, \ldots, 16\}$, and the selected value of N_b^θ is numerically optimized for each codebook size N_b, to maximize the average array factor. The gNBs are assumed to be equipped with 32 × 8 planar arrays. The performance is evaluated for a variety of UE distributions. First, the UE angles, seen from the gNB local co-ordinate system, are assumed to be uniformly distributed over the field-of-view ($\overline{\phi} = 180°, \overline{\theta} = 90°$). Fig. 4.7 depicts the average power array factor and its 10th percentile in dB versus the beam codebook size. The proposed DA-based

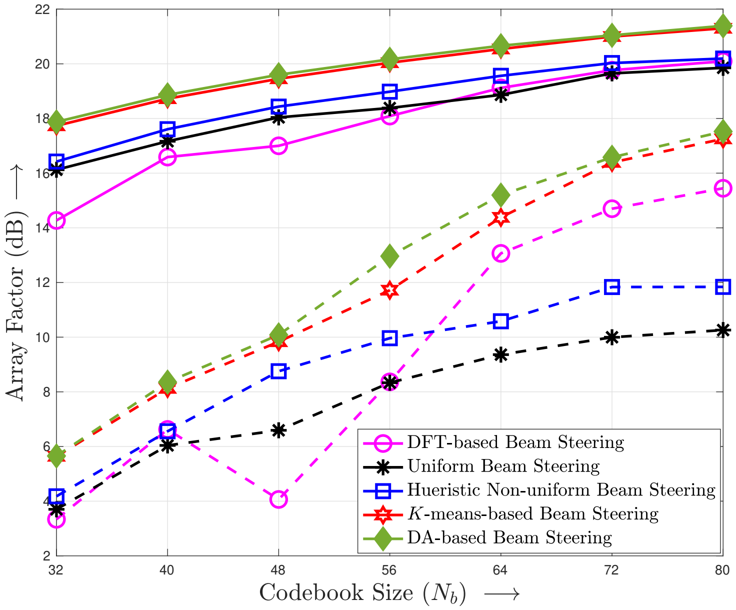


Figure 4.7: The UEs' angles are assumed uniformly distributed. Average (sold lines) and 10th percentile (dashed lines) of power array factor for competing beam steering design methods. Reprinted, with permission, from [63] © 2021 IEEE.

beam steering approach offers gains of up to 4 dB and 7.2 dB, in the average power array factor and its 10th percentile, respectively, when compared with the baseline methods. Note that the codebooks are designed to maximize the average power array factor over all users, which sometimes results in a degraded 10th percentile performance, as seen for DFT-based codebook at $N_b = 48$.

Next, to test the approaches in a less simplistic scenario, the users' angles were distributed as a mixture of bi-variate Gaussians in the angular field-of-view ($\overline{\phi} = 180°, \overline{\theta} = 90°$). The underlying premise of this model is that users often tend to cluster around certain locations such as shops, traffic lights, bus stops, etc. The average power array factor and its 10th percentile are plotted for this scenario in Fig. 4.8. Note that in this case, the proposed DA-based codebook design offers up to 6 dB and 12.5 dB improvements in the average power array factor and its 10th percentile, respectively, when compared with uniform or DFT-based beam steering approaches.

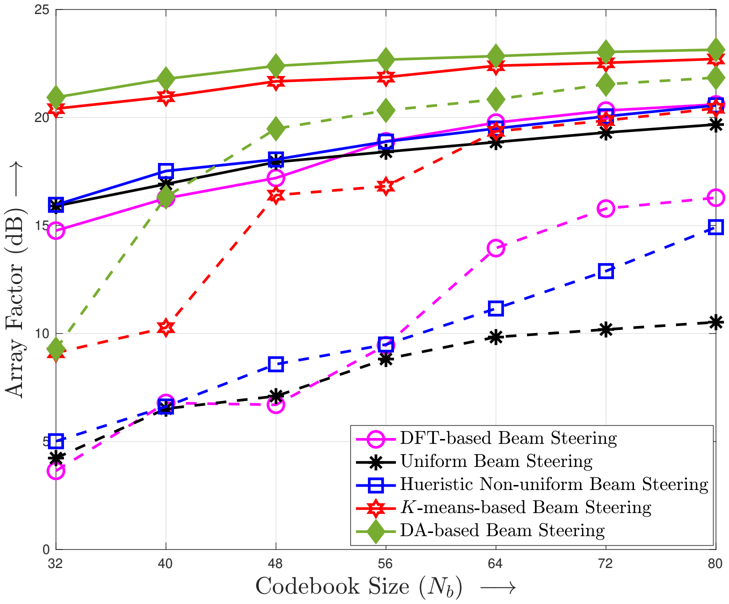


Figure 4.8: The UEs' angles are assumed to be distributed as a mixture of bi-variate Gaussians. Average (sold lines) and 10th percentile (dashed lines) of power array factor for competing beam steering design methods. Reprinted, with permission, from [63] © 2021 IEEE.

We next consider the simple UE distribution suggested in [64] for outdoor Urban Micro (UMi) system scenarios, where UE positions are uniformly distributed on the horizontal plane. Under this UE distribution assumption, two network layouts were simulated: i) The gNBs are placed in a Manhattan-like grid, and sectorized into 4 sectors, or ii) The gNBs are placed in a hexagonal grid, and sectorized into 3 sectors. The inter-site distance for both network layouts is 200 m. The average power array factor and its 10th percentile are plotted for this scenario in Fig. 4.9 and Fig. 4.10. The proposed DA-based design method outperforms the baseline methods by up to 5.5 dB and 13 dB in the average power array factor and its 10th percentile, respectively. It is noteworthy that the DA algorithm offers larger gains over the baseline schemes when the UE angles are non-uniformly distributed. This is to be expected because DA can adapt and exploit irregularities in the UE distribution, for example by placing more beams at the angular

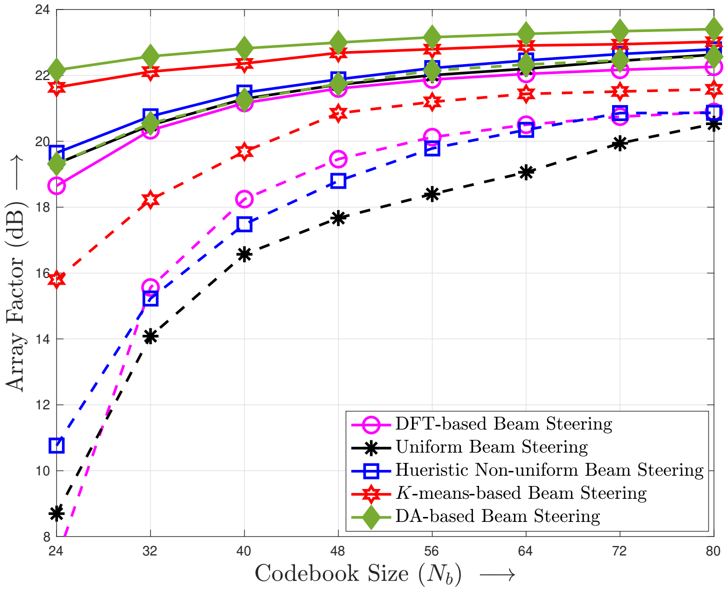


Figure 4.9: The UEs' positions are uniformly distributed across the horizontal plane in a Manhattan-like network grid. Average (sold lines) and 10th percentile (dashed lines) of power array factor for competing beam steering design methods. Reprinted, with permission, from [63] © 2021 IEEE.

directions pointing at areas that are more densely populated by UEs. This flexibility is not available to the uniform beam steering method or the DFT-based beam steering method, thus putting them at significant disadvantage in likely scenarios of non-uniform UE distribution.

To provide further evidence for the practical benefits of the proposed beam placement algorithms, a full-fledged system simulation was carried out for outdoor cellular 5G settings. The simulation assumptions are summarized in Table 4.1. A random TDM scheduler is employed per base station sector, where each sector schedules randomly one of the active users. For each gNB-UE link, the transmit beam that maximize the received SNR is enabled, where beams are selected from a predefined beamforming codebook that is designed offline. The average SNR performance, calculated using (4.9), is depicted in Fig. 4.11. The proposed beam steering algorithms offer up to 4.5 dB and 6.5 dB improvements in the average SNR seen over all users for Manhattan-like, or hexagonal network

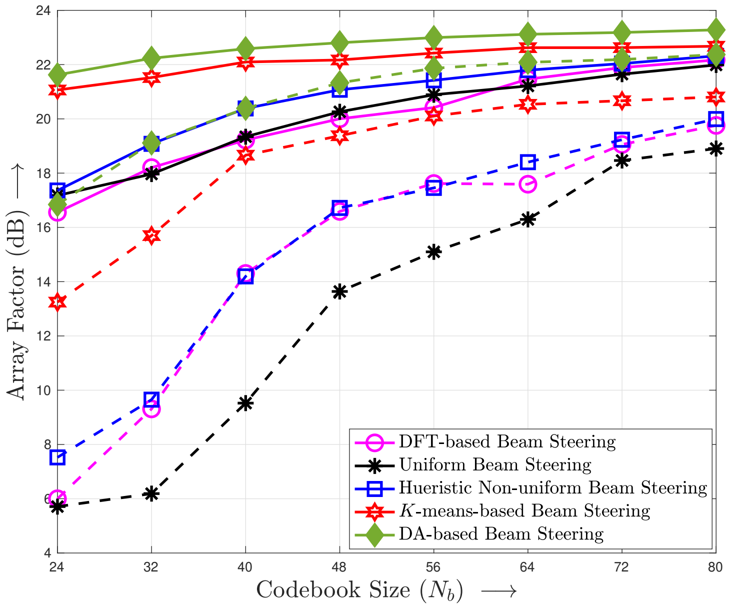


Figure 4.10: The UEs' positions are uniformly distributed across the horizontal plane in a hexagonal network grid. Average (sold lines) and 10th percentile (dashed lines) of power array factor for competing beam steering design methods. Reprinted, with permission, from [63] © 2021 IEEE.

grids, respectively. Note that while the simulation is for the simple channel (consisting of one ray), the results and conclusions are readily extendable to more complex channels. It is further important to emphasize that the performance gains are achieved at no operational cost, because typical beam steering codebooks are designed offline and stored in memory. Thus, the operational complexity of deploying any of the competing codebooks is the same. On the other hand, during their design phase, both the k-means and DA-based algorithms require prior information (or assumptions) on user statistics, which is implicit in the training data used. If the system experiences a dynamic user distribution, DA-based and k-means algorithm would require additional operational complexity in order to track user statistics and update codebooks accordingly.

Table 4.1: Summary of System Simulation Assumptions. Reprinted, with permission, from [63] © 2021 IEEE

Metric	**Value**
System Scenario	UMi
Direction of Transmission	Downlink (gNB to UE)
Carrier Frequency & Bandwidth	$f_c = 28$ GHz, $B = 100$ MHz
Sub-carrier Spacing	$\Delta f = 120$ kHz
Number of Clusters & Rays	$N_\text{c} = 1$ and $M_\text{r} = 1$
Path-loss Model	3GPP model in [64]
Network Layout	Manhattan-like or Hex. grid
Inter-site Distance	$D = 200$ meters
Number of gNBs	25 sites or 19 sites
Number of UEs per site	10 UEs
Avg. TX Power Per PA	23 dBm [62, 69]
gNB Antenna Array Size	$N_\text{tx} = 256$ elements
gNB Element Power Model	According to [64]
gNB Max. Element Gain	$G_\text{tx} = 8$ dBi
UE Antenna Array Size	$N_\text{rx} = 1$ element
UE Element Power Model	Omni Antenna Element
UE Max. Element Gain	$G_\text{rx} = 0$ dBi
UE Noise Figure	$10\log_{10}(F_n) = 8$ dB

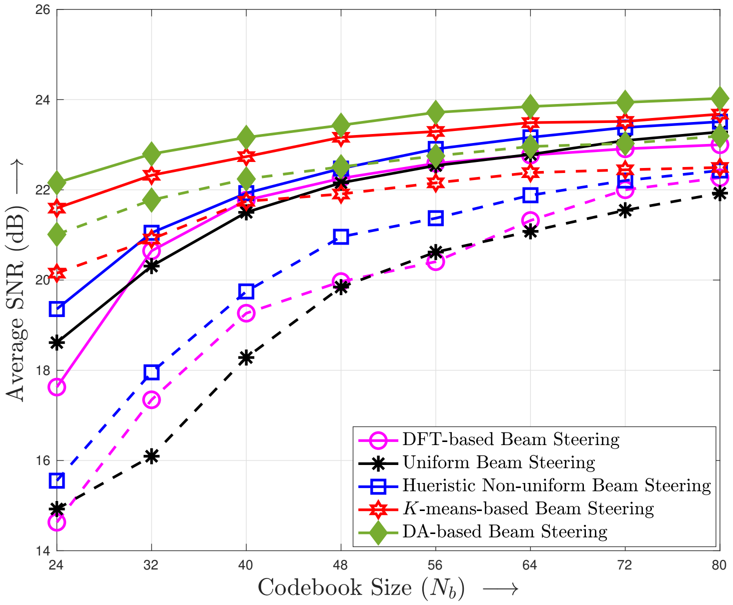


Figure 4.11: System SNR performance for competing beam steering design methods in Manhattan-like network gird (solid lines) or hexagonal network grid (dashed lines). The UEs' positions are uniformly distributed across the horizontal plane. Reprinted, with permission, from [63] © 2021 IEEE.

Chapter 5

Reformulation of Supervised Learning System into Stochastic Rate-Distortion Framework

5.1 A Reformulated System Model for Efficient Machine Learning

We propose to reformulate the supervised learning system within a rate-distortion framework. The main question posed is: what is the minimum amount of *information (in bits)* that can be extracted by a learning system to produce desired outputs at a specified *accuracy or distortion requirement*? The proposed system model, depicted in Fig. 5.1, is composed of a set of L simpler learning system, e.g., DNN, with L denoting the length of the input's super-symbols. The DNNs are tasked to extract the *minimum* amount of information necessary to enable reading the desired outputs from a codebook at a required accuracy. Let S be the number of layers in the overall equivalent DNN, where

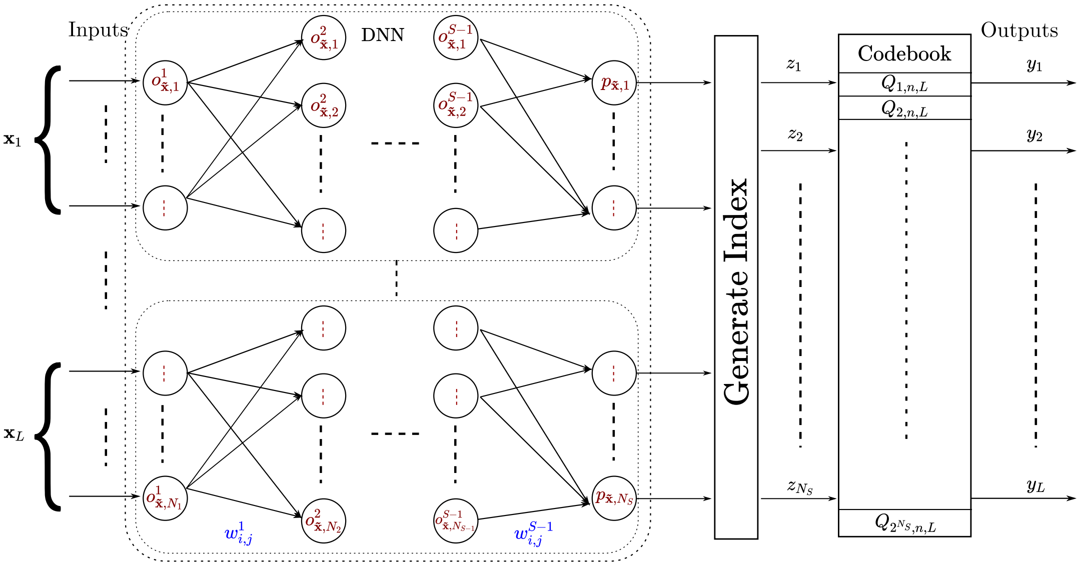

Inputs
DNN
Outputs
$\mathbf{x}_1$
$\mathbf{x}_L$
$o^1_{\tilde{\mathbf{x}},1}$
$o^2_{\tilde{\mathbf{x}},1}$
$o^2_{\tilde{\mathbf{x}},2}$
$o^{S-1}_{\tilde{\mathbf{x}},1}$
$o^{S-1}_{\tilde{\mathbf{x}},2}$
$p_{\tilde{\mathbf{x}},1}$
$o^1_{\tilde{\mathbf{x}},N_1}$
$o^2_{\tilde{\mathbf{x}},N_2}$
$o^{S-1}_{\tilde{\mathbf{x}},N_{S-1}}$
$p_{\tilde{\mathbf{x}},N_S}$
$w^1_{i,j}$
$w^{S-1}_{i,j}$
Generate Index
z_1
z_2
z_{N_S}
Codebook
$Q_{1,n,L}$
$Q_{2,n,L}$
$Q_{2^{N_S},n,L}$
y_1
y_2
y_L

Figure 5.1: System model of the proposed reformulated supervised learning system within rate-distortion framework.

the number of nodes in each of the layers is N_s, $s \in \{1, 2, \ldots, S\}$. Furthermore, let $w^s_{i,j}$, $i \in \{1, 2, \ldots, N_s\}$, $j \in \{1, 2, \ldots, N_{s+1}\}$, be the weight associated with the link connecting the i-th node in a layer s to the j-th node in the subsequent layer. The k-th input vector in the training set consists of L sub-vectors or super symbols, each of length M, i.e., $\mathbf{x}(k) = [\tilde{\mathbf{x}}_1(k), \ldots, \tilde{\mathbf{x}}_L(k)]$, where $\tilde{\mathbf{x}}_\ell(k)$ is the ℓ-th M-length sub-vector, with $k \in \{1, 2, \ldots, K\}$. Source sub-vectors $\{\tilde{\mathbf{x}}_\ell(k)\}$ are generated independently, with the stationary and ergodic M-th joint distribution $P_M = P_{\tilde{\mathbf{x}}} = \{P_M(\tilde{\mathbf{x}}) : \tilde{\mathbf{x}} \in \mathcal{X}^M\}$. The output at each node of the DNN at layer s, for the input vector $\mathbf{x}$, denoted as $o^s_{x,j}$, is computed as,

$$o^s_{\mathbf{x},j} = f(q^s_{\mathbf{x},j}), \quad q^s_{\mathbf{x},j} = \sum_{i=1}^{N_{s-1}} w^{s-1}_{i,j} o^{s-1}_{\mathbf{x},i}, \tag{5.1}$$

where $f(\cdot)$ is the activation function, $o^1_{\mathbf{x},j} = x_j$, and x_j is the j-th element in the input vector $\mathbf{x}$. For ease of notation, denote the outputs at the last DNN layer as $\{p_{\mathbf{x},j} = o^S_{\mathbf{x},j}\}$, which are each bounded between 0 and 1.

5.2 The Mechanism of the Proposed Reformulated Learning System Model

In this section, we explain how an output class is generated for the k-th input source vector $\mathbf{x}(k)$. First, the DNN produce an index vector $\mathbf{z}_{\mathbf{x}(k)}$, which is used afterwards to retrieve a codeword that represents the given input from the codebook. Denote the codebook obtained, after the training stage is completed, as $\mathcal{C} = \{\mathbf{y}(j), j \in \{1, \ldots, 2^{N_S}\}\}$. Hence, the codeword that represents the k-th input vector, is $\mathbf{y}(\mathbf{z}_{\mathbf{x}(k)})$. We assume that the codewords are of width L, i.e., the ℓ-th sub-vector in the k-th input pattern $\mathbf{x}_\ell(k)$ is mapped to ℓ-th sample y_ℓ in the codeword. For example, if we consider a digit classification learning system, where the system's goal is to classify a sequence of L images of handwritten digits, each into one of the predefined digit classes, then the ℓ-th letter in the codeword, with $\ell = 1, \ldots, L$, indicates the classified digit of the ℓ-th input sub-vector. In this example, the reproduction alphabet $\mathcal{Y}$ would obviously be the space of all possible digits, i.e., $\mathcal{Y} = \{0, 1, \ldots, 9\}$. One should now train both components of the system, i.e., the DNN and the codebook, to generate outputs that minimize a specified cost function averaged over all input patterns. The training of the proposed system operates in an iterative manner, i.e., first a DNN is trained for a given codebook, then a codebook is trained for a fixed DNN, and these two steps are repeated until convergence.

Note that the output of the DNN are soft in nature during the training stage, i.e., it produces a *probability distribution* over codebook locations, unlike the majority of learning system where a hard output is obtained for each input, specifying the output class with probability one. However, after the training stage is completed, the soft outputs of the DNN in the proposed learning system converges to hard outputs, i.e., the probability distribution induced by the probability vector $[p_{\mathbf{x}(k),1} \ldots p_{\mathbf{x}(k),N_S}]$ collapses onto a single codeword for any input pattern. The mechanism of such convergence is governed by the deterministic annealing framework and is explained in detail in Section 5.4. Similarly, the codewords in the codebook are soft in nature during the training stage, i.e., the codebook contains a set of L-dimensional distributions $Q_{\mathbf{z},n,L} = \{[Q^{(1)}_{\mathbf{z},n,L}(y), \ldots, Q^{(L)}_{\mathbf{z},n,L}(y)] : y \in \mathcal{Y}\}$, where $\mathbf{z} \in \mathcal{B}^{N_S}$ is the codebook location, and n is the training iteration index. This set of distributions specifies the probability of an output letter $y \in \mathcal{Y}$ at the ℓ-th codeword positions and any the $\mathbf{z}$-th codebook location. After the training stage is completed, the codebook distributions are replaced with hard codewords, and the mechanism of this replacement is illustrated in detail in Section 5.3. In the next sections, the training techniques of the codebook and the DNN are illustrated.

5.3 Codebook Regeneration within the NTS Framework

First, it is worth noting that an optimal length-constrained codebook, will maximize the end-to-end accuracy achieved by the learning system. Due to the soft nature of the DNN outputs, the DNN ultimately performs a probabilistic mapping of the input vectors into codebook locations according to the distribution $P_{\mathbf{z}|\mathbf{x}(k)}(\mathbf{z}) = \prod_{j=1}^{N_S} p^{z_j}_{\mathbf{x}(k),j}(1-p_{\mathbf{x}(k),j})^{(1-z_j)}$. For this reason, in an optimal system, each codebook location must therefore have its unique codewords' statistics to best represent, in the minimum distortion sense, the source vectors that are mapped or clustered into that codebook location. Hence, in the training stage, we assume that there exists a set of temporarily mini-codebooks at each location $\mathbf{z}$, denoted as $\mathcal{C}_{\mathbf{z}}$, each is used to capture the statistics of the input vectors that are clustered to location $\mathbf{z}$. The statistics of the codewords in $\mathcal{C}_{\mathbf{z}}$ are recorded in the form of location-dependent codeword distributions $Q_{\mathbf{z},n,L}$. Furthermore, define $Q_{n,L} = \{Q_{n_L}(y), y \in \mathcal{Y}\}$ as the memoryless codebook reproduction distribution that is used to generate codewords in any of the codebook locations $\mathbf{z}$, which consequently populate the mini-codebooks $\{\mathcal{C}_{\mathbf{z}}\}$. Throughout this paper, we assume that the learning system is supervised, let $y^*_\ell(k)$ be the label or true output for the ℓ-th sub-vector in the k-th input pattern, i.e., $\tilde{\mathbf{x}}_\ell(k)$. In this setup, we denote $\rho : \mathcal{Y}\times\mathcal{Y} \to [0, \infty)$ for an arbitrary non-negative distortion measure or cost function, where the distortion over vectors is assumed as average distortion over sub-vectors, i.e.,

$$\rho\left(\mathbf{x}(k), \mathbf{y}\right) = \frac{1}{L}\sum_{\ell=1}^{L} \rho(\tilde{\mathbf{x}}_\ell(k), y_\ell) = \frac{1}{L}\sum_{\ell=1}^{L} \rho(y^*_\ell(k), y_\ell). \tag{5.2}$$

In order to train or update the set distributions $Q_{\mathbf{z},n,L}$ in iteration n, first the mini-codebooks $\{\mathcal{C}_{\mathbf{z}}\}$ need to be populated with codewords as follows: For a given DNN, and the k-th input source word $\mathbf{x}(k)$, the system rolls the dice according to $P_{\mathbf{z}|\mathbf{x}(k)}(\mathbf{z})$, to generate the r-th index vector $\mathbf{z}_{\mathbf{x}(k)}(r)$. Then, the system generates an r-th random codeword $\mathbf{y}(r)$ according to the current memoryless codebook reproduction distribution $Q_{n,L}$. If $\mathbf{y}(r)$ d-matches the current source vector $\mathbf{x}(k)$, i.e., $\mathbf{y}(r)$ satisfies $\rho(\mathbf{x}(k), \mathbf{y}(r)) \leq d$, then this codeword is recorded in a "*mini-codebook*" at location $\mathbf{z}_{\mathbf{x}(k)}(r)$, and the system proceeds to the next training input pattern. If the d-match event is not achieved, the system generates a new index vector $\mathbf{z}_{\tilde{\mathbf{x}}(k)}(r+1)$ and a new random codeword $\mathbf{y}(r+1)$ until a d-matching codeword is found and recorded in one of the mini-codebooks. After all the training sequences have been processed, i.e., after K source words have been processed, the system updates the set of distributions $\{Q_{\mathbf{z},n,L}, \forall \mathbf{z} \in \mathcal{B}^{2^{N_S}}\}$ by finding the *maximum likelihood* distributions that would have generated the set of codewords that d-matched the source words in each of the mini-codebooks $\{\mathcal{C}_{\mathbf{z}}\}$. Let $\mathcal{C}_{\mathbf{z}} = \{\mathbf{y}^{(\mathbf{z})}(j_{k_{\mathbf{z}}}), k_{\mathbf{z}} \in \{1, \ldots, K_{\mathbf{z}}\}\}$, and $\sum_{\mathbf{z}} K_{\mathbf{z}} = K$, be the set of codewords that have d-matched source words at codebook location $\mathbf{z}$.

Lemma 4: The maximum likelihood codebook reproduction distribution, that would have generated the set of d-matching codewords at every codebook locations, is computed as, i.e.,

$$\hat{Q}^{\mathrm{ML}} = Q_{\mathbf{z},n+1,L} = \{[Q^{(1)}_{\mathbf{z},n,L}(y), Q^{(2)}_{\mathbf{z},n,L}(y), \ldots, Q^{(L)}_{\mathbf{z},n,L}(y)]\} \tag{5.3}$$

$$Q^{(\ell)}_{\mathbf{z},n,L}(y) = \frac{1}{K_{\mathbf{z}}} \sum_{k=1}^{K_{\mathbf{z}}} \mathbb{I}_{y^{(\mathbf{z})}_{\ell}(j_{k_{\mathbf{z}}}),y}, \tag{5.4}$$

where the indicator function $\mathbb{I}_{y^{(\mathbf{z})}_{\ell},y}$ equals one, if the ℓ-th element in the d-matching codeword $\mathbf{y}^{(\mathbf{z})}(j_{k_{\mathbf{z}}})$, denoted as $y^{(\mathbf{z})}_{\ell}(j_{k_{\mathbf{z}}})$, is equal to y, and equals zero otherwise. Note that Lemma 4 is a direct generalization of Lemma 1 for nonidentical distributions across the L codeword symbols. Similarly, the system updates the codebook reproduction distribution $Q_{n,L}$, that generates i.i.d. code letters, by taking the average of all location-dependent codebook distributions (across codebook locations $\mathbf{z}$ and super-symbol positions ℓ, i.e.,

$$Q_{n,L}(y) = \frac{1}{LK} \sum_{\mathbf{z}} \sum_{k=1}^{K_{\mathbf{z}}} \sum_{\ell=1}^{L} \mathbb{I}_{y^{(\mathbf{z})}_{\ell}(j_{k_{\mathbf{z}}}),y}. \tag{5.5}$$

After the training stage is completed, each of the codebook reproduction distributions $Q_{\mathbf{z},n,L}$ is replaced with the most likely codeword to be generated from that distribution. Next, we show the asymptotic performance of the proposed codebook that is designed within an NTS framework.

Corollary 4 *The sequence of codebook reproduction distributions $Q_{n,L}$ converges asymptotically as $K \to \infty$, $n \to \infty$, $L \to \infty$, and for fixed sub-vector length M, to the distribution $Q^*_{P_{\tilde{\mathbf{x}}},d}$ in probability, where $Q^*_{P_{\tilde{\mathbf{x}}},d}$ is the optimal codebook reproduction distribution that achieves the M-th order rate-distortion bound $R(P_{\tilde{\mathbf{x}}}, d)$.*

Note that Corollary 4 immediately follows Corollaries 1 and 2 on convergence of NTS algorithm for sources with memory. Corollary 4 establishes that, asymptotically, the stochastic codebook tends to a codebook generated by the optimal distribution that achieves the rate-distortion bound. It is hence (by Shannon) the shortest codebook possible guaranteed to contain an entry, within the required distortion (accuracy) for randomly generated source examples. It thus minimizes the number of (output) bits learned by the DNN. In other words, a smaller codebook implies a simpler network that is used to index this codebook and hence a more efficient learning system.

5.4 Deep Neural Network Training within the Deterministic Annealing Framework

Now, we turn our attention to the other part of the system training. For a given set of distributions that represents the statistics of the d-matching codewords at locations $\mathbf{z} \in \mathcal{B}^{N_S}$, i.e., $\{Q_{\mathbf{z},n,L}\}$, how is the DNN trained? To answer this question, first let us define the average end-to-end distortion seen over all input patterns in the training set,

$$\begin{aligned} D = & \sum_{k=1}^{K} P_{\mathbf{x}}(\mathbf{x}(k)) D(\mathbf{x}(k)) = \\ & \sum_{k=1}^{K} P_{\mathbf{x}}(\mathbf{x}(k)) \sum_{\mathbf{z}\in\mathcal{B}^{N_S}} P_{\mathbf{z}|\mathbf{x}(k)}(\mathbf{z}) \left(\frac{1}{L} \sum_{\ell=1}^{L} \sum_{y\in\mathcal{Y}} Q^{\ell}_{\mathbf{z},n,L}(y) \rho\left(y^*_\ell(k), y\right) \right), \end{aligned} \tag{5.6}$$

where $P_{\mathbf{x}}(\mathbf{x}(k))$ and $P_{\mathbf{z}|\mathbf{x}(k)}(\mathbf{z})$ denote the probability of generating source vector $\mathbf{x}(k)$ and the conditional probability of generating index vector $\mathbf{z}$, respectively. The latter is computed as $P_{\mathbf{z}|\mathbf{x}(k)}(\mathbf{z}) = \prod_{j=1}^{N_S} p_{\mathbf{x}(k),j}^{z_j} (1 - p_{\mathbf{x}(k),j})^{(1-z_j)}$, where $p_{\mathbf{x}(k),j}$ are the DNN outputs, and z_j is the j-th element in $\mathbf{z}$. The classical back propagation methodology in [70] performs gradient descent on the weights, i.e., $\Delta w^s_{i,j} \propto -\frac{\partial D(\mathbf{x}(k))}{\partial w^s_{i,j}}$. However, back-propagation's gradient decent only guarantees convergence to a locally optimal solution, while in many cases of interest the cost surface is riddled with poor local minima. Multiple heuristic methods have been proposed over the decades to combat this difficulty [71, 72, 73, 74].

They range from repeated optimization with different initialization, to modification of weight update steps based on second derivatives, or weights update based on momentum and more. However, significant gains are yet to be recouped by a principled attack on the problem. This motivates the use of powerful optimization tools, i.e. the DA.

Here, DA embeds DNN training within a rate-distortion (or statistical physics) optimization framework. The Lagrangian or free energy is given by,

$$\mathcal{L} = D - TH = \sum_{k=1}^{K} P_{\mathbf{x}}(\mathbf{x}(k))\mathcal{L}(\mathbf{x}(k)), \tag{5.7}$$

$$H = -\sum_{k=1}^{K} \sum_{\mathbf{z}\in\mathcal{B}^{N_S}} P_{\mathbf{x}}(\mathbf{x}(k))P_{\mathbf{z}|\mathbf{x}(k)}(\mathbf{z}) \log\left(P_{\mathbf{x}}(\mathbf{x}(k))P_{\mathbf{z}|\mathbf{x}(k)}(\mathbf{z})\right). \tag{5.8}$$

For the ease of notation, the index of the input k will be dropped in subsequent analysis, hence, $\mathbf{x} = \mathbf{x}(k)$. Gradient descent is performed on the effective cost function or Lagrangian, i.e., accounting for the entropy or randomness constraint: $\Delta w_{i,j}^{s} \propto -\frac{\partial\mathcal{L}(\mathbf{x})}{\partial w_{i,j}^{s}}$, and by applying the chain rule:

$$-\frac{\partial\mathcal{L}(\mathbf{x})}{\partial w_{i,j}^{s}} = -\frac{\partial\mathcal{L}(\mathbf{x})}{\partial q_{\mathbf{x},j}^{s+1}}\frac{\partial q_{\mathbf{x},j}^{s+1}}{\partial w_{i,j}^{s}} = -\frac{\partial\mathcal{L}(\mathbf{x})}{\partial q_{\mathbf{x},j}^{s+1}} o_{\mathbf{x},i}^{s}, \tag{5.9}$$

i.e., it can be rewritten as,

$$-\frac{\partial\mathcal{L}(\mathbf{x})}{\partial w_{i,j}^{s}} = \delta_{\mathbf{x},j}^{s+1} o_{\mathbf{x},i}^{s}, \text{ where, } \delta_{\mathbf{x},j}^{s+1} = -\frac{\partial\mathcal{L}(\mathbf{x})}{\partial q_{\mathbf{x},j}^{s+1}}, \tag{5.10}$$

$$\text{and, } \delta_{\mathbf{x},j}^{s+1} = -\frac{\partial\mathcal{L}(\mathbf{x})}{\partial o_{\mathbf{x},j}^{s+1}}\frac{\partial o_{\mathbf{x},j}^{s+1}}{\partial q_{\mathbf{x},j}^{s+1}} = -\frac{\partial\mathcal{L}(\mathbf{x})}{\partial o_{\mathbf{x},j}^{s+1}} f'\left(q_{\mathbf{x},j}^{s+1}\right) \tag{5.11}$$

Next, it is straightforward to show that at the output layer,

$$\frac{\partial \mathcal{L}(\mathbf{x})}{\partial o^S_{\mathbf{x},j}} = \sum_{\mathbf{z}\in\mathcal{B}^{N_S}} \frac{\partial P_{\mathbf{z}|\mathbf{x}}(\mathbf{z})}{\partial o^S_{\mathbf{x},j}} \Bigg(\Bigg(\frac{1}{L} \sum_{\ell=1}^{L} \sum_{y\in\mathcal{Y}} Q^{\ell}_{\mathbf{z},n,L}(y) \rho\left(y^*_\ell(k), y\right) \Bigg) + T\left(1 + \log(P_{\mathbf{x}}(\mathbf{x}) P_{\mathbf{z}|\mathbf{x}}(\mathbf{z}))\right) \Bigg), \tag{5.12}$$

$$\frac{\partial P_{\mathbf{z}|\mathbf{x}}(\mathbf{z})}{\partial o^S_{\mathbf{x},j}} = (-1)^{1-z_j} \prod_{i\neq j}^{N_S} p^{z_i}_{\mathbf{x},i} (1 - p_{\mathbf{x},i})^{(1-z_i)}. \tag{5.13}$$

Finally, $\delta^s_{\mathbf{x},j}$ at any non-output layer can be computed recursively, or back-propagated as,

$$\delta^s_{\mathbf{x},j} = f'(q^s_{\mathbf{x},j}) \sum_{i=1}^{N_{s+1}} \delta^{s+1}_{\mathbf{x},i} w^s_{j,i}. \tag{5.14}$$

Conditioned on a given set of codebook reproduction distributions $\{Q_{\mathbf{z},n,L}\}$, the unrestricted conditional distribution $P_{\mathbf{z}|\mathbf{x}}(\mathbf{z})$ that minimizes the Lagrangian function $\mathcal{L}$ is the Gibbs Distribution [6], i.e.,

$$P^*_{\mathbf{z}|\mathbf{x}}(\mathbf{z}) = \frac{1}{A_{\mathbf{x}}} \exp\left[-\frac{1}{T} \left(\frac{1}{L} \sum_{\ell=1}^{L} \sum_{y\in\mathcal{Y}} Q^{\ell}_{\mathbf{z},n,L}(y) \rho\left(y^*_\ell(k), y\right) \right) \right], \tag{5.15}$$

where $A_{\mathbf{x}}$ is the normalization constant. However, the conditional distribution $P_{\mathbf{z}|\mathbf{x}}(\mathbf{z})$ that can actually be obtained by the DNN is constrained by system topology and parameters, such as weights, number of nodes per layer, activation function, etc. We note in passing that an optimal system should further adapt these parameters based on the current temperature, an extension left for future work. At very high temperature, the optimal conditional distribution is the uniform distribution over all codebook locations. However, at zero temperature, the optimal conditional distribution collapses on the codebook location with the smallest distortion [6].

5.5 Toy Example

As preliminary validation of our proposed learning system model, we consider a digit classification toy example using the MNIST handwritten digit data-set [75]. In this toy example, we assume that each input vector consists of either one, two, or three blocks of 28×28 gray-scale pixel images containing handwritten digits, i.e., $L \in \{1, 2, 3\}$, and $M = 784$. The distortion function considered is the average Hamming distortion measure, i.e., if the digit is classified correctly, the distortion vanishes, otherwise the distortion per incorrect digit classification is equal to one. The distortion threshold for d-match events of codebook training is set to zero. We compare two learning systems:

$S1$) A simple fully-connected DNN that contains one hidden layers, with 100 nodes. The output of this DNN is 4 bits, estimating the binary representation of the input the digit(s) for $L = 1$. As L increases, each input sub-vector is fed into an independent sub-DNN with the same complexity, e.g. for $L = 3$, three independent sub-DNNs are used for every input sub-vector, with one hidden layer, 100 nodes and 4 output nodes, as shown in Fig. 5.1. This system contains no codebook, and the network weights are trained via the traditional back-propagation algorithm with gradient descent, as introduced in [70]. This system is considered the baseline.

$S2$) The second learning system contains the same DNN as $S1$, for fair comparison. This system additionally includes a codebook of outputs which is trained within the NTS framework proposed in Section 5.3. Similar to $S1$, the DNN weights are trained via the traditional back-propagation algorithm with gradient descent. Thus, this system immediately reflects the benefit of introducing a codebook in the learning process.

$S3$) Finally, the third system contains the same DNN and codebook as $S2$. However, the DNN in this system is trained within the DA framework as proposed in Section 5.4.

Table 5.1: Average training (left) and test (right) digit classification accuracy results for the 3 competing systems

	$S1$: Baseline		$S2$: NTS		$S3$: NTS+DA	
$L{=}1$	97.9%	94.1%	97.9%	94.3%	98.6%	94.4%
$L{=}2$	51.0%	48.0%	83.3%	80.9%	98.9%	94.6%
$L{=}3$	36.5%	35.2%	79.7%	76.4%	99.0%	94.65%

Thus, $S3$ immediately shows the benefit of the DA training framework in comparison to $S2$ that employs the standard back-propagation technique.

It is worth noting that the set of output nodes of the DNN, per input digit, is *compressed* to cardinality of only $\lceil \log_2(|\mathcal{Y}|) \rceil$. This is unlike the conventional “winner-take-all” DNNs in [76, 77, 78], where the set of output nodes has cardinality that equals the size of the output alphabet $|\mathcal{Y}|$. While this might result in some performance degradation, it is the only viable solution in practical examples in which the cardinality of the output alphabet or the number of possible classes is very large. It is also worth noting that as L increases, the cardinality of the set of all possible label combinations (i.e., the set of all output nodes in the winner-takes-all scenario) increases exponentially as, $|\mathcal{Y}^L|$, which provides a further compelling reason to compress the output of the DNN as proposed in systems $S1$, $S2$ and $S3$.

Table 5.1 shows the training and test accuracy achieved by the three competing systems. It can be shown that the systems with trained codebooks, i.e., $S2$ and $S3$, outperforms the baseline system $S1$ by up to 61.4% and 58.9% in the training and test set accuracy, respectively. It is worth noting that $S2$ and $S3$ have the same DNN as $S1$. Furthermore, it can be seen that at higher input dimension, when the cost function is riddled with poor local minima, the proposed non-convex optimization DA framework for DNN training significantly outperforms the standard back-propagation descent algorithm in the training and test set accuracy, respectively, i.e. the performance gains are pronounced as the input length L increases. Note that the discrepancy between training and test sets accuracy results exists for higher input dimension, due to the limited availability of input images in the training set.

Chapter 6

Conclusion

The core of this work presents methods and techniques to optimally design codebooks. More specifically, a tractable and asymptotically optimal stochastic codebook generating and adaptation algorithm is devised which is applicable to the vast majority of sources. We propose a complete overhaul to the original iterative NTS algorithm in [10] in order to address its fundamental flaws. First, an ML-framework is designed and leveraged at each NTS iteration n, to estimate the most likely codebook reproduction distribution that would have generated a sequence of K d-matching codewords to a respective sequence of independently generated source words. In Theorems 1-2, and Corollaries 1-2, it is proven that for sources, potentially with memory, over discrete alphabet spaces, given a fixed source word length L, and memory-depth M, the reproduction distribution of the generalized recursive NTS algorithm converges, in probability, asymptotically, as $K \to \infty$, and $n \to \infty$, to the optimal "achievable" codebook reproduction distribution among a set of distributions constrained by the string length L. It was further shown that, if L is sent to infinity, the proposed NTS algorithm finds the optimal codebook reproduction distribution $Q^*_{P_M,d}$ that achieves the M-th order rate-distortion bound $R(P_M, d)$. Afterwords, we developed a variant of the NTS algorithm for sources with finite memory depth, i.e., source with finite order Markov property over discrete alphabet spaces. In Theorems 3-4, we further show that convergence to the optimal constrained codebook reproduction distribution is achieved, without the recourse of sending the memory depth M to infinity. Next, we expand the NTS algorithm to more general sources over abstract or continuous alphabet spaces. It is important to emphasize that the standard concept of types, which is at the heart of the NTS work on discrete alphabet sources, and was specifically instrumental in showing asymptotic convergence to the reconstruction distribution that achieves the rate-distortion bound, does not apply to continuous alphabet sources. Hence, the generalization of the NTS algorithm to continuous alphabet sources is, in fact, fundamentally more challenging. For this type of source, we start working with probability measures over abstract alphabet spaces rather than the method of types. In Theorems 5-7, we extend the NTS convergence to the optimal results to sources over abstract alphabet spaces, by showing that the reproduction distribution, obtained by the generalized NTS algorithm, converges in the weak convergence sense almost surely, to the optimal distribution which achieves the M-th order rate-distortion bound asymptotically in K, n, and L. Furthermore, for a fixed and finite string length L, the codebook reproduction distribution on $\mathcal{Y}^{ML}$ converges to the optimal distribution $Q^*_{P^L_M,\gamma}$, in the weak convergence sense, that achieves the rate distortion function $R(P^L_M,\gamma)$ albeit for an auxiliary distortion measure $\rho^{(d)}$, and the extreme distortion constraint $\gamma = 0$. Afterwards, to provide a further compelling evidence on the practicality of the proposed NTS algorithm, the rate of convergence of the NTS algorithm is studied with respect to i the number of NTS iterations n, ii) the statistical depth K and finally iii) the source word length L. Moreover, toy examples, which consider binary asymmetric sources, are shown to provide further evidence of the fact that the proposed NTS algorithm in addition to being asymptotically optimal is also tractable.

Next, in order to further assess the effectiveness of the proposed codebook design and adaptation techniques, methodologies for optimal codebook generation and adaptation are developed and employed in two promising example applications, that can greatly benefit of such algorithms, in the areas of *i*) wireless communications and *ii*) machine learning. In particular, for millimeter wave wireless systems, we investigate the problem of finding the optimal beam steering codebook to match user statistics. Ultimately, a powerful non-convex optimization technique is derived within the framework of deterministic annealing, to avoid poor local minima on the cost surface (that might result from the state-of-the-art k-means beam steering codebook design approach). The proposed DA-based beam steering algorithm outperforms the baseline uniform steering approaches by up to 6 dB and 12.5 dB in the average and the 10th percentile of power array factor, respectively. Additionally, in a full-fledged system simulation for an outdoor cellular 5G setting, the DA-based algorithms yields SNR gains of up to 6.5 dB. It is noted that the gains in power array factor or in SNR can be traded for significantly reduced codebook size. This would, in turn, reduce the beam management complexity, and hence enhance robustness to user dynamics.

Finally, for machine learning applications, we reformulate the classical supervised learning problem within a rate-distortion framework, by dividing this problem into two parts (Fig. 5.1). The first part of the proposed framework extracts, i.e., learns, the minimal necessary number of information bits from the source examples by a simplified learning system (e.g., deep network). The learned bits are used to generate an index in order to retrieve the desired outputs from a trained codebook, which satisfies a distortion or accuracy requirement. Note that the fewer bits of information we require the system to learn from the source, the easier the learning task in terms of system complexity, generalization and training data requirements. The system optimizes its components through an iterative setup alternating between two main steps: *i*) regenerating the codebook within the asymptotically optimal NTS framework for a fixed DNN, and *ii*) optimizing the DNN parameters within the powerful non-convex DA optimization framework for a fixed codebook. A toy example shows compelling evidence of the superior performance of the proposed system model, thus providing concrete confirmation of its effectiveness.

Appendix A

Asymptotic Performance Evaluation of Natural Type Selection Algorithm

A.1 Lemma 1: Maximum Likelihood Estimation of Codebook Reproduction Distribution

We prove this lemma for memory depth $M = 1$, while noting that specialization to $M \geq 1$ is straightforward. Let $\mathcal{C}_L(d) = \{\mathbf{y}(j_1), \mathbf{y}(j_2), \ldots, \mathbf{y}(j_K)\}$ be the set of L-length d-matching i.i.d. codewords to the P_1-distributed input source words over $\mathcal{X}$: $\{\mathbf{x}(i_1), \mathbf{x}(i_2), \ldots, \mathbf{x}(i_K)\}$ (with n being the NTS iteration index), i.e.,

$$\rho\left(\mathbf{x}(i_k), \mathbf{y}(j_k)\right) \leq d, \quad \mathbf{x}(i_k) \in \mathcal{X}^L, \quad \mathbf{y}_{j_k} \in \mathcal{Y}^M, \quad \forall k \in \{1, 2, \ldots, K\}. \tag{A.1}$$

The i.i.d. property of codewords includes two aspects, independence and identical distribution, so it would be great, beyond independence, to compute the ML generating distribution. Next, we assume that the input alphabet $\mathcal{X}$ and reproduction alphabet $\mathcal{Y}$ are discrete alphabet spaces. The ML estimator of the codebook generating distribution, in the next NTS iteration, finds the most likely distribution that generates the set of codewords in $\mathcal{C}_L(d)$. Consequently, the ML estimator would maximize the *joint* probability of generating the codewords in $\mathcal{C}_L(d)$, and hence can be mathematically written as,

$$Q_{n+1,1,L,K} = \arg\max_{\hat{Q}\in\mathcal{Q}} \mathbb{P}(\mathbf{y}(j_1), \mathbf{y}(j_2), \ldots, \mathbf{y}(j_K)|\hat{Q}), \tag{A.2}$$

where the subscript "1" stands for $M = 1$, $\mathbb{P}(\mathbf{y}(j_1), \mathbf{y}(j_2), \ldots, \mathbf{y}(j_K)|\hat{Q})$ is the joint probability of generating the codewords $\{\mathbf{y}(j_k)\}$ conditioned on the generating distribution $\hat{Q}$, and $\mathcal{Q}$ is the set of valid distributions that produce i.i.d. symbols over $\mathcal{Y}$, i.e.,

$$\mathcal{Q} = \left\{ Q \in \mathbb{R}^{|\mathcal{Y}|} : \sum_{y\in\mathcal{Y}} Q\,(y) = 1, Q(y) \geq 0 \ \ \forall y \in \mathcal{Y} \right\}. \tag{A.3}$$

It is worth noting that we are restricting the set distributions to distributions that only generate i.i.d. code letters (or super-symbols when generalized to $M > 1$). The likelihood function shown in (A.2) depends on the codewords $\mathbf{y}(j_k)$, $1 \leq k \leq K$ only through the codewords' types [45]. Let $Q_{n,1,L}(\mathbf{y}(j_k))$ be the type of the d-matching codeword $\mathbf{y}(j_k)$ at iteration n, i.e., the frequency of occurrence of every code letter $y \in \mathcal{Y}$ as seen in the codeword $\mathbf{y}(j_k)$. Then, the ML formulation in (A.2) can be written as,

$$Q_{n+1,1,L,K} = \arg\max_{\hat{Q}\in\mathcal{Q}} \mathbb{P}(Q_{n,1,L}(\mathbf{y}(j_1)), \ldots Q_{n,1,L}(\mathbf{y}(j_K))|\hat{Q}). \tag{A.4}$$

where $\mathbb{P}(Q_{n,1,L}(\mathbf{y}(j_1)), \ldots Q_{n,1,L}(\mathbf{y}(j_K))|\hat{Q})$ is the joint probability of generating the types of the codewords $\{\mathbf{y}(j_k)\}$ conditioned on the generating distribution $\hat{Q}$. For the ease of notation, denote $Q_{n,1,L}(\mathbf{y}(j_k))$ as Q_k. Taking the property of statistical independence between codewords into consideration, we get,

$$Q_{n+1,1,L,K} = \arg\max_{\hat{Q}\in\mathcal{Q}} \prod_{k=1}^{K} \mathbb{P}(Q_k|\hat{Q}). \tag{A.5}$$

The probability of generating a codeword of type Q given that the generating distribution is $\hat{Q}$ is written as, [45],

$$\mathbb{P}(Q|\hat{Q}) = \exp\left\{-L\left(H(Q) + \mathcal{D}\left(Q||\hat{Q}\right)\right)\right\}, \tag{A.6}$$

where $H(Q)$ denotes the entropy function calculated over the type Q, and $\mathcal{D}(\cdot||\cdot)$ is the KL divergence function. Combining (A.5) and (A.6) yields,

$$Q_{n+1,1,L,K} = \arg\max_{\hat{Q}\in\mathcal{Q}} \prod_{k=1}^{K} \exp\left\{-L\left(H\left(Q_k\right) + \mathcal{D}\left(Q_k||\hat{Q}\right)\right)\right\}, \tag{A.7}$$

$$Q_{n+1,1,L,K} = \arg\max_{\hat{Q}\in\mathcal{Q}} \exp\left\{-L\sum_{k=1}^{K}\left(H\left(Q_k\right)+\mathcal{D}\left(Q_k||\hat{Q}\right)\right)\right\}, \tag{A.8}$$

The $\log_e(\cdot)$ function is monotonically increasing, and the entropy term $H(Q_k)$ doesn't depend on $\hat{Q}$, hence the expression in (A.8) simplifies to,

$$Q_{n+1,1,L,K} = \arg\min_{\hat{Q}\in\mathcal{Q}} \sum_{k=1}^{K}\left(\mathcal{D}\left(Q_k||\hat{Q}\right)\right) \tag{A.9}$$

In summary, the ML estimate of the codebook reproduction distribution is the one that minimizes the sum of KL divergences towards the types of the d-matching codewords subject to the constraints in (A.3). The Lagrangian function to be minimized, that takes into account the constraint $\sum_{y} \hat{Q}(y) = 1$, can thus be written as,

$$\mathcal{L} = \sum_{k=1}^{K}\left(\mathcal{D}\left(Q_k||\hat{Q}\right)\right) + \beta\left(\sum_{y\in\mathcal{Y}} \hat{Q}\left(y\right) - 1\right), \tag{A.10}$$

where β is the Lagrange multiplier. Hence,

$$\mathcal{L} = \sum_{k=1}^{K} \left(\sum_{y \in \mathcal{Y}} Q_k(y) \log \left(\frac{Q_k(y)}{\hat{Q}(y)} \right) \right) + \beta \left(\sum_{y \in \mathcal{Y}} \hat{Q}\,(y) - 1 \right). \tag{A.11}$$

Next, taking partial derivative with respect to $\hat{Q}(y)$, for any $y \in \mathcal{Y}$, yields,

$$\frac{\partial \mathcal{L}}{\partial \hat{Q}(y)} = -\sum_{k=1}^{K} \frac{Q_k(y)}{\hat{Q}_{\mathrm{ML}}(y)} + \beta = 0. \tag{A.12}$$

$$\hat{Q}_{\mathrm{ML}}(y) = \frac{1}{\beta} \sum_{k=1}^{K} Q_k(y). \tag{A.13}$$

Finally, setting the constraint $\sum\limits_{y \in \mathcal{Y}} \hat{Q}_{\mathrm{ML}}(y) = 1$ results in,

$$Q_{n+1,1,L,K} = \hat{Q}_{\mathrm{ML}} = \frac{1}{K} \sum_{k=1}^{K} Q_{n,1,L}(\mathbf{y}(j_k)). \tag{A.14}$$

■

A.2 Lemma 2: Variance of the Unbiased ML Estimate of Codebook Reproduction Distribution with Finite Statistical Depth

As shown in Lemma 1, and upon generalization for $M \geq 1$, given a discrete reproduction alphabet space $\mathcal{Y}$, the ML estimate of the codebook reproduction distribution in every NTS iteration n simplifies to,

$$\hat{Q}_{\mathrm{ML}}(\mathbf{y}) = \frac{1}{K} \sum_{k=1}^{K} Q_k(\mathbf{y}), \quad \forall \mathbf{y} \in \mathcal{Y}^M, \tag{A.15}$$

where Q_k is the M-th order type of the k-th d-matching codeword, and K is the number of d-match operations the algorithm observes before updating the codebook reproduction distribution, i.e., the statistical depth. For statistically independent d-matching codewords, we have,

$$\mathbb{E}[\hat{Q}_{\mathrm{ML}}(\mathbf{y})] = \mathbb{E}[Q_k(\mathbf{y})], \quad \forall \mathbf{y} \in \mathcal{Y}^M, \tag{A.16}$$

where $\mathbb{E}[\cdot]$ denotes the expectation of the argument. The expectation in (A.16) is taken over the distribution of the d-matching codewords' types. Thus, this establishes that the estimator is unbiased. Finally, the variance of the maximum likelihood estimate decays proportional to $1/K$ as follows,

$$\mathbb{VAR}[\hat{Q}_{\mathrm{ML}}(\mathbf{y})] = \frac{1}{K}\mathbb{VAR}[Q_k(\mathbf{y})], \quad \forall \mathbf{y} \in \mathcal{Y}^M. \tag{A.17}$$

■

A.3 Lemma 3: ML Estimate of Codebook Reproduction Distribution with Markov Property

Given a set of d-matching L-length codewords $\mathbf{y}(j_k)$, $k = 1, \ldots, K$, where $\mathbf{y}(j_k) = (y_1(j_k), y_2(j_k), \ldots, y_L(j_k))$ and $y_\ell(j_k) \in \mathcal{Y}$, the maximum likelihood estimator for the codebook distribution transition matrix, having an M-th order Markov property is formulated as,

$$\hat{\mathbf{Q}}^{\mathrm{ML}} = \arg\max_{\hat{\mathbf{Q}} \in \mathcal{Q}} \mathbb{P}(\mathbf{y}(j_1), \ldots, \mathbf{y}(j_k) | \hat{\mathbf{Q}}), \tag{A.18}$$

where $\mathcal{Q}$ is the set of all valid transition distribution matrices that satisfy the stationary assumption, and $\mathbb{P}(\mathbf{y}(j_1), \ldots, \mathbf{y}(j_k) | \hat{\mathbf{Q}})$ is the joint probability of generating the types of the codewords $\{\mathbf{y}(j_k)\}$ conditioned on the generating distribution transition matrix $\hat{\mathbf{Q}}$. Next, by the independence of the d-match events, the maximum likelihood formulation is written as,

$$\hat{\mathbf{Q}}^{\mathrm{ML}} = \arg\max_{\hat{\mathbf{Q}}\in\mathcal{Q}} \prod_{k=1}^{K} \mathbb{P}(\mathbf{y}(j_k)|\hat{\mathbf{Q}}). \tag{A.19}$$

Next, we have,

$$\mathbb{P}(\mathbf{y}|\hat{\mathbf{Q}}) = \mathbb{P}(Y_1 = y_1) \prod_{\ell=2}^{L} Q_{y_{\ell-1},y_\ell}, \tag{A.20}$$

where y_ℓ is the ℓ-th element in the codeword $\mathbf{y}$. Let the number of transition from state i to state j in the codeword $\mathbf{y}$ be denoted as $N(i \to j|\mathbf{y})$. Then the probability of generating a codeword $\mathbf{y}$ conditioned on the state transition probability matrix $\hat{\mathbf{Q}}$ is simplified to,

$$\mathbb{P}(\mathbf{y}|\hat{\mathbf{Q}}) = \prod_{m=1}^{M} \mathbb{P}(Y_m = y_m \,|\, Y_{m-1} = y_{m-1}, \ldots, Y_1 = y_1) \prod_{(i,j)\in\mathcal{S}^2} \hat{Q}_{j|i}^{N(i \to j|\mathbf{y})}. \tag{A.21}$$

Taking the log of $P(\mathbf{y}|\hat{\mathbf{Q}})$ results in,

$$\begin{aligned}\log(\mathbb{P}(\mathbf{y}|\hat{\mathbf{Q}})) = \sum_{m=1}^{m} \log(\mathbb{P}(Y_m = y_m \,|\, Y_{m-1} = y_{m-1}, \ldots, Y_1 = y_1)) \, + \\ \sum_{(i,j)\in\mathcal{S}^2} N(i \to j|\mathbf{y}) \log(\hat{Q}_{j|i}).\end{aligned} \tag{A.22}$$

Hence,

$$\begin{aligned}\hat{\mathbf{Q}}^{\mathrm{ML}} = \arg\max_{\hat{\mathbf{Q}}\in\mathcal{Q}} \Bigg\{ \sum_{k=1}^{K} \log(\mathbb{P}(Y_m = y_m(j_k) \,|\, Y_{m-1} = y_{m-1}(j_k), \ldots, Y_1 = y_1(j_k))) \, + \\ \sum_{k=1}^{K} \sum_{(i,j)\in\mathcal{S}^2} N(i \to j|\mathbf{y}(j_k)) \log(\hat{Q}_{j|i}) \Bigg\}.\end{aligned} \tag{A.23}$$

For sufficiently large codeword lengths L, one can ignore the first sum term in (A.23), hence,

$$\hat{\mathbf{Q}}^{\mathrm{ML}} \approx \arg\max_{\hat{\mathbf{Q}}\in\mathcal{Q}} \left\{ \sum_{k=1}^{K} \sum_{(i,j)\in\mathcal{S}^2} N(i \to j|\mathbf{y}(j_k)) \log(\hat{Q}_{j|i}) \right\}. \tag{A.24}$$

The Lagrangian function that enforces the set of constraints, $\sum\limits_{j\in\mathcal{S}} \hat{Q}_{j|i} = 1$, can be written as,

$$\mathcal{L} = \sum_{k=1}^{K} \sum_{(i,j)\in\mathcal{S}^2} N(i{\to}j|\mathbf{y}(j_k)) \log(\hat{Q}_{j|i}) - \sum_{i\in\mathcal{S}} \lambda_i \left(\sum_{j\in\mathcal{S}} \hat{Q}_{j|i} - 1 \right). \tag{A.25}$$

Differentiating with respect to $\hat{Q}_{j|i}$ results in,

$$\frac{\partial \mathcal{L}}{\partial \hat{Q}_{j|i}} = \sum_{k=1}^{K} \frac{N(i{\to}j|\mathbf{y}(j_k))}{\hat{Q}^{\mathrm{ML}}_{j|i}} - \lambda_i = 0 \tag{A.26}$$

$$\hat{Q}^{\mathrm{ML}}_{j|i} = \frac{\sum\limits_{k=1}^{K} N(i{\to}j|\mathbf{y}(j_k))}{\sum\limits_{k=1}^{K} \sum\limits_{j'\in\mathcal{S}} N(i{\to}j'|\mathbf{y}(j_k))}. \tag{A.27}$$

A.4 Theorem 1: Conditional Limit Theorem for Tractable NTS Algorithm over Discrete Alphabets

Let $\mathbf{x}(i_k)$ and $\mathbf{y}(j_k)$, $k = 1, 2, \ldots, K$, be a sequence of d-matching L-length source words and codewords, where source words are generated with letter distributions P over discrete alphabets $\mathcal{X}$, and the code letters are distributed (prior to d-match events) according to Q over the discrete alphabet $\mathcal{Y}$. The distribution of code letter might obviously be altered post d-match events. Each codeword $\mathbf{y}(j_k)$ d-matches source word $\mathbf{x}(i_k)$ independent of other codewords, i.e., the d-matching events are obviously independent. Furthermore, the empirical distributions of the d-matching codewords' letters on $\mathcal{Y}$, depend on the source distribution P, the codebook reproduction distribution Q, the string length L, and the set of d-matching balls around source realizations $\mathbf{x} \in \mathcal{X}^L$, defined as,

$$\mathcal{B}_L(\mathbf{x}, d) = \left\{ \mathbf{y} : \mathbf{y} \in \mathcal{Y}^L, \rho(\mathbf{x}, \mathbf{y}) \le d \right\}, \quad \forall \mathbf{x} \in \mathcal{X}^L. \tag{A.28}$$

Hence, this concludes that the empirical distributions of d-matching codewords are i.i.d. because P, Q, L and the set of d-matching balls $\{\mathcal{B}_L(\mathbf{x}, d),\ \forall \mathbf{x} \in \mathcal{X}^L\}$ are unchanged across the d-matching events. This together with weak law of large numbers implies the first part of Theorem 1. In order to show the second part of Theorem 1, define $\mathcal{U}_L(d)$ as the set of all possible pairs of L-length source words and codewords that can d-match, i.e.,

$$\mathcal{U}_L(d) \triangleq \left\{(\mathbf{x}, \mathbf{y}) : \mathbf{x} \in \mathcal{X}^L, \mathbf{y} \in \mathcal{Y}^L, \rho(\mathbf{x}, \mathbf{y}) \leq d\right\}. \tag{A.29}$$

Without loss of generality, we assume that the source distribution over the discrete $\mathcal{X}^L$ is strictly positive, i.e., $P^L(\mathbf{x}) > 0, \forall \mathbf{x} \in \mathcal{X}^L$. Note that, by Lemma 1, the ML codebook reproduction distribution is the y-marginal of a K-average input-output joint distribution over d-matching source words and codewords. Hence, the ML distribution is the y-marginal of a joint distribution over the set $\mathcal{U}_L(d)$. Next, for every source realization $\mathbf{x} \in \mathcal{X}^L$, define $\mathcal{W}_L(\mathbf{x}, d)$ as the set of all conditional distributions that would generate a joint distribution $\left(P^L(\mathbf{x})W(\mathbf{y}|\mathbf{x})\right)$ which guarantee d-matching codewords with probability one, i.e.,

$$\mathcal{W}_L(\mathbf{x}, d) = \left\{W(\mathbf{y}|\mathbf{x}) : W(\mathbf{y}|\mathbf{x}) \geq 0, W(\mathbf{y}|\mathbf{x}) = 0, \forall \mathbf{y} \notin \mathcal{B}_L(\mathbf{x}, d), W(\mathcal{Y}^L|\mathbf{x}) = 1\right\}. \tag{A.30}$$

Hence, the set of all possible K-average joint distributions that is obtained after observing a set of K independent d-match event, for sufficiently large K, is calculated as,

$$\begin{aligned} E_{L,K}(P, d) = \Big\{V : V\left(\mathcal{X}^L, \mathcal{Y}^L\right) = 1,\ V(\mathbf{x}, \mathbf{y}) = \frac{n(\mathbf{x}, \mathbf{y})P^L(\mathbf{x})W(\mathbf{y}|\mathbf{x})}{K}, \\ W(\mathbf{y}|\mathbf{x}) \in \mathcal{W}_L(\mathbf{x}, d), n(\mathbf{x}, \mathbf{y}) \in \{0, \ldots, K\}\Big\} \end{aligned} \tag{A.31}$$

This implies that the set $E_{L,K}(P, d)$ contains all possible L-th order types with denominator K that generates d-matching source and code pairs with probability one. It is important to reiterate that by Lemma 1, the ML estimate of the next iteration codebook reproduction distribution $\hat{Q}^{\mathrm{ML}}$, that most likely generates the set of d-matching codewords, is the K-average of the d-matching codewords' types, and hence the joint input-output L-th order type, with y-marginal equals to $\hat{Q}^{\mathrm{ML}}$, must belong to $E_{L,K}(P,d)$. Next, we turn our attention to the concatenated source and code blocks. Let $\bar{\mathbf{x}}$ and $\bar{\mathbf{y}}$ be the KL-length blocks constructed by concatenating $(\mathbf{x}(i_1),\ldots,\mathbf{x}(i_K))$ source words, and $(\mathbf{y}(j_1),\ldots,\mathbf{y}(j_K))$ codewords, respectively. We will show that for any $\delta>0$, and sufficiently large K,

$$\mathbb{P}\left(\mathcal{D}(Q_{\bar{\mathbf{y}}}||Q_L^*(P,Q,d)) > 3\delta | V_{\bar{\mathbf{x}},\bar{\mathbf{y}}} \in E_{L,K}(P,d)\right) \leq (K+1)^{2|\mathcal{X}^L||\mathcal{Y}^L|}e^{-K\delta}. \tag{A.32}$$

In other words, if we condition on the event that the joint distribution of $(\bar{\mathbf{x}},\bar{\mathbf{y}})$, denoted as $V_{\bar{\mathbf{x}},\bar{\mathbf{y}}}$ over $\mathcal{X}^L\times\mathcal{Y}^L$, belongs to $E_{L,K}(P,d)$, the distribution of $\bar{\mathbf{y}}$, denoted as $Q_{\bar{\mathbf{y}}}$, is with high probability close in the divergence-sense to $Q_L^*(P,Q,d)$, defined in (3.10). Since closeness in divergence also implies closeness in the $\mathcal{L}_1$ sense [45, Lemma 11.6.1], and by part $i)$ of Theorem 1, i.e., the average type of the d-matching codeword converges to the expected type $\mathbb{E}[Q_k]$, this establishes part $ii)$ of Theorem 1. We start by verifying that, as $K\to\infty$, $E_{L,K}(P,d)$ approaches $E_L(P,d)$ by (3.13) and (A.31), hence define,

$$D^* = \min_{V\in E_L(P,d)} \mathcal{D}\left(V\middle|\middle|P^L\times Q^L\right), \tag{A.33}$$

where P^L and Q^L denotes the L-dimensional product distributions over $\mathcal{X}^L$ and $\mathcal{Y}^L$, respectively. Then following [51, 45, Th. 11.6.2], we obtain,

$$\mathbb{P}\left(\mathcal{D}\left(V_{\bar{\mathbf{x}},\bar{\mathbf{y}}}||P^L\times Q^L\right) > D^* + 3\delta, V_{\bar{\mathbf{x}},\bar{\mathbf{y}}}\in E_{L,K}(P,d)\right) = \sum_{\substack{V'\in E_{L,K}(P,d)\cap\mathcal{P}_{L,K}\times\mathcal{Q}_{L,K}:\\ \mathcal{D}(V'||P^L\times Q^L)>D^*+3\delta}} \mathbb{P}(T_{\mathrm{y}}(V')), \tag{A.34}$$

where the probability of type class of V' is denoted by $\mathbb{P}(T_{\mathrm{y}}(V'))$, $\mathcal{P}_{L,K}$, and $\mathcal{Q}_{L,K}$ are the sets of all possible L-th order input and output types with denominator K. Then, by [45, Th. 11.1.4] which bounds the probability of type classes,

$$\begin{aligned}\mathbb{P}\Bigg(\mathcal{D}\left(V_{\bar{\mathbf{x}},\bar{\mathbf{y}}}||P^L\times Q^L\right)>D^{*}+3\delta, V_{\bar{\mathbf{x}},\bar{\mathbf{y}}}\in E_{L,K}(P,d)\Bigg) \leq \\ \sum_{\substack{V'\in E_{L,K}(P,d)\cap\mathcal{P}_{L,K}\times\mathcal{Q}_{L,K}:\\ \mathcal{D}(V'||P^L\times Q^L)>D^{*}+3\delta}} \exp\left(-K\mathcal{D}\left(V'||P^L\times Q^L\right)\right),\end{aligned} \tag{A.35}$$

$$\begin{aligned}\mathbb{P}\Bigg(\mathcal{D}\left(V_{\bar{\mathbf{x}},\bar{\mathbf{y}}}||P^L\times Q^L\right)>D^{*}+3\delta, V_{\bar{\mathbf{x}},\bar{\mathbf{y}}}\in E_{L,K}(P,d)\Bigg) \leq \\ \sum_{\substack{V'\in E_{L,K}(P,d)\cap\mathcal{P}_{K}\times\mathcal{Q}_{K}:\\ \mathcal{D}(V'||P^L\times Q^L)>D^{*}+3\delta}} \exp\left(-K(D^{*}+3\delta)\right),\end{aligned} \tag{A.36}$$

$$\begin{aligned}\mathbb{P}\Bigg(\mathcal{D}\left(V_{\bar{\mathbf{x}},\bar{\mathbf{y}}}||P^L\times Q^L\right)>D^{*}+3\delta, V_{\bar{\mathbf{x}},\bar{\mathbf{y}}}\in E_{L,K}(P,d)\Bigg) \leq \\ (K+1)^{|\mathcal{X}^L||\mathcal{Y}^L|}\exp\left(-K\left(D^{*}+3\delta\right)\right),\end{aligned} \tag{A.37}$$

since there are only a polynomial number of joint types. Then, again by [45, Th. 11.1.4], we observe that,

$$\mathbb{P}\left(\mathcal{D}\left(V_{\bar{\mathbf{x}},\bar{\mathbf{y}}}||P^L\times Q^L\right)\leq D^{*}+2\delta, V_{\bar{\mathbf{x}},\bar{\mathbf{y}}}\in E_{L,K}(P,d)\right) = \sum_{\substack{V'\in E_{L,K}(P,d)\cap\mathcal{P}_{K}\times\mathcal{Q}_{K}:\\ \mathcal{D}(V'||P^L\times Q^L)\leq D^{*}+2\delta}} \mathbb{P}(T_{\mathrm{y}}(V')), \tag{A.38}$$

$$\begin{aligned}\mathbb{P}\Bigg(\mathcal{D}\left(V_{\bar{\mathbf{x}},\bar{\mathbf{y}}}||P^L\times Q^L\right)\leq D^{*}+2\delta, V_{\bar{\mathbf{x}},\bar{\mathbf{y}}}\in E_{L,K}(P,d)\Bigg) \geq \\ \sum_{\substack{V'\in E_{L,K}(P,d)\cap\mathcal{P}_{K}\times\mathcal{Q}_{K}:\\ \mathcal{D}(V'||P^L\times Q^L)\leq D^{*}+2\delta}} \frac{\exp\left(-K\mathcal{D}\left(V'||P^L\times Q^L\right)\right)}{(K+1)^{|\mathcal{X}^L||\mathcal{Y}^L|}},\end{aligned} \tag{A.39}$$

$$\mathbb{P}\left(\mathcal{D}\left(V_{\bar{\mathbf{x}},\bar{\mathbf{y}}}||P^L\times Q^L\right)\leq D^*+2\delta, V_{\bar{\mathbf{x}},\bar{\mathbf{y}}}\in E_{L,K}(P,d)\right)\geq \sum_{\substack{V'\in E_{L,K}(P,d)\cap\mathcal{P}_K\times\mathcal{Q}_K:\\ \mathcal{D}(V'||P^L\times Q^L)\leq D^*+2\delta}}\frac{\exp\left(-K(D^*+2\delta)\right)}{(K+1)^{|\mathcal{X}^L||\mathcal{Y}^L|}}, \tag{A.40}$$

$$\mathbb{P}\left(\mathcal{D}\left(V_{\bar{\mathbf{x}},\bar{\mathbf{y}}}||P^L\times Q^L\right)\leq D^*+2\delta, V_{\bar{\mathbf{x}},\bar{\mathbf{y}}}\in E_{L,K}(P,d)\right)\geq\frac{\exp\left(-K\left(D^*+2\delta\right)\right)}{(K+1)^{|\mathcal{X}^L||\mathcal{Y}^L|}}, \tag{A.41}$$

since for sufficiently large K, there exists at least one term in the summation, i.e., there exists one joint type V' in $E_{L,K}(P,d)$ such that,

$$\mathcal{D}\left(V'||P^L\times Q^L\right)\leq D^*+2\delta. \tag{A.42}$$

Next, taking into account that the probability of one event is larger than or equal to the probability of the intersection, we have,

$$\mathbb{P}\left(V_{\bar{\mathbf{x}},\bar{\mathbf{y}}}\in E_{L,K}(P,d)\right)\geq\frac{\exp\left(-K\left(D^*+2\delta\right)\right)}{(K+1)^{|\mathcal{X}^L||\mathcal{Y}^L|}}. \tag{A.43}$$

By Bayes' law we get,

$$\mathbb{P}\left(\mathcal{D}\left(V_{\bar{\mathbf{x}},\bar{\mathbf{y}}}||P^L\times Q^L\right)>D^*+3\delta\;\middle|\;V_{\bar{\mathbf{x}},\bar{\mathbf{y}}}\in E_{L,K}(P,d)\right)\leq(K+1)^{2|\mathcal{X}^L||\mathcal{Y}^L|}\exp\left(-K\delta\right). \tag{A.44}$$

By the "Pythagorean" theorem [45, Th. 11.6.1], we have,

$$\mathcal{D}(V_{\bar{\mathbf{x}},\bar{\mathbf{y}}}||V_L^*)+\mathcal{D}\left(V_L^*||P^L\times Q^L\right)\leq\mathcal{D}\left(V_{\bar{\mathbf{x}},\bar{\mathbf{y}}}||P^L\times Q^L\right), \tag{A.45}$$

where, for ease of notation, $V_L^*=V_L^*(P,Q,d)$. Hence, $\mathcal{D}\left(V_{\bar{\mathbf{x}},\bar{\mathbf{y}}}||P^L\times Q^L\right)\leq D^*+3\delta$ implies that,

$$\mathcal{D}(V_{\bar{\mathbf{x}},\bar{\mathbf{y}}}||V_L^*)\leq 3\delta. \tag{A.46}$$

Next, by the data processing inequality, we have,

$$\mathcal{D}(Q_{\bar{\mathbf{y}}}||Q_L^*(P,Q,d)) \leq \mathcal{D}(V_{\bar{\mathbf{x}},\bar{\mathbf{y}}}||V_L^*) \tag{A.47}$$

$$\mathcal{D}\left(Q_{\bar{\mathbf{y}}}^{\text{Marg.}} \;\middle|\middle|\; Q_L^*(P,Q,d)^{\text{Marg.}}\right) \leq \mathcal{D}(Q_{\bar{\mathbf{y}}}||Q_L^*(P,Q,d)) \tag{A.48}$$

since in (A.47), both are the respective L-dimensional y-marginals of the joint distributions on $\mathcal{Y}^L$, and in (A.48), both are the respective marginal distributions on $\mathcal{Y}$. Finally, it is important to note that by Lemma 1, the next iteration ML codebook reproduction distribution is the average type of d-matching codewords, hence,

$$Q_{n,1,L,K} = Q_{\bar{\mathbf{y}}}^{\text{Marg.}} \tag{A.49}$$

Consequently, part *ii*) of Theorem 1 follows from (A.44) and part *i*) of Theorem 1 as follows,

$$Q_{n,1,L,K} \to \mathbb{E}[Q_k] = Q_L^*(P,Q,d)^{\text{Marg.}} \quad \text{as } K \to \infty \quad \text{w.p. 1.} \tag{A.50}$$

■

A.5 Theorem 2: Discrete Alphabet NTS Algorithm, Alternating Minimization over Convex Sets

we can write $R_L(P,d)$ as a double minimization over *convex* sets, i.e.,

$$R_L(P,d) = \min_{Q} \min_{V \in E_L(P,d)} \mathcal{D}\left(V \;||\; P^L \times Q^L\right). \tag{A.51}$$

Next we show that, for a fixed V, the reproduction distribution Q on $\mathcal{Y}$ which minimizes $\mathcal{D}\left(V \;||\; P^L \times Q^L\right)$ is the y-marginal of V on $\mathcal{Y}$. Define the Lagrangian function to be minimized as,

$$\mathcal{L} = \mathcal{D}\left(V||P^L \times Q^L\right) + \beta\left(\sum_{y\in\mathcal{Y}} Q(y) - 1\right), \tag{A.52}$$

$$\mathcal{L} = \sum_{\mathbf{x}\in\mathcal{X}^L}\sum_{\mathbf{y}\in\mathcal{Y}^L} -V(\mathbf{x},\mathbf{y})\log\left(\frac{\prod\limits_{\ell=1}^{L} P(x_\ell)\prod\limits_{\ell=1}^{L} Q(y_\ell)}{V(\mathbf{x},\mathbf{y})}\right) + \beta\left(\sum_{y\in\mathcal{Y}} Q(y) - 1\right), \tag{A.53}$$

where $\mathbf{x} = (x_1, \ldots, x_L)$ is a source word realization, $\mathbf{y} = (y_1, \ldots, y_L)$ is a codeword realization, β is the Lagrangian multiplier, and the second term (A.53) enforces the necessary constraint of a valid distribution, i.e., $\sum\limits_{y\in\mathcal{Y}} Q(y) = 1$. Performing straight forward partial differentiation with respect to $Q(y')$ (with $y' \in \mathcal{Y}$) yields,

$$\frac{\partial\mathcal{L}}{\partial Q(y')} = \sum_{\mathbf{x}\in\mathcal{X}^L}\sum_{\mathbf{y}\in\mathcal{Y}^L} -V(\mathbf{x},\mathbf{y})\frac{L\ N(y'|\mathbf{y})}{Q(y')} + \beta, \tag{A.54}$$

where $N(y'|\mathbf{y})$ is the frequency of occurrence of y' as seen in $\mathbf{y}$. Next, setting (A.54) to 0, results in,

$$\sum_{\mathbf{y}\in\mathcal{Y}^L} \frac{L\ N(y'|\mathbf{y})}{Q(y')} \sum_{\mathbf{x}\in\mathcal{X}^L} V(\mathbf{x},\mathbf{y}) = \beta, \tag{A.55}$$

$$Q(y') = \sum_{\mathbf{y}\in\mathcal{Y}^L} N(y'|\mathbf{y}) \sum_{\mathbf{x}\in\mathcal{X}^L} V(\mathbf{x},\mathbf{y}), \tag{A.56}$$

which is exactly the y-marginal of V on $\mathcal{Y}$. On the other hand, for a fixed Q and distortion constraint d, the joint distribution which minimizes $\mathcal{D}\left(V\ ||P^L \times Q^L\right)$ over $E_L(P,d)$ will induce $Q_L^*(P,Q,d)^{\text{Marg.}}$. By the results of Theorem 1, the recursion in Algorithm 5 performs exactly this minimization over the convex sets, as shown in Fig. A.1, i.e.,

$$\begin{gathered} V_L^*(P, Q_{0,1,L}, d) \to (P^L \times Q_L^*(P, Q_{0,1,L}, d)) \to \\ V_L^*(P, Q_{1,1,L}, d) \to (P^L \times Q_L^*(P, Q_{1,1,L}, d)) \ldots \end{gathered} \tag{A.57}$$

It should be noted that the distance in the alternating minimization is measured by divergence. Hence, by [79, Th. 3], the sequences of divergences and distributions will converge to the minimum divergence, i.e., $R_L(P,d)$, and the corresponding reproduction distribution $Q^*_L(P,d)^{\text{Marg.}}$. Next, to show part *ii)* of Theorem 2 stated in (3.16), first verify that the minimum coding rate with constrained reproduction distribution Q, denoted as $R(P,Q,d)$, is written as [10],

$$R(P,Q,d) = \min_{W:\rho(P,W)\leq d} I(P,W) + \mathcal{D}([P\circ W]_y \,||Q), \tag{A.58}$$

$$R(P,Q,d) = \min_{W:\rho(P,W)\leq d} \mathcal{D}(P\circ W||P\times Q), \tag{A.59}$$

Hence, $R(P,d)$ follows from (A.59) as,

$$R(P,d) = \min_{Q} \min_{W:\rho(P,W)\leq d} \mathcal{D}(P\circ W||P\times Q). \tag{A.60}$$

Now, as $L\to\infty$, and by law of large numbers, it is straight forward to show that,

$$E_L(P,d) \to \big\{V : V = P^L\circ W', \mathbb{E}[\rho(\mathbf{X},\mathbf{Y})] \leq d\big\}, \tag{A.61}$$

where the expectation in $\mathbb{E}[\rho(\mathbf{X},\mathbf{Y})]$ is over the joint distribution $P^L\circ W'$. Consequently, as $L\to\infty$, and from (3.17), (A.51), and (A.60),

$$R_L(P,Q,d) \to \min_{W:\mathbb{E}[\rho(\tilde{\mathbf{X}},\tilde{\mathbf{Y}})]\leq d} \mathcal{D}\left(P^L\circ W||P^L\times Q^L\right), \tag{A.62}$$

$$\frac{1}{L}R_L(P,d) \to R(P,d). \tag{A.63}$$

Thus, by the definition of $Q^*_L(P,d)$ in (3.17), and the definition of the rate-distortion

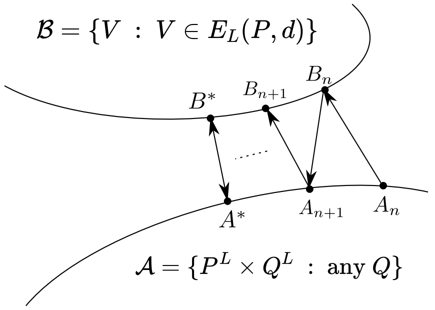

$\mathcal{B} = \{V : V \in E_L(P,d)\}$
B^*
B_{n+1}
B_n
A^*
A_{n+1}
A_n
$\mathcal{A} = \{P^L \times Q^L : \text{any } Q\}$

Figure A.1: Alternating Minimization over Convex Sets Induced by the Natural Type Selection Algorithm.

function achieving distribution $Q^*_{P,d}$, part *ii*) of Theorem 2 follows. ■

A.6 Theorem 3: Conditional Limit Theorem for Tractable NTS Algorithm for Markov Sources

In the subsequent analysis, without loss of generality, we will only consider the case for which $K = 1$. Note that the distribution of the long source and code blocks $\mathbf{s}$ and $\mathbf{c}$ (formed by concatenating K d-matching source and code words, respectively) do not change for $K > 1$, as $L \to \infty$, because it is the K-average of identical converging distributions, as will be shown by this theorem. Let the length $L_{\mathbf{x},\mathbf{y}}$ be the number of letters of the source sub-stream $\mathbf{s_x}$ that is reproduced by the code sub-stream $\mathbf{c_y}$, such that $\sum L_{\mathbf{x},\mathbf{y}} = L$. For asymptotically large $L_{\mathbf{x},\mathbf{y}}$, we define the set $E_{\mathbf{x},\mathbf{y}}(d_{\mathbf{x},\mathbf{y}})$ as in (3.29). Thus $E_{\mathbf{x},\mathbf{y}}(d_{\mathbf{x},\mathbf{y}})$ denotes a set of all joint distributions on $\mathcal{X} \times \mathcal{Y}$ that satisfies the source distribution $P(X|\mathbf{x})$ and distortion level $d_{\mathbf{x},\mathbf{y}}$. Note that, by the strong law of large numbers, the realizations of the instantaneous source types $P_{L_{\mathbf{x},\mathbf{y}}}$ of the source letters in the sub-stream $\mathbf{s_x}$ that were represented by code letters in the sub-stream $\mathbf{c_y}$, converge almost surely to $P(X|\mathbf{x})$. Next, define the minimum divergence $\mathcal{D}^*$ as,

$$\mathcal{D}^* = \min_{\substack{\mathbb{M}(\mathbf{y}|\mathbf{x}) \\ d_{\mathbf{x},\mathbf{y}}, V_{\mathbf{x},\mathbf{y}}}} \sum_{\mathbf{x}\in\mathcal{X}^M} \mathbb{M}(\mathbf{x}) \sum_{\mathbf{y}\in\mathcal{Y}^M} \mathbb{M}(\mathbf{y}|\mathbf{x}) \; \mathcal{D}(V_{\mathbf{x},\mathbf{y}}||P(X|\mathbf{x}) \times Q(Y|\mathbf{y})), \tag{A.64}$$

such that $\mathbb{M}(\mathbf{y}|\mathbf{x})$, $d_{\mathbf{x},\mathbf{y}}$ and $V_{\mathbf{x},\mathbf{y}}$ satisfy,

$$\sum_{\mathbf{x},\mathbf{y}} \mathbb{M}(\mathbf{x})\mathbb{M}(\mathbf{y}|\mathbf{x})\rho(x_1,y_1) = \sum_{\mathbf{x},\mathbf{y}} \mathbb{M}(\mathbf{x})\mathbb{M}(\mathbf{y}|\mathbf{x})d_{\mathbf{x},\mathbf{y}} \leq d,\ V_{\mathbf{x},\mathbf{y}} \in E_{\mathbf{x},\mathbf{y}}(d_{\mathbf{x},\mathbf{y}}), \tag{A.65}$$

with x_1 and y_1 being the left most letters in $\mathbf{x}$ and $\mathbf{y}$, respectively. Let $\mathbb{M}^*(\mathbf{y}|\mathbf{x})$, $d^*_{\mathbf{x},\mathbf{y}}$, and $V^*_{\mathbf{x},\mathbf{y}}$ be the optimization variables that achieve the minimum in (A.64). Furthermore, define $\mathcal{D}^*_{\mathbf{y}} \triangleq \sum_{\mathbf{x}} \mathbb{M}^*(\mathbf{x}|\mathbf{y})\mathcal{D}(V^*_{\mathbf{x},\mathbf{y}}||P(X|\mathbf{x}) \times Q(Y|\mathbf{y})), \forall \mathbf{y} \in \mathcal{Y}^M$, where $\mathbb{M}^*(\mathbf{x}|\mathbf{y})$ can be calculated from $\mathbb{M}(\mathbf{x})$ and $\mathbb{M}^*(\mathbf{y}|\mathbf{x})$ from Bayes' law. Hence, we have, $\mathcal{D}^* = \sum_{\mathbf{y}} \mathbb{M}(\mathbf{y})\mathcal{D}^*_{\mathbf{y}}$, with $\mathbb{M}(\mathbf{y}) = \sum_{\mathbf{x}'} \mathbb{M}(\mathbf{x}')\mathbb{M}^*(\mathbf{y}|\mathbf{x}')$. We will show that for any $\delta > 0$, and sufficiently large $L = \sum_{\mathbf{x},\mathbf{y}} L_{\mathbf{x},\mathbf{y}}$,

$$\begin{aligned} &\mathbb{P}\left(\sum_{\mathbf{x}} \mathbb{M}(\mathbf{x}|\mathbf{y})\mathcal{D}(Q_{\mathbf{C}|\mathbf{x},\mathbf{y}}||Q^*(P(X|\mathbf{x}), Q(Y|\mathbf{y}), d^*_{\mathbf{x},\mathbf{y}})) > 3\delta | V_{\mathbf{S},\mathbf{C}} \in E_L(d)\right) \\ &\qquad\qquad \leq \prod_{\mathbf{x}} (L\mathbb{M}(\mathbf{y})\mathbb{M}^*(\mathbf{x}|\mathbf{y}) + 1)^{2|\mathcal{X}||\mathcal{Y}|} e^{-L\delta}. \end{aligned} \tag{A.66}$$

In other words, if we condition on the event that the joint distribution of the random source and code block pair (with $K = 1$) $\mathbf{S}$, and $\mathbf{C}$, denoted as $V_{\mathbf{S},\mathbf{C}}$ over $\mathcal{X} \times \mathcal{Y}$, belongs to $E_L(d)$, the average conditional distribution of the codewords is with high probability close, in the divergence-sense, to $\sum_{\mathbf{x}} \mathbb{M}(\mathbf{x}|\mathbf{y})Q^*(P(X|\mathbf{x}), Q(Y|\mathbf{y}), d^*_{\mathbf{x},\mathbf{y}})$. Since closeness in divergence also implies closeness in the $\mathcal{L}_1$ sense [45, Lemma 11.6.1], this establishes Theorem 1. Then following [51, 45, Th. 11.6.2], we obtain,

$$\begin{aligned} &\mathbb{P}\left(\sum_{\mathbf{x}} \mathbb{M}(\mathbf{x}|\mathbf{y})\mathcal{D}\left(V_{\mathbf{S},\mathbf{C}|\mathbf{x},\mathbf{y}}||P(X|\mathbf{x}) \times Q(Y|\mathbf{y})\right) > D^* + 3\delta, V_{\mathbf{S},\mathbf{C}} \in E_L(d)\right) = \\ &\qquad \sum_{\substack{V'_{\mathbf{x},\mathbf{y}} \in E_{\mathbf{x},\mathbf{y}}(d_{\mathbf{x},\mathbf{y}}) \cap \mathcal{P}_{L_{\mathbf{x},\mathbf{y}}} \times \mathcal{Q}_{L_{\mathbf{x},\mathbf{y}}}: \\ \sum_{\mathbf{x}} \mathbb{M}(\mathbf{x}|\mathbf{y})\mathcal{D}(V'_{\mathbf{x},\mathbf{y}}||P(X|\mathbf{x}) \times Q(Y|\mathbf{y})) > D^*_{\mathbf{y}} + 3\delta \\ \sum_{\mathbf{x},\mathbf{y}} \mathbb{M}(\mathbf{x},\mathbf{y})\rho(x_1,y_1) = \sum_{\mathbf{x},\mathbf{y}} \mathbb{M}(\mathbf{x},\mathbf{y})d_{\mathbf{x},\mathbf{y}} \leq d}} \prod_{\mathbf{x}} \mathbb{P}(T_{\mathrm{y}}(V'_{\mathbf{x},\mathbf{y}})), \end{aligned} \tag{A.67}$$

where the probability of type class of $V'_{\mathbf{x},\mathbf{y}}$ is denoted by $\mathbb{P}(T_{\mathrm{y}}(V'_{\mathbf{x},\mathbf{y}}))$, $\mathcal{P}_{L_{\mathbf{x},\mathbf{y}}}$, and $\mathcal{Q}_{L_{\mathbf{x},\mathbf{y}}}$ are the sets of all possible input and output types with denominator $L_{\mathbf{x},\mathbf{y}}$. Then, by [45, Th. 11.1.4] which bounds the probability of type classes,

$$\mathbb{P}\left(\sum_{\mathbf{x}}\mathbb{M}(\mathbf{x}|\mathbf{y})\mathcal{D}\left(V_{\mathbf{S},\mathbf{C}|\mathbf{x},\mathbf{y}}||P(X|\mathbf{x})\times Q(Y|\mathbf{y})\right)>D^*_{\mathbf{y}}+3\delta, V_{\mathbf{S},\mathbf{C}}\in E_L(d)\right)\leq \prod_{\mathbf{x}}(L_{\mathbf{x},\mathbf{y}}+1)^{|\mathcal{X}||\mathcal{Y}|}\exp\left(-L\left(\mathcal{D}^*_{\mathbf{y}}+3\delta\right)\right), \tag{A.68}$$

since there are only a polynomial number of joint types. Then, again by [45, Th. 11.1.4], we observe that,

$$\mathbb{P}\left(\sum_{\mathbf{x}}\mathbb{M}(\mathbf{x}|\mathbf{y})\mathcal{D}\left(V_{\mathbf{S},\mathbf{C}|\mathbf{x},\mathbf{y}}||P(X|\mathbf{x})\times Q(Y|\mathbf{y})\right)\leq D^*_{\mathbf{y}}+2\delta, V_{\mathbf{S},\mathbf{C}}\in E_L(d)\right)= \sum_{\substack{V'_{\mathbf{x},\mathbf{y}}\in E_{\mathbf{x},\mathbf{y}}(d_{\mathbf{x},\mathbf{y}})\cap\mathcal{P}_{L_{\mathbf{x},\mathbf{y}}}\times\mathcal{Q}_{L_{\mathbf{x},\mathbf{y}}}:\\ \sum_{\mathbf{x}}\mathbb{M}(\mathbf{x}|\mathbf{y})\mathcal{D}(V'_{\mathbf{x},\mathbf{y}}||P(X|\mathbf{x})\times Q(Y|\mathbf{y}))\leq D^*_{\mathbf{y}}+2\delta\\ \sum_{\mathbf{x},\mathbf{y}}\mathbb{M}(\mathbf{x},\mathbf{y})\rho(x_1,y_1)=\sum_{\mathbf{x},\mathbf{y}}\mathbb{M}(\mathbf{x},\mathbf{y})d_{\mathbf{x},\mathbf{y}}\leq d}}\prod_{\mathbf{x}}\mathbb{P}(T_{\mathrm{y}}(V'_{\mathbf{x},\mathbf{y}}))\geq\frac{\exp(-L(\mathcal{D}^*_{\mathbf{y}}+2\delta))}{\prod_{\mathbf{x}}(L_{\mathbf{x},\mathbf{y}}+1)^{|\mathcal{X}||\mathcal{Y}|}}, \tag{A.69}$$

since for sufficiently large L, there exists at least one term in the summation, i.e., there exists a set of joint types $V'_{\mathbf{x},\mathbf{y}}$ in $E_{L_{\mathbf{x},\mathbf{y}}}(d_{\mathbf{x},\mathbf{y}}), \forall\mathbf{x}\in\mathcal{X}^M, \forall\mathbf{y}\in\mathcal{Y}^M$, such that,

$$\sum_{\mathbf{x}}\mathbb{M}(\mathbf{x}|\mathbf{y})\mathcal{D}\left(V'_{\mathbf{x},\mathbf{y}}||P(X|\mathbf{x})\times Q(Y|\mathbf{y})\right)\leq D^*_{\mathbf{y}}+2\delta,\text{ and }\sum_{\mathbf{x},\mathbf{y}}\mathbb{M}(\mathbf{x},\mathbf{y})d_{\mathbf{x},\mathbf{y}}\leq d. \tag{A.70}$$

Next, taking into account that the probability of one event is larger than or equal to the probability of the intersection, we have,

$$\mathbb{P}\left(V_{\mathbf{S},\mathbf{C}}\in E_L(d)\right)\geq\frac{\exp\left(-L\left(D^*_{\mathbf{y}}+2\delta\right)\right)}{\prod_{\mathbf{x}}(L_{\mathbf{x},\mathbf{y}}+1)^{|\mathcal{X}||\mathcal{Y}|}}. \tag{A.71}$$

By Bayes' law we get,

$$\mathbb{P}\left(\sum_{\mathbf{x}}\mathbb{M}(\mathbf{x}|\mathbf{y})\mathcal{D}\left(V_{\mathbf{S},\mathbf{C}|\mathbf{x},\mathbf{y}}||P(X|\mathbf{x})\times Q(Y|\mathbf{y})\right)>D^*_{\mathbf{y}}+3\delta\;\middle|\;V_{\mathbf{S},\mathbf{C}}\in E_L(d)\right)\leq \prod_{\mathbf{x}}(L_{\mathbf{x},\mathbf{y}}+1)^{2|\mathcal{X}||\mathcal{Y}|}\exp\left(-L\delta\right). \tag{A.72}$$

By the "Pythagorean" theorem [45, Th. 11.6.1], we have,

$$\mathcal{D}(V_{\mathbf{S},\mathbf{C}|\mathbf{x},\mathbf{y}}||V^*_{\mathbf{x},\mathbf{y}})+\mathcal{D}\left(V^*_{\mathbf{x},\mathbf{y}}||P(X|\mathbf{x})\times Q(Y|\mathbf{y})\right)\leq\mathcal{D}\left(V_{\mathbf{S},\mathbf{C}|\mathbf{x},\mathbf{y}}||P(X|\mathbf{x})\times Q(Y|\mathbf{y})\right). \tag{A.73}$$

Hence, $\sum_{\mathbf{x}} \mathbb{M}(\mathbf{x}|\mathbf{y})\mathcal{D}\left(V_{\mathbf{S},\mathbf{C}|\mathbf{x},\mathbf{y}}||P(X|\mathbf{x}) \times Q(Y|\mathbf{y})\right) \leq D^*_{\mathbf{y}} + 3\delta$ implies that,

$$\sum_{\mathbf{x}} \mathbb{M}(\mathbf{x}|\mathbf{y})\mathcal{D}(V_{\mathbf{S},\mathbf{C}|\mathbf{x},\mathbf{y}}||V^*_{\mathbf{x},\mathbf{y}}) \leq 3\delta. \tag{A.74}$$

Finally, by the data processing inequality, we have,

$$\sum_{\mathbf{x}} \mathbb{M}(\mathbf{x}|\mathbf{y})\mathcal{D}(Q_{\mathbf{C}|\mathbf{x},\mathbf{y}}||Q^*(P(X|\mathbf{x}), Q(Y|\mathbf{y}), d^*_{\mathbf{x},\mathbf{y}})) \leq \sum_{\mathbf{x}} \mathbb{M}(\mathbf{x}|\mathbf{y})\mathcal{D}(V_{\mathbf{S},\mathbf{C}|\mathbf{x},\mathbf{y}}||V^*_{\mathbf{x},\mathbf{y}}), \tag{A.75}$$

since, both are the respective y-marginals of the joint types. Hence, Theorem 3 follows from (A.72) as desired. Furthermore, it is worth noting that,

$$R(P(X|\mathbf{x}), Q(Y|\mathbf{y}), d_{\mathbf{x},\mathbf{y}}) = \min_{V \in E_{\mathbf{x},\mathbf{y}}(d_{\mathbf{x},\mathbf{y}})} \mathcal{D}(V||P(X|\mathbf{x}) \times Q(Y|\mathbf{y})). \tag{A.76}$$

Hence, by [38], the minimum in (A.64) is achieved by adding the output-constrained rate-distortion functions at points of equal slopes in all co-ordinates, implying (3.30). ■

A.7 Theorem 4: NTS Algorithm Alternating Minimization over Convex Sets for Markov Sources

We can write the average rate-distortion function over all source-code cross sub-streams $\{\mathbf{s}_{\mathbf{x}}, \mathbf{c}_{\mathbf{y}}\}$, that can be achieved by an output distribution with M-th order Markov property, as an average of double minimization over *convex* sets, i.e.,

$$\overline{R}(d) = \min_{Q(Y|\mathbf{y})} \min_{\substack{\mathbb{M}(\mathbf{y}|\mathbf{x}) \\ d_{\mathbf{x},\mathbf{y}}, V_{\mathbf{x},\mathbf{y}}}} \sum_{\mathbf{x},\mathbf{y}} \mathbb{M}(\mathbf{x})\mathbb{M}(\mathbf{y}|\mathbf{x})\ \mathcal{D}\left(V_{\mathbf{x},\mathbf{y}}\ \middle|\middle|\ P(X|\mathbf{x}) \times Q(Y|\mathbf{y})\right). \tag{A.77}$$

such that $\mathbb{M}(\mathbf{y}|\mathbf{x})$, $d_{\mathbf{x},\mathbf{y}}$ and $V_{\mathbf{x},\mathbf{y}}$ satisfy conditions in (A.65). For ease of notation, let,

$$\overline{D} = \sum_{\mathbf{x},\mathbf{y}} \mathbb{M}(\mathbf{x})\mathbb{M}(\mathbf{y}|\mathbf{x})\ \mathcal{D}\left(V_{\mathbf{x},\mathbf{y}}\ \middle|\middle|\ P(X|\mathbf{x}) \times Q(Y|\mathbf{y})\right). \tag{A.78}$$

It should be noted that all constraints on optimization variables in (A.77) are convex, and it is easy to verify that this yields convex sets. It is easy to show that, for a fixed set of joint distributions $\{V_{\mathbf{x},\mathbf{y}}\}$, the reproduction conditional distribution $Q(Y|\mathbf{y})$ which minimizes the average divergence $\overline{D}$ is the y-marginal of $\sum_{\mathbf{x}} \mathbb{M}(\mathbf{x}|\mathbf{y})V_{\mathbf{x},\mathbf{y}}$ on $\mathcal{Y}$. On the other hand, for a fixed set of conditional distributions $\{Q(Y|\mathbf{y})\}$ and distortion constraint d, the joint distribution which minimizes $\overline{D}$ under constraints in (A.65) will induce $Q(Y|\mathbf{y}) = \sum_{\mathbf{x}} \mathbb{M}^*(\mathbf{x}|\mathbf{y})Q^*(P(X|\mathbf{x}), Q(Y|\mathbf{y}), d^*_{\mathbf{x},\mathbf{y}})$. By the results of Theorem 3, the recursion in Algorithm 6 performs exactly this minimization over the convex sets. It should be noted that the distance in the alternating minimization is measured by divergence. Hence, by [79, Th. 3], the sequences of divergences and distributions will converge to the minimum average divergence, i.e., $\overline{R}(d)$, and the corresponding conditional reproduction distributions. ■

A.8 Theorem 5: Conditional Limit Theorem For Tractable NTS Algorithm over Abstract Alphabets

We assume that the alphabet spaces $\mathcal{X}$ and $\mathcal{Y}$ are complete separable metric spaces (often called Polish spaces), equipped with their associated Borel σ-field $\mathcal{X}'$ and $\mathcal{Y}'$, respectively. Let $\mathbf{x}(i_k)$ and $\mathbf{y}(j_k)$, $k = 1, 2, \ldots, K$, be a sequence of d-matching ML-length words that are generated with the product probability measure P_M^L over $\mathcal{X}^{ML}$, and $Q_{n-1,M,L}$ over $\mathcal{Y}^{ML}$, respectively. For the ease of notation, denote $Q_{n-1,M,L}$ as Q_M^L. In other words, $\rho(\mathbf{x}(i_k), \mathbf{y}(j_k)) \leq d, \forall k \in \{1, 2, \ldots, K\}$. Now let us consider the realizations of the concatenated d-matching source and code vectors, similar to Theorem 1, (also called blocks here and after) $\overline{\mathbf{x}} = (\mathbf{x}(i_1)\ \mathbf{x}(i_2)\ \ldots \mathbf{x}(i_K))$, and $\overline{\mathbf{y}} = (\mathbf{y}(j_1)\ \mathbf{y}(j_2)\ \ldots \mathbf{y}(j_K))$. Furthermore, let $Q_{\overline{\mathbf{Y}}}$ be the random empirical distribution of the random concatenated d-matching code block $\overline{\mathbf{Y}}$. Note that the source block realization $\overline{\mathbf{x}}$ and code block realization $\overline{\mathbf{y}}$ satisfies a stricter distortion requirement, due to the inherent maximum distortion constraint over sub-blocks. In order to capture such stricter distortion requirement, we define a scalar-valued auxiliary distortion function as follows:

$\left(\rho^{(d)} : \mathcal{X}^{ML} \times \mathcal{Y}^{ML} \to \{0,1\}\right)$, which is additive across the K ML-length sub-blocks, i.e.,

$$\rho^{(d)}\left(\mathbf{x}(i_k), \mathbf{y}(j_k)\right) = \begin{cases} 0 & \text{if } \rho\left(\mathbf{x}(i_k), \mathbf{y}(j_k)\right) \leq d, \\ 1 & \text{otherwise} \end{cases} \tag{A.79}$$

$$\rho^{(d)}(\overline{\mathbf{x}}, \overline{\mathbf{y}}) = \frac{1}{K}\sum_{k=1}^{K} \rho^{(d)}\left(\mathbf{x}(i_k), \mathbf{y}(j_k)\right), \tag{A.80}$$

Note that by setting $\rho^{(d)}(\overline{\mathbf{x}}, \overline{\mathbf{y}}) = 0$, we impose a requirement of maximum distortion d per sub-block, over the K sub-blocks. Hence, the condition $\rho\left(\mathbf{x}(i_k), \mathbf{y}(j_k)\right) \leq d$ implies that $\rho^{(d)}(\overline{\mathbf{x}}, \overline{\mathbf{y}}) = 0$, or in other words, the auxiliary distortion function $\rho^{(d)}(\cdot)$ is satisfied between $\overline{\mathbf{x}}$ and $\overline{\mathbf{y}}$ with *zero* distortion constraint. Next, we show that the empirical distribution of the code block $\overline{\mathbf{y}}$ (formed by concatenating the d-matching codewords) is unaltered regardless if the d-matching events for each source word and codeword pair, occurred independently or jointly. In view of (A.79) and by the independent generation of every ML-length part of the source blocks, and code blocks, as well as the definition of the distortion measure $\rho^{(d)}$ at distortion level $\gamma = 0$, we have,

$$\mathbb{P}\left(\rho^{(d)}\left(\overline{\mathbf{X}}, \overline{\mathbf{Y}}\right) = 0 \;\middle|\; \overline{\mathbf{X}} = \overline{\mathbf{x}}\right) = \prod_{k=1}^{K} \mathbb{P}\left(\rho^{(d)}\left(\mathbf{X}(i_k), \mathbf{Y}(j_k)\right) = 0 \;\middle|\; \mathbf{X}(i_k) = \mathbf{x}(i_k)\right). \tag{A.81}$$

$$\mathbb{P}\left(\rho^{(d)}\left(\overline{\mathbf{X}}, \overline{\mathbf{Y}}\right) = 0 \;\middle|\; \overline{\mathbf{X}} = \overline{\mathbf{x}}\right) = \prod_{k=1}^{K} \mathbb{P}\left(\rho\left(\mathbf{X}(i_k), \mathbf{Y}(j_k)\right) \leq d \middle|\; \mathbf{X}(i_k) = \mathbf{x}(i_k)\right). \tag{A.82}$$

Hence, the δ-match event $\Big($for $\rho^{(d)}$ with $\gamma = 0\Big)$ between the MLK-length random source block $\overline{\mathbf{X}}$ and code block $\overline{\mathbf{Y}}$ implies a sequence of d-match events for every ML-length respective sub-blocks $\mathbf{X}(i_k)$, and $\mathbf{Y}(j_k)$. This, together with the independent generation of every ML-length source and code sub-blocks, immediately shows that for every (measurable) $E \subset \mathcal{Y}^{ML}$, $Q_{\overline{\mathbf{Y}}}(E)$ is unchanged whether the d-match events occurred independently or jointly across the K source and respective code sub-blocks. Next define $Q^*_{P^L_M,Q^L_M,\gamma}$ as the optimal distribution that minimize the coding rate for a given current codebook reproduction distribution Q^L_M, and for the auxiliary distortion measure $\rho^{(d)}(\cdot)$ at distortion level γ, as in (3.40), where the definition is repeated here for better readability,

$$Q^*_{P^L_M,Q^L_M,\gamma} = \arg\inf_{Q'} \left\{ I_{\min}(P^L_M||Q',\gamma) + D(Q'||Q^L_M) \right\}, \tag{A.83}$$

$$I_{\min}\left(P^L_M||Q',\gamma\right) = \inf_{\substack{V:[V]_x=P^L_M,\ [V]_y=Q', \\ \mathbb{E}_V\left(\rho^{(d)}(\mathbf{X},\mathbf{Y})\right)\leq\gamma}} I\left(\mathbf{X},\mathbf{Y}\right), \tag{A.84}$$

Now we invoke Theorem 3 in [41], with straight forward extension to sources with memory, for every (measurable) $E \subset \mathcal{Y}^{ML}$, the probability,

$$\mathbb{P}\Big(\big|\hat{Q}_{\overline{\mathbf{Y}}}(E) - Q^*_{P^L_M,Q^L_M,\gamma}(E)\big| > \epsilon \,\big|\, \rho^{(d)}(\overline{\mathbf{X}},\overline{\mathbf{Y}}) = 0, \overline{\mathbf{X}} = \overline{\mathbf{x}} \Big) \to 0, \tag{A.85}$$

as $K \to \infty$, exponentially fast, where $\hat{Q}_{\overline{\mathbf{Y}}}$ is the random empirical distribution of the random code block $\overline{\mathbf{Y}}$ on $\mathcal{Y}^{ML}$. Thus conditioning on the P^L_M-almost every realization $\overline{\mathbf{x}}$ (as $K \to \infty$) and the γ-match event $\rho^{(d)}(\overline{\mathbf{X}},\overline{\mathbf{Y}}) = \gamma = 0$, the probability that the difference between empirical distributions $\hat{Q}_{\overline{\mathbf{Y}}}$ and $Q^*_{P^L_M,Q^L_M,\gamma}$ over any (measurable) $E \subset \mathcal{Y}^{ML}$ is larger than ϵ, with $\epsilon > 0$, goes to zero asymptotically in K. Note that the effective lengths of the source or code blocks are $\overline{L} \triangleq LK$, hence sending $K \to \infty$, obviously implies that $\overline{L} \to \infty$. Furthermore, by [41], we have,

$$\begin{aligned} \mathbb{P}\Big(\big|\hat{Q}_{\overline{\mathbf{Y}}}(E) - Q^*_{P^L_M,Q^L_M,\gamma}(E)\big| > \epsilon \,\big|\, \rho^{(d)}(\overline{\mathbf{X}},\overline{\mathbf{Y}}) = 0, \overline{\mathbf{X}} = \overline{\mathbf{x}} \Big) \ = \\ \mathbb{P}\Big(\big|Q_{\overline{\mathbf{Y}}}(E) - Q^*_{P^L_M,Q^L_M,\gamma}(E)\big| > \epsilon \,\big|\, \overline{\mathbf{X}} = \overline{\mathbf{x}} \Big). \end{aligned} \tag{A.86}$$

This together with Borel-Cantelli lemma, and [41], we conclude that for any measurable set $E \subset \mathcal{Y}^{ML}$,

$$Q_{\overline{\mathbf{Y}}}(E) \to Q^*_{P^L_M, Q^L_M, \gamma}(E), \quad \text{as } K \to \infty, \quad \text{w.p. } 1. \tag{A.87}$$

Since $\mathcal{Y}^{ML}$ is a Polish space, then there exists a countable convergence determining class $\mathcal{E} = \{E_i\} \subset \mathcal{Y}^{ML}$. Therefore, with probability one we have,

$$Q_{\overline{\mathbf{Y}}}(E_i) \to Q^*_{P^L_M, Q^L_M, \gamma}(E_i), \quad \text{as } K \to \infty, \quad \forall i, \tag{A.88}$$

which subsequently implies Theorem 5. ■

A.9 Theorem 6: Abstract Alphabet NTS Algorithm, Alternating Minimization over Convex Sets

Using the same arguments as Theorem 2 and Theorem 4, it is straightforward to verify that the sets of joint distributions $\{P^L_M \times Q^L_M : \text{any } Q^L_M\}$, and the γ-constrained set $\left\{V : [V]_x = P^L_M, \ \mathbb{E}_V \left(\rho^{(d)}(\mathbf{X}, \mathbf{Y})\right) \leq \gamma\right\}$ are convex sets. Furthermore, it should be noted that for a fixed joint distribution V, the reproduction distribution which minimizes $\mathcal{D}(V || P^L_M \times Q^L_M)$ is the y-marginal of V on $\mathcal{Y}^{ML}$. On the other hand, for a fixed Q^L_M and distortion constraint γ, the joint distribution which minimizes $\mathcal{D}(V || P^L_M \times Q^L_M)$ over $\left\{V : [V]_x = P^L_M, \ \mathbb{E}_V \left(\rho^{(d)}(\mathbf{X}, \mathbf{Y})\right) \leq \gamma\right\}$ will induce $Q^*_{P^L_M, Q^L_M, \gamma}$. Hence, by Theorem 5, the recursion in Algorithm 7, achieves a sequence of alternating minimization across convex sets.

$$V^*_{P^L_M, Q_{0,M,L}, \gamma} \to \left(P^L_M \times Q^*_{P^L_M, Q_{0,M,L}, \gamma}\right) \to V^*_{P^L_M, Q_{1,M,L}, \gamma} \to \left(P^L_M \times Q^*_{P^L_M, Q_{1,M,L}, \gamma}\right) \ldots, \tag{A.89}$$

where,

$$V^*_{P^L_M, Q_{n,M,L}, \gamma} \triangleq \arg \inf_{\substack{V:[V]_x = P^L_M, \\ \mathbb{E}_V\left(\rho^{(d)}(\mathbf{X},\mathbf{Y})\right) \le \gamma}} \mathcal{D}\left(V || P^L_M \times Q_{n,M,L}\right). \tag{A.90}$$

It should be noted that the distance in the alternating minimization is measured by divergence. Hence, by [79, Th. 3], the sequences of divergences and distributions will converge to the minimum divergence, i.e., $R(P^L_M, 0)$, and the corresponding optimum reproduction distribution $Q^*_{P^L_M,0}$ on $\mathcal{Y}^{ML}$ asymptotically in K and n. ■

A.10 Theorem 7: Convergence of the NTS Codebook Reproduction Distribution for Abstract Alphabet Sources

We assume that the alphabet spaces $\mathcal{X}$ and $\mathcal{Y}$ are complete separable metric spaces (often called Polish spaces), equipped with their associated Borel σ-field $\mathcal{X}'$ and $\mathcal{Y}'$, respectively. Let $\mathbf{x}(i_k)$ and $\mathbf{y}(j_k)$, $k = 1, 2, \dots, K$, be a sequence of d-matching ML-length words that are generated with the product probability measure P^L_M over $\mathcal{X}^{ML}$, and $Q_{n-1,M,L}$ over $\mathcal{Y}^{ML}$, respectively. For the ease of notation, denote $Q_{n-1,M,L}$ as Q^L_M. In other words, $\rho(\mathbf{x}(i_k), \mathbf{y}(j_k)) \le d, \forall k \in \{1, \dots, K\}$. It can be shown by [41, Th. 3] and straight forward generalization to sources with memory, i.e., $M > 1$, that the marginal probability measure of the d-matching codeword converges in the weak convergence sense to $Q^*_{P_M, Q^{\text{Marg.}}_{n-1,M,L}, d}$, defined in (3.40), as L goes to infinity, i.e.,

$$\mathbf{z} \triangleq \mathbf{y}(j_k), \quad Q^{\text{Marg.}}_{\mathbf{y}(j_k)} = \frac{1}{L}\sum_{\ell=1}^{L} \delta_{\mathbf{z}_\ell}, \quad \forall k, \tag{A.91}$$

$$Q^{\text{Marg.}}_{\mathbf{y}(j_k)} \Longrightarrow Q^*_{P_M, Q^{\text{Marg.}}_{n-1,M,L}, d}\,, \quad \text{as } L \to \infty, \tag{A.92}$$

where $\mathbf{z}_\ell$ is the ℓ-th super-symbol in the codeword $\mathbf{z}$, and $Q^{\text{Marg.}}_{\mathbf{y}(j_k)}$ is the marginal empirical probability measure of $\mathbf{y}(j_k)$ on the alphabet $\mathcal{Y}^M$. Hence, by the definition of $Q_{n,M,L}$ in

(3.36), the marginal probability measure $Q_{n,M,L}^{\text{Marg.}}$ converges weakly to $Q^*_{P_M,Q_{n-1,M,L}^{\text{Marg.}},d}$, as $L \to \infty$, as well. The M-th order joint rate-distortion function in (2.7) can be rewritten as [10, 41]

$$R(P_M, d) = \inf_{Q} \inf_{\substack{V:[V]_x = P_M, \\ \mathbb{E}_V(\rho(\mathbf{X},\mathbf{Y}))\leq d}} \mathcal{D}\left(V || P_M \times Q\right), \tag{A.93}$$

here the inner infimum is taken over all joint distributions V of the random M-length sub-vectors or super-symbols $(\mathbf{X}, \mathbf{Y})$ such that the x-marginal of V is P_M, and the expected distortion $\mathbb{E}_V(\rho\left(\mathbf{X}, \mathbf{Y}\right)) \leq d$. Finally, similar to Theorems 2, 4, 6, the marginal distributions obtained by the recursion in (3.36), as $L \to \infty$, result in a sequence of alternating minimization across convex sets, i.e.,

$$V^*_{P_M,Q_{0,M}^{\text{Marg.}},d} \to \left(P_M \times Q^*_{P_M,Q_{0,M}^{\text{Marg.}},d}\right) \to V^*_{P_M,Q_{1,M}^{\text{Marg.}},d} \to \left(P_M \times Q^*_{P_M,Q_{1,M}^{\text{Marg.}},d}\right) \ldots, \tag{A.94}$$

where, $Q_{n,M}^{\text{Marg.}}$ is the marginal probability measure of $Q_{n,M}$, and,

$$V^*_{P_M,Q_{n,M}^{\text{Marg.}},d} \triangleq \arg \min_{\substack{V:[V]_x = P_M, \\ \mathbb{E}_V(\rho(\mathbf{X},\mathbf{Y}))\leq d}} \mathcal{D}\left(V || P_M \times Q_{n,M}^{\text{Marg.}}\right). \tag{A.95}$$

The sequence of divergences will converge to the minimum divergence, i.e., $R(P_M, d)$, and the marginal probability measure $Q_{n,M,L}^{\text{Marg.}}$ will converge to the corresponding optimum reproduction distribution $Q^*_{P_M,d}$ asymptotically in n and L. ■

Bibliography

[1] S. P. Lloyd, *Least Squares Quantization in PCM*, *IEEE Trans. on Inform. Theory* **28** (1982) 129–137.

[2] J. Max, *Quantizing for Minimum Distortion*, *IEEE Trans. on Inform. Theory* **6** (1960) 7–12.

[3] Y. Linde, A. Buzo, and R. M. Gray, *An Algorithm for Vector Quantizer Design*, *IEEE Trans. on Commun.* **28** (1980) 84–95.

[4] G. Ball and D. Hall, *A Clustering Technique for Summarizing Multivariate Data*, *Behavioral Science* **12** (1967) 153–155.

[5] J. MacQueen, *Some Methods for Classification and Analysis of Multivariate Observations*, *Proc. 5th Berkeley Symp. Math. Statistics and Probability* **1** (1967) 281–297.

[6] K. Rose, *Deterministic Annealing for Clustering, Compression, Classification, Regression, and Related Optimization Problems*, *Proceedings of the IEEE* **86** (1998), no. 11 2210–2239.

[7] J. Ziv and A. Lempel, *A universal algorithm for sequential data compression*, *IEEE Trans. on Inf. Theory* **23** (1977), no. 3 337–343.

[8] J. Ziv and A. Lempel, *Compression of individual sequences via variable rate coding*, *IEEE Trans. on Inf. Theory* **IT-24** (1978) 530–536.

[9] Z. Zhang and V. Wei, *An on-line universal lossy data compression algorithm via continuous codebook refinement—part i: Basic results*, *IEEE Trans. on Inf. Theory* **IT-42** (1996) 803–821.

[10] R. Zamir and K. Rose, *Natural type selection in adaptive lossy compression*, *IEEE Trans. on Inf. Theory* **47** (2001) 99–110.

[11] Y. Kochman and R. Zamir, *Adaptive parametric vector quantization by natural type selection*, in *Data Compression Conference (DCC)*, 2002.

[12] Y. Steinberg and M. Gutman, *An algorithm for source coding subject to a fidelity criterion, based on string matching*, *IEEE Trans. on Inf. Theory* **IT-39** (May 1993) 877–886.

[13] R. Zamir and K. Rose, *Towards lossy Lempel-Ziv: Natural type selection*, in *Proc. of the Inf. Theory Workshop, Haifa, Israel*, p. pp. 58, June 1996.

[14] R. Zamir and K. Rose, *A type generation model for adaptive lossy compression*, in *Proc. of ISIT97*, p. 186, Ulm, Germany, June 1997.

[15] K. Rose, *A mapping approach to rate-distortion computation and analysis*, *IEEE Trans. on Inform. Theory* **40** (Nov., 1994).

[16] M. Agiwal, A. Roy, and N. Saxena, *Next Generation 5G Wireless Networks: A Comprehensive Survey*, *IEEE Commun. Surveys and Tut.* **18** (3rd Quart. 2016) 1617–1655.

[17] M. Shafi *et. al.*, *5G: A Tutorial Overview of Standards, Trials, Challenges, Deployment, and Practice*, *IEEE J. on Sel. Areas in Commun.* **35** (2017) 1201–1221.

[18] T. S. Rappaport *et. al.*, *Millimeter Wave Mobile Communications for 5G Cellular: It Will Work!*, *IEEE Access* **1** (2013) 335–349.

[19] J. G. Andrews *et. al.*, *What Will 5G Be?*, *IEEE J. on Sel. Areas in Commun.* **32** (2014) 1065–1082.

[20] S. Rajagopal, *Beam Broadening for Phased Antenna Arrays using Multi-beam Subarrays*, in *IEEE Int. Conf. on Commun.*, June, 2012.

[21] M. Giordani *et. al.*, *A Tutorial on Beam Management for 3GPP NR at mmWave Frequencies*, *IEEE Commun. Surveys and Tut.* **21** (2018) 173–196.

[22] V. Raghavan *et. al.*, *Beamforming Tradeoffs for Initial UE Discovery in Millimeter-Wave MIMO Systems*, *IEEE J. on Sel. Topics Signal Process.* **10** (2016) 543–559.

[23] A. Elshafiy and A. Sampath, *Beam Broadening for 5G Millimeter Wave Systems*, in *IEEE Wireless Commun. and Net. Conf.*, April, 2019. Reprinted, with permission. © 2019 IEEE.

[24] V. Desai *et. al.*, *Initial Beamforming for mmWave Communications*, in *48th Asilomar Conf. on IEEE Sig., Systems and Comp.*, Nov., 2014.

[25] C. N. Barati, S. A. Hosseini, S. Rangan, P. Liu, T. Korakis, S. S. Panwar, and T. S. Rappaport, *Directional Cell Discovery in Millimeter Wave Cellular Networks*, *IEEE Trans. on Wireless Communications* **14** (2015) 6664–6678.

[26] A. Elshafiy, K. Rose, and A. Sampath, *On Optimal Beam Steering Directions in Millimeter Wave Systems*, in *International Conference on Acoustics, Speech, and Signal Processing*, April, 2019. Reprinted, with permission. © 2019 IEEE.

[27] F. Rosenblatt, *Principles of Neurodynamics: Perceptrons and the Theory of Brain Mechanisms.* Spartan Press, 1961.

[28] M. Minsky and S. A. Papert, *Perceptrons: An Introduction to Computational Geometry.* MIT Press, 1961.

[29] S. J. Nowlan and G. E. Hinton, *Simplifying neural networks by soft weight-sharing*, *Neural computation* **4** (1992), no. 4 473–493.

[30] Y. Gong, L. Liu, M. Yang, and L. Bourdev, *Compressing deep convolutional networks using vector quantization*, *arXiv preprint arXiv:1412.6115* (2014).

[31] S. A. Janowsky, *Pruning versus clipping in neural networks*, *Physical Review A* **39** (1989), no. 12 6600.

[32] E. D. Karnin, *A simple procedure for pruning back-propagation trained neural networks*, *IEEE transactions on neural networks* **1** (1990), no. 2 239–242.

[33] M. C. Mozer and P. Smolensky, *Skeletonization: A technique for trimming the fat from a network via relevance assessment*, in *Advances in neural information processing systems*, pp. 107–115, 1989.

[34] T. N. Sainath, B. Kingsbury, V. Sindhwani, E. Arisoy, and B. Ramabhadran, *Low-rank matrix factorization for deep neural network training with high-dimensional output targets*, in *2013 IEEE international conference on acoustics, speech and signal processing*, pp. 6655–6659, IEEE, 2013.

[35] J. Han, V. Melkote, and K. Rose, *Transform-domain temporal prediction in video coding: Exploiting correlation variation across coefficients*, in *2010 IEEE International Conference on Image Processing*, pp. 953–956, 2010.

[36] S. Li, T. Nanjundaswamy, and K. Rose, *Transform domain temporal prediction with extended blocks*, in *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 1476–1480, 2016.

[37] T. Berger, *Rate Distortion Theory.* Prentice-Hall, 1971.

[38] R. M. Gray, *Conditional rate-distortion theory*, Tech. Rep. 6502-2, Stanford Electronics Lab, 1973.

[39] T. Berger, *Rate distortion theory for sources with abstract alphabets and memory*, *Information and Control* **13** (September, 1968).

[40] E. H. Yang and J. Kieffer, *On the performance of data compression algorithms based upon string matching*, *IEEE Trans. on Inf. Theory* **44** (1998) 47–65.

[41] I. Kontoyiannis and R. Zamir, *Mismatched codebooks and the role of entropy coding in lossy data compression*, *IEEE Tras. on Inform. Theory* **52** (May, 2006).

[42] A. Dembo and I. Kontoyiannis, *Source coding, large deviations, and approximate pattern matching*, *IEEE Tras. on Inform. Theory* **48** (June, 2002).

[43] I. Csiszar, *I-Divergence Geometry of Probability Distributions and Minimization Problems*, *Annals of Probability* **3** (1975), no. 1 146–158.

[44] R. Gallager, *Information Theory and Reliable Communication.* John Wiley and Sons, Inc., 1968.

[45] T. M. Cover and J. A. Thomas, *Elements of Inf. Theory.* Wiley-Interscience, 2006.

[46] A. Elshafiy, M. Namazi, R. Zamir, and K. Rose, *On-the-fly stochastic codebook re-generation for sources with memory*, in *IEEE Information Theory Workshop (ITW)*, 2021. Reprinted, with permission. © 2021 IEEE.

[47] V. Kostina and S. Verdú, *Fixed-length lossy compression in the finite blocklength regime*, *CoRR* **abs/1102.3944** (2011) [arXiv:1102.3944].

[48] R. E. Blahut, *Computation of channel capacity and rate-distortion functions*, *IEEE Transactions on Information Theory* **IT-18** (1972) 460–473.

[49] S. Geman and D. Geman, *Stochastic Relaxation, Gibbs Distribution, and the Bayesian Restoration of Images*, *IEEE Trans. on Pattern Analysis Machine Intelligence* **6** (1984) 721–741.

[50] S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi, *Optimization by Simulated Annealing*, *Science* **220** (1983) 671–680.

[51] A. Elshafiy, M. Namazi, and K. Rose, *On effective stochastic mechanisms for on-the-fly codebook regeneration*, in *IEEE International Symposium on Inf. Theory (ISIT)*, 2020. Reprinted, with permission. © 2020 IEEE.

[52] A. Elshafiy, M. Namazi, R. Zamir, and K. Rose, *Stochastic codebook regeneration for sequential compression of continuous alphabet sources*, in *IEEE International Symposium on Inf. Theory (ISIT)*, 2021, in press. Reprinted, with permission. © 2021 IEEE.

[53] A. Elshafiy and K. Rose, *On stochastic mechanisms codebook regeneration for markov sources*, in *Submitted to IEEE Data Compression Conference*, 2023. Reprinted, with permission. © 2023 IEEE.

[54] M. Harrison and I. Kontoyiannis, *Maximum likelihood estimation for lossy data compression*, in *Proc. 40th Ann. Allerton Conf. Comm. Contr. Comp.*, pp. 596–604, 10, 2002.

[55] M. Madiman, M. Harrison, and I. Kontoyiannis, *Minimum description length vs. maximum likelihood in lossy data compression*, in *International Symposium on Information Theory, 2004.*, pp. 461–, 2004.

[56] A. Dembo and O. Zeitouni, *Refinements of the gibbs conditioning principle*, *Probability Theory and Related Fields* **104** (1996) 1–14.

[57] M. Bartlett, *The frequency goodness of fit test for probability chains*, *Mathematical Proc. of the Cambridge Philosophical Society* **47** (1951), no. 1 86–95.

[58] P. Boukris, *An upper bound on the speed of convergence of the blahut algorithm for computing rate-distortion functions (corresp.)*, *IEEE Transactions on Information Theory* **19** (1973), no. 5 708–709.

[59] C. Léonard and J. Najim, *An extension of sanov's theorem: application to the gibbs conditioning principle*, *Bernoulli* (2002) 721–743.

[60] A. Dembo and O. Zeitouni, *Large Deviations Techniques and Applications.* Springer Science and Business Media, 1998.

[61] J. Kuelbs and A. Meda, *Rates of convergence for the nummelin conditional weak law of large numbers*, *Stochastic Processes and their Applications* **98** (2001), no. 2 229–252.

[62] A. Elshafiy and A. Sampath, *System Performance of Indoor Office Millimeter Wave Communications*, in *IEEE Wireless Communications and Networking Conference*, April, 2019. Reprinted, with permission. © 2019 IEEE.

[63] A. Elshafiy, A. Sampath, and K. Rose, *A Clustering Approach to Optimizing Beam Steering Directions in Wireless Systems*, in *Wireless Communications and Networking Conference (WCNC)*, May, 2021. Reprinted, with permission. © 2021 IEEE.

[64] 3GPP, *Study on Channel Model for Frequency Spectrum Above 6 GHz*, tech. rep., 3rd Generation Partnership Project (3GPP), TR 38.900 V15.0.0, 2018.

[65] C. Balanis, *Antenna Theory, Analysis, and Design.* New Jersey: Wiley, 3rd ed., 2005.

[66] M. Cheng, J.-B. Wang, J.-Y. Wang, M. Lin, Y. Wu, and H. Zhu, *A Fast Beam Searching Scheme in mmWave Communications for High-Speed Trains*, in *IEEE International Conf. on Communications*, May, 2019.

[67] M. Giordani *et. al.*, *Initial Access Frameworks for 3GPP NR at mmWave Frequencies*, in *2018 17th Annual Mediterranean Ad Hoc Networking Workshop (Med-Hoc-Net)*, June, 2018.

[68] D. Yang, L.-L. Yang, and L. Hanzo, *DFT-Based Beamforming Weight-Vector Codebook Design for Spatially Correlated Channels in the Unitary Precoding Aided Multiuser Downlink*, in *IEEE International Conference on Communications*, May, 2010.

[69] A. Chakrabarti and H. Krishnaswamy, *High power, High Efficiency Stacked mmWave Class-E-like CMOS Power Amplifiers: Theory and Implementation*, *IEEE Trans. on Microwave Theory and Techniques* **62** (2014) 1686–1704.

[70] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, *Parallel distributed processing: explorations in the microstructure of cognition.* MIT Press, 1986.

[71] G. D. Magoulas, M. N. Vrahatis, and G. S. Androulakis, *Effective backpropagation training with variable stepsize*, *Neural Networks* **10** (1997), no. 1 69–82.

[72] A. A. Hameed, B. Karlik, and M. S. Salman, *Back-propagation algorithm with variable adaptive momentum*, *Knowledge-Based Systems* **114** (2016) 79–87.

[73] X.-H. Yu and G.-A. Chen, *Efficient backpropagation learning using optimal learning rate and momentum*, *Neural Networks* **10** (1997), no. 3 517–527.

[74] D. P. Kingma and J. Ba, *Adam: A method for stochastic optimization*, in *ICLR (Poster)*, 2015.

[75] L. Deng, *The mnist database of handwritten digit images for machine learning research*, *IEEE Signal Processing Magazine* **29** (2012), no. 6 141–142.

[76] S. S. Kadam, A. C. Adamuthe, and A. B. Patil, *Cnn model for image classification on mnist and fashion-mnist dataset*, *Journal of scientific research* **64** (2020), no. 2 374–384.

[77] O. Kaziha and T. Bonny, *A comparison of quantized convolutional and lstm recurrent neural network models using mnist*, in *2019 International Conference on Electrical and Computing Technologies and Applications (ICECTA)*, pp. 1–5, IEEE, 2019.

[78] S. An, M. Lee, S. Park, H. Yang, and J. So, *An ensemble of simple convolutional neural network models for mnist digit recognition*, *arXiv preprint arXiv:2008.10400* (2020).

[79] I. Csiszar and G. Tusnady, *Information geometry and alternating minimization procedures*, *Statist. Decision* (1984), no. 1 205–237.